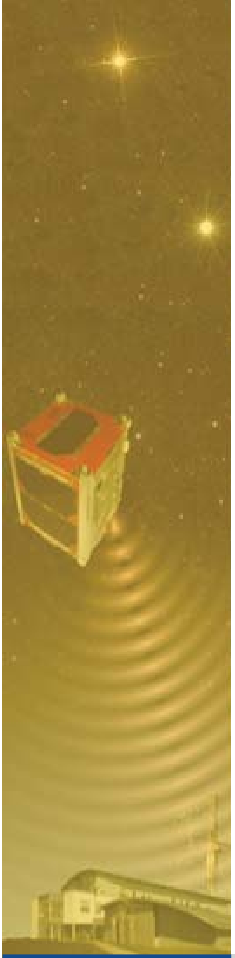
Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling



Würzburger Forschungsberichte
in Robotik und Telematik

Uni Wuerzburg Research Notes
in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Band 9

Zhihao Xu

Cooperative Formation Controller Design for Time-Delay and Optimality Problems

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.

- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85906

opus@bibliothek.uni-wuerzburg.de
http://opus.bibliothek.uni-wuerzburg.de

ISSN 1868-7474 (online)
ISSN 1868-7466 (print)
ISBN: 978-3-923959-96-9 (online)

Zitation dieser Publikation

XU, Z. (2014). Cooperative Formation Controller Design for Time-Delay and Optimality Problems. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 9. Würzburg: Universität Würzburg.

Cooperative Formation Controller Design for Time-Delay and Optimality Problems

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius-Maximilians-Universität Würzburg

vorgelegt von

Zhihao Xu

aus Shanghai, China

Würzburg, Oktober 2013

Eingereicht am: 23.10.2013

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Klaus Schilling
2. Gutachter: Prof. Dr. Dan Necsulescu

Tag der mündlichen Prüfung: 17.10.2014

To my dear parents.

ACKNOWLEDGEMENTS

First of all, I'm deeply grateful to my doctoral supervisor, Prof. Dr. Klaus Schilling, for his successive guidance and motivation over the last four and a half years. If it had not been for his insightful suggestions and remarks to my dissertation work, I would not have been able to accomplish my research so far. I'm also thankful to my second supervisor, Dr. Dan Necsulescu from University of Ottawa, for sharing his research experience with me at the beginning of my doctoral period, as well as providing constructive comments to my dissertation. I also wish to thank Dr. Magnus Egerstedt from Georgia Institute of Technology, for his helpful and intensive discussions with me during my visit at the GRITS lab, together with Dr. Hiroaki Kawashima from Kyoto University, for his extraordinary hands-on instructions to my research work. I'm also indebted to Prof. Dr. Lei Ma from Southwest Jiao Tong University, for his constant support and knowledge transfer during my whole doctoral period.

Secondly, I'm thankful to all the members in our research team: Robin Heß, Philip Necsulescu, Daniel Eck, and our former group members: Dr. Martin Hess, Dr. Martin Saska, Zhongyang Wu for exchanging ideas on research topics and helping out on every possible situation. I'd like to extend my special gratitude to Kaipeng Sun, for his constant help and detailed explanation during each of our discussions. I also wish to thank all the other members in our chair, Stephan Busch, Dr. Christian Herrmann, Florian Leutert, Karthik Ravandoor, Julian Scharnagl, Dr. Marco Schmidt for creating such a nice and warm atmosphere. My special thanks also go to the technical staff Dieter Ziegler, whom I can always count on whenever I have a hardware problem, and Lakshminarasimhan Srinivasan, who is able to provide IT support at every possible moment. I'd also like to thank our secretary Heidi Frankenberger, who has been supportive in all the administrative work.

I'm also greatly thankful to Greg Droge, who has been constantly sharing valuable research ideas with me, and helping me out on various computer programming issues. My deep gratitude also goes to Dr. Martin Schröter, who worked with me on a large amount of mathematical problems in my research work, and helped to establish our joint publications.

Finally, I'd like to thank my dear friend Kechun Zhang for her constant and strong support to me, which has accompanied and encouraged me through all the ups and downs. I also wish to thank my Mom and Dad for their undoubted trust in me during my complete study period in Germany. I wish I could make up for the time that I could not spend with them in the past four and a half years.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . **iii**

LIST OF FIGURES . . . . **ix**

SUMMARY . . . . **xiii**

1 INTRODUCTION **1**

1.1 Motivation and Contribution 3

1.2 Outline 4

2 BACKGROUND **7**

2.1 Control of Multi-Agent System 7

2.1.1 Feedback Linearization 7

2.1.2 Time-delay Problem 9

2.1.3 Consensus Problem 11

2.2 Optimal Control 14

2.2.1 Model Predictive Control 15

2.2.2 Parameterized Model Predictive Control 16

2.3 Experimental Platform 17

2.3.1 Mathematical Model 17

2.3.2 Hardware 18

3 TIME-DELAYED FORMATION CONTROLLER **21**

3.1 Problem Formulation and Related Work 21

3.1.1 Problem Description 21

3.1.2 Related Work 23

3.2 System Modeling 25

3.3 Formation Feedback Control 27

3.3.1 Input-Output Feedback Linearization 27

3.3.2 Stability and Internal Dynamics 28

3.4 Time-delayed Formation System 31

3.4.1 P-type Controller for Delay Compensation 31

3.4.2 Improved PD-type Controller 36

3.5 Parameter Design and Stability Analysis 38

3.5.1 Parameter Selection 38

3.6 Convergence and Internal Dynamics 44

3.6.1 Exponential Convergence 46

3.6.2 Internal Dynamics in Time-delayed Case 48

3.7 Simulation and Experimental Validation 51

3.7.1 Simulation 52

3.7.2 Cooperative V-Formation Experiment 57

3.7.3 Non-cooperative Formation Tracking using Visual Data . . . 60

3.8 Summary and Discussion 64

4 OPTIMAL INITIAL CONDITION – READINESS ANALYSIS 67

4.1 Definition 67

4.1.1 Inspiration 67

4.1.2 Readiness Formulation 69

4.2 Case Study 71

4.2.1 Problem Formulation 71

4.2.2 System Modeling and Formation Control Design 72

4.2.3 Readiness Optimization 76

4.2.4 Simulation Results 79

4.3 Summary and Discussion 82

5 CONTROLLER PARAMETER OPTIMIZATION 83

5.1 Scenario and Problem Definition 83

5.2 Related Work 85

5.2.1 Cooperative Systems 85

5.2.2 Kuramoto Model 87

5.2.3 Parameterized MPC for Multi-Robot Coordination 89

5.3 Model Design for Balanced Deployment 91

5.3.1 Cosine-Kuramoto Model 91

5.3.2 Examples 93

5.4 Optimal Convoy Control 96

5.4.1 Design Requirements and Technical Assumptions 96

5.4.2 Model and Control of Car-like Robot Convoy 97

5.4.3 Parameterized Model Predictive Control 100

5.5 Case Studies 105

5.5.1 Case 1: Centralized 106

5.5.2 Case 2: Decentralized 109

5.6 Experimental Validation 114

5.7 Summary and Discussion 118

6 CONCLUSIONS 119

APPENDIX A — COSINE POSITIVE 121

APPENDIX B — PROOF ON OPTIMALITY CONDITIONS FOR OPTIMAL READINESS (THEOREM 4.1) 123

APPENDIX C — PROOF ON OPTIMALITY CONDITION FOR PARAMETERIZED MPC (THEOREM 5.2) 127

LIST OF FIGURES

1.1 Area Coverage and Monitoring; the orange area is the target ground to monitor; the dashed disc is the effective monitoring perimeter of each UAV. 2

1.2 Multiple Vehicles Transport Large Load; the vehicles are coordinated locally to maintain the shape when the path changes. 2

2.1 Car-like mobile robot model 18

2.2 MERLIN Outlook 18

2.3 Overview of the hardware architecture on MERLIN 19

3.1 Illustration of formation teleoperation with communication delays in various links 22

3.2 Model of leader-follower formation for car-like vehicles; leader robot is either teleoperated or driven autonomously with predefined commands, while the follower robot is driving autonomously based on the designed controller. 26

3.3 Formation control scheme under leader-follower approach; the controller block refers to Eq. (3.3) and Eq. (3.4); the kinematics model is described in Eq. (2.20); and the LF-system states are from Eq. (3.2). 28

3.4 Follower tracks the delayed leader. At the time when the follower receives the leader information, the actual leader is T time ahead. Thus the follower is only tracking the delayed version of the leader. $\tilde{\rho}^{\mathrm{d}}$ is defined in Eq. (3.9). 32

3.5 Follower tracks the delayed leader using its own delayed version. ρ^{d} is defined in Eq. (3.10). 32

3.6 Formation control scheme of time-delayed system. The extra delay component T in blue is added intentionally into the system. 33

3.7 Stability chart of system (3.13); the shaded area indicates the stable region where all the roots are located in the $\mathbb{C}^-$ plane; the curves are described in Eq. (3.17). 35

3.8 Stability chart of system (3.20); the curves are defined in Eq. (3.25); the shaded area indicates the stable region where all the roots are located in the $\mathbb{C}^-$ plane. 38

3.9 The zero solutions of function $g_{k,d}(s)$ with different values of d. 46

3.10 Real part of the rightmost root; the curve with $d = -e^{-2}$ illustrates Theorem 3.3. 48

3.11 Parameter domain for global stability; the color map represents the maximum of delay T_a in [s] 51

3.12 Formation tracking without delay, $k_\rho = 3, k_\alpha = 4$, where leader drives along a predefined circle, and the followers are driving autonomously along under the controller from Eq. (3.4). 53

3.13 Formation tracking with delayed leader states and current follower states, $k_\rho = 1.5, k_\alpha = 2$; delay=0.6s. The steady-state error is clearly observed. 54

3.14 Formation tracking with delayed leader states and delayed follower states to compensate for steady-state error; P-type controller, with $k_\rho = 1.5, k_\alpha = 2$; delay=0.6s. 55

3.15 Formation tracking with PD-Type controller under delayed leader states; $k_\rho = 0.676, k_\alpha = 0.676, d_\rho = -0.135, d_\alpha = -0.135; \text{delay} = 0.4 - 0.8s$. 56

3.16 Snapshots of MERLIN in leader-follower experiment 57

3.17 MERLIN formation drive with delayed leader states; the parameters used in the experiment are as follows: $\rho_d = 1.5m, \alpha_d = \pm\pi/6, k_\rho = 1.2, k_\alpha = 1.2; \text{delay} = 0.4 - 0.8s$. Leader is driving at 0.5 m/s, with steering angle at 5 deg during circular motion. 58

3.18 MERLIN formation with PD-Type controller; the parameters used in the experiment are as follows: $\rho_d = 1.5m, \alpha_d = \pm\pi/6, k_\rho = 1.2, k_\alpha = 1.2, d_\rho = -0.135, d_\alpha = -0.135; \text{delay} = 0.4 - 0.8s$. Leader speed is regulated at 0.5 m/s and steering angle at 5 deg during circular motion. 59

3.19 Hardware setup on Outdoor MERLIN for formation tracking using visual data, and high-precision measuring unit for evaluation. 61

3.20 Snapshot of the initial formation setup. The camera perspective from the follower robot is also displayed in the figure. The bounding box in color is the estimated 6D posed of the target robot. 61

3.21 Formation tracking results under P-type controller given in Eq. (3.12) to achieve the linear system of Eq. (3.11). Controller parameters are set as: $\rho_d = 1.5m, \alpha_d = 0, k_\rho = 1.5, k_\alpha = 1.5$. The target robot is driving at 0.3 m/s. 62

3.22 Formation tracking results under P-type controller given in Eq. (3.12) to achieve the linear system of Eq. (3.11). Controller parameters are set as: $\rho_d = 1.5m, \alpha_d = 0, k_\rho = 1.5, k_\alpha = 1.5, d_\rho = -0.135, d_\alpha = -0.135$. The target robot is driving at 0.3 m/s, with steering angle at 5 deg during circular motion. 63

4.1 Depiction of a leader-follower network, with the followers lining up on a circle and the leader at the center; blue discs represent the nonholonomic follower robots, with their headings indicated by the red arrows; the red disc in the middle is the leader robot; blue dashed lines are the communication links. 71

4.2 Car-like mobile robot model with off-center point z as reference . . . 74

4.3 Edge-tension energy distribution with optimal initial orientations; θ_l is used in Eq. (4.24) to generate instantaneous movement of the leader, which is also considered as the exogenous input. 79

4.4 Edge-tension energy distribution with uniform orientations. 80

4.5 Cost comparison; the costs for tangential and optimal setups approach zero much faster than the other two configurations, indicating the faster recovery of the formation shape. 81

4.6 Optimal initial orientations of the convoy; the red arrows indicate the orientations, and the blue dashed lines are the communication or sensing links. 81

5.1 Illustration of mobile robots in circular convoy. The moving target is represented as a disc in the center of the circle. 84

5.2 Phase synchronization example, where $\omega_i = 0$ in Eq. (5.1) 89

5.3 Frequency synchronization example, where $\omega_i \neq 0$ in Eq. (5.1) 89

5.4 Balanced distribution of 10 agents 94

5.5 Self-organizing behavior in balanced distribution with added agents. . 95

5.6 Convoy Protection with Car-like Robots; the protected target is denoted by the brown disc in the center, and the dashed circle indicates the desired convoy shape with the protection perimeter being r. . . . 98

5.7 Block diagram of the consensus-based parameterized MPC in convoy protection scenario, where the operations in the gray area are performed on each individual agent. $j \in \mathcal{N}_i$ represents one of agent i's neighbors, and $t \in (t_0, t_f]$ indicates the time horizon of the optimization. $c^{(k)}$ is the position of the target at time k. 106

5.8 Convoy Protection with Static Target (scenario 1 in centralized case). 107

5.9 Convoy protection with decreasing convoy radius (scenario 2 in centralized case). 108

5.10 Convoy protection with moving target (scenario 3 in centralized case). 110

5.11 Decentralized optimization control in convoy protection; the dashed line indicates the desired convoy location; the green square is the path traveled by the target in red disc. 4 mobile robots are protecting the target on the desired circle. 112

5.12 Optimal control signals in the decentralized control strategy; The gray dashed lines indicate the transition between different phases of motion; Note that the steering angles are bounded by ±0.3 rad due to mechanical limits. 113

5.13 Snapshots of the optimal convoy protection experiment with three MERLIN vehicles. 116

5.14 Speed and steering profiles in the convoy protection experiment. . . . 117

SUMMARY

This dissertation presents controller design methodologies for a formation of cooperative mobile robots to perform trajectory tracking and convoy protection tasks. Two major problems related to multi-agent formation control are addressed, namely the time-delay and optimality problems. For the task of trajectory tracking, a leader-follower based system structure is adopted for the controller design, where the selection criteria for controller parameters are derived through analyses of characteristic polynomials. The resulting parameters ensure the stability of the system and overcome the steady-state error as well as the oscillation behavior under time-delay effect. In the convoy protection scenario, a decentralized coordination strategy for balanced deployment of mobile robots is first proposed. Based on this coordination scheme, optimal controller parameters are generated in both centralized and decentralized fashion to achieve dynamic convoy protection in a unified framework, where distributed optimization technique is applied in the decentralized strategy. This unified framework takes into account the motion of the target to be protected, and the desired system performance, for instance, minimal energy to spend, equal inter-vehicle distance to keep, etc.

Both trajectory tracking and convoy protection tasks are demonstrated through simulations and real-world hardware experiments based on the robotic equipments at Department of Computer Science VII, University of Wüzburg.

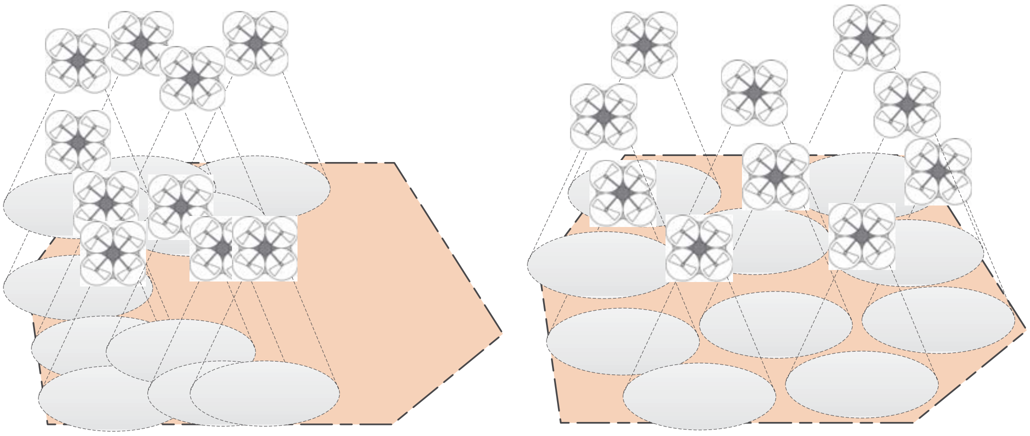
CHAPTER 1

INTRODUCTION

Over the past two decades, the research of multi-agent systems for a variety of human activities has tremendously increased. The concept of multi-agent systems provides solutions to many scientific and social problems that impose either complex structures, or structures that are difficult to handle by a single-agent system. For example, in Figure 1.1, a group of intelligent agents (or Unmanned Aerial Vehicles, UAV) are organized to work cooperatively with each other in order to perform area coverage and monitoring. These agents can be of small sizes equipped with less expensive sensing units. The coordination between the agents are implicit, which means that there is no central coordinator to issue commands to each agent. They only exchange information with their neighboring agents locally, while achieving the mission on a global level[1]. In Figure 1.2, multiple ground vehicles are coordinated to transport large payload, which requires the vehicles to maintain a certain formation during the transportation, especially during the path changes. This type of robotic application is inspired by the biological phenomenon in ants [9]. More examples include Earth remote sensing with satellite constellations, oceanographic currents determination over long distance with underwater robotic swarms, etc.

The issue of decentralized formation control thus arises in the application of multi-agent systems. One essential step in designing the formation control is the selection of controller parameters. Therefore, this dissertation presents two methodologies for selecting controller parameters, that deal with time-delay and optimality problems, respectively. The first design methodology ensures stability of the system with a stable parametric region for the controller parameters, whereas the second one utilizes optimization techniques to tune the controller parameters online. These two frameworks are demonstrated through two exemplary applications, the formation tracking control based on leader-follower structure, and the dynamic convoy protection with multiple ground mobile robots. In the rest of this chapter, we elaborate on the motivations for the dissertation, as well as its scientific contributions.

[1] One real-world implementation of area coverage and intervention with unmanned aerial and ground vehicles can be found in [2].



(a) Initial deployment of 10 UAVs **(b)** Final distribution of 10 UAVs covering the complete area

Figure 1.1: Area Coverage and Monitoring; the orange area is the target ground to monitor; the dashed disc is the effective monitoring perimeter of each UAV.

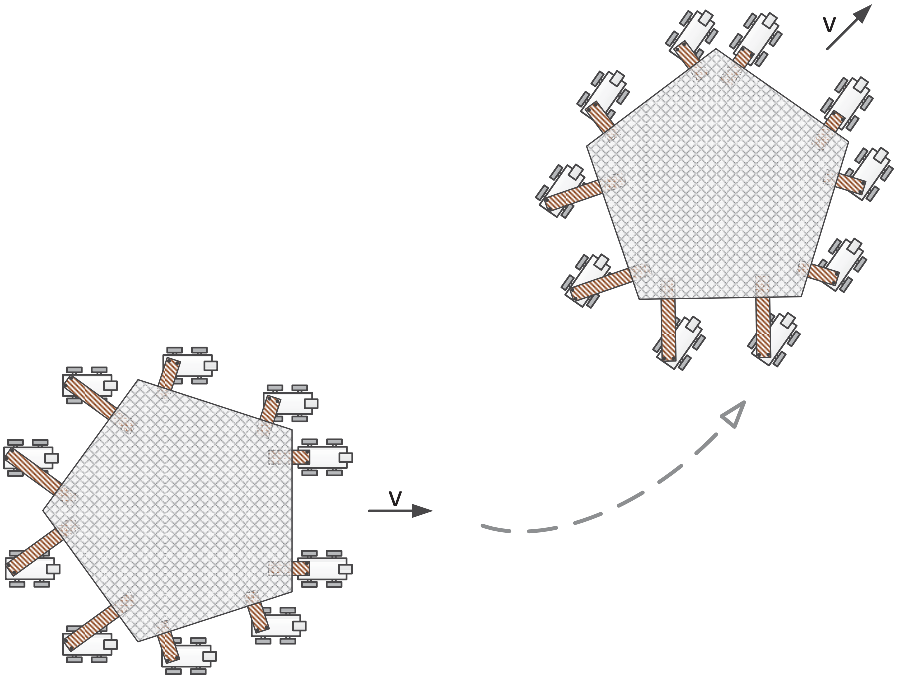


Figure 1.2: Multiple Vehicles Transport Large Load; the vehicles are coordinated locally to maintain the shape when the path changes.

1.1 Motivation and Contribution

The ever-growing interest in applications of multi-agent systems is largely attributed to the development of distributed networked control theories, which involve the information exchange between the systems through a communication network. Among the numerous networked control tools, the methods to overcome the communication constraints in the network are studied intensively. These constraints have various forms, such as time delays in long-distance communication, packet loss and disorder in congestive links, etc. In this dissertation, we focus on one aspect of these constraints, namely the time-delay effect. In particular, we design the formation controller for a leader-follower tracking system, that maintains a certain fixed formation shape with time-delay compensation. Without this compensation, the formation system will fail to keep the desired shape due to the delayed information exchange in the communication channel. We aim to provide a set of parameter selection rules for such a stable formation controller that can be utilized in various collaborative tasks, such as large load transportation shown in Figure 1.2. This controller can also be applied to non-cooperative multi-agent systems, such as adaptive cruise control (ACC) for automotive industry, where the distance and aligning orientation between two driving cars are expected to be safely maintained. A test scenario to demonstrate the usage of the controller in non-cooperative tracking system is presented in Chapter 3, where the main source of time delay in the system is from the computation involved in image processing.

On the other hand, apart from keeping the formation shape, most multi-agent systems are expected to operate with many other requirements, such as low energy spent across the system, fast mission completion time, etc. For these purposes, optimal control is one admissible and effective technique. This control approach considers the system dynamics and generates the optimal control actions based on desired performance indices. To demonstrate the utility of optimal control in the coordination of multi-agent system, we present a dynamic convoy protection scenario with multiple mobile robots. More specifically, we consider parameterized model predictive control method to overcome the computational demands commonly arising from closed-loop optimal control problems. Using distributed optimization technique to optimize controller parameters, we intend to show how multi-agent systems can benefit from the nonlinear optimal control in real-world scenarios.

The contributions of the dissertation are threefold:

- For the time-delayed formation tracking system, a stable PD-type controller is proposed and applied in real-world mobile robotic systems. This controller ensures the stability of the time-delayed formation system, and overcomes the steady-state error and oscillating behavior of the system, which are often neglected in the existing literature. Controller parameters are chosen according to the derived parametric stable region. Meanwhile, the internal dynamics of a feedback linearized system are also proved to be bounded. Apart from the application in cooperative formation tracking system, this type of controller is also suitable for non-cooperative missions, such as adaptive cruise control (ACC). A demonstration of ACC using two car-like mobile robots is provided as an extended application.

- For a general nonholonomic multi-agent system, the notion of readiness is newly proposed to characterize the initial conditions of the system. This notion can be considered as an extension to the existing indices, such as manipulability, responsiveness, which are used to characterize the holonomic multi-agent systems. Classical optimization approach based on calculus of variations is used to derive the optimal readiness. This provides the criteria for the initialization of the multi-agent system to accomplish the cooperative tasks.
- A parameterized model predictive control framework is proposed and demonstrated through a dynamic convoy protection scenario with multiple mobile robots. Conventional model predictive control method is converted into a parameter optimization problem, which largely reduces the computational complexity, thus being implementable online. A new cosine-Kuramoto model is designed for the distributed coordination of multi-agent systems, which allows the system to be scaled and controlled in a decentralized fashion. Optimal control technique is further applied to obtain the optimal controller parameters, which satisfy the desired performance of the convoy protection system. The real-world experiment showcases the feasibility and applicability of the proposed framework.

1.2 Outline

The rest of the dissertation is structured as follows[2]: Chapter 2 presents some background knowledge on the control of multi-agent systems, as well as the experimental

[2] The references given in this section are the publications authored or co-authored by the author of this dissertation. They are associated with the corresponding chapters.

platform we use for demonstration. Chapter 3 deals with the time-delayed formation tracking system, where the controller parameter tuning methods are discussed. We also analyze the stability of the formation controller, and the internal dynamics of the time-delayed system [150, 151]. Chapter 4 introduces a new notion to evaluate the initialization of multi-agent systems, especially nonholonomic systems, based on certain performance indices [149]. Chapter 5 demonstrates the parameterized optimal control approach in multi-agent system to achieve a dynamic convoy protection mission. The overall implementation is based on the optimal initialization studied in Chapter 4, as well as the distributed optimization technique, which enables the online optimization of controller parameters [148]. Chapter 6 presents a summary of the dissertation, as well as some research aspects that are considered appropriate extensions to the results from this work.

CHAPTER 2

BACKGROUND

This chapter is intended to provide background knowledge in the robotics research fields concerning two major topics, namely multi-agent formation control and optimal control. Some of the state-of-the-art research is also presented along. The contents of this chapter can be considered as laying the conceptual basis for the rest of this dissertation. At the end of this chapter, we introduce our experimental platform, which we use to evaluate the developed techniques.

2.1 Control of Multi-Agent System

The study of Multi-Agent System (MAS) has gained widespread recognition in numerous research areas over the past two decades. In general, MAS refers to the system consisting of multiple interacting elements, also known as agents, to perform certain activities collectively. Due to its extensive applicability, MAS framework is utilized to outline the interesting issues arising from, e.g., robotics [137], sensor networks [141], computer vision [132], software engineering [89], even sociological [45] and economical [10] studies. One of the foci of this dissertation is on the formation control aspect of multi-agent systems, especially nonholonomic mobile robot systems. To address this issue, in this section we introduce one control methodology, *feedback linearization*, and point out two related problems, namely the *time-delay* and *consensus* problem.

2.1.1 Feedback Linearization

Feedback linearization is an approach in nonlinear control design, where the nonlinear system dynamics are transformed algebraically into the equivalent simplified ones, either fully or partially linear, often depending on the control objectives. This approach differs completely from the conventional (Jacobian) linearization, which uses linear approximation of the nonlinear dynamics to simplify the system.

Consider the problem of designing control law to stabilize a class of nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x) + G(x)u \\ y &= h(x). \end{aligned} \tag{2.1}$$

If this system is feedback linearizable [69, Chapter 13], through the change of variable (or so-called *diffeomorphism*)

$$z = T(x),$$

the system can be transformed into the form

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)], \tag{2.2}$$

where (A, B) controllable and $\gamma(x)$ nonsingular on the domain of interest. Further, we can pick the following state feedback control law:

$$u = \alpha(x) + \beta(x)v \tag{2.3}$$

where $\beta(x) = \gamma^{-1}(x)$, to convert the system into an equivalent linear system

$$\dot{x} = Ax + Bv.$$

By designing the new control input v using linear system theories, the original nonlinear system can be stabilized. If we design the control v such that the system input-output map is linearized, while the system states may be partially linearized, we call the control law *Input-Output Feedback Linearization.*

Input-output feedback linearization is a typical technique used in tracking control of robotic navigation, where the control objective is often for the robot to track a desired trajectory while keeping the overall system states bounded. Kim and Oh [71] present a well-defined tracking control design for a wheeled mobile robot using input-output linearization. Due to the nonsquare (underactuated) property of the system, the generalized inverse technique is used to obtain $\beta(x)$ as in Eq. (2.3). Some other recent work based on this technique can be found in [67, 154].

Typically the design procedure is performed in the following three steps [127]:

a) differentiate the system output until the input appears explicitly;

b) choose the input to cancel the nonlinearities and guarantee the tracking convergence;

c) study the stability of the internal dynamics.

Internal dynamics are the “unobservable” dynamics of the system, which cannot be seen from the input-output relationship as in Eq. (2.2). The stability of internal dynamics is required in solving the complete tracking control problem. In practice, the stability of internal dynamics is proved in the BIBO (Bounded-Input Bounded-Output) sense by analyzing system's zero dynamics, while zero dynamics is defined to be the internal dynamics of the system when the system output is kept at zero by the input.

The three steps mentioned above are the procedures we follow for designing the formation controller in Chapter 3, with the extension of compensating for the time-delay effect in the formation system.

2.1.2 Time-delay Problem

Multi-Agent Systems are often modeled as Networked Control Systems (NCSs) for the purpose of analyzing certain aspects in the communication channel which affect system performance. These aspects are generally referred in NCSs as network imperfections, such as time delays, packet losses and disorder, and data quantization, etc. In this section, we focus on the problem of time delay in NCSs, which is one of the main topics addressed in the rest of the dissertation. Recent studies on the other aspects of network imperfections include [35, 119, 144, 157] for packet losses and disorder, and [14, 43, 83] for data quantization, to name a few.

The time delays in an NCS are generally caused by two sources, the transmission distance and the network traffic congestion, while the computational delay is negligible compared to these two classes. As shown in the early studies of time-delay systems from Lee and Dianat [80], the system performance and stability are often degraded by the delays, although the fact that delays can have stabilizing effects on the system is also reported in some literature (see [126]). The time delays in NCSs are commonly categorized into constant versus time-varying delays, and deterministic versus stochastic delays. This categorization is mostly dependent on the applications in different networks. For instance, the constant delay is supposed to hold in CAN protocols, or in situations where data buffers are used at each network node to create an upper-bound of the delays, so that the system design is simplified. In most Internet-based NCSs, the varying time delays are considered to be more practical, although certain conditional bounds on the varying delays are often placed to allow the modeling of the systems.

Various control strategies to cope with the time-delay problem mentioned above are developed over the past few decades. Tipsuwan and Chow [138] provide a summary on the typical methodologies used in literature up to 2003, including modeling, analysis and synthesis of NCSs involved with delay problem. Later on, Naghshtabrizi et al. [102] complement the earlier work by addressing the advances from 2003 to 2007 in the control community. The approaches of modeling NCSs as delayed differential equations (DDEs), or as switched systems, are reviewed for that period. The most recent survey on NCSs from Zhang et al. [155] reports the developments from 2007 to 2012, where the authors classify the control design into *robustness* and *adaptation* frameworks. For more details on the control strategies, we refer the readers to the surveys mentioned above, and the references therein.

In order to be self-contained, we state the following definitions and theorems from other references, which are indicated correspondingly.

1. The following theorem formulates the boundedness of the solutions to a general linear time-invariant delay differential equation, which is an excerpt from [50]. It is the basis for the Theorem provided in Section 3.6.

   Theorem 2.1 (Theorem 1.5 from [50]). *Given a general linear time-invariant delay differential equation*

   $$\dot{x}(t) = \int_{-r}^{0} x(t+T)\,dF(T) \tag{2.4}$$

   and its fundamental solution $\Phi(t, t_0)$ *with the initial condition*

   $$\Phi(t_0+T, t_0) = \begin{cases} I & T = 0 \\ 0 & -r \leq T < 0 \end{cases} . \tag{2.5}$$

   Let $\Delta(s)$ *be the Laplace transform of* $\Phi(t, t_0)$*, and define*

   $$\lambda_0 = \max\{\mathrm{Re(s)} \,|\, \det[\Delta(\mathrm{s})] = 0\}, \tag{2.6}$$

   then the following three statements are true:

   (i) The system (2.4) *is stable if and only if* $\lambda_0 < 0$.

   (ii) For any $\lambda > \lambda_0$*, there exists a* $K > 0$*, s.t. any solution* $x(t)$ *of Eq.* (2.4) *with the initial condition* (2.5) *is bounded as*

   $$\|x(t)\| \leq Ke^{\lambda t}. \tag{2.7}$$

   (iii) λ_0 *is continuous with respect to* r_k*, for all* $r_k > 0, k = 1, 2, ...K$.

   The number λ_0 is known as the stability exponent of the system.

2. The following theorem (Elsgolts and Norkin [see 36, pp.160, 161]) provides the conditions for asymptotic stability of a delay-differential equation. It is the basis for the proof in Section 3.6.2.

Theorem 2.2. *The null solution of the system*

$$\dot{x}(t) = \sum_{j=1}^{n} \sum_{i=0}^{m} a_{ij} x_j(t - \tau_i) + R(t, x_1(t), \ldots, x_1(t - \tau_1), \ldots, x_n(t - \tau_m)) \quad (2.8)$$

is asymptotically stable if:

(a) all roots of the characteristic equation for the first approximation system for (2.8),

$$\det \left(\sum_{i=0}^{m} A_i e^{-\lambda \tau_i} - \lambda I \right) = 0, \quad (2.9)$$

where $[A_i]_{jj} = a_{ij}$, *have negative real parts;*

(b) R is such bounded that:

$$\| R(t, u_1, \ldots, u_{n(m+1)}) \| \leq \alpha \sum_{i=1}^{n(m+1)} \| u_i \|, \quad (2.10)$$

where $\alpha > 0$ *is a sufficiently small constant, and* $\|u_i\|$ *are sufficiently small.*

This theorem indicates that if the first approximation (linear or linearized part) of a nonlinear delay-differential equation has asymptotically stable null solution, and the nonlinear part is bounded, then the original equation has asymptotically stable null solution.

2.1.3 Consensus Problem

The collaborative phenomena in biological, physical and social systems have greatly inspired the research of multi-agent systems. Birds migrating in flocks, fish chasing in school, and people applausing consistently are some typical natural examples where a single agent acts cooperatively with each other to achieve certain common objectives at the formation level. This has been understood as a technological direction of going from simple local rules to complex global behaviors. An early study from Reynolds [123] interprets three principles to achieve this type of cooperation, which are termed as *separation*, *alignment* and *cohesion* (also known as *collision avoidance*, *velocity matching* and *flock centering*). Based on these principles, a great number of literature has emerged ever since, revealing different facets of cooperative multi-agent systems and some underlying theories, as well as their new applications in industrial areas. Murray [101] presents a broad overview of the research developments in this regard till

2006. The author covers the application areas, such as military systems, mobile sensor networks and transportation systems, reviews the technological basics supporting these applications, including formation control, cooperative tasking, spatial-temporal planning and consensus algorithms, and indicates some future research topics, e.g. the synthesis of control, communication and computation. The study from Cao et al. [17] provides a survey on distributed multi-agent systems from 2006 to 2012, focusing on the following five directions: consensus-related topics, distributed formation, optimization, estimation and control, and task assignment. We refer the readers to the surveys mentioned above for more details, as well as the references therein.

Up till now, certain aspects of multi-agent coordination problem are well understood. A prime example is the consensus problem, where the agents are expected to reach an agreement on some quantities by locally communicating with or sensing each other. Olfati-Saber et al. [108] present a cohesive overview of the theories and applications of consensus problems in a unified framework, which combines the basic notions in information consensus (e.g. graph theories [98]) and control designs (e.g. distributed formation control [39],) for system convergence and performance analysis. Under this framework, various consensus algorithms, such as nearest-neighbor rules from Jadbabaie et al. [59], flocking algorithm from Olfati-Saber and Murray [109], and gossip algorithm from Boyd et al. [11] can be synthesized based on similar mathematical tools.

Some other topics, such as rendezvous and synchronization, are derived from consensus algorithms. Rendezvous in space [22] is equivalent to reaching a consensus in position with a number of agents based on position-induced topology, e.g. proximity graph. Both Lin et al. [85] and Cortés et al. [22] consider this as an unconstrained consensus problem with variations in network topology. Hess [54], however, tackles the rendezvous problem of nonholonomic wheeled mobile robots under constrained kinematics, and generates discontinuous control laws for a group of car-like robots to achieve the goal.

To understand the contents presented in Chapter 4 and 5, in the following, we provide some basics in algebraic graph theory according to the convention used by Mesbahi and Egerstedt [98].

The model for the interaction topology of a network of N agents can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{e_1, \ldots, e_M\} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, where M is the number of edges.

The adjacency matrix $A(\mathcal{G})$ is the symmetric $N \times N$ matrix indicating the adjacency relationships in the graph $\mathcal{G}$, where

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

Thus the neighborhood set $\mathcal{N}_i$ of the node i is defined by:

$$\mathcal{N}_i = \{v_j | [A(\mathcal{G})]_{ij} = 1\}. \tag{2.12}$$

The incidence matrix $D(\mathcal{G})$, encoding not only the adjacency map of the nodes but also the orientation of the edges, is defined by an $N \times M$ matrix, i.e.

$$[D(\mathcal{G})]_{ij} = \begin{cases} -1 & \text{if } v_i \text{ is the tail of } e_j, \\ +1 & \text{if } v_i \text{ is the head of } e_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.13}$$

The direction of the edges is used to define the information flow, or communication links. For instance, in an undirected cycle graph $\mathcal{G}_c$, where each node can communicate with its two neighbors, the incidence matrix $D(\mathcal{G}_c)$ can be described by:

$$D = \begin{pmatrix} -1 & 0 & \cdots & +1 \\ +1 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & +1 & -1 \end{pmatrix}. \tag{2.14}$$

Note that the directions of edges in a typical graph, e.g. cycle, star, tree or all-to-all, are such defined by convention, while in principle they can be arbitrarily given.

The graph *Laplacian* $L(\mathcal{G})$ is defined as:

$$L(\mathcal{G}) = D(\mathcal{G}) D(\mathcal{G})^\top, \tag{2.15}$$

where $D(\mathcal{G})$ is the incidence matrix in Eq. (2.13). $L(\mathcal{G})$ is a symmetric, positive semidefinite matrix, as revealed by the definition above.

Denoting the scalar state of the node i as $x_i \in \mathbb{R}$ and the stack form as $x = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$, then the well-known *consensus equation*, or *agreement protocol*, can be written as:

$$\dot{x}(t) = -L(\mathcal{G}) x(t), \tag{2.16}$$

where $L(\mathcal{G})$ is the graph *laplacian* given in Eq. (2.15).

2.2 Optimal Control

Optimal control theory has been developed and formulated as an extension to the calculus of variations over the past few centuries. Both being mathematical optimization methods, the principle of optimal control is readily derived from calculus of variations, which involves the minimization or maximization of certain functionals subject to constraints. Most commonly, optimal control derives the control policies for dynamic systems aiming at minimizing or maximizing certain performance criteria (cost functionals), utilizing directly some fully-fledged mathematical theories, e.g. Pontryagin's Maximum Principle [117]. A typical optimal control problem can be represented in the *Bolza form* [84], where we consider the real-valued cost functional of the form:

$$J(u) \triangleq \int_{t_0}^{t_f} L(t, x(t), u(t))\mathrm{d}t + \Psi(t_f, x(t_f)), \tag{2.17}$$

where L and Φ are given functions (so-called *instantaneous cost* and *terminal cost*, respectively), t_0 and t_f are initial and final (terminal) time, $x(t_f)$ is the final state, and $u(t)$ is the control signal. Hence, we are looking for the optimal control $u^*(t)$, which is defined as:

$$\begin{aligned} u^*(t) &= \arg\min_u J(u) && (2.18)\\ \text{subject to:}\quad \dot{x}(t) &= f(x, u), \\ x(t_0) &= x_0. \end{aligned}$$

The First Order Necessary Condition (FONC) for optimality to the *Bolza problem* mentioned above is given by the following differential equations in the Hamiltonian structure:

$$\dot{\lambda} = -\frac{\partial \mathcal{H}}{\partial x}(t, x, u, \lambda), \tag{2.19a}$$

$$\lambda(t_f) = -\frac{\partial \Psi}{\partial x}(t_f, x(t_f)), \tag{2.19b}$$

$$0 = \frac{\partial \mathcal{H}}{\partial u}(t, x, u, \lambda), \tag{2.19c}$$

for all $t_0 \leq t \leq t_f$, with $\mathcal{H} \triangleq L + \lambda^\top f$. $\mathcal{H}$ is called Hamiltonian, and λ is termed as costate, whose dynamics and boundary condition are given by Eq. (2.19a) and Eq. (2.19b), respectively.

Note that this optimal control technique produces only an open-loop solution for the time interval $[t_0, t_f]$, which means the outcome of the procedures above is a control signal $u(t)$ in terms of time t. One option to convert this open-loop signal to a closed-loop signal ($u(x)$ in terms of state x) is to perform Model Predictive Control, which is introduced in the next section.

2.2.1 Model Predictive Control

Model Predictive Control (MPC) is a well-known control technique in process control, especially for chemical industry, where the plant dynamics are often slow and difficult to characterize in time [118]. This technique, as the name indicates, uses the model of control plant to predict or simulate its behavior at future time instants (horizon), by calculating a sequence of control commands on-line based on the minimization of a certain cost function. Due to the fact that only the first control command in the sequence is actually applied to the plant at each step with the horizon for prediction moving along, this technique is also called Receding Horizon Control (RHC). In the subsequent time interval, the control command is recalculated according to the same procedure as mentioned above. In principle, MPC introduces feedback into the open-loop optimal control problem, at the cost of being computationally intensive. Usually MPC requires solving a set of differential equations with given initial and final conditions, which is known as a two-point boundary value problem [73]. In the past, this typically constrains the applicability of MPC, where only slow dynamics are allowed [40, 93]. However, with the fast development of computing technologies, it is suitable to apply the MPC in many other industrial and research areas, especially energy [15, 143] and robotics [18, 94, 142] fields. Two recent dissertations on MPC from Kanjanawanishkul [62] and Saska [125] present comprehensive overviews on the variation, development and application of MPC in many different directions, while a detailed survey of MPC theoretical works on stability and optimality issues is provided by Mayne et al. [93]. Here we briefly introduce the concept of MPC in mathematical terms.

Similar as many other optimization techniques, MPC minimizes a cost metric which characterizes the desired goal of the control problem. Take the cost from Eq. (2.17) for example, where typically $t_f = t_0 + \Delta$, Δ being a short time horizon length. The basic MPC algorithm is described in Algorithm 1.

The complexity of the MPC algorithm comes from the cross-dependency between the system state x, costate λ, and input u in the two-point boundary value problem from Step 1), where some of the differential equations depend on the initial system states, while the others depend on the terminal costs. Under the system nonlinearities, a family of numerical methods, called Direct Methods [16], such as direct shooting

Algorithm 1 Model Predictive Control

```
1) Minimize the cost (2.17) with respect to control signal $u(t)$, $\forall t \in [t_0, t_f]$. The
   Hamiltonian structure in Eq. (2.19) for deriving the first-order necessary con-
   ditions is one commonly-used approach.
2) Apply only the first control $u(t_0)$ to the system, whose dynamics are given as
   in Eq. (2.18).
3) Repeat step 1) and 2) in each of the subsequent control cycles.
```

method in Nonlinear Programming [124], are utilized to get the optimal control signal in every control step, which results in large computation time. In the next section, we present a few approaches that can be used to circumvent this problem.

2.2.2 Parameterized Model Predictive Control

Although the computational complexity of MPC applied on-line is greatly compensated by the computing hardware itself, people are still seeking remedies to reduce the burden through other sources. One typical attempt is to search for sub-optimal solutions in exchange for faster computation, or even real-time performance. Falcone et al. [38] propose a suboptimal MPC design to achieve real-time vehicle steering control. They rely on the accurate modeling of the vehicle nonlinearities, and the analysis of the constraints and performance index in the optimal control problem, so that the online linearization of the nonlinear vehicle model is valid during each optimization step.

Parameterized control is another way of compensating the heavy computation in MPC. This type of controller generates the control action as a function of a variable vector in addition to time and/or state [56]. It is applied in various applications with complex models, including parameterized gait control for snake robots [136], collision avoidance and interactive control of soccer robots [70], path following for air vehicles [107], adaptive motion planning and navigation of fast-moving mobile robots [56, 81], optimal trajectory generation for nonholonomic vehicles [68].

In parameterized MPC, feedback control law is designed for the overall control objective. During optimization step, the control law is utilized to simulate the corresponding actions forward in time, by only optimizing the parameters in the control law, which means the actual control law is tuned, instead of tuning the output of the control law as in conventional MPC [57]. Thus, solving a two-point boundary value problem becomes a parameter optimization problem, which largely reduces the computation effort. Naus et al. [103] introduce the parameterized MPC into the adaptive cruise control (ACC) in automobile industry. The parameters considered in the MPC optimization represent the key characteristics in safety, comfort and fuel economy aspects, which makes it intuitive even for the drivers to tune online. Alamir et al. [4] present two real-world implementation of parameterized Nonlinear MPC on automobiles. The nominal computation time using Control Parameterization Approach (CPA) in automated manual transmission (AMT)-problem is under 0.35 ms, with model and torque constraints under consideration. This certainly demonstrates the feasibility of real-time application of MPC in automotive industry.

2.3 Experimental Platform

In the last part of the chapter, we introduce the experimental platform used in the context of this dissertation, including the mathematical model of the mobile robot, and the basic hardware components.

2.3.1 Mathematical Model

The type of vehicle we consider throughout the dissertation is a standard rear-wheel drive, front-wheel steerable (Ackermann Steering) car-like mobile robot, whose kinematic equation is given by:

$$f_q(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) \cdot v(t) \\ \sin\theta(t) \cdot v(t) \\ \dfrac{[\tan\beta(t)] \cdot v(t)}{l} \end{bmatrix}, \tag{2.20}$$

where v is the translational driving speed; β is the equivalent steering angle of the front wheels (see Figure 2.1); (x, y) is the position of the robot in global frame, located at the mid-point of the rear-wheel axle, and θ is termed as the orientation, which is the angular difference between the global frame and mobile robot local frame; l is the distance between front and rear wheel axle. The Instantaneous Center of Curvature (ICC) lies on the line of rear-wheel axle. The turning radius $r = \dfrac{l}{\tan\beta}$ if $\beta \neq 0$. Note that the first two scalar equations represent the nonholonomic constraints of the car-like mobile robot, which indicates that no sideward movement of the vehicle is allowed due to wheel slippage and rolling constraints. (Refer to [54] for more details on kinematic models of wheeled mobile robots.)

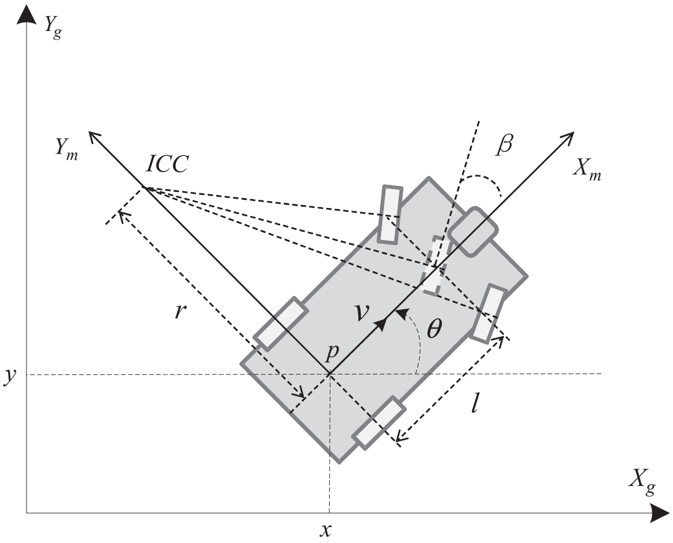


Figure 2.1: Car-like mobile robot model

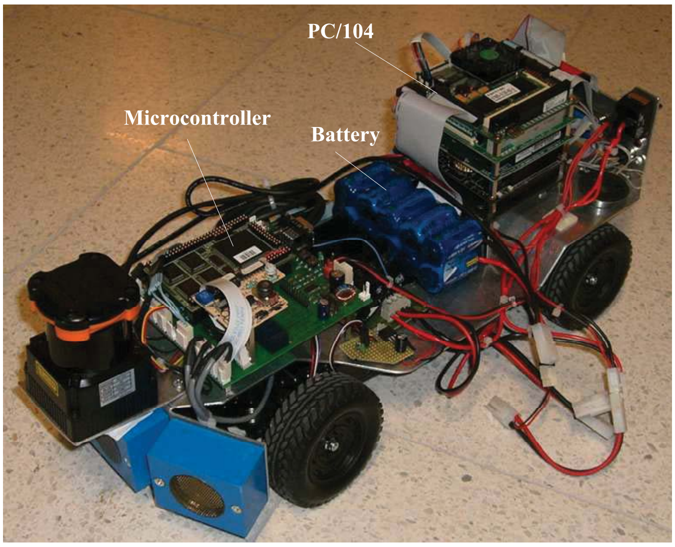


Figure 2.2: MERLIN Outlook

2.3.2 Hardware

The experiments to validate the developed techniques in this dissertation are performed on a group of car-like mobile robots, MERLIN (Mobile Experimental Robot for Localization and Intelligent Navigation), as shown in Figure 2.2. The 16-bit microcontroller C167CR-LM, running an in-house developed operating system, is used onboard mainly for interfacing with the drive system, i.e. the brushless DC motor for driving and the servo module for steering. Various functionalities to acquire and process sensor data are also integrated in the operating system, with a sampling rate

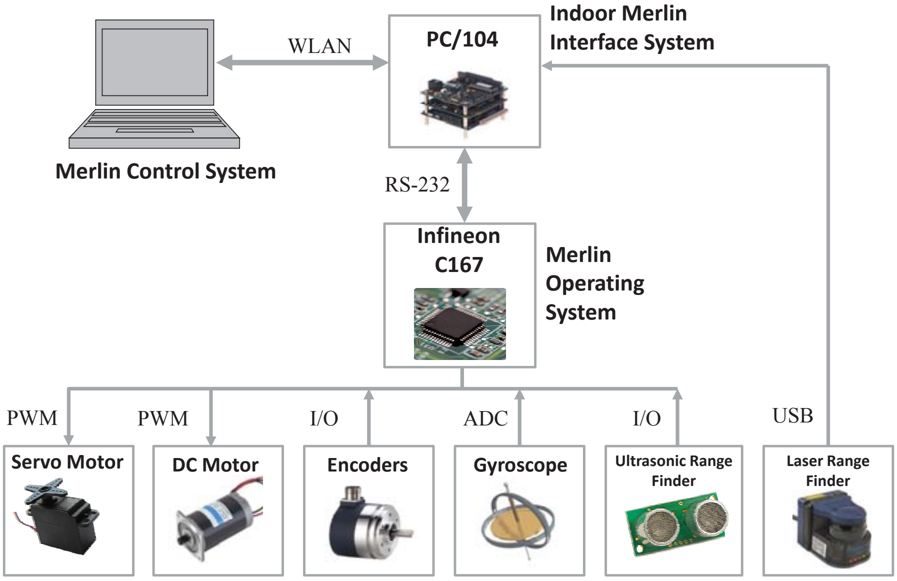


Figure 2.3: Overview of the hardware architecture on MERLIN

of 10Hz. Some low-level controls to ensure the safety and smooth driving are implemented in the operating system, such as the emergency stop in case the robot loses data connection during teleoperation, steering correction to compensate for the imperfection in mechanical mounting based on gyroscope, etc.

Figure 2.3 shows the various hardware components onboard MERLIN and their connection interfaces. The embedded PC (PC/104) platform is equipped for communicating with the control station running Merlin Control System (MCS), and the other MERLIN robots via wireless network. It also executes the developed control algorithms in the Indoor Merlin Interface System (IMIS). The data acquired from the microcontroller, running Merlin Operating System (MOS), is transmitted through serial port to PC/104, and the PC/104 is in charge of broadcasting all the necessary information as UDP packets in the local Ad-Hoc network.

It should be noted that, the odometry of the vehicles relies on the dead reckoning principle based on sensor readings of the wheel encoders and onboard gyroscope. During the experiments, in order to have a unified coordinate system, each vehicle is started in one known position, such that the odometry data is qualified for localization without external positioning systems[1].

[1] Since the dead reckoning principle suffers from long-term accumulated errors, we carry out most experiments in a short period of time, as well as in a stable indoor environment.

CHAPTER 3

TIME-DELAYED FORMATION CONTROLLER

In this chapter, we focus on the parameter design for a class of formation control laws under communication delays. By constructing *Leader-Follower* framework, which enables the formation system to scale and be controlled in a decentralized fashion, we aim to automate the follower robots to track the leader robot with desired relative distances and orientations. Taking into account the time delay in data transmission between the robots, we formulate both stable parametric regions for the formation control law and its corresponding exponential convergence property. We illustrate both the advantages of the designed controller in comparative simulation results, and the applications of the control scheme in real-world robotic vehicle tracking scenarios.

In the following sections, we first present a brief review of the research work on formation control problem under time delays (Section 3.1). Secondly, we derive our feedback control law for both time-delay and delay-free cases. (Sections 3.2 to 3.4). Then the controller parameter selection criteria are analyzed (Section 3.5), followed by the proof on system convergence and internal dynamics (Section 3.6). The comparative simulation and experimental studies are shown afterward (Section 3.7). Finally we provide a short summary along with several remarks to conclude this chapter (Section 3.8).

3.1 Problem Formulation and Related Work

In this section, the problem of time-delayed formation control is presented. We also provide an overview of the closely related work that address this problem from different aspects than our proposed approach.

3.1.1 Problem Description

Consider the the scenario of teleoperating a robot formation depicted in Figure 3.1, where the local and remote sites can be far from each other, resulting in a certain amount of transmission delay in the communication channel. On the remote site, we assign one robot as the leader, and the rest as followers. We require the follower robots to communicate only with the leader, where the communication is also affected by time delays. The control task we focus on in this chapter is to design the stable

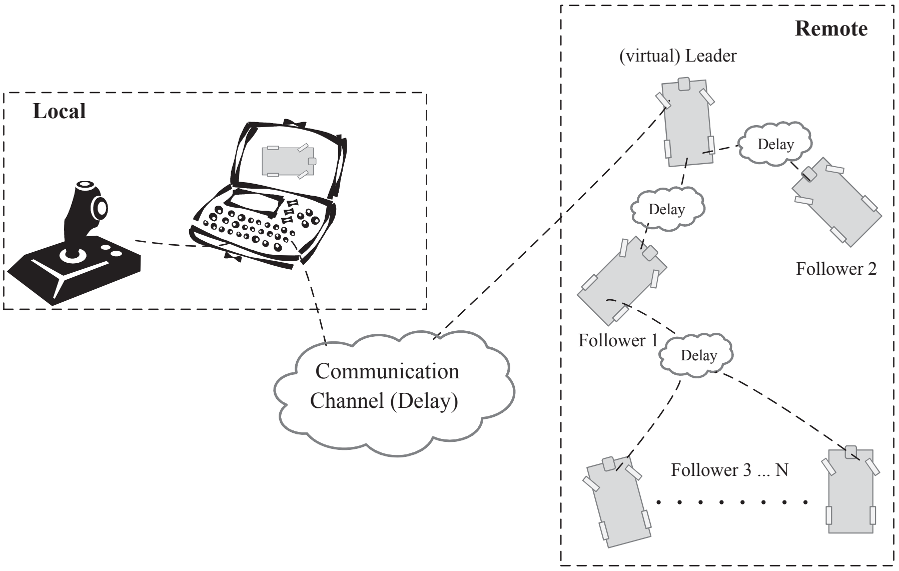


Figure 3.1: Illustration of formation teleoperation with communication delays in various links

decentralized controller on the follower robots for them to track the leader, where we teleoperate the leader robot from our local control station. We address mainly two issues in nonlinear tracking systems with time-delay effect, namely the tracking steady-error and the oscillation problem.

There are a few related theoretical details to this setup, which we state in the following:

- The scalability of the formation under leader-follower framework is realized by considering a set of leader-follower robots as an interconnected system, where multiple interconnected systems are to be coupled with each other. In principle, this forms a tree hierarchy, where the follower robot can be treated as the leader robot for the subsequent followers. Refer to Stipanović et al. [131] for detailed technicalities on expanding interconnected systems.

- As indicated in Figure 3.1, the leader can be non-physical robot, which we call *virtual leader*, in which case the status of the leader is simulated on the local site and transmitted to the first follower robots in the formation hierarchy. One of the earliest studies on applying virtual leader concept in robotic control is from Egerstedt et al. [33], where the path tracking control is designed using virtual vehicle for a car-like robot. A recent dissertation from Saska [125] covers the path planning algorithms for virtual leader to guide the robot formation to the desired destination.

- Among the information flows from the local operation site to the remote leader robot, as well as to the followers, there exist other time-delay sources than the communication channel that affect system performance, for instance, the sensor-to-controller delay and controller-to-actuator delay. However, under the assumption of time-invariant control laws, these different causes of delays can be combined into one single delay T for analysis purposes. In more practical cases, T can also be considered as the upper bound of the time-varying delays, or the piece-wise constant delays in digital data transmission [156].

3.1.2 Related Work

In order to cope with the time-delay effect, numerous control strategies have been developed. The surveys mentioned in Section 2.1.2 cover the recent progress in this direction from the past decade. In this section, we provide a brief overview on these strategies, and point out some interesting research work that are closely related to our approach in two aspects, namely formation control and time-delay compensation.

3.1.2.1 Nonlinear formation control

For the first aspect in terms of the control of nonlinear systems, many design tools can be considered, among which two popular ones are Lyapunov-based analysis and feedback linearization method [69].

Mastellone et al. [92] develop a decentralized feedback control law based on Lyapunov analysis, which achieves both collision avoidance and trajectory tracking for a formation of differential-drive robots using leader-follower structure. The overall design using Lyapunov analysis naturally integrates the avoidance function with the tracking control, and the stability of internal dynamics is also guaranteed, since the complete error dynamics of the formation can be included in the Lyapunov-like function candidate, which is proved to be non-increasing. This work also presents sufficiently extensive experiments, which show the robustness of the controller against communication imperfections, such as delays, quantization, and packet losses. Panimadai Ramaswamy and Balakrishnan [112] extend the Lyapunov-like analysis to the car-like robot formation, analyzing the formation error dynamics and applying the control through car-like dynamic model using an online neural network. Some other research work on formation control using Lyapunov analysis, especially for nonholonomic systems, are [19, 82, 120], to name just a few. However, the analysis on the delay effect within Lyapunov framework is absent in most literature, which otherwise would require Lyapunov-Krasovskii functionals [44] that largely complicate the control design and stability proof. One exemplary study using Lyapunov-Krasovskii functionals to analyze the stability of general time-delay nonlinear systems is provided by [116].

On the other hand, Input-Output feedback linearization (see Section 2.1.1), one of the state feedback (model-based) control laws, is another typical approach in the control of nonlinear systems, especially nonholonomic systems [3]. Eghtesad and Necsulescu [34] study the internal dynamics of a three-wheel mobile robot with nonholonomic constraints in addition to the feedback linearization control on the system external dynamics, and they point out the conditions under which the internal dynamics are stable. The similar procedure as in [34] is performed in our analysis on time-delayed formation control. One variation of input-output feedback linearization, dynamic feedback linearization method, is also successfully applied to the trajectory tracking control of nonholonomic mobile robots. The difference between these two methods is that dynamic feedback linearization gives full-state linearized equations by defining new intermediate control inputs. It is suitable for handling high-dimensional systems, while restrained from its structural singularity in the dynamic extension process (see Luca et al. [88]). Yang et al. [153] design a singularity-free formation model in order to apply dynamic feedback linearization. They choose the robot coordinate to be located on the steerable castor wheel, instead of the driving wheel, as most other work have done, to avoid the structural singularity. The stability of internal dynamics is also provided therein. Gamage [46] emphasize on the use of dynamic feedback linearization as low-level behavior-based controller, and develop a high-level discrete event system to interact with the environment. The structural singularity restricts the robots to move without stopping. More applications of feedback linearization can be found in [63, 86, 90, 99, 128].

3.1.2.2 Time-delayed formation tracking

The second aspect we are interested in is the time-delay effect in formation control. Most of the existing work dealing with time-delay problem of controlling multiple robots focus on the communication channel between the user and remote site in the bilateral teleoperation setup [42, 78, 79]. Palafox and Spong [110, 111] consider the time delays in the communication link, as well as the delays on both local and remote sites. The focus is on the asymptotical stability of the delayed bilateral teleoperation system. Nevertheless, the authors are comparing the master position with the delayed slave positions, which means the steady-state error problem on the remote formation driving is not handled. This issue corresponds to the steady-state error of the formation tracking for the followers in leader-follower framework, as also neglected by Goi et al. [49], where the delayed leader position is directly set as the tracking object for the immediate follower on purpose. Therefore, the tracking error is between the current follower's position and the delayed leader's position, which, on the other hand, facilitates the autonomous convoying as pointed out in their paper, since the leader's states between the current and the delayed moments are used to estimate its speed and heading. This is a feasible solution, since the objective of convoying mission is path following, which differs from the control objective of this work. The steady-state error in the separation distance from [49] still exists due to the image processing delay. The solution to compensate for the steady-state error is elaborately addressed in Section 3.4 in this dissertation.

3.1.2.3 Various formation control approaches

In addition, as a horizontal overview, there are quite a few new techniques developed for formation control that are often extended to deal with communication delays. For instance, consensus-based control (Ding et al. [25], Liu et al. [87], Olfati-Saber and Murray [109], Papachristodoulou et al. [113]), which mainly applies to linear systems or nonlinear systems that are either feedback linearizable or locally passive; event-based control (Chen et al. [20]) and non-time based control (Jia and Xi [61]), which dispose the time-dependency in the system dynamics, thus making it effective in dealing with time-delayed systems.

3.2 System Modeling

In order to solve the problem mentioned in Section 3.1.1, we need to derive the analytical model for the formation system. In this section, we introduce the kinematic model for a formation of car-like mobile robots under leader-follower scheme.

In the leader-follower formation system, the control objective is for the followers to maintain constant distances and relative bearing angles with respect to the leader. Following Section 2.3.1 and Figure 3.2, we denote $q \triangleq [x \quad y \quad \theta]^\top$ as the vehicle state, and $u \triangleq [v \quad \beta]^\top$ as the control inputs. Define the system output $z \triangleq [\rho, \alpha, \phi]^\top$, and system inputs [1] $u_L \triangleq [v_L, \beta_L]^\top$ and $u_F \triangleq [v_F, \beta_F]^\top$. The output z is calculated based

[1] subscript L, F stand for leader and follower, respectively.

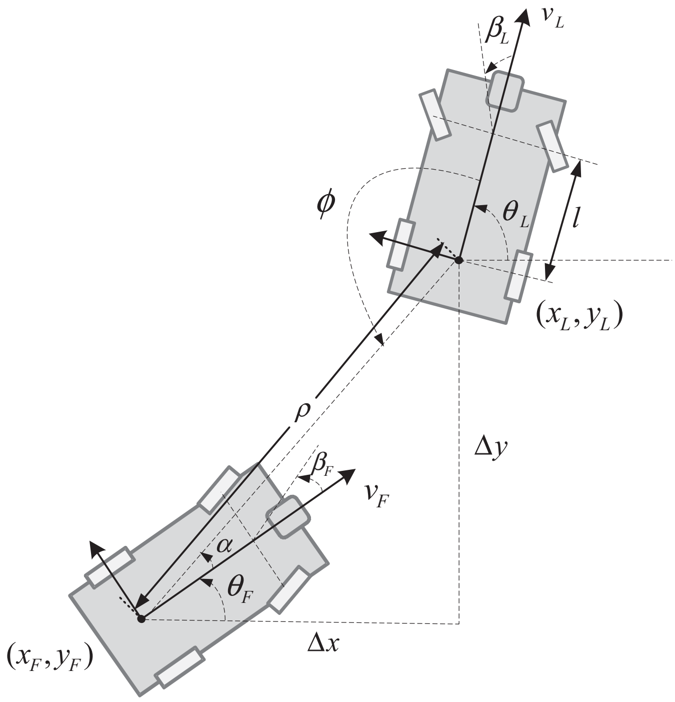


Figure 3.2: Model of leader-follower formation for car-like vehicles; leader robot is either teleoperated or driven autonomously with predefined commands, while the follower robot is driving autonomously based on the designed controller.

on the geometry of the leader-follower formation as follows:[2]

$$\begin{cases} \rho = \sqrt{(x_L - x_F)^2 + (y_L - y_F)^2} \\ \alpha = \text{atan2}\left(y_L - y_F, x_L - x_F\right) - \theta_F \\ \phi = \alpha + \theta_F - \theta_L + \pi \end{cases} \tag{3.1}$$

where ρ is the distance between the mid-points of the rear-wheel axle of the leader and follower vehicles; α is the angle from the line of follower's orientation to the line

[2] The function atan2$(\cdot,\cdot)$ denotes the arctangent of two arguments, which is defined as:

$$\text{if } y \neq 0: \quad \text{atan2}(y,x) = \begin{cases} \arctan(|y/x|)\cdot\text{sgn}(y) & x > 0 \\ \pi/2\cdot\text{sgn}(y) & x = 0 \\ (\pi - \arctan(|y/x|))\cdot\text{sgn}(y) & x < 0 \end{cases},$$

$$\text{if } y = 0: \quad \text{atan2}(y,x) = \begin{cases} 0 & x \geq 0 \\ \pi & x < 0 \end{cases}.$$

where sgn$(\cdot)$ is the sign function.

of ρ counter-clockwise; and ϕ is the angle from the line of the leader's orientation to the line of ρ counter-clockwise.

By differentiating Eq. (3.1) with respect to time and after some equation manipulations [104], the kinematics of the leader-follower formation system is given as:

$$\begin{cases} \dot{\rho} = -v_F \cdot \cos\alpha - v_L \cdot \cos\phi \\ \dot{\alpha} = \dfrac{v_F \cdot \sin\alpha}{\rho} + \dfrac{v_L \cdot \sin\phi}{\rho} - \dfrac{v_F \cdot \tan\beta_F}{l} \\ \dot{\phi} = \dfrac{v_F \cdot \sin\alpha}{\rho} + \dfrac{v_L \cdot \sin\phi}{\rho} - \dfrac{v_L \cdot \tan\beta_L}{l} \end{cases} \tag{3.2}$$

where u_L is considered as an exogenous input and u_F is the endogenous control input for the formation system.

Note that a fixed system output z defines a unique formation shape. However, the system given in Eq. (3.2) shows that it has two independent control inputs but three states, which results in an underactuated system. The controller design in the next section will cover this issue.

3.3 Formation Feedback Control

In this section, the controller for achieving stable leader-follower formation is developed under the formation kinematics given in Eq. (3.2). Meanwhile, the stability issue of the controller with respect to the internal dynamics of the formation is also addressed.

3.3.1 Input-Output Feedback Linearization

Following Section 2.1.1, we design the formation controller based on input-output feedback linearization method. The control objective in the leader-follower formation system is to keep desired relative distances between the leader and follower vehicles, as well as to maintain desired relative heading directions. Therefore, we select the control output to be ρ and α, so that the shape of the formation from the follower's perspective is maintained once these two variables are kept constant.

Let $z_d \triangleq [\rho_d, \alpha_d]$ be the desired control output, where ρ_d is the desired distance between two vehicles, and α_d is the desired relative bearing angle. Hence, the desired linearized system is given by:

$$\begin{cases} \dot{\rho} = k_\rho \cdot (\rho_d - \rho) \\ \dot{\alpha} = k_\alpha \cdot (\alpha_d - \alpha), \end{cases} \tag{3.3}$$

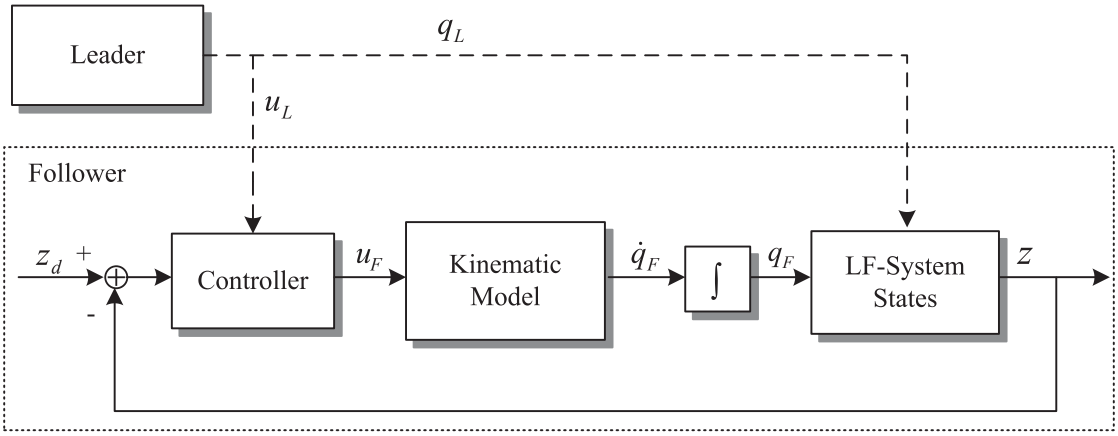


Figure 3.3: Formation control scheme under leader-follower approach; the controller block refers to Eq. (3.3) and Eq. (3.4); the kinematics model is described in Eq. (2.20); and the LF-system states are from Eq. (3.2).

where $k_\rho > 0$, $k_\alpha > 0$ are tuning parameters. Based on input-output feedback linearization, the control inputs for the follower vehicle are so derived as to eliminate the nonlinear terms of the system in Eq. (3.2) to achieve the linearized model given in Eq. (3.3), thus resulting in the following form:

$$\begin{cases} v_F = -\dfrac{\dot{\rho} + v_L \cos\phi}{\cos\alpha} \\ \beta_F = \operatorname{atan}\left[l \left(\dfrac{\sin\alpha}{\rho} + \dfrac{\dot{\alpha}\cos\alpha}{\dot{\rho} + v_L\cos\phi} - \dfrac{v_L \sin\varphi \cos\alpha}{\rho\,(\dot{\rho} + v_L \cos\phi)} \right) \right] \end{cases} \tag{3.4}$$

The overall leader-follower formation system is shown in Figure 3.3, where q_L and q_F are the states of the leader and follower, respectively. Note that the velocity and steering angle inputs for the robot are limited in the implementation according to the following condition:

$$|v| \leq v_{max}, \quad |\beta| \leq \beta_{max} \tag{3.5}$$

where $v_{max} > 0$ and $\beta_{max} > 0$ are the maximum allowable speed and steering inputs due to mechanics of the robot.

3.3.2 Stability and Internal Dynamics

As indicated in Section 3.1, the stability problem in the controlled formation system is analyzed through the convergence of internal dynamics of the system.

We first define the system error $e \triangleq [e_1, e_2, e_3]^\top = [\rho - \rho_d, \alpha - \alpha_d, \phi - \phi_d]^\top$, which represents the tracking error of the follower vehicle[3]. The error dynamics is simply

[3] Note that ϕ_d can be any constant value, considering we have no control over the output ϕ. We

the time derivative of the system error, given as:

$$\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -k_\rho e_1 \\ -k_\alpha e_2 \\ \dot{\phi} \end{bmatrix}. \tag{3.6}$$

Using the controller given in Eq. (3.4), it is trivial to prove that the linearized system in Eq. (3.3) converges to zero exponentially in terms of e_1 and e_2, which means $\rho \to \rho_d$ and $\alpha \to \alpha_d$ as $t \to \infty$ (refer to [24] for the proof).

However, due to the fact that the system given in Eq. (3.2) is under-actuated, the third system state ϕ is left in open loop and it is thus referred as the internal dynamics of the system. In the following, we derive the stability condition of the internal dynamics of the formation system, which is equivalent to showing the convergence of error e_3 to a constant value as $t \to \infty$. Note that the internal dynamics here is analyzed in the formation system for the case of a moving leader. A variation of the internal dynamics analysis can be found in Eghtesad and Necsulescu [34], where the stabilization of a single mobile robot is considered.

The internal dynamics are analyzed by looking at the zero dynamics of the system (as explained in Section 2.1.1, which can be obtained by setting ρ, α and their derivatives in Eq. (3.2) to zeros [58, 139]), to which the solution is:

$$\dot{e}_3 = A \cdot \sin\left(e_3 - \alpha_d\right) - B, \tag{3.7}$$

where $A = \dfrac{v_L}{\rho_d \cos \alpha_d}$, and $B = \dfrac{v_L \cdot \tan \beta_L}{l}$. B represents the angular velocity of the leader robot, which means $0 \leq |B| \leq B_{max}$ with B_{max} being a positive upper bound. Also, under the assumption that $-\frac{\pi}{2} < \alpha_d < \frac{\pi}{2}, v_L \neq 0$, we have $0 < |A| \leq A_{max}$ with A_{max} being a positive upper bound.

Solving this equation by using *dsolve* function in MATLAB® gives us:

$$e_3 = \begin{cases} \alpha_d + 2 \cdot \operatorname{atan}\left(\dfrac{A - \sqrt{B^2 - A^2} \tan\left(\frac{\sqrt{B^2-A^2}(c+t)}{2}\right)}{B} \right) & \text{if } B \neq 0 \text{ and } B \neq A \\ \alpha_d + 2 \cdot \operatorname{atan}\left(\dfrac{2}{c + A \cdot t} + 1 \right) & \text{if } B \neq 0 \text{ and } B = A \\ \alpha_d + 2 \cdot \operatorname{atan}\left(e^{c + A \cdot t}\right) & \text{if } B = 0 \end{cases} \tag{3.8}$$

where c is a constant depending on the initial value of this error e_3. This solution is further divided into different cases:

denote this variable here only for completeness.

1. $B = 0$: This means the leader is moving along a straight line, either forward or backward, depending on the sign of A.

   (a) $A > 0$: The leader is moving forward. The error $\lim_{t\to\infty} e_3 = \alpha_d + \pi$. This proves that the internal dynamics is stable. From Eq. (3.1), we know that the orientation of the follower will be equivalent to the orientation of the leader, as $t \to \infty$.

   (b) $A < 0$: The leader is moving backward, $v_L < 0$. And the error $\lim_{t\to\infty} e_3 = \alpha_d$. This also proves the internal dynamics is stable. In this case, there is an orientation difference of π between the leader and follower based on our definitions from Figure 3.2.

2. $B \neq 0$: the problem is further divided into the following three subcases:

   (a) if $|B| < |A|$: Eq. (3.8) becomes:

$$e_3 = \alpha_d + 2 \cdot \operatorname{atan}\left(\frac{A - i\sqrt{A^2 - B^2}\tan(\eta(t)i)}{B}\right),$$

   where $\eta(t) = \dfrac{\sqrt{A^2 - B^2}(c + t)}{2}$. Recall that $\tan\theta = \dfrac{e^{-i\theta} - e^{i\theta}}{e^{-i\theta} + e^{i\theta}}i$, the equation above becomes:

$$e_3 = \alpha_d + 2 \cdot \operatorname{atan}\left(\frac{A + \sqrt{A^2 - B^2}\frac{e^{\eta} - e^{-\eta}}{e^{\eta} + e^{-\eta}}}{B}\right).$$

   Since $\lim_{t\to\infty} \dfrac{e^{\eta} - e^{-\eta}}{e^{\eta} + e^{-\eta}} = 1$, it can be concluded that the error e_3 will converge as $t \to \infty$: $\lim_{t\to\infty} e_3 = \alpha_d + 2 \cdot \operatorname{atan}\left(\dfrac{A + \sqrt{A^2 - B^2}}{B}\right)$.

   (b) if $|B| = |A|$: the error e_3 converges to $\lim_{t\to\infty} e_3 = \alpha_d + \operatorname{sgn}(A) \cdot \dfrac{\pi}{2}$, which also means stability is ensured on the internal dynamics.

   (c) if $|B| > |A|$: Eq. (3.8) is not converging to a constant value, which gives no guarantee on system stability.

Remarks:

a) In the case where $|B| < |A|$, and $|B| \neq 0$, i.e. $|\dfrac{v_L \cdot \tan\beta_L}{l}| < |\dfrac{v_L}{\rho_d \cos\alpha_d}|$, the subsequent requirement on the leader robot is:

$$|\beta_L| < \operatorname{atan}\frac{l}{\rho_d|\cos\alpha_d|}.$$

b) In order to avoid $A \to \infty$, the following correction will be made in the implementation.

$$\cos(\alpha) = \begin{cases} \epsilon & \text{if } \alpha = \pm\frac{\pi}{2}, \\ \cos(\alpha) & \text{otherwise.} \end{cases}$$

where ϵ is a positive infinitesimal. Obviously this correction does not affect the system behavior. Meanwhile it also applies to the control inputs in Eq. (3.4), which are further limited by Eq. (3.5).

3.4 Time-delayed Formation System

As mentioned in Section 3.1, we need to improve the controller from Eq. (3.4) after taking into account the assumed constant time delay T of the information flow between the leader and the follower robots. In this section, we extend from Eq. (3.3) and Eq. (3.4) to two types of controllers for this purpose. Meanwhile, the stability analyses on these two controllers are also provided.

3.4.1 P-type Controller for Delay Compensation

Following the input-output feedback linearization control scheme discussed in Section 3.3.1, the time delay of data transmission between the leader and follower robots is considered in this section. The resulting steady-state error problem and its solution are illustrated. A P-type controller (similar to Eq. (3.3)) and an improved PD-type controller to compensate for the delay effects are proposed and analyzed.

3.4.1.1 Steady-state error

We first consider the time delay T of the information exchange between the leader and the follower robot. Accordingly, the system output $\tilde{z}^{\mathrm{d}}$ becomes[4]:

$$\tilde{z}^{\mathrm{d}} = \begin{bmatrix} \tilde{\rho}^{\mathrm{d}} \\ \tilde{\alpha}^{\mathrm{d}} \\ \tilde{\phi}^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \sqrt{({x_L}^{\mathrm{d}} - x_F)^2 + ({y_L}^{\mathrm{d}} - y_F)^2} \\ \operatorname{atan2}\left({y_L}^{\mathrm{d}} - y_F, {x_L}^{\mathrm{d}} - x_F\right) - \theta_F \\ \alpha^{\mathrm{d}} + \theta_F - {\theta_L}^{\mathrm{d}} + \pi \end{bmatrix}, \tag{3.9}$$

where the follower states at the current time instance t is compared with the delayed leader states at time $t - T$. Apparently, this will cause steady-state error in tracking, because follower's motion is only regulated to keep up with the delayed leader's motion. This is illustrated in Figure 3.4. A direct impact from this delayed tracking is

[4] $\mathit{variable}^{\mathrm{d}} \triangleq \mathit{variable}(t - T)$, where T is the constant time delay as introduced in Section 3.1, which can be achieved by using an appropriate network protocol.

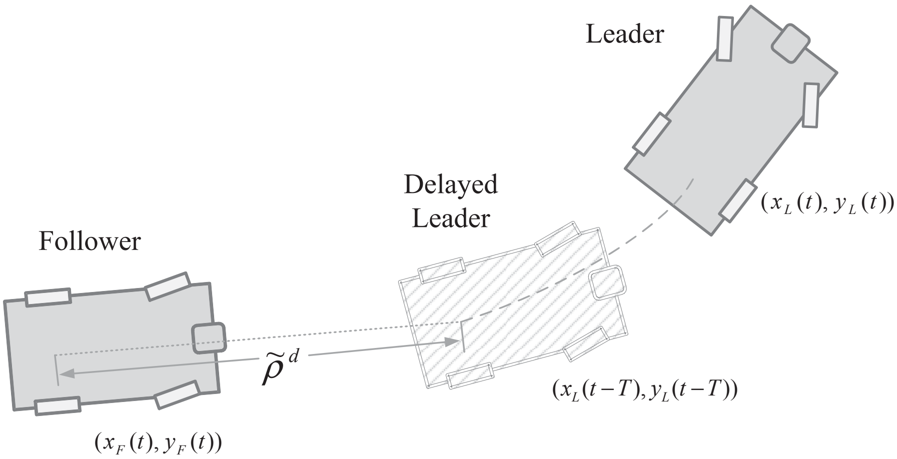


Figure 3.4: Follower tracks the delayed leader. At the time when the follower receives the leader information, the actual leader is T time ahead. Thus the follower is only tracking the delayed version of the leader. $\tilde{\rho}^{\mathrm{d}}$ is defined in Eq. (3.9).

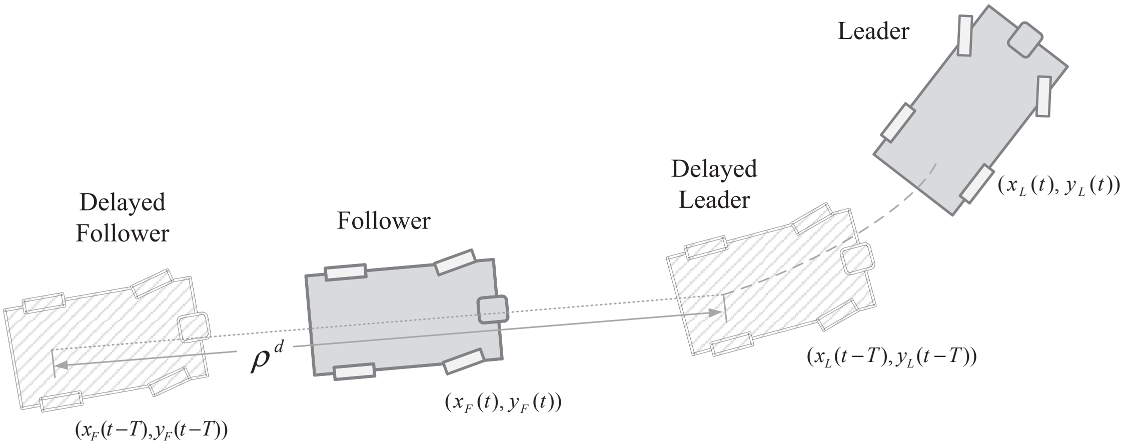


Figure 3.5: Follower tracks the delayed leader using its own delayed version. ρ^{d} is defined in Eq. (3.10).

that a desired formation shape cannot be maintained. Moreover, the tracking error increases with the increase of time delay in the communication. This problem of steady-state error is visualized through simulations in Section 3.7.1.

3.4.1.2 Delay-compensation scheme

One way to compensate for the steady-state error due to the delay effect is to modify the computation of system output from Eq. (3.9) by using the follower states at time

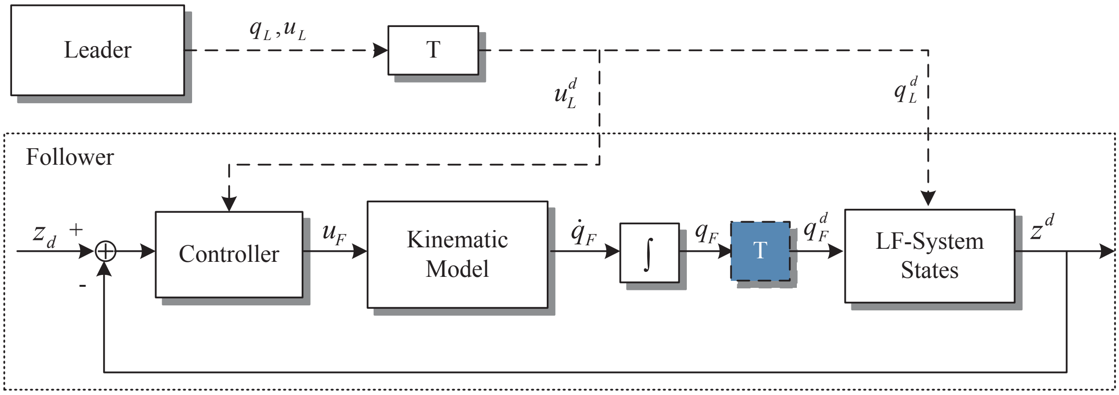


Figure 3.6: Formation control scheme of time-delayed system. The extra delay component T in blue is added intentionally into the system.

$t - T$ instead of at time t, as illustrated in Figure 3.5. Therefore, we obtain:

$$z^{\mathrm{d}} = \begin{bmatrix} \rho^{\mathrm{d}} \\ \alpha^{\mathrm{d}} \\ \phi^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \sqrt{({x_L}^{\mathrm{d}} - {x_F}^{\mathrm{d}})^2 + ({y_L}^{\mathrm{d}} - {y_F}^{\mathrm{d}})^2} \\ \mathrm{atan2}\left({y_L}^{\mathrm{d}} - {y_F}^{\mathrm{d}}, {x_L}^{\mathrm{d}} - {x_F}^{\mathrm{d}}\right) - {\theta_F}^{\mathrm{d}} \\ \alpha^{\mathrm{d}} + {\theta_F}^{\mathrm{d}} - {\theta_L}^{\mathrm{d}} + \pi \end{bmatrix}. \tag{3.10}$$

Now this delayed output can be used in the same controller designed in Section 3.3.1, where the linearized closed-loop delay system takes the form:

$$\begin{cases} \dot{\rho} = k_\rho \cdot \left(\rho_d - \rho^{\mathrm{d}}\right) \\ \dot{\alpha} = k_\alpha \cdot \left(\alpha_d - \alpha^{\mathrm{d}}\right) \end{cases} \tag{3.11}$$

and the control inputs for the follower vehicle are given similar to Eq. (3.4) as follows:

$$\begin{cases} v_F = -\dfrac{\dot{\rho} + {v_L}^{\mathrm{d}} \cos\phi^{\mathrm{d}}}{\cos\alpha^{\mathrm{d}}} \\ \beta_F = \mathrm{atan}\left[l\left(\dfrac{\sin\alpha^{\mathrm{d}}}{\rho^{\mathrm{d}}} + \dfrac{\dot{\alpha}\cos\alpha^{\mathrm{d}}}{\dot{\rho} + {v_L}^{\mathrm{d}}\cos\phi^{\mathrm{d}}} - \dfrac{{v_L}^{\mathrm{d}}\sin\phi^{\mathrm{d}}\cos\alpha^{\mathrm{d}}}{\rho^{\mathrm{d}}\left(\dot{\rho} + {v_L}^{\mathrm{d}}\cos\phi^{\mathrm{d}}\right)}\right)\right] \end{cases} \tag{3.12}$$

where $\dot{\rho}$ and $\dot{\alpha}$ are from Eq. (3.11). The overall leader-follower formation system is shown in Figure 3.6. The only change of the control scheme on the delayed system from the one shown in Figure 3.3 is that the states of the follower robot will be stored for a time period of T, and once the delayed leader states arrive at the follower, the stored data will be used and the current states of the followers will be stored.

3.4.1.3 Stability Analysis on P-type Controller

The stability of the linearized closed-loop delayed system described in Eq. (3.11) is investigated in this section. The delayed differential equations can be rewritten for simplicity in the form (with a slight abuse of notation on x):

$$\dot{x}(t) = -kx(t - T). \tag{3.13}$$

For instance, let $x(t) \triangleq \rho_d - \rho(t)$, and $k \triangleq k_\rho \neq 0$. To determine the stability of the system, we follow the theorem from Sipahi et al. [126] to analyze the transcendental characteristic equation of Eq. (3.13):

$$f_{k,T}(s) \triangleq s + k \cdot e^{-Ts} = 0. \tag{3.14}$$

Theorem 3.1 (Stability condition for P-type controller [126])**.** *System defined in Eq.* (3.13) *is asymptotically stable if the parameters k and T satisfy the conditions given in Eq.* (3.15)*, i.e.*

$$0 < kT < \frac{\pi}{2}, \quad k > 0. \tag{3.15}$$

Proof. We first derive the stability condition on the parameter k, and then show in detail for the condition on T.

1. Suppose $k < 0$ in Eq. (3.13), and we can define two curves: $y_1(s) := s; y_2(s) := -ke^{-Ts}$ with $k < 0$. It is obvious that $y_1(0) < y_2(0)$ and $\lim_{s\to\infty} y_1(s) > \lim_{s\to\infty} y_2(s)$. Consequently, there must be at least one intersection point of these two curves on the right-hand side of the plane. Therefore, $k > 0$ is a necessary condition for the stability of the system.

2. When $T \neq 0$, because of the periodicity in the exponential term, there exists an infinite number of roots in the complex $\mathbb{C}$ plane. These are called characteristic roots. In order to find out whether the real part of the right-most root is in the $\mathbb{C}^-$ plane, we will analyze Eq. (3.14) on the imaginary axis. Let $s = ib$, and separate real and imaginary parts of Eq. (3.14) as follows:

$$\begin{cases} \cos(bT) = 0, \\ k\sin(bT) = b. \end{cases} \tag{3.16}$$

   The result gives the infinite number of delays as $T_{c,l} = \dfrac{\pi}{2k} + \dfrac{2\pi l}{b_c}, l = 0, 1, 2, \ldots,$ all of which yield the crossing frequency $b_c = k$. By continuity, the asymptotic stability of the system given in Eq. (3.13) is guaranteed for all delays satisfying $T \in (0, T_c)$, where $T_c = \dfrac{\pi}{2k}$. Equivalently, this gives the first condition in Eq. (3.15), i.e. $0 < kT < \dfrac{\pi}{2}$.

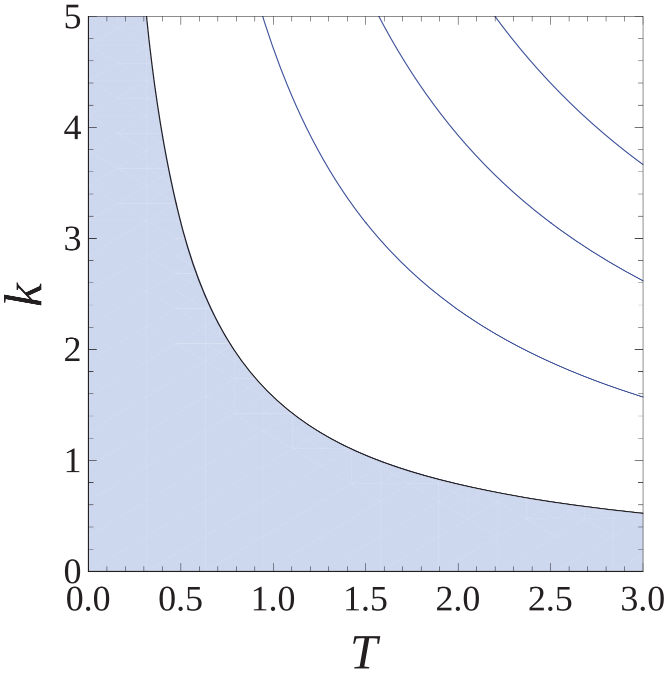


Figure 3.7: Stability chart of system (3.13); the shaded area indicates the stable region where all the roots are located in the $\mathbb{C}^-$ plane; the curves are described in Eq. (3.17).

3. Eq. (3.15) is depicted in Figure 3.7, which is referred as the stability chart. The curves in the figure can be described in the form:

$$C_n = \{(T, k) = (b, \frac{\pi/2 + m\pi}{b}), n\pi < b < (n+1)\pi\}, \tag{3.17}$$

   where $m, n \in \mathbb{N}_0$. Further investigation on the quantity $\Re\left\{\frac{ds}{dT}|_{s=ib}\right\}$ gives:

$$\Re\left\{\frac{ds}{dT}|_{s=ib}\right\} = \frac{b^2}{(kT)^2 + \sin^2(bT)} > 0. \tag{3.18}$$

   Therefore the increase in delay T beyond each curve C_n will bring the roots on imaginary axis to the $\mathbb{C}^+$ plane, and the system will become unstable.

This completes the proof. □

Remark: Even if the condition (3.15) is satisfied, the dynamic response of the system (3.13) can be either slow or oscillating, because its dominant root can be quite close to the imaginary axis in the s-plane (See Figure 3.10, where $d = 0$). Our desired goal in the next step is to improve the dynamic response by placing the dominant roots (rightmost roots) in $\mathbb{C}^-$ plane far from the imaginary axis with an extra tuning parameter.

3.4.2 Improved PD-type Controller

In the previous controller, the dynamic response of the stable system still has large oscillations or low convergence rate due to the location of the dominant root with respect to the imaginary axis. Therefore, in this section, an extra derivative term is to be designed in the controller to compensate for this oscillating behavior. The overall system with PD-type controller is given as follows:

$$\begin{cases} \dot{\rho}(t) = k_\rho \cdot (\rho_d - \rho(t-T)) + d_\rho \cdot \dot{\rho}(t-T) \\ \dot{\alpha}(t) = k_\alpha \cdot (\alpha_d - \alpha(t-T)) + d_\alpha \cdot \dot{\alpha}(t-T), \end{cases} \tag{3.19}$$

where d_ρ, d_α are parameters to be designed. These two equations can be rewritten in a generalized delay-differential equation form as:

$$\dot{x}(t) = -kx(t-T) + d\dot{x}(t-T), \tag{3.20}$$

where delay $T > 0$. Before we present the stability theorem for this delayed system, some simplifications need to be introduced for the proof of this theorem.

The characteristic equation corresponding to Eq. (3.20) is a quasi-polynomial of the form:

$$h_{T,k,d}(s) \triangleq s + ke^{-sT} - dse^{-sT} = 0. \tag{3.21}$$

Without loss of generality, through change of variable $s \mapsto sT$ and $k \mapsto kT$ in Eq. (3.21) (refer to [13]), the simplified characteristic equation is described by the function:

$$g_{k,d} : \mathbb{C} \to \mathbb{C}, \quad g_{k,d}(s) \triangleq s + ke^{-s} - dse^{-s} = 0, \tag{3.22}$$

which indicates $g_{k,d}(Ts) = Th_{T,k,d}(s)$, meaning that s is a root of $h_{T,k,d}(s)$ if and only if Ts is a root of $g_{k,d}(s)$. Now the following theorem describes the stability conditions for the delayed system above.

Theorem 3.2 (Stability condition for PD-type controller)**.** *The system defined in Eq.* (3.20) *is asymptotically stable if the parameters k, d and T satisfy the conditions described in Eq.* (3.23)*, i.e.*

$$-1 < d < 1, \quad 0 < kT < r\sin(r), \quad k > 0, \tag{3.23}$$

where $r = \arccos(d)$.

Proof. Similar to the proof in Section 3.4.1.3, we can plot the curves in stability chart with the roots of the simplified characteristic equation (3.22) having zero real parts.

1. Suppose $k < 0$ in Eq. (3.22), and again we define two curves: $y_1(s) \triangleq s$; and $y_2(s) \triangleq -(k - d \cdot s) \cdot e^{-s}$ with $k < 0$. We have $y_1(0) < y_2(0)$ and $\lim\limits_{s\to\infty} y_1(s) > \lim\limits_{s\to\infty} y_2(s)$. Then there must be at least one intersection point of these two curves in the $\mathbb{C}^+$ plane. Therefore, $k > 0$ is a necessary condition for the stability of the system (3.20).

2. Our next aim is to find out the parametric region for k, d, T from $h(s)$ in Eq. (3.21), which is equivalent to finding the parametric region for k, d from $g(s)$ in Eq. (3.22), where it is still guaranteed that all the roots of Eq. (3.21) will have their real parts in the $\mathbb{C}^-$ plane. Let $s = ib$ and separate real and imaginary parts. It follows:

$$\begin{cases} k = b\sin(b) \\ d = \cos(b) \end{cases} . \tag{3.24}$$

The stability chart based on Eq. (3.24) is shown in Figure 3.8[5]. The shaded area indicates the stability region for the parameters k, d, because when $d = 0$, the system (3.20) is the same as the system (3.13), hence the same stability condition applies.

3. Following the same analysis from Section 3.4.1.3, by determining the sign of the quantity $\Re\left\{\frac{ds}{dk}|_{s=ib}\right\}$ across the curves of the form:

$$C_n = \{(d, k) = (\cos(b), b\sin(b)), n\pi < b < (n+1)\pi\} \tag{3.25}$$

where $n \in \mathbb{N}_0$, we have:

$$\Re\left\{\frac{ds}{dk}|_{s=ib}\right\} = \frac{k}{k^2 + (d_\rho b + \sin(b))^2} > 0 \tag{3.26}$$

This shows that the increase in k beyond each curve C_n will bring the roots on the imaginary axis to the $\mathbb{C}^+$ plane and the system will be unstable.

This completes the proof. □

Inside the stable region, the system performance is affected by the the location of the dominant root of Eq. (3.21), which depends on the tuning parameters k, d and time delay T. This is addressed in the following section.

[5] The figure is generated using Mathematica®.

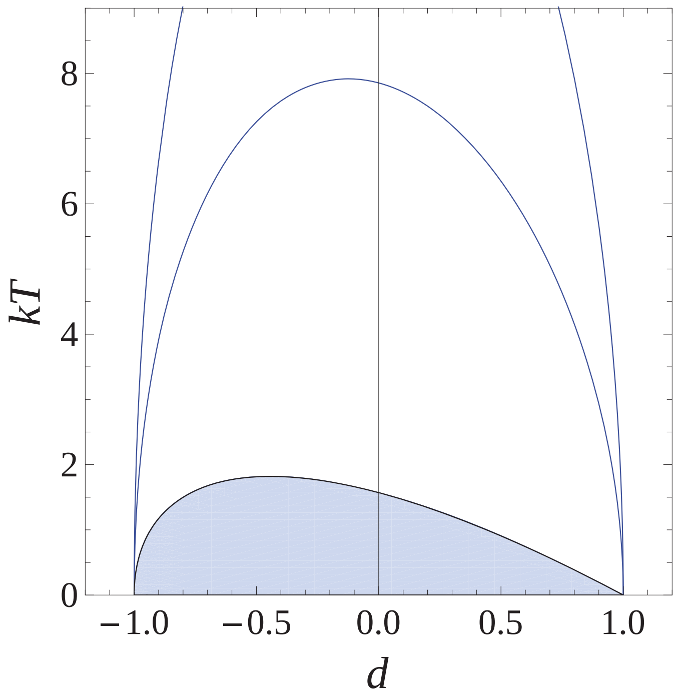


Figure 3.8: Stability chart of system (3.20); the curves are defined in Eq. (3.25); the shaded area indicates the stable region where all the roots are located in the $\mathbb{C}^-$ plane.

3.5 Parameter Design and Stability Analysis

In this section, we elaborate the parameter selection rules on k, d, T in the system (3.20) using D-partition method [36, pp.132][6]. The optimal values for the parameters in terms of system response are derived in a systematic way.

3.5.1 Parameter Selection

D-partition, or D-decomposition, is a well-known method for investigations in the asymptotic stability of a system. It makes use of the simple fact that the roots of a characteristic equation are continuous functions of the coefficients in the equation. Thus the space of coefficients is decomposed by the hyper-surface of all points which generate at least one root on the imaginary axis, i.e. on the line L_0, where

$$L_a = \{a + ib \mid b \in \mathbb{R}\}, \quad a \in \mathbb{R}. \tag{3.27}$$

Since the analysis of stability consists mainly of finding the coefficients which generate only roots with negative real part, the decomposition idea from D-partition simplifies

[6] Similar work for deriving stable PI controller using D-partition method can be found in Xue and Li [152].

this question to the problem of finding the appropriate region in the space of coefficients. In Theorem 3.1 and 3.2, we already applied this method to the systems (3.13) and (3.20), respectively.

Here we extend this methodology in the following way: consider $a \leq 0$ in Eq. (3.27) as an additional parameter, and search for coefficients which generate at least one root in L_a. In order to do so, we need the following Lemmas.

Lemma 3.1. *Every complex number $s \triangleq a + ib \in \mathbb{C}$ is one root of $g_{k,d}$ if the real part a and the imaginary part b of s satisfy the following two equations (with any given $d \in (-1,\ 1)$ and $k > 0$):*

$$k = da + e^{a} b \sin(b) - e^{a} a \cos(b) \tag{3.28}$$

$$0 = db - e^{a} b \cos(b) - e^{a} a \sin(b) \tag{3.29}$$

Proof. The equation $g_{k,d}(a + ib) = 0$ is equivalent to the following two equations, separated into the real and imaginary parts:

$$a + ke^{-a}\cos(b) - dae^{-a}\cos(b) - dbe^{-a}\sin(b) = 0 \tag{3.30}$$

$$b - ke^{-a}\sin(b) + dae^{-a}\sin(b) - dye^{-a}\cos(b) = 0 \tag{3.31}$$

- If $\cos(b) = 0$, then $\sin(b) = \pm 1$. It is obvious that Eq. (3.30) is equivalent to Eq. (3.29) and Eq. (3.31) is equivalent to Eq. (3.28).

- If $\cos(b) \neq 0$, by multiplying Eq. (3.30) with $\cos(b)$ and subtracting Eq. (3.31) multiplied by $\sin(b)$, the following equation is derived:

  $$a\cos(b) + ke^{-a} - dae^{-a} - b\sin(b) = 0$$

  Then Eq. (3.28) is obtained by solving the equation above according to k. Now insert Eq. (3.28) into Eq. (3.31):

  $$b - b(\sin(b))^2 + a\cos(b)\sin(b) - dbe^{-a}\cos(b) = 0.$$

  After canceling $\cos(b)$, Eq. (3.29) is finally derived.

□

Lemma 3.1 suggests to decompose the analysis of roots in two steps. First, we consider the trajectory of all roots by varying a, with $b = 0$ and fixed $d \in (-1, 1)$. Second, we fix $d \in (-1, 1)$ and consider the trajectory of all roots by varying b. In both steps, the trajectories of the roots form continuous curves in the $k - a$ plane. These curves are defined as follows.

Definition 3.1. *According to Eq.* (3.28)*, every root with zero imaginary part (i.e.* $b = 0$*) in the* $k - a$ *plane lies on the following curve:*

$$\Psi : \mathbb{R} \to \mathbb{R}^2, \quad \Psi(a) \triangleq (k(a),\ a)^\top = (da - ae^a,\ a)^\top$$

If the root has nonzero imaginary part (i.e. $b \neq 0$*), Eq.* (3.29) *cannot be solved explicitly with respect to* a*. In this case, the roots form a level set in the* $k - a$ *plane, which is defined as*

$$\Upsilon := \{(k, a) \mid \exists b \neq 0 \ \ s.t.\ \ (a, k, b) \ \ satisfies\ Eq.\ (3.28)\ \ and\ \ (3.29)\} \tag{3.32}$$

However, it is possible to solve Eq. (3.29) *locally with respect to* a*, i.e.* $a = a(b)$*. Thus, the level set* Υ *can be parameterized with curves of the form:*

$$\begin{aligned} &\Phi : \mathbb{R} \to \mathbb{R}^2 \\ &\Phi(b) \triangleq (k(a, b),\ a(b))^\top \\ &\quad\;\; = (da(b) + e^{a(b)} b \sin(b) - e^{a(b)} a(b) \cos(b),\ a(b))^\top. \end{aligned}$$

In the following part, the intersection point of the curves Ψ and Φ will be studied. This intersection point is where the roots of $g_{k,d}$ in the $\mathbb{C}^-$ plane have the same real part a with given k and d on both curves, which will reveal the dependence of the real part of the rightmost root on these two parameters.

Lemma 3.2. *Suppose* $(\hat{k}, \hat{a})$ *is an intersection point of the curves* Ψ *and* Φ *in the* $k - a$ *plane, and suppose* $\hat{b}$ *is the imaginary part of the intersection point.*

- *If* $\hat{b} = 0$*, then the following two equations will hold:*

$$\hat{k} = d\hat{a} - \hat{a}e^{\hat{a}} \tag{3.33}$$

$$0 = 1 + \hat{a} - de^{-\hat{a}} \quad . \tag{3.34}$$

Note that the equations above are only solvable for $d \geq -e^{-2}$*.*

- *If* $\hat{b} \neq 0$*, the following equalities hold:*

$$\hat{a} = \ln(|d|) \quad and \quad \hat{k} = \max\{2d \ln(|d|), 0\}$$

Note that there will be no intersection point for $d = 0$ *in the case of* $\hat{b} \neq 0$*.*

Proof. Depending on whether the intersection point has imaginary part, two cases are analyzed here.

- If $\hat{b} = 0$, $g_{\hat{k},d}$ will have multiple zeros at the intersection point where $s = \hat{a}$. Hence it holds that $g_{\hat{k},d}(\hat{a}) = 0$ and $g'_{\hat{k},d}(\hat{a}) = 0$. This gives us:

$$\begin{aligned} 0 &= \hat{a} + \hat{k}e^{-\hat{a}} - d\hat{a}e^{-\hat{a}} \\ 0 &= 1 - \hat{k}e^{-\hat{a}} - de^{-\hat{a}} + d\hat{a}e^{-\hat{a}}. \end{aligned}$$

Insert the first equation above into the second, Eq. (3.34) is derived, and multiply the first equation with e^{a}, it gives Eq. (3.33). As an example, Figure 3.9a shows the relation between k and a, with various regions for b.

- If $\hat{b} \neq 0$, $\hat{a} + i\hat{b}$ and $\hat{a}$ are both zeros of function $g_{\hat{k},d}$. According to Lemma 3.1:

$$d\hat{a} - e^{\hat{a}}\hat{a} = k = d\hat{a} + e^{\hat{a}}\hat{b}\sin(\hat{b}) - e^{\hat{a}}\hat{a}\cos(\hat{b}) \tag{3.35}$$

$$d\hat{b}e^{-\hat{a}} = \hat{a}\sin(\hat{b}) + b\cos(\hat{b}) \tag{3.36}$$

From Eq. (3.35), it gives $-\hat{a} = \hat{b}\sin(\hat{b}) - \hat{a}\cos(\hat{b})$.

a) If $\hat{b} \notin 2\pi\mathbb{Z}$, Eq. (3.36) becomes:

$$\begin{aligned} d\hat{b}e^{-\hat{a}} &= \frac{\hat{b}\sin(\hat{b})^2}{\cos(\hat{b}) - 1} + \hat{b}\cos(\hat{b}) \\ &= \frac{\hat{b}\sin(\hat{b})^2(\cos(\hat{b}) + 1)}{\cos(\hat{b})^2 - 1} + \hat{b}\cos(\hat{b}) \\ &= -\hat{b} \end{aligned}$$

Since $\hat{b} \neq 0$, it follows that $\hat{a} = \ln(-d)$. This equation is only solvable for $d < 0$. Insert this solution into (3.33), it gives $k = 2d\ln(-d)$. Two examples of this case are shown in Figure 3.9b and Figure 3.9c.

b) If $\hat{b} \in 2\pi\mathbb{Z}$, Eq. (3.29) becomes $0 = d\hat{b} - \hat{b}e^{\hat{a}}$. Since $\hat{b} \neq 0$ still holds, it gives $\hat{a} = \ln(d)$. Moreover, this equation is only solvable for $d > 0$. Insert this into Eq. (3.28), then $k = 0$. This means the intersection point is on the a axis in the $k - a$ Plot (as shown in Figure 3.9a with the smaller a), where all zero solutions have the same real part ($\hat{a} = \ln(d)$) but different imaginary parts ($\hat{b} \in 2\pi\mathbb{Z}$) in the s-plane.

Note that for the case when $d = 0$, the only intersection point is located on the real axis, where $\hat{b} = 0$. The corresponding plot is shown in Figure 3.9d. This completes the proof for Lemma 3.2. □

Lemma 3.2 provides the relationship between the real part a of the roots and the two tuning parameters k and d at the point where curve Ψ and Φ intersects. In order to find the solution to the smallest real part of the rightmost root, the following Lemma is necessary, which proves the monotonicity of the curve Φ in a certain region of interest.

Lemma 3.3. *The trajectory of the curve Φ is strictly monotonously increasing for $k > da - ae^a$ and $a < 0$ in $k - a$ plane.*

Proof. According to Eq. (3.28), the following partial derivative over a and b can be derived:

$$\begin{cases} \dfrac{\partial k}{\partial a} & = d + e^a b \sin(b) - e^a a \cos(b) - e^a \cos(b) \\ \dfrac{\partial k}{\partial b} & = e^a \sin(b) + e^a b \cos(b) + e^a a \sin(b) \end{cases} \tag{3.37}$$

A total differentiation of Eq. (3.29) over b yields:

$$de^{-a} - dbe^{-a}\frac{da}{db} = \sin(b)\frac{da}{db} + \cos(b)a + \cos(b) - b\sin(b),$$

and it follows:

$$d + e^a b\sin(b) - e^a a\cos(b) - e^a \cos(b) = \left(db + e^a \sin(b)\right)\frac{da}{db}. \tag{3.38}$$

Combining Eq. (3.37), (3.38) with Eq. (3.29):

$$\frac{\partial k}{\partial a} = \frac{\partial k}{\partial b}\frac{da}{db}. \tag{3.39}$$

Since the curve Φ is strictly monotonously increasing in $k - a$ Plot if $\dfrac{dk}{da} \in (0, +\infty)$, under Eq. (3.39), it is equivalent to $\dfrac{\partial k}{\partial a} > 0$, because

$$\frac{dk}{da} = \frac{\partial k}{\partial a} + \frac{\partial k}{\partial b}\frac{db}{da} = \frac{\partial k}{\partial a}\left(1 + \frac{\left(\frac{\partial k}{\partial b}\right)^2}{\left(\frac{\partial k}{\partial a}\right)^2}\right)$$

Without loss of generality, let $b \geq 0$. In the region $k > da - ae^a$:

$$\begin{aligned} be^a\frac{\partial k}{\partial a} &= bde^{-a} - b\cos(b) + y\left(b\sin(b) - a\cos(b)\right) \\ &= a\sin(b) + be^{-a}\left(e^a a + e^a b\sin(b) - e^a a\cos(b)\right) - ba \\ &= a\sin(b) + be^{-a}\left(e^a a - da + k\right) - ba \\ &> a\sin(b) - ba \\ &= (-a)\left(b - \sin(b)\right) \\ &\geq 0. \end{aligned}$$

This indicates that $\dfrac{\partial k}{\partial a} > 0$ in the region described above and hence $\dfrac{dk}{da} \in (0, \infty)$. □

Now the following theorem can be concluded based on Lemma 3.1 to 3.3, which states the main result on the criteria of tuning parameters in the PD-type controller to adjust the location of the dominant root.

Theorem 3.3 (Global Minimum of the Rightmost Root)**.** *Let $\tilde{a}(k, d)$ denote the largest real part of all the roots of equation $g_{k,d}(s)$, with the parameters $k > 0$ and $d \in (-1, 1)$, i.e.,*

$$\tilde{a}(k, d) \triangleq \max\{\Re(s) \mid g_{k,d}(s) = 0\}.$$

For a given $d \in (-1, 1)$, the trajectory of $k \mapsto (k, \tilde{a}(k, d))^\top$ has two parts. The first part starts from the curve of Ψ and the other part is on the curve Φ. The minimum of $k \mapsto \tilde{a}(k, d)$ is therefore the intersection point of the two curves.

If d varies in $(-1, 1)$, then the global minimum of function $\tilde{a}(k, d)$ is -2 with $k = 4e^{-2}$ and $d = -e^{-2}$, i.e.

$$\tilde{a}(4e^{-2}, -e^{-2}) = \min_{k>0, |d|<1} \tilde{a}(k, d) = -2.$$

Proof. We derive the global minimum point through the monotonicity property of curves Ψ and Φ, and search for the minimum value in the set of all possible intersection points.

With given k and d, where $k > 0$ and $d \in (-1, 1)$, the value $\tilde{a}(k, d)$ is the real part of one zero solution of function $g_{k,d}(s)$, and the point $(k, \tilde{a}(k, d))$ must lie on either the curve Φ or Ψ according to Lemma 3.1 and Definition 3.1. Since $|dse^{-s}| = |d||s|e^{-\Re\{s\}} < |s|$ for $\Re\{s\} > 0$ and $g_{0,d}(0) = 0$, $s = 0$ is hence one zero with the largest real part of function $g_{0,d}$, and the point $(0, \tilde{a}(0, d))$ is on the curve Ψ. Since $\tilde{a}$ is continuous, point $(k, \tilde{a}(k, d))$ for $k > 0$ stays on the curve Ψ until the first intersection point of Ψ and Φ. Since $\dfrac{dk}{da} > 0$ after the intersection point, $(k, \tilde{a}(k, d))$ will switch onto the curve Φ.

On curve Ψ, $k(a) \triangleq da - ae^a$ is decreasing before the intersection point, and therefore $k \mapsto \tilde{a}(k, d)$ is decreasing. After the intersection point, the function $k \mapsto \tilde{a}(k, d)$ is monotonously increasing on curve Φ according to Lemma 3.3. Therefore, the point $(k, \tilde{a}(k, d))$ cannot come back to the curve Ψ after the intersection point, and this intersection point is therefore the minimum of $k \mapsto \tilde{a}(k, d)$.

Let $(\hat{k}_{i,d}, \hat{a}_{i,d})$, $i \in I$ be all the intersection points with a given value of $d \in (-1, 1)$, and the following equation can be drawn from the argument above:

$$\min_{k>0} \tilde{a}(k, d) = \max_{i \in I} \hat{a}_{i,d}.$$

This means:

$$\min_{k>0,|d|<1} \tilde{a}(k,d) = \min_{|d|<1} \max_{i \in I} \hat{a}_{i,d}.$$

According to Lemma 3.2, it can be written as:

$$\begin{aligned}&\min_{k>0,|d|<1} \tilde{a}(k,d)\\ &= \min_{|d|<1} \left(\max \left\{ \hat{a} \mid 0 = 1 + \hat{a} - de^{-\hat{a}} \cup \hat{a} = \ln(|d|) \right\} \right) \end{aligned} \tag{3.40}$$

- If $d > -e^{-2}$, then $\ln|d| < -2$. Meanwhile, $1 + a - de^{-a} < 0$ for $a = -2$ and $\lim\limits_{a \to \infty} (1 + a - de^{-a}) = \infty$. By continuity, there must be one $\hat{a} > -2$, which satisfies the equation $0 = 1 + a - de^{-a}$.

- If $d < -e^{-2}$, the equation $0 = 1 + a - de^{-a}$ has no solution and $\ln|d| > -2$.

- If $d = -e^{-2}$, $\hat{a} = -2$.

Therefore, $\tilde{a}(k,d) = -2$ is the global minimum of Eq. (3.40), with $d = -e^{-2}$ and $k = 4e^{-2}$. The corresponding $k - a$ Plot is shown in Figure 3.9e. □

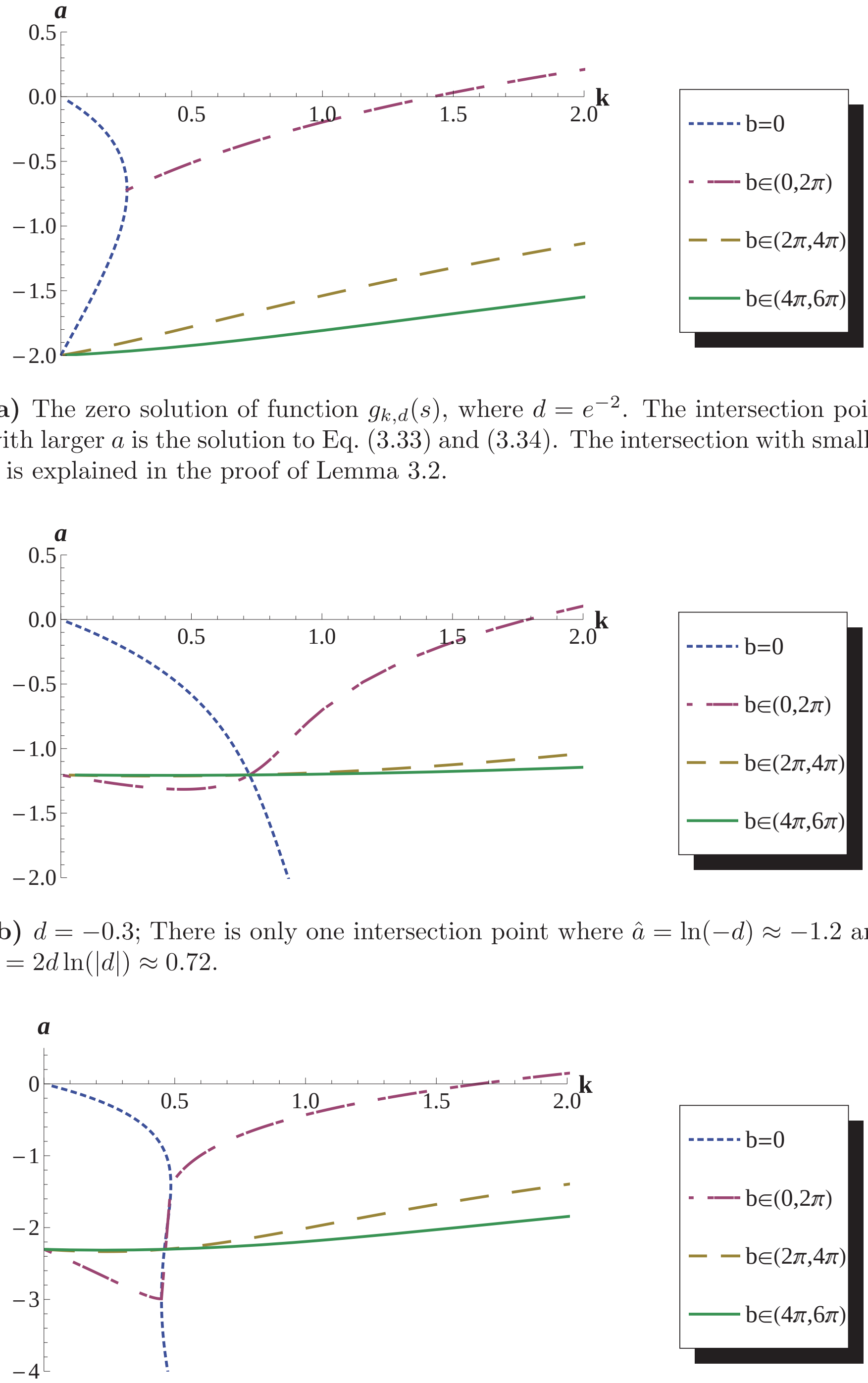
Remark:

1. Recall that in the simplified characteristic equation (3.22), we have the change of variable $k \mapsto kT$. Under Theorem 3.3, it means that the system can tolerate a large delay T by adjusting the proportional gain k in the controller, where the same minimum dominant root can be achieved.

2. The method of D-partition allows us to analyze the real part of the roots in terms of the coefficients in the characteristic equations. This approach is generally applicable to parameter selection of other types of controllers, provided that the transcendental characteristic equations exist.

3.6 Convergence and Internal Dynamics

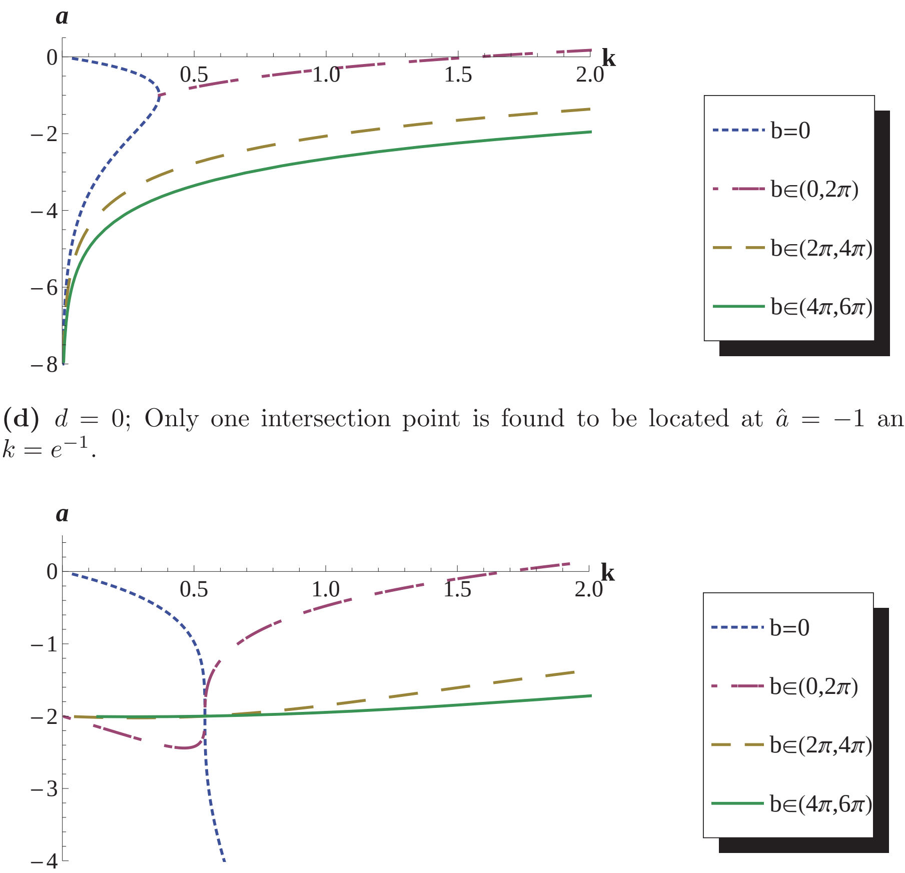
In the section above, we analyzed the relationship between the real part of roots of the characteristic equation (3.22) and its coefficients. The analysis is valid based on the change of variables $s \mapsto sT$ and $k \mapsto kT$ in Section 3.4.2. In this section, we will address the convergence rate of the formation system described in the general delay differential equation (3.20), which fundamentally determines the performance of the system. The advantage of our proposed PD-type controller will be summarized



(a) The zero solution of function $g_{k,d}(s)$, where $d = e^{-2}$. The intersection point with larger a is the solution to Eq. (3.33) and (3.34). The intersection with smaller a is explained in the proof of Lemma 3.2.

(b) $d = -0.3$; There is only one intersection point where $\hat{a} = \ln(-d) \approx -1.2$ and $k = 2d\ln(|d|) \approx 0.72$.

(c) The zero solution of function $g_{k,d}(s)$, where $d = -0.1$. There are three intersection points, two of which are from the first part of Lemma 3.2 where $\hat{b} = 0$ and $\hat{a}$ is the solution to Eq. (3.33) and (3.34). The other intersection point is located at $\hat{a} = \ln(-d) \approx -2.3$ and $k = 2d\ln(|d|) \approx 0.46$ according to the second part of Lemma 3.2 where $\hat{b} \neq 0$.



(d) $d = 0$; Only one intersection point is found to be located at $\hat{a} = -1$ and $k = e^{-1}$.

(e) $d = -e^{-2}$; The global minimum of $\tilde{a}$ is found at $a = -2$, where $k = 4e^{-2}$.

Figure 3.9: The zero solutions of function $g_{k,d}(s)$ with different values of d.

through the following two theorems, which are based on the Theorem 2.1 for a general linear time invariant system. Meanwhile, the analysis on internal dynamics in the case of time-delayed formation system is also provided to complete the overall controller design.

3.6.1 Exponential Convergence

For the P-type controller discussed in Section 3.4.1, the following theorem can be concluded.

Theorem 3.4 (Convergence rate of P-type Controller)**.** *The trivial solution of the delay differential equation* (3.13) *is unstable for* $k < 0$ *and for* $kT > \frac{\pi}{2}$*. In the case* $0 < kT < \frac{\pi}{2}$*, all solutions* $x(t)$ *of Eq.* (3.13) *are asymptotically stable according to the* *exponential rule*

$$|x(t)| \le Qe^{\lambda t}$$

for any $\lambda > \dfrac{\tilde{a}(kT, 0)}{T} \triangleq \lambda_0$ *and* $Q > 0$.

If $T > 0$ *is fixed, the smallest value for the stability exponent* λ_0 *is* $-\dfrac{1}{T}$. *This value is only achieved for the system parameter* $k = \dfrac{1}{T}e^{-1}$.

Proof. Theorem 3.4 is a trivial deduction of Theorem 3.1, Lemma 3.2 and 3.3 for $d = 0$. Thus λ_0 and k can be computed using Eq. (3.33) and (3.34), where the intersection point has zero imaginary part. □

For the PD-type controller presented in Section 3.4.2, the following theorem can be drawn:

Theorem 3.5 (Convergence rate of PD-type Controller)**.** *The trivial solution of the delay equation* (3.20) *is unstable for* $k < 0$ *and for* $kT > r\sin(r)$ *with* $r = \arccos(d)$. *In the case* $0 < kT < r\sin(r)$, *all solutions* $x(t)$ *of Eq.* (3.20) *are asymptotically stable according to the exponential rule*

$$|x(t)| \le Qe^{\lambda t}$$

for any $\lambda > \dfrac{\tilde{a}(kT, d)}{T} \triangleq \lambda_0$ *and* $Q > 0$.

If $T > 0$ *is fixed, the smallest value for the stability exponent* λ_0 *is* $-\dfrac{2}{T}$. *This value is only achieved for the system parameter* $k = \dfrac{4}{T}e^{-2}$ *and* $d = -e^{-2}$.

Proof. Theorem 3.5 is also a trivial deduction of Lemma 3.2, Lemma 3.3 and Theorem 3.3 for $d \neq 0$. □

These two theorems indicate that switching from the P-type control in Eq. (3.13) to the PD-type control in Eq. (3.20) makes it possible to double the stability exponent λ_0 (from $-\dfrac{1}{T}$ to $-\dfrac{2}{T}$), which gives higher convergence rate and more stable response to the respective systems. Figure 3.10 shows the plot of roots which illustrates the two theorems above.

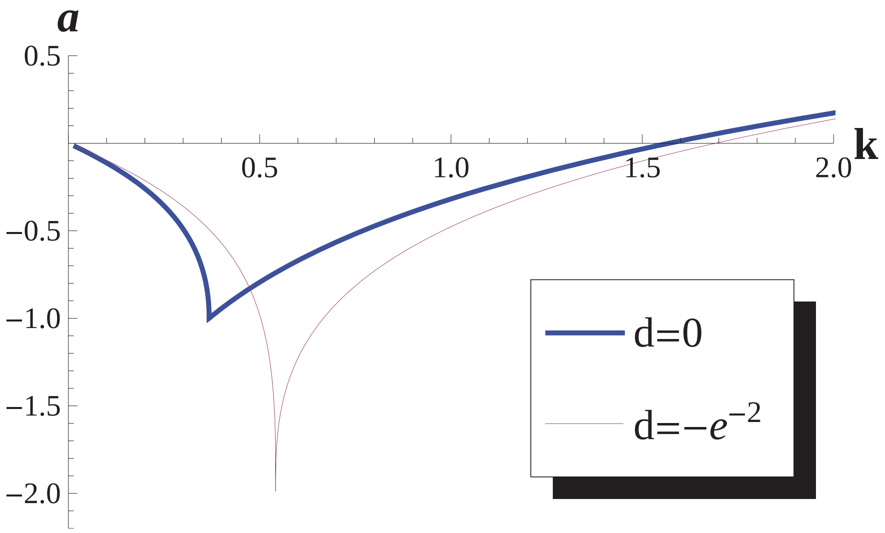


Figure 3.10: Real part of the rightmost root; the curve with $d = -e^{-2}$ illustrates Theorem 3.3.

3.6.2 Internal Dynamics in Time-delayed Case

The other part of the controller stability analysis includes the convergence proof of the internal dynamics of the delayed formation system, similar to the analysis in Section 3.4.1.3. The theorem developed in this section will address the local stability conditions and global attraction domain for the internal dynamics.

In order to simplify notation assuming no ambiguity arises on the variable x, let $x(t) \triangleq e_3(t) - \alpha_d$ in the internal dynamics (3.7), and we get the nonlinear delay differential equation

$$\dot{x}(t) = M \sin(x(t-T)) - N, \quad M, N \in \mathbb{R},\ T \geq 0, M \neq 0, \tag{3.41}$$

where $M = \dfrac{{v_L}^{\mathrm{d}}}{\rho_d \cos \alpha_d}, N = \dfrac{{v_L}^{\mathrm{d}} \tan {\beta_L}^{\mathrm{d}}}{l}$.

Recall that a system like (3.41) yields a unique solution if we have an initial function $\Lambda : [-T, 0] \to \mathbb{R}$ in advance and let $x(t) = \Lambda(t)$ for all $t \in [-T, 0]$. An equilibrium point $x_e \in \mathbb{R}$ of a system like (3.41) is called locally (asymptotically) stable if there exists $\mu_a > 0, T_a > 0$, such that $\sup_{t\in[-T,0]} |\Lambda(t) - x_e| < \mu_a$ implies $x(t) \to x_e$ for $T \leq T_a$. If there exists $T_a > 0$, such that $x(t) \to x_e$ holds for any Λ and any $0 \leq T \leq T_a$, we call the system globally (asymptotically) stable (see Driver [28], Melchor-Aguilar and Niculescu [96] for the definitions and Theorem 3.1 in [47] for the proof.).

Theorem 3.6. *Consider the nonlinear delay-differential equation* (3.41)

- *if* $|M| < |N|$, *it has no equilibrium point.*
- *if* $|M| = |N|$, *then* $x_e = \text{sgn}(\frac{N}{M})\frac{\pi}{2} \mod 2\pi$ *is the only equilibrium point in the interval* $[0, 2\pi)$.
- *if* $|M| > |N|$, *there are two equilibrium points* $x_1 = \arcsin(N/M) \mod 2\pi$ *and* $x_2 = \pi - \arcsin(N/M) \mod 2\pi$ *in the interval* $[0, 2\pi)$.

Define:

$$C := \text{sgn}(M)T\sqrt{M^2 - N^2}. \tag{3.42}$$

The point x_1 *is locally asymptotically stable if* $-C \in [0,\ \frac{\pi}{2})$, *and unstable if* $-C \notin [0,\ \frac{\pi}{2}]$.

The point x_2 *is locally asymptotically stable if* $C \in [0,\ \frac{\pi}{2})$, *and unstable if* $C \notin [0,\ \frac{\pi}{2}]$.

Proof. Three cases are analyzed separately, depending on the values of M and N.

- If $|M| < |N|$, for the right-hand side of Eq. (3.41), it is obvious that

  $$|M\sin(x(t-T)) - N| > |N| - |M| > 0.$$

  Therefore, $\lim_{t\to\infty} |x(t)| = \infty$ and Eq. (3.41) has no equilibrium point.

- If $|M| > |N|$, let $\sigma = \arcsin(\frac{N}{M})$. The equilibrium points of Eq. (3.41) are the same as the zeros of the function

  $$\begin{aligned} f(z) &\triangleq M\sin(z) - N \\ &= M\left(\sin(z) - \sin(\sigma)\right) \\ &= 2M\sin\left(\frac{z-\sigma}{2}\right)\cos\left(\frac{z+\sigma}{2}\right). \end{aligned} \tag{3.43}$$

  The two zeros of $f(z)$ are $z_1 = \sigma \mod 2\pi$ and $z_2 = \pi - \sigma \mod 2\pi$ in the interval $[0, 2\pi)$, which gives the two equilibrium points $x_1 = z_1$ and $x_2 = z_2$. The linearization of Eq. (3.41) at these two equilibrium points is:

  $$\begin{cases} \dot{y}_1(t) = M\cos(\sigma)y_1(t-T) & \text{at point } x_1 \\ \dot{y}_2(t) = -M\cos(\sigma)y_2(t-T) & \text{at point } x_2 \end{cases}$$

where $y_1(t) = x(t) - x_1$, and $y_2(t) = x(t) - x_2$. Let $\hat{M} \triangleq M\cos(\sigma)$, then the corresponding characteristics equation of these two linearized equations are:

$$s - \hat{M}e^{-Ts} = 0 \quad \text{and} \quad s + \hat{M}e^{-Ts} = 0. \tag{3.44}$$

Following Zhang et al. [156], the stability regions for the characteristic equations above are well-defined, which are

$$\begin{cases} 0 \leq -T\hat{M} < \dfrac{\pi}{2} & \text{at point } x_1 \\ 0 \leq T\hat{M} < \dfrac{\pi}{2}. & \text{at point } x_2. \end{cases} \tag{3.45}$$

Therefore, the stability conditions of $-C \in [0,\ \frac{\pi}{2})$ on equilibrium point x_1 and $C \in [0,\ \frac{\pi}{2})$ on equilibrium point x_2 are the results from Eq. (3.45), with $\hat{M} = M\sqrt{1-(N/M)^2} = \text{sgn}(M)\sqrt{M^2 - N^2}$. Standard results on stability in first approximation [see 36, pp.160, 161] complete the proof from the stability of the linearized function to the stability of the original nonlinear delay-differential equation (3.41).

- If $|M| = |N|$, the two zeros from Eq. (3.43) merge together, and the stability of the equilibrium point of $x = \pi/2$ is not well-defined.

This completes the proof. □

Theorem 3.6 provides the parameter domain in M, N, T which defines local stability conditions for the internal dynamics of the delayed formation system.

The parameter domain in M, N, T for the global stability property of the system (3.41) is numerically computed under Mathematica®, which is shown in Figure 3.11. We approximate the function $T_a(M, N)$, such that each function $x(t)$ with the following two properties:

- $x(t)$ is a solution of (3.41) for given parameters M, N, $0 \leq T \leq T_a(M, N)$ and
- $x(t) = c,\ \forall t \in [-T, 0]$ and $\forall c \in \mathbb{R}$

satisfies $x(t) \to x_e$. (Due to the second property, this is a weaker formulation than the one of the global stability.) Note that Figure 3.11 only captures the case where $M < 0$. When $M > 0$, the analogous figure appears, which is symmetric to Figure 3.11 with respect to $M = 0$ axis.

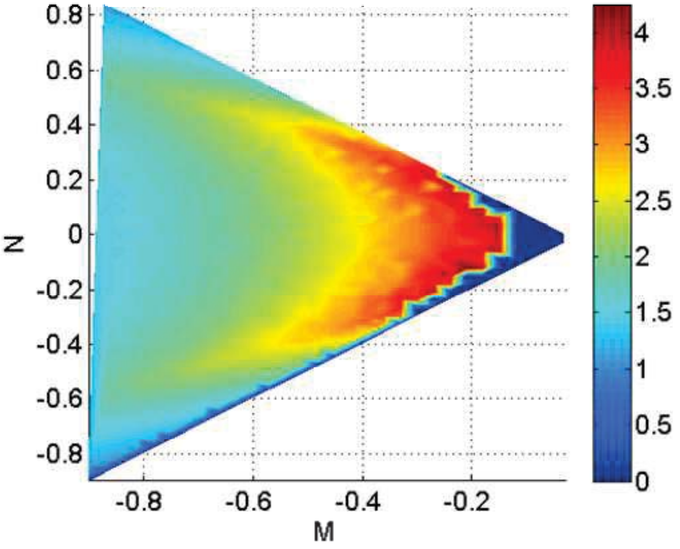


Figure 3.11: Parameter domain for global stability; the color map represents the maximum of delay T_a in [s]

3.7 Simulation and Experimental Validation

In this section, we present both simulation and experimental results for a formation system composed of multiple car-like robots.

In the simulations, the leader robot (marked in red) moves at a constant speed $v_L = 1$ m/s, with a steering angle $\beta_L = 0.05$ radian, while two followers (in yellow) start from different initial positions and try to achieve a V-formation with a desired separation distance $\rho_{d1} = 2$ m and relative bearing angles $\alpha_{d1} = \pm\frac{\pi}{4}$, and two other sub-followers (in yellow) try to achieve a subsequent V-formation with one of the followers. For these two sub-followers, the desired separation distance is $\rho_{d2} = 3.5$ m and relative bearing angles are $\alpha_{d2} = \pm\frac{\pi}{6}$.

In the hardware experiments, the leader robot is driving semi-autonomously at the speed of $v_L = 0.5$ m/s for cooperative V-formation, and at $v_L = 0.3$ m/s for the non-cooperative formation tracking using visual data. The steering angle is set as $\beta_L = 0.087$ radian (5 degree) in circular motion. These values are properly selected according to the size of the robots and the experiment field. The sampling period of the onboard sensors is $T_s = 0.1$ s.

The constant time delay T in both cases is simulated using data buffer. In the case of time-varying delays, we apply the following rules and technical assumptions to form piecewise constant delays:

1. The variable delays are upper rounded to $N \cdot T_s$, where $N \in \mathbb{N}$, and T_s is the

simulation step size or sampling period of the robots. Each data packet has its own delay value.

2. The data packets sent out from the leader are dropped by the follower if the follower considers the packets are older than the ones it has received before.

3. The follower keeps the newest packet from the leader as the reference at every processing step.

4. The follower is assumed to be able to determine how much delay is induced on each packet based on packet numbering or timestamps.

In the experiment presented in Section 3.7.2, the localization between the mobile robots is based on the dead reckoning technique, as pointed out in Section 2.3.2. It suffers from long-term accumulation of measurement errors from onboard sensors, such as gyroscope and optical encoders. Nevertheless, short-term experiments for proving the concepts are feasible. Such demonstration based on dead reckoning can also be found in vision-based vehicle-following system from Benhimane et al. [8] and laser range-based system from Vasseur et al. [140], as well as the rendezvous system from Hess [54].

3.7.1 Simulation

The simulation results from Figure 3.12 to Figure 3.15 cover the overall study from this chapter. The car-like mobile robots denoted with Follower 1 and Follower 2 are assigned to keep a distance $\rho_{d1} = 2$m and bearing angle $\alpha_{d1} = \pm\pi/4$ from the Leader robot, while Follower 2.1 and Follower 2.2 are expected to keep the V-shape formation with Follower 2 at a distance $\rho_{d2} = 3.5$m and bearing angle $\alpha_{d2} = \pm\pi/6$.

3.7.1.1 Feasibility

To evaluate the performance of the proposed controllers from this chapter, a cascade leader-follower formation system without delay is simulated and shown in Figure 3.12. The controller applied is from Eq. (3.4). A smooth and stable behavior of the system achieving two cascading V-formations can be observed in the figure, which proves the feasibility of the input-output feedback linearization method based on scalable leader-follower structure.

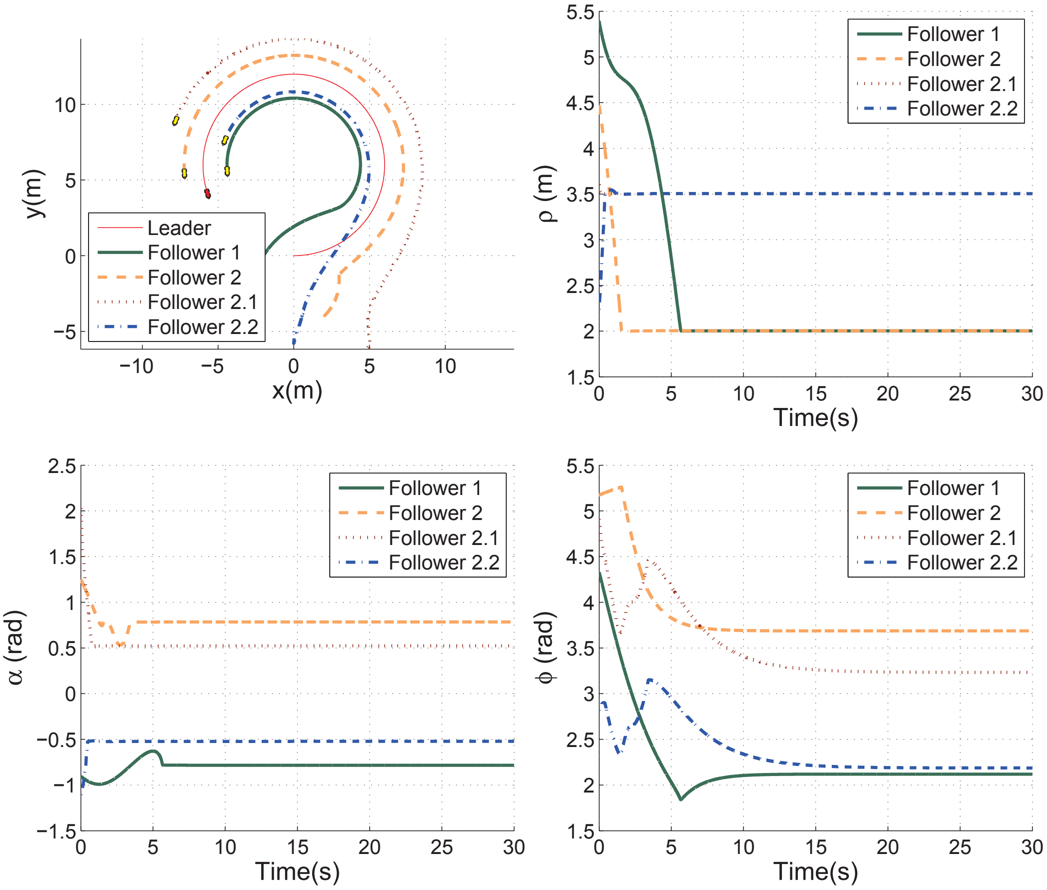


Figure 3.12: Formation tracking without delay, $k_\rho = 3, k_\alpha = 4$, where leader drives along a predefined circle, and the followers are driving autonomously along under the controller from Eq. (3.4).

3.7.1.2 Steady-state error

In order to illustrate the problem of steady-state error related to the formation tracking system, a second simulation based on the controller from Eq. (3.12) is shown in Figure 3.13. This controller incorporates Eq. (3.9), which utilizes current states of follower and delayed states of the leader. Clearly, the steady-state error can be observed on ρ and α as discussed in section 3.4.1. Figure 3.4 illustrates the reason behind this phenomenon.

3.7.1.3 Oscillation with delay-compensation

After the design of the delay-compensating controller given by Eq. (3.12), which uses Eq. (3.10) as the input to achieve the linear system behavior given in Eq. (3.11),

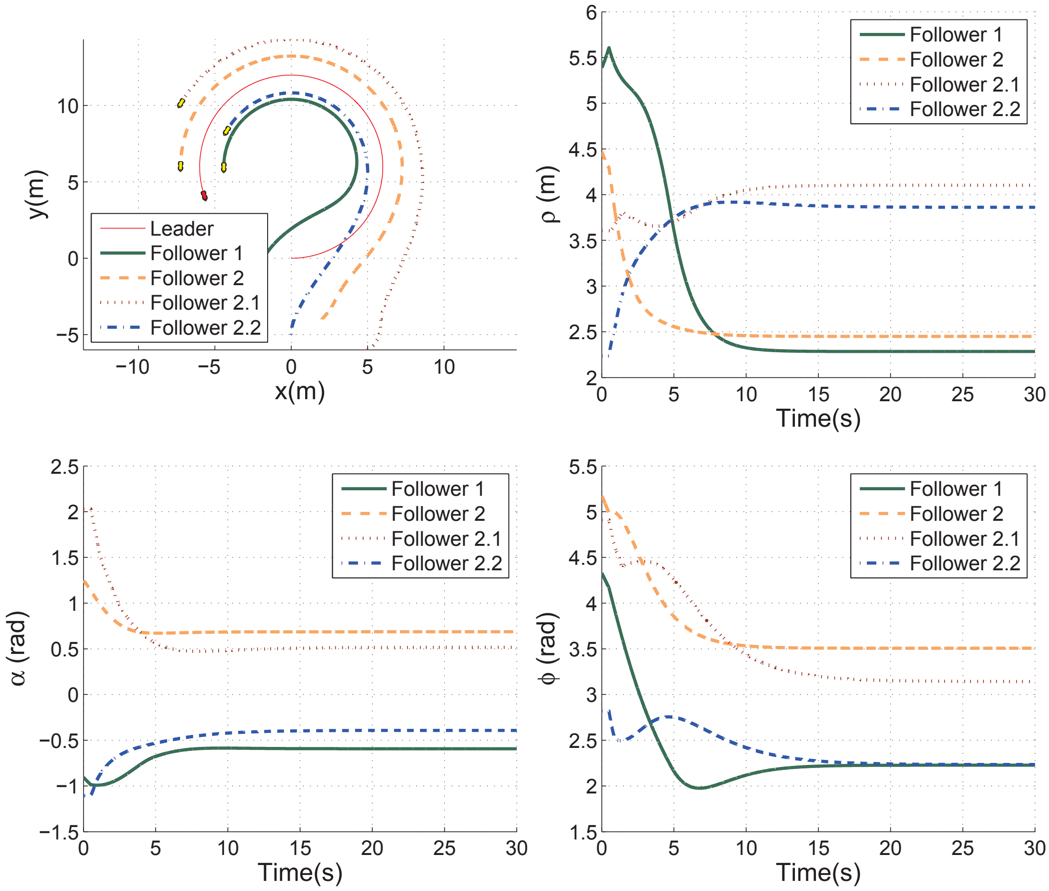


Figure 3.13: Formation tracking with delayed leader states and current follower states, $k_\rho = 1.5, k_\alpha = 2$; delay=0.6s. The steady-state error is clearly observed.

the same formation system as before is simulated. The results are shown in Figure 3.14. Although the convergence condition from Eq. (3.15) still holds in this case, the formation system presents a very oscillating behavior due to the location of the dominant root close to the imaginary axis, which can be found in Figure 3.10 (where $d = 0$). This oscillation can be suppressed through the derivative term in the PD-type controller, as shown in the following simulation. Nevertheless, the steady-state error does not show up in this case due to the delay-compensating controller design, which is explained in Figure 3.5.

3.7.1.4 Stable delay-compensation

Under the PD-type controller derived from Eq. (3.19), we have the simulated results in Figure 3.15, with the parameters derived in Section 3.5.1. Additionally, we applied variable time delays (piecewise constant) between 0.4 to 0.8 sec. To achieve fastest

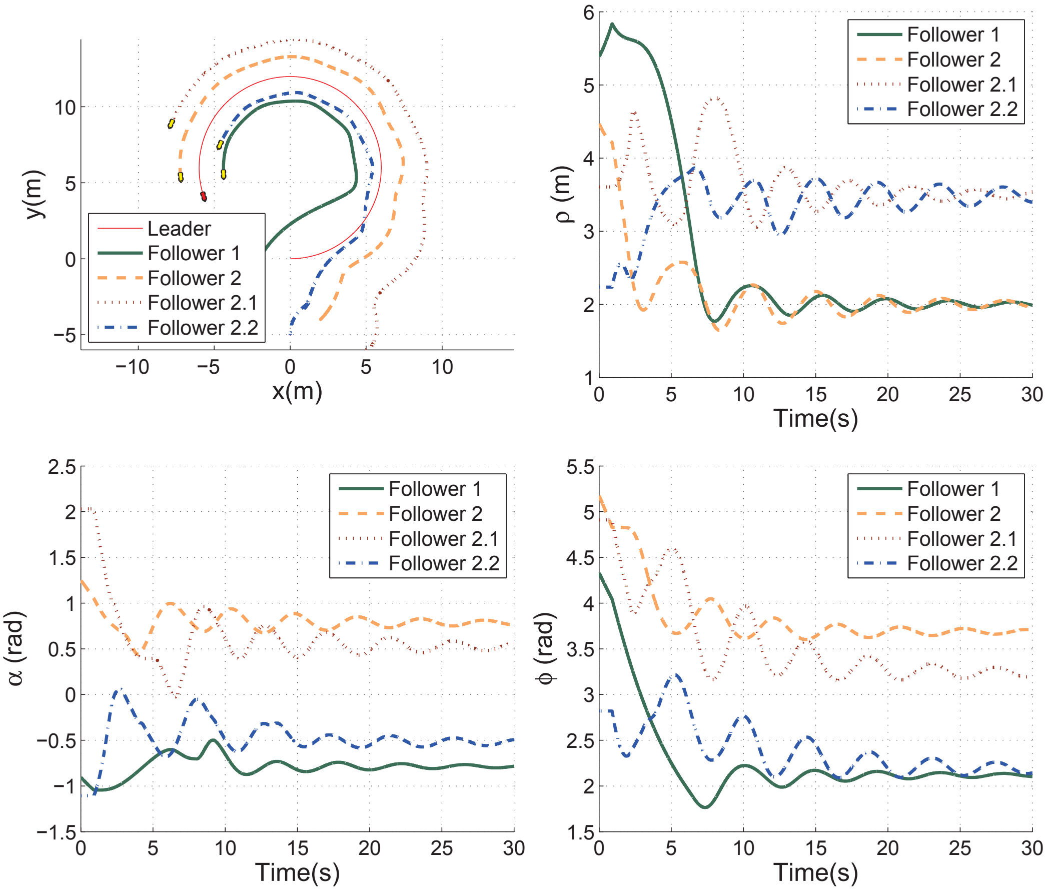


Figure 3.14: Formation tracking with delayed leader states and delayed follower states to compensate for steady-state error; P-type controller, with $k_\rho = 1.5, k_\alpha = 2$; delay=0.6s.

convergence according to Theorem 3.5, $k_\rho = k_\alpha = \dfrac{4}{T}e^{-2} = 0.6767$ and $d_\rho = d_\alpha = -e^{-2} = -0.1353$, where $T = 0.8s$ is the upper-bound of the delay. The system has a much more stable behavior than in Figure 3.14, because the location of the dominant root in this case is placed far enough to the left-side of the imaginary axis. Obviously, this demonstrates the usage of the derived theorems on selection criteria of stable parameters in Section 3.6.

In the next section, experiments of various formation tracking systems are presented to further validate and showcase the developed delay-compensating formation controller.

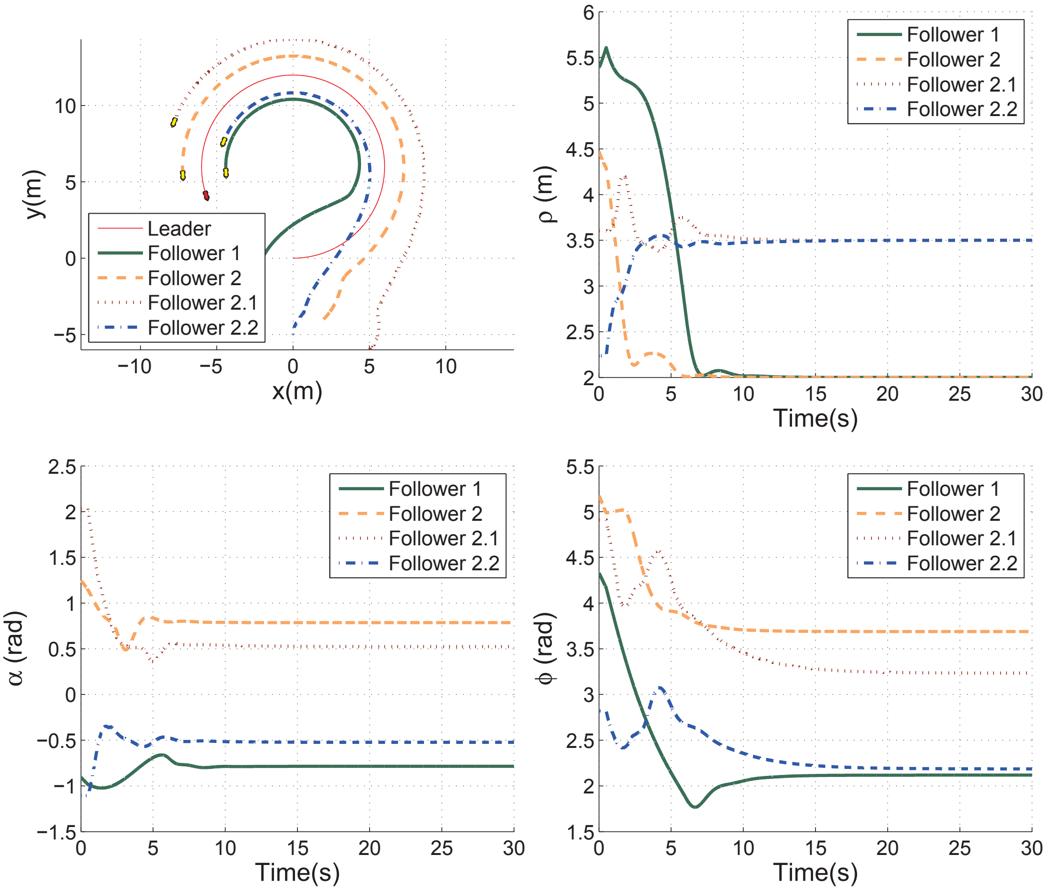
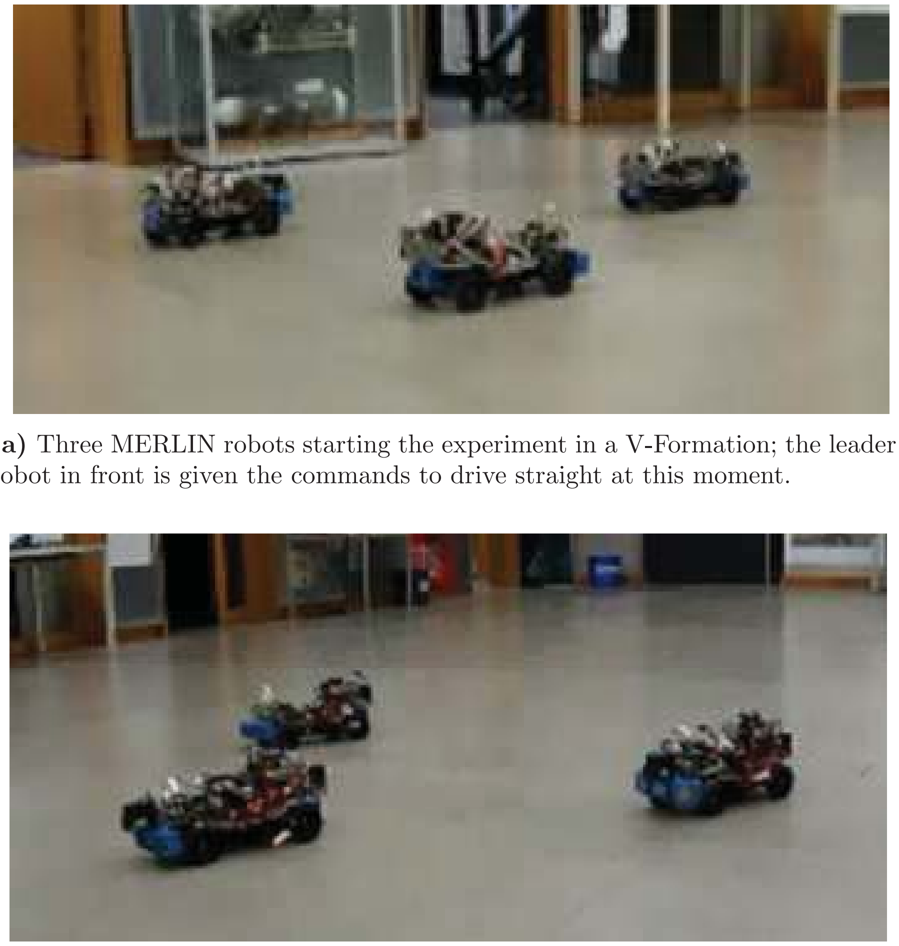


Figure 3.15: Formation tracking with PD-Type controller under delayed leader states; $k_\rho = 0.676, k_\alpha = 0.676, d_\rho = -0.135, d_\alpha = -0.135$; delay $= 0.4 - 0.8s$



(a) Three MERLIN robots starting the experiment in a V-Formation; the leader robot in front is given the commands to drive straight at this moment.

(b) Three MERLIN robots during the experiment after a circular turnaround.

Figure 3.16: Snapshots of MERLIN in leader-follower experiment

3.7.2 Cooperative V-Formation Experiment

For the real-world experimental study on our MERLN robots (which are described in Section 2.3.2), we compare the delay-compensating formation controller with the initial feedback linearization method from Section 3.3.1, which is affected by steady-state error due to the communication delays. In order to be able to manually bound the communication delays between the robots, we create data buffers on the leader robot in the software. Before it broadcasts its localization information using UDP in an Ad-Hoc network, the data buffer mechanism checks if the desired delay is reached for each packet. We also add bounded randomness to the delay timer. In this way, we are able to create piece-wise constant delays within the range from 0.4 to 0.8 seconds. To visualize the conducted experiments, two snapshots of the second experiment performed in this section are shown in Figure 3.16.

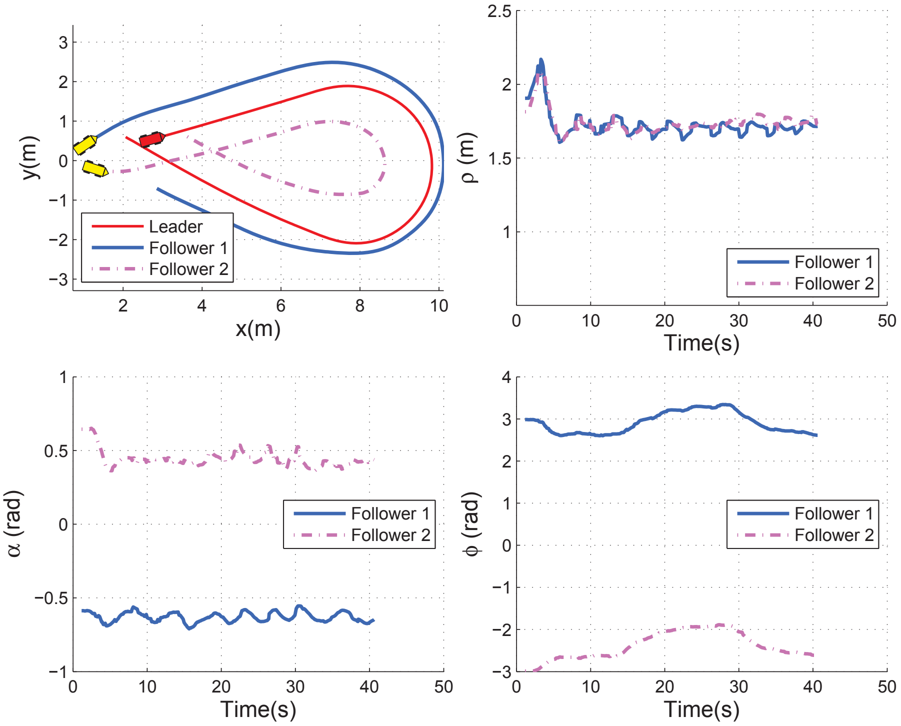


Figure 3.17: MERLIN formation drive with delayed leader states; the parameters used in the experiment are as follows: $\rho_d = 1.5m, \alpha_d = \pm\pi/6, k_\rho = 1.2, k_\alpha = 1.2;\text{delay} = 0.4 - 0.8s$. Leader is driving at 0.5 m/s, with steering angle at 5 deg during circular motion.

The first experiment consists of three MERLIN robots, where the two followers are running the controller from Eq. (3.4), without considering the communication delays. The results are shown in Figure 3.17. As illustrated in the upper-left plot, the followers first track the leader's motion in a line, and perform a circular motion to turn around, followed again by a straight line path. The switch signal between different leader's motion is given by the operator. The formation parameters ρ, α, ϕ are calculated according to the recorded status packets of the robots with synchronized timestamps. With the desired separation distance set to be 1.5 meters, both followers are maintaining the distances ρ around 1.7 meters under the delay from 0.4 to 0.8 seconds. The followers are simply trying to keep a formation with the delayed leader, as illustrated in Figure 3.4. Similar deviations from the desired values are to be found for the bearing angle α. In principle, this steady-state error is caused by data inconsistency, which is also illustrated through the result in previous simulation study shown in Figure 3.13. This points out one of the problems in formation tracking under time-delay in data transmission.

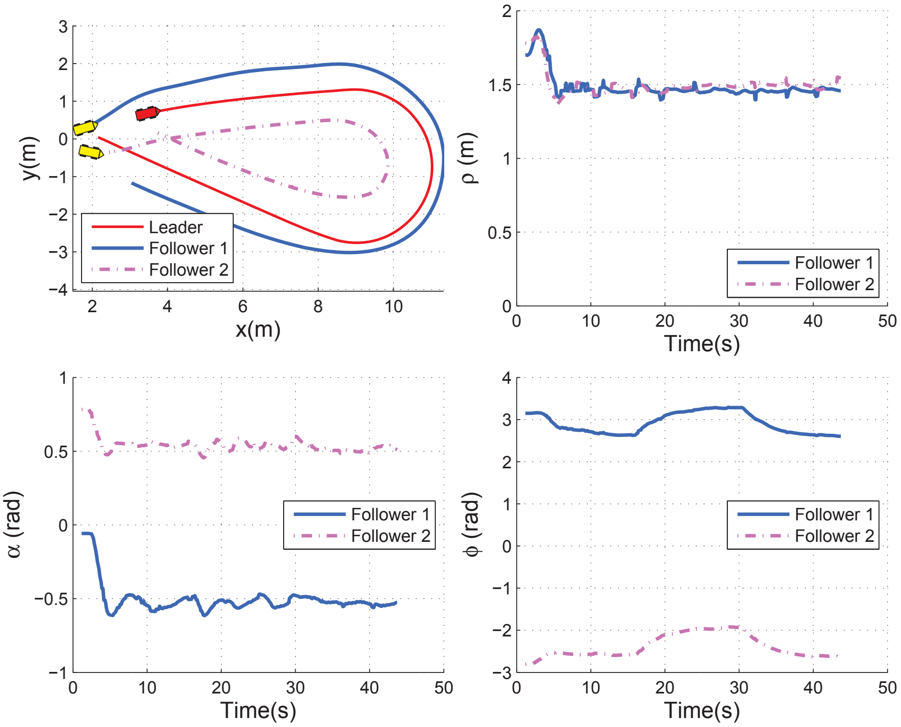


Figure 3.18: MERLIN formation with PD-Type controller; the parameters used in the experiment are as follows: $\rho_d = 1.5m, \alpha_d = \pm\pi/6, k_\rho = 1.2, k_\alpha = 1.2, d_\rho = -0.135, d_\alpha = -0.135;$ delay $= 0.4 - 0.8s$. Leader speed is regulated at 0.5 m/s and steering angle at 5 deg during circular motion.

To demonstrate the effectiveness of the proposed controller, a similar experiment with three MERLIN robots under the delay-compensating control law is implemented, which corresponds to the idea from Figure 3.5. Figure 3.18 shows the resulting improvement. Clearly, the tracking performance is achieved according to the given specifications ($\rho \to 1.5m$ and $\alpha \to \pm\pi/6$), as shown in Table 3.1. The fact that no steady-state error and oscillating behavior arise during the experiment has demonstrated the applicability and effectiveness of the proposed control scheme for delayed formation tracking systems.

Note that the controller parameters for the proportional gain are larger than the ones given in the comparative simulation shown in Figure 3.15. One reason is that the smaller proportional gain leads to a longer convergence time, especially when the robots need to conquer the static friction from the ground and compensate for the actuator delays. Although the higher proportional gain brings less smoothness to the system (some small overshoots can be observed in the experiments), the faster reaction under the implemented controller is also desired in real-world experiment.

Table 3.1: Tracking errors of the two followers in the experiment show in Figure 3.18. The nominal value for ρ is $\rho_d = 1.5$m, and for α is $\alpha_d = \pm 30$deg.

	Separation Distance [m] (mean ± standard deviation)	Bearing Angle [deg] (mean ± standard deviation)
Follower 1	1.4586 ± 0.02	-30.31 ± 1.8
Follower 2	1.4806 ± 0.03	30.50 ± 1.5

This trade-off should be balanced according to the specific high-level mission together with low-level hardware requirements.

Table 3.1 shows the statistical evaluation of the formation tracking experiment in Figure 3.18. The values are computed over the time period from approximately 5 seconds to the end of the experiment. The data at the beginning of the experiment are not considered for this evaluation, since the followers need to form the desired formation from the initial positions. The small tracking offsets from the desired values shown in the table are resulted from two sources. One is the systematic offsets in robot speed and steering control, as these two states can be affected by the calibration errors in mechanical construction. The other source is the sampling period of the sensors at 0.1s, which brings offset to the measured values. This value on sampling period is limited by the capability of the onboard microcontroller. Nevertheless, with the nominal speed of the leader at 0.5 m/s, the standard deviations for separation distance at the magnitude of centimeter and for bearing angle under 2 degrees show the reliable performance of the designed controller in the time-delayed leader-follower formation system.

3.7.3 Non-cooperative Formation Tracking using Visual Data

The developed PD-type control law under communication delays in formation control is not restricted in compensating steady-state errors for cooperative missions. To further showcase the applicability and the value of the proposed control scheme, we present a non-cooperative formation tracking scenario using visual data. The importance of the derivative term in the controller, together with the parameter selection criteria, is demonstrated through comparative experimental studies in this section.

Figure 3.19 illustrates the outdoor version of the MERLIN robot, equipped with high-precision measuring unit, iSpace [55], and a visual system consisting of two different types of cameras. The iSpace system is used for providing ground truth data of the formation experiment. Both the leader and follower robots are equipped with

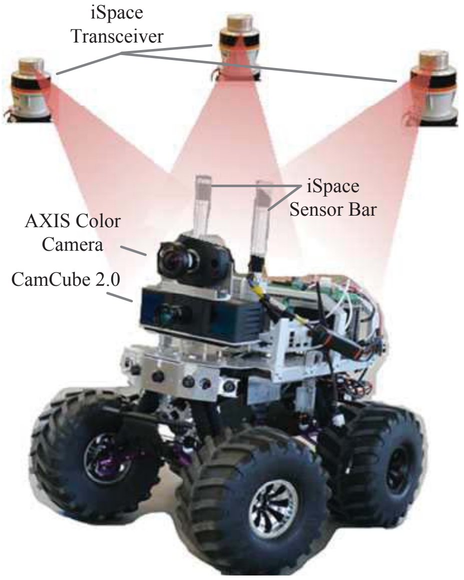


Figure 3.19: Hardware setup on Outdoor MERLIN for formation tracking using visual data, and high-precision measuring unit for evaluation.

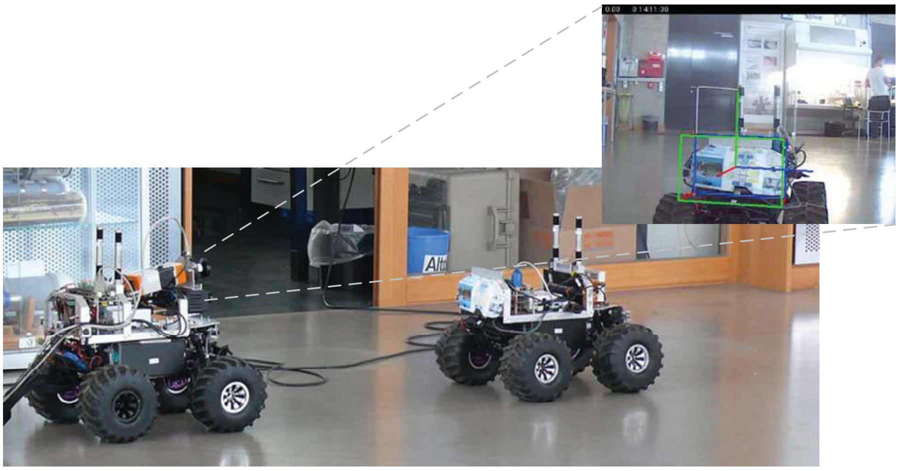
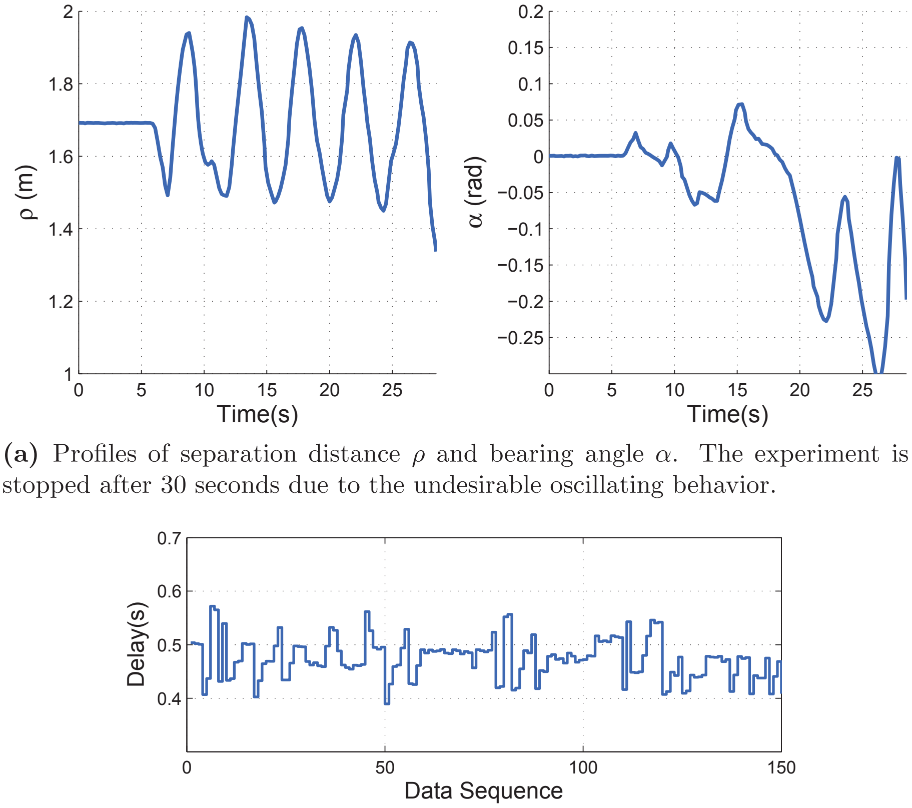


Figure 3.20: Snapshot of the initial formation setup. The camera perspective from the follower robot is also displayed in the figure. The bounding box in color is the estimated 6D posed of the target robot.

the laser transmitters, which can be positioned under millimeter accuracy through the fixed laser headers in the test field. The visual system, providing the 6 degrees of freedom (DoF) pose estimation of the target robot, processes RGB-Depth data based on the sensor fusion between a 3D Time-of-Flight (ToF) camera and a 2D color camera. Figure 3.20 illustrates the initial setup of the formation tracking system. The estimated pose of the target robot is shown through the colored bounding box. The relative pose between the two robots (for the calculation of ρ, α, ϕ) and the estimated



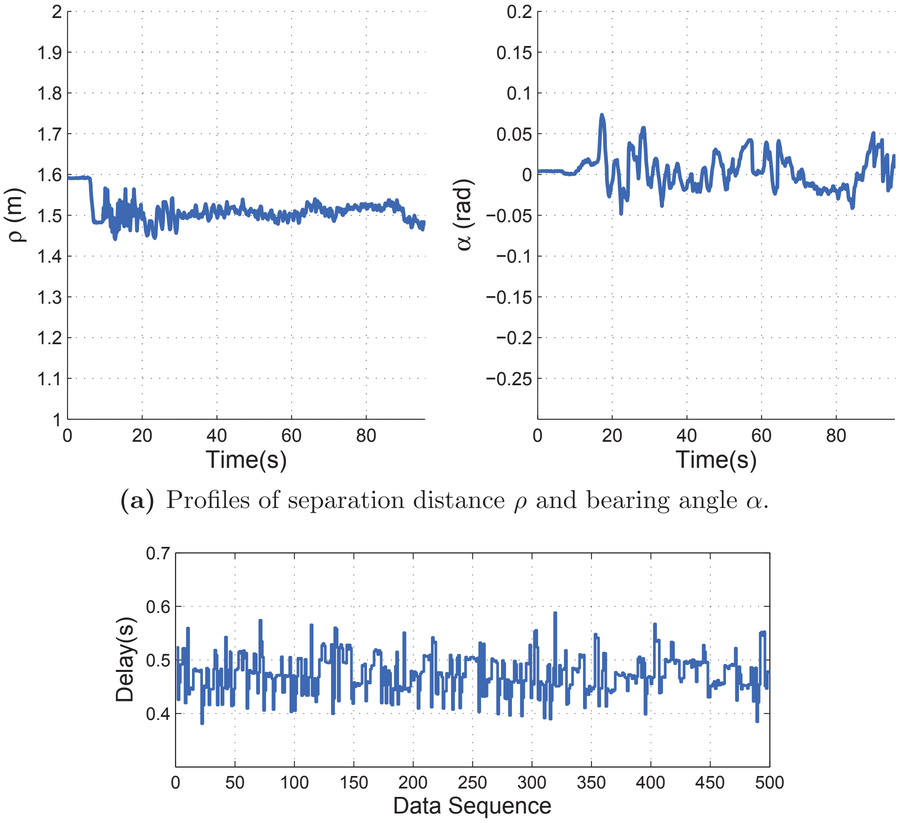
(a) Profiles of separation distance ρ and bearing angle α. The experiment is stopped after 30 seconds due to the undesirable oscillating behavior.

(b) The varying time delays from the captured visual frame till the computed driving commands, ranging approximately from $0.4 - 0.6s$.

Figure 3.21: Formation tracking results under P-type controller given in Eq. (3.12) to achieve the linear system of Eq. (3.11). Controller parameters are set as: $\rho_d = 1.5m, \alpha_d = 0, k_\rho = 1.5, k_\alpha = 1.5$. The target robot is driving at 0.3 m/s.

target speed (v_L) are considered as the inputs for the delay-compensating formation tracking controller proposed in Section 3.4.2. The details on estimating the 6D pose of an arbitrary object can be found in the paper by Sun et al. [134].

Note that the delay source in this experimental setup is different from the one discussed in Section 3.7.2, since there is no communication between the two mobile robots. Instead, as pointed out in [134], the vision-based pose estimation algorithm under this setup creates a noticeable delay in computation. Therefore, the captured target information is considered delayed due to this high computational demand when it is used for calculation of control commands. Since the pose estimation algorithm directly produces the relative pose information, i.e. $\rho^{\mathrm{d}}, \alpha^{\mathrm{d}}$, and ϕ^{d}, as well as the target speed ${v_L}^{\mathrm{d}}$, there is no data inconsistency between the target and the follower states. Therefore, no steady-state error in the tracking is to be expected. However, the



(a) Profiles of separation distance ρ and bearing angle α.

(b) The varying time delays from the captured visual frame till the computed driving commands, ranging approximately from $0.4 - 0.6s$.

Figure 3.22: Formation tracking results under P-type controller given in Eq. (3.12) to achieve the linear system of Eq. (3.11). Controller parameters are set as: $\rho_d = 1.5m, \alpha_d = 0, k_\rho = 1.5, k_\alpha = 1.5, d_\rho = -0.135, d_\alpha = -0.135$. The target robot is driving at 0.3 m/s, with steering angle at 5 deg during circular motion.

analysis on the stability region of the controller parameters is utilized to understand the oscillating behavior of the tracking system, as well as to improve the desired formation tracking specifications.

In the following, we compare the P-type controller with the PD-type controller through the experimental setup described above, so as to show the effectiveness of the proposed scheme on suppressing the oscillating behavior of the system by referring to the derived stability regions. The trajectory of the semi-autonomous target robot is similar to the one from indoor MERLIN experiment in Section 3.7.2, which consists of both linear and circular motion.

Figure 3.21 provides the resulting data during the formation tracking experiment. Similar to the simulated case in Figure 3.14, the P-type controller causes undesired

Table 3.2: Tracking errors of the follower in the experiment show in Figure 3.22. The nominal value for ρ is $\rho_d = 1.5$m, and for α is $\alpha_d = 0$deg.

	Separation Distance [m] (mean ± standard deviation)	Bearing Angle [deg] (mean ± standard deviation)
Follower	1.5049 ± 0.02	0.22 ± 1.24

oscillating behavior in both ρ and α states due to the location of the dominant root of the linearized system. Note that the system is not diverging yet, since the product of $k \cdot T$ is within the stable parametric region shown in Figure 3.10. Therefore, the next experiment is performed with the PD-type controller to relax the constraint on the choices of controller parameters.

Figure 3.22 shows the experimental results from the formation tracking using PD-type controller. The initial separation distance is approximately $1.6m$. Both robots start moving after about 5 seconds. Obviously, the follower robot is able to maintain the desired separation distance and bearing angle. The deviations in the figure come from the sensor noises, as well as the ground frictions during driving.

Table 3.2 provides the statistical data of the experiment performed with PD-type controller. Different from the experiments in Section 3.7.2, the evaluated states ρ and α are obtained through the high-precision iSpace system, rather than using dead-reckoning method. Therefore, the tracking offsets are largely reduced compared to Table 3.1. The mean and standard deviation values are computed over the time period from approximately 10 seconds to the end of the experiment, which represents the nominal operation time frame of the controller.

3.8 Summary and Discussion

We have studied the time-delay problem in the formation control of multiple car-like robots, and developed a stable PD-type controller to achieve trajectory tracking using feedback linearization method in a leader-follower formation framework. The stability of the system is guaranteed by the parameter selection rules for the controller, and by the convergence of internal dynamics. We derived the convergence theorems of the parameters based on the analysis of transcendental characteristic equations of time-delayed systems. The robustness of the controller against varying time delays is shown through both simulations and experiments.

The parameter design methodology introduced in this chapter also applies to general systems that are feedback linearizable and suffer from time-delay effects. The

PD-type controller can be substituted by other types of controllers, such as sliding mode controller, although the stability analysis with the advanced controllers would be more complicated.

As mentioned in Section 3.1, the delay-compensating framework developed in this chapter can also be applied to robotic teleoperation, including near-space satellite operation, where the time-delay effect is inevitable. In this case, the delay between the ground control center and the satellites can be modeled as the delay between the leader and follower satellites, where the leader satellite can be virtually simulated on the ground side. This closely corresponds to the experimental study given by Palafox and Spong [111], where the steady-state error problem is not considered. With the delay-compensating PD-type controller extended to the teleoperation scheme, the slave formation can track the master positions without offsets.

Another extension of the proposed control scheme is the non-cooperative vehicle-following systems, such as the one studied by Goi et al. [49]. Section 3.7.3 demonstrates the ability of the controller in this similar direction, although the experiment is performed under relatively low speed and close separation distance. More applications include adaptive cruise control (ACC) technology in automotive industry, where the complex sensor systems on the vehicle may create some undesired information delay due to the processing time. This amount of delay can be similarly handled by the proposed control scheme, as shown in Section 3.7.3, so that the real-time cruise control can be further improved.

CHAPTER 4

OPTIMAL INITIAL CONDITION – READINESS ANALYSIS

This chapter explores the performance for a formation system in response to inputs (or disturbances) from outside of the formation. In order to analyze the system performance, the notion of *readiness* is proposed, which describes the initial conditions of the formation in terms of a certain set of input space that stems from external sources. A higher readiness means the system is better prepared with the corresponding set of initial conditions to maintain or recover the original formation shape against exogenous inputs (or disturbances). For deriving the optimality of the initial conditions, optimization method based on calculus of variations is utilized. The meaning and features of readiness notion are demonstrated on car-like mobile robot formation through simulations.

In the following sections, we first present an overview of the research work which inspire us in the derivation of readiness and put forth the mathematical representation (Section 4.1). Then we provide a case study of readiness on car-like mobile robot formation using a distance-based control approach (Sections 4.2.1 and 4.2.2). Afterward we formulate the optimality conditions for the optimal initial orientations in the formation (Section 4.2.3). Different initial conditions are compared with each other through simulations (Section 4.2.4). At the end of this chapter, we provide a short summary and discussion on the potential applications of the readiness notion (Section 4.3).

4.1 Definition

This section reviews some recent literatures on the multi-agent system control, which form the inspiration for the proposition of readiness notion. We also define the readiness notion with a mathematical formulation and provide the optimality conditions in a general form that solve the readiness optimization problem.

4.1.1 Inspiration

The studies on multi-agent system control in general are reflected in all sorts of social aspects (refer to Section 2.1). In particular, robotic formation control has been an attractive topic, especially for traffic control, underwater or space missions. One interesting problem in formation control is the study on the relationship between the system response and certain conditions in the system, e.g. positions of agents or system topology. Many existing concepts can be used to characterize this relationship, such as controllability [121, 135], rigidity and stiffness [72, 158], manipulability [64], and responsiveness [66].

Controllability describes the point-to-point property of a network system. It examines whether or not the agents in the system can be moved from one configuration to another by certain set of permissible controls. This issue is introduced in leader-follower multi-agent systems by Tanner [135], and later on Rahmani et al. [121] provide an analysis on this issue from a graph-theoretic point of view, which shows the relation between the symmetry structure of the network topology and its controllability. Other than the point-to-point property of the system, the measure of instantaneous response of the system to external perturbation is also studied under several concepts. The notion of rigidity is used to describe whether or not the system behaves as a rigid body with fixed-length links between agents in a network. Zhu and Hu [158] extend this notion in a rigid graph to stiffness and rigidity indices, which measure the ability of control laws in maintaining the shape of the formation system. Another property of multi-agent systems, manipulability, is introduced by Kawashima and Egerstedt [64] in the leader-follower formation context. It indicates the amount of effort to move the leader agent in a certain direction given various network topologies. The concept of manipulability is further applied to provide criteria for selecting effective leaders in leader-follower networks by identifying the influence of leader's inputs on the network centroid [65]. These notions to interpret instantaneous network response are unified by Kawashima et al. [66] in responsiveness, which provides comprehensive characterization of the choices of both leader nodes and types of interaction topology in designing the control laws for the system.

Building upon these concepts for evaluating the performance of multi-agent systems, we propose the notion of readiness. It is yet another index to characterize the properties of multi-agent systems. Readiness refers to the initial conditions of a formation system in terms of a certain set of input space. It describes how well the system is prepared for a variety of external disturbances. An optimal readiness means the system is prepared with the optimized set of initial conditions to maintain or recover the original formation shape against exogenous disturbances in the shortest possible period of time. On the other hand, if the system fails to recover the original formation shape due to the external perturbations in a decent amount of time, it has therefore low readiness.

Readiness notion differs from the above-mentioned existing concepts, where only single-integrator models with two-dimensional states (position) are assumed. For a more general type of robotic models, such as nonholonomic robots, the notion of readiness provides the measure on extra dimensions of states. For instance, in a car-like robot formation, apart from the positions of the agents, their headings or orientations are also part of the states, which affect the system performance. Previous indices, such as manipulability or responsiveness, fail to describe the properties of this type of systems due to the nonholonomic constraints. However, with the readiness notion, a car-like robot formation can be characterized in terms of the initial orientations, which are part of the initial conditions of the system.

In the following, we interpret the readiness notion from a mathematical point of view, and provide the optimality conditions in a general form for solving the readiness optimization problem.

4.1.2 Readiness Formulation

The following notations are needed in order to formulate the mathematical form of readiness .

Consider a group of N agents, and the state $x_i(t) \in \mathbb{R}^d$ is associated with each agent at time t, where $i = 1, \ldots, N$. Denote the overall system configuration in the stack form by $x \triangleq [x_1^\top, ..., x_N^\top]^\top$. The dynamics of the agents are given by:

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

with the initial states $x(0) = x_0 \in \mathbb{R}^d$, where $u \in U$ is the exogenous input to the formation system, and U represents the set of input space.

Take the following cost functional for the optimal initial condition problem [73]:

$$\tilde{J}(x_0, u) = \int_0^T L(x(t), u)dt + \Psi(x(T), u), \tag{4.2}$$

where $L(x(t), u)$ is an instantaneous cost on states and input u, and $\Psi(x(T), u)$ is a terminal cost function. These two cost functions are designed according to the system requirements. For instance, the instantaneous cost can punish the control effort spent in the system, or the inter-vehicle distances to keep the vehicles apart from each other during movement, while the terminal cost can be used to penalize the difference between desired formation shape and the shape at time T.

Now we formulate the readiness by taking the integral (parameterized family) of the cost functionals $\{\tilde{J}(x_0, u) | u \in U\}$ over the input space to evaluate the overall response of the system. The following cost functional is defined for readiness:

$$J(x_0) = \int_U \left(\int_0^T L(x(t), u) dt + \Psi(x(T), u) \right) du. \tag{4.3}$$

$J(x_0)$ punishes the general integral cost under inputs $u \in U$. If $J(x_0)$ is small, then the system with the corresponding x_0 has high readiness. In other words, we would like to examine how the overall system is prepared for a variety of perturbation, parameterized by u acting on the system. We would also like to know what the optimal initial conditions are, which meets the system design requirements indicated by the cost functional $J(x_0)$.

We can follow the approach for solving the optimal initial conditions from [73], and write the solution as x_0^* given by:

$$\begin{aligned} x_0^* &= \arg\min_{x_0} J(x_0) \qquad (4.4)\\ \text{s.t.} \quad \dot{x}(t) &= f(x(t), u) \\ x(0) &= x_0. \end{aligned}$$

The following theorem states the optimality condition of readiness with respect to the initial condition x_0, which can be used to solve the problem given in Eq. (4.4) by different numerical methods, such as gradient decent method [12] .

Theorem 4.1. *The optimality conditions in terms of first-order necessary conditions (FONC) for the optimization problem described in Eq.* (4.4) *are given by:*

$$\frac{\partial J}{\partial x_0} = \int_U \lambda^\top(0, u) du = 0, \tag{4.5}$$

where

$$\begin{cases} \dot{\lambda}(t, u) &= -\dfrac{\partial L}{\partial x}^\top (x(t), u) - \dfrac{\partial f}{\partial x}^\top (x(t), u)\lambda(t, u) \\ \lambda(T, u) &= \dfrac{\partial \Psi}{\partial x(T)}^\top (x(T), u) \end{cases} \tag{4.6}$$

with the costate $\lambda \in \mathbb{R}^{dN}$.

Proof. We take the advantage of the integral form of the parameter u in Eq. (4.3), and derive the optimality conditions using calculus of variations. See Appendix B for the details. □

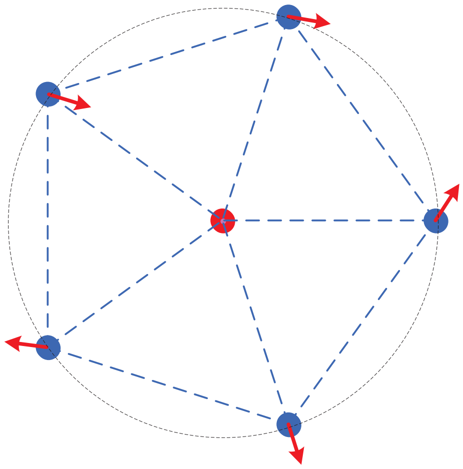


Figure 4.1: Depiction of a leader-follower network, with the followers lining up on a circle and the leader at the center; blue discs represent the nonholonomic follower robots, with their headings indicated by the red arrows; the red disc in the middle is the leader robot; blue dashed lines are the communication links.

4.2 Case Study

In this section, we present an example of multi-robot control system to illustrate the usage of the readiness notion. We first apply a distance-based control to a leader-follower formation system, and demonstrate the optimization process of readiness. We also compare the results of optimized readiness with other initial conditions to show the importance of the readiness in a multi-agent system.

4.2.1 Problem Formulation

Consider the scenario depicted in Figure 4.1, where the nonholonomic robots form a circular formation in a leader-follower structure, with the leader in the middle of the circle. The communication topology is established among the direct neighbors of the follower robots, as well as between the leader and each follower. Suppose the follower robots are evenly distributed on the circle with different orientations. The formation system is expected to maintain the communication topology by holding the shape indicated in the figure. In case the leader robot at the centroid of the circle moves in some direction on the plane, the follower robots are supposed to move accordingly to locate the leader robot back to the centroid.

Denote the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for Figure 4.1, where $\mathcal{V} = \{\mathsf{v}_1, \ldots, \mathsf{v}_N\}$ denotes the set of robots in the graph, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, indicated by the blue dashed lines. The cardinality of $\mathcal{E}$ is $|\mathcal{E}| = M$, which is the number of edges in the graph. Based on the readiness definition given in Eq. (4.3), we consider the movement of the leader as the perturbation to the system, which is represented by the arbitrary input $u \in U$. We also reformulate the dynamics of the agents by separating followers and leader, as well as positions and orientations, respectively:

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_f(t) \\ \dot{x}_l(t) \end{bmatrix} = \begin{bmatrix} g_x(x, \theta_f) \\ g_{xl}(x_l, u) \end{bmatrix}, \qquad \dot{\theta}(t) = \begin{bmatrix} \dot{\theta}_f(t) \\ \dot{\theta}_l(t) \end{bmatrix} = \begin{bmatrix} g_\theta(x, \theta_f) \\ g_{\theta l}(x_l, u) \end{bmatrix}, \tag{4.7}$$

where we assign the agent with the last index as the leader, i.e., $x_l \triangleq x_N$, and the remaining $N_f = N - 1$ agents as followers, i.e., $x_f = [x_1^\top, \dots, x_{N_f}^\top]^\top$, $x_i \in \mathbb{R}^2$. $\theta_f = [\theta_1, \dots, \theta_{N_f}]^\top$ and $\theta_l \triangleq \theta_N, \theta_i \in \mathbb{R}, i = 1, \dots, N$. Here the initial condition in readiness notion refers to the initial orientations of the followers, i.e.: $\theta_0 \triangleq \theta_f(0)$, and the initial positions are predefined, i.e. $x(0) \in X \triangleq \{x \in \mathbb{R}^{2N}\}$. If the agents are evenly distributed on the circle, we define the position set X^* as

$$X^* \triangleq \{x \in \mathbb{R}^{2N} | \|x_i - x_j\| = d_{ij}, \{\mathsf{v}_i, \mathsf{v}_j\} \in \mathcal{E}\}, \tag{4.8}$$

where d_{ij} is the desired distance between agent i and j.

4.2.2 System Modeling and Formation Control Design

Before we analyze the optimal readiness in terms of the orientations of the agents, we need to introduce our control strategy, which is often referred to as edge-tension energy based formation control [98]. The basic idea is to form an energy function based on the distances of each pair of neighboring agents. The energy increases if the desired formation shape is violated, thus forcing the agents to maintain the formation to keep the "tension" as low as possible.

4.2.2.1 Edge-tension Energy

Similar to the formulation in [66], we introduce the following general edge-tension energy:

$$E(x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} E_{ij}(x_i(t), x_j(t)), \tag{4.9}$$

where

$$E_{ij}(x_i, x_j) = \begin{cases} \frac{1}{2} \left(e_{ij}(||x_i - x_j||)\right)^2 & \{\mathsf{v}_i, \mathsf{v}_j\} \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

Here, $e_{ij} : \mathbb{R}_+ \to \mathbb{R}$ is a strictly increasing, twice differentiable function such that $e_{ij}(d_{ij}) = 0$ and $e'_{ij}(d_{ij}) \neq 0$, where $d_{ij} > 0$ is the desired distance between agents i

and j, and $e'_{ij}(r) \triangleq \dfrac{de_{ij}(r)}{dr}$, if $\{\mathsf{v}_i, \mathsf{v}_j\} \in \mathcal{E}$. A typical example of e_{ij} is

$$e_{ij}(||x_i - x_j||) = c_{ij} \cdot (||x_i - x_j|| - d_{ij}) \, , \tag{4.11}$$

where $c_{ij} > 0$ is a weighting factor assigned to the edge $\{\mathsf{v}_i, \mathsf{v}_j\}$.

Let $D(\mathcal{G}) \in \{-1, 0, 1\}^{N\times M}$ be the incidence matrix of graph $\mathcal{G}$. The first and the second derivatives of the edge-tension energy is given by the following[1]: (see [64] for the details of the derivation):

$$\begin{aligned} \frac{\partial E(x)}{\partial x}^{\top} &= ((DW_1(x)D^{\top}) \otimes I_d)x, \\ \frac{\partial^2 E(x)}{\partial x^2} &= (DW_1(x)D^{\top}) \otimes I_d + R(x)^{\top} W_2(x) R(x) \end{aligned}$$

where $W_1(x)$ and $W_2(x)$ are $M \times M$ diagonal matrices, whose diagonal elements are

$$\begin{aligned} [W_1(x)]_{kk} &= w_{i_k j_k}(||x_{i_k} - x_{j_k}||), \\ [W_2(x)]_{kk} &= \frac{w'_{i_k j_k}(||x_{i_k} - x_{j_k}||)}{||x_{i_k} - x_{j_k}||}, \\ & \quad k = 1, ..., M, \quad \{\mathsf{v}_{i_k}, \mathsf{v}_{j_k}\} : \text{edge } k, \end{aligned}$$

where we define

$$\begin{aligned} e''_{ij}(r) &\triangleq \frac{d^2 e_{ij}(r)}{dr^2}, \\ w_{ij}(r) &\triangleq \frac{e_{ij}(r) e'_{ij}(r)}{r}, \\ w'_{ij}(r) &\triangleq \frac{dw_{ij}}{dr} = \frac{\{e'_{ij}(r)^2 + e_{ij}(r)e''_{ij}(r)\}r - e_{ij}(r)e'_{ij}(r)}{r^2}. \end{aligned}$$

Here we assume that the indices of the edges are consistent between $W_1(x)$ and the incidence matrix D, and between $W_2(x)$ and the rigidity matrix R, respectively.

Remark: If all the desired distances are satisfied at $x = x^*$, then $\left.\dfrac{\partial E}{\partial x}\right|_{x^*} = 0$ and

$$\left.\frac{\partial^2 E}{\partial x^2}\right|_{x^*} = R(x^*)^{\top} W_2(x^*) R(x^*). \tag{4.12}$$

By the definition of the function e_{ij} and the fact that $||x_i - x_j|| = d_{ij} > 0$ at $x = x^*$, $W_2(x^*)$ is always positive definite.

Example 4.1. *If the edge-tension energy is given by (4.11), then* $e'_{ij}(r) = c_{ij}$, $e''_{ij}(r) = 0$, $[W_1(x)]_{kk} = c^2_{i_k j_k}(1 - d_{i_k j_k}/||x_{i_k} - x_{j_k}||)$, *and* $[W_2(x)]_{kk} = c^2_{i_k j_k} d_{i_k j_k}/||x_{i_k} - x_{j_k}||^3$.

[1] operator $\otimes$ indicates *Kronecker product.*

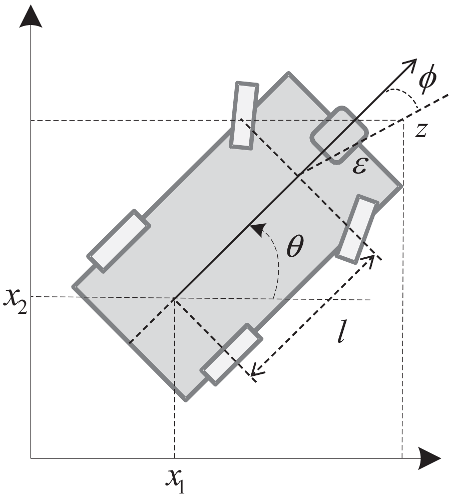


Figure 4.2: Car-like mobile robot model with off-center point z as reference

Hence, when the desired distances are satisfied at $x = x^*$, the k-th diagonal elements of the weight matrices become

$$[W_1(x^*)]_{kk} = 0, \quad [W_2(x^*)]_{kk} = \left(\frac{c_{i_k j_k}}{d_{i_k j_k}}\right)^2. \tag{4.13}$$

Finally, the edge-tension energy from Eq. (4.9) is applied to the system in (4.7) through the following dynamics:

$$\mu_i(x) \triangleq \dot{x}_i = -\frac{\partial E}{\partial x_i}^\top(x) = -\sum_{j \in \mathcal{N}_i} w_{ij}(x_i - x_j). \tag{4.14}$$

In terms of the followers, the dynamics can be derived as:

$$\dot{x}_f = -\frac{\partial E}{\partial x_f}^\top(x) = -((D_f W_1(x) D^\top) \otimes I_2)x. \tag{4.15}$$

4.2.2.2 Control Design with Car-like Robots

The model of the vehicle we use in this example is the same car-like robot mentioned in Section 2.3.1. Due to the nonholonomic constraints, we need to use an off-center point of the model as the reference (for complete controllability) in order to apply the distance-based control described in Section 4.2.2.1. Rewrite the kinematic equation of car-like robot as:

$$\begin{bmatrix} \dot{x}_{1i}(t) \\ \dot{x}_{2i}(t) \\ \dot{\theta}_i(t) \end{bmatrix} = \begin{bmatrix} \cos\theta_i(t) \\ \sin\theta_i(t) \\ \dfrac{\tan\beta_i}{l} \end{bmatrix} v_i(t), \tag{4.16}$$

A typical choice of the off-center point is defined by Luca et al. [88, pp.200], which is indicated in Figure 4.2 with the point z. Now we can represent z_i for each agent in the formation by:

$$z_i = \begin{bmatrix} x_{1i} + l\cos(\theta_i) + \epsilon\cos(\theta_i - \phi) \\ x_{2i} + l\sin(\theta_i) + \epsilon\sin(\theta_i - \phi) \end{bmatrix}, \tag{4.17}$$

where $-\dfrac{\pi}{2} < \phi < \dfrac{\pi}{2}$ is the off-center angle, and $\epsilon > 0$ is the off-center length, which is typically small enough. The time-derivative of the off-center point follows:

$$\dot{z}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) - \epsilon\sin(\theta_i - \phi) \\ \sin(\theta_i) & \cos(\theta_i) + \epsilon\cos(\theta_i - \phi) \end{bmatrix} \begin{bmatrix} v_i \\ v_i \dfrac{\tan\beta_i}{l} \end{bmatrix} \tag{4.18}$$

Combining Eq. (4.14) and Eq. (4.18), we can achieve $\dot{z}_i = \mu_i$ by the following control law:

$$\begin{bmatrix} v_i \\ v_i \dfrac{\tan\beta_i}{L} \end{bmatrix} = \frac{1}{1 + \epsilon\cos(\phi)} \begin{bmatrix} p_1(\theta_i)^\top \\ p_2(\theta_i)^\top \end{bmatrix} \mu_i, \tag{4.19}$$

where

$$\begin{aligned} p_1(\theta_i) &\triangleq [\cos(\theta_i) + \epsilon\cos(\theta_i - \phi), \sin(\theta_i) + \epsilon\sin(\theta_i - \phi)]^\top, \\ p_2(\theta_i) &\triangleq [-\sin\theta_i, \cos\theta_i]^\top. \end{aligned}$$

Now we can write the formation kinematics in terms of the followers with the control in Eq. (4.19) in a stack form:

$$\begin{cases} \dot{x}_f = g_x(x, \theta_f) \triangleq -\dfrac{1}{1 + \epsilon\cos(\phi)} P_3(\theta_f) P_1(\theta_f)^\top \dfrac{\partial E}{\partial x_f}^\top (x) \\ \dot{\theta}_f = g_\theta(x, \theta_f) \triangleq -\dfrac{1}{1 + \epsilon\cos(\phi)} P_2(\theta_f)^\top \dfrac{\partial E}{\partial x_f}^\top (x) \end{cases} \tag{4.20}$$

where $P_j \in \mathbb{R}^{2N_f \times N_f}, j = 1, 2, 3$, is defined as:

$$P_j(\theta_f) \triangleq \begin{bmatrix} p_j(\theta_1) & & & \\ & p_j(\theta_2) & & \\ & & \ddots & \\ & & & p_j(\theta_{N_f}) \end{bmatrix}, \tag{4.21}$$

and further

$$p_3(\theta_i) \triangleq [\cos\theta_i, \sin\theta_i]^\top.$$

In the following, we prove the local stabilization of the formation system under the dynamics of Eq. (4.20).

Proposition 4.1 (Formation stabilization under edge-tension energy)**.** *The formation system with car-like mobile robot model given in Eq.* (4.16) *is locally stable under the control law given in Eq.* (4.20)*.*

Proof. Consider the Lyapunov function candidate:

$$V(x) = E(x),$$

where $E(x)$ is the edge-tension energy defined in Eq. (4.9). Differentiating V with respect to time:

$$\begin{aligned}\dot{V}(x) = \dot{E}(x) &= \frac{\partial E}{\partial x_f}\dot{x}_f \\ &= -\frac{\partial E}{\partial x_f} P_1(\theta_f) P_3(\theta_f)^\top \frac{\partial E}{\partial x_f}^\top \\ &= -\frac{\partial E}{\partial x_f} P_1(\theta_f) \left[P_1(\theta_f)^\top + \epsilon P_1(\theta_f - \phi\mathbf{1})^\top\right] \frac{\partial E}{\partial x_f}^\top \\ &= -\left\| P_1(\theta_f)^\top \frac{\partial E}{\partial x_f}^\top \right\|^2 - \epsilon \frac{\partial E}{\partial x_f} P_1(\theta_f) P_1(\theta_f - \phi\mathbf{1})^\top \frac{\partial E}{\partial x_f}^\top . \end{aligned} \tag{4.22}$$

The eigenvalues of the matrix $P_1(\theta_f)P_1(\theta_f - \phi\mathbf{1})^\top$ can be calculated as 0 and $\cos(\phi)$. Since $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$, we know that $P_1(\theta_f)P_1(\theta_f - \phi\mathbf{1})^\top \succeq 0$. Therefore, $\dot{V} \leq 0$, which ensures local stability of the system in Eq. (4.20). □

Remark: While in general the formation system in Eq. (4.20) may be driven toward a local equilibrium set, we here consider only small perturbations of the agents from their desired formation, which is defined by the set X^* in Eq. (4.8). Based on Example 4.1, we know $\|x_i - x_j\| = d_{ij} \Rightarrow \dot{V} = 0$. Therefore, the set X^* is one equilibrium set.

4.2.3 Readiness Optimization

Following the scenario depicted in Figure 4.1, each agent in the system has the dynamics of a car-like robot model given in Eq. (4.16). Suppose the system initial configuration satisfies the desired distances of:

$$x = x^* = \begin{bmatrix} x_f^* \\ x_l^* \end{bmatrix}, \tag{4.23}$$

with $\|x_i - x_j\| = d_{ij}, \forall\{\mathsf{v}_i, \mathsf{v}_j\} \in \mathcal{E}$. Note that the edge set $\mathcal{E}$ represents the communication link shown in Figure 4.1, which does not change with time. Let δx_l be the instantaneous movement of the leader at time $t = 0$, given by:

$$\delta x_l = \gamma \begin{bmatrix} \cos\theta_l \\ \sin\theta_l \end{bmatrix}, \tag{4.24}$$

where $\gamma > 0$ is the small perturbation distance, and θ_l is the direction of the movement. The leader will not move after this displacement, i.e.,

$$x(0) = x^* + \begin{bmatrix} 0 \\ \delta x_\ell \end{bmatrix}, \quad x(t) = x^* + \begin{bmatrix} \delta x_f(t) \\ \delta x_\ell \end{bmatrix}, \tag{4.25}$$

where $\delta x_f(t) \triangleq x_f(t) - x_f^*$ denotes the position change of the followers.

Now we associate the following cost with the set of initial orientations of all followers:

$$J(x_0) = \int_U \Psi(x(T), u)du, \tag{4.26}$$

where we let $L = 0$ in Eq. (4.3), since we only focus on the final formation shape for simplicity. An example of $f(x, u)$ and $\Psi(x, u)$ is $\dot{x} = -\frac{\partial E}{\partial x}^\top(x, u)$ and $E(x(T), u)$, respectively. The exogenous input u is defined as the direction of leader's movement, i.e.

$$u \triangleq \theta_l, \qquad \text{and } U \triangleq [0, 2\pi].$$

In this case, $J(x_0)$ is an index to measure the change of edge-tension energy E in the system under all given $u \in U$. Note that although u is part of the formation state x, we write explicitly in the energy function as $E(x, u)$ for clarification purpose.

The optimal initial condition problem in terms of optimizing the readiness of the system depicted in Figure 4.1 is given by:

$$\begin{aligned} \min_{\theta_0} \quad & J(\theta_0) = \int_0^{2\pi} \Psi(x(T), u)du \\ \text{s.t.} \quad & \dot{x} = \begin{bmatrix} \dot{x}_f \\ \dot{x}_l \end{bmatrix} = \begin{bmatrix} g_x(x, \theta_f, u) \\ 0 \end{bmatrix} \\ & x_f(0) = x_f^*, \quad \theta_f(0) = \theta_0 \\ & x_l(0) = \delta x_l, \quad \theta_l(0) = \theta_l \end{aligned} \tag{4.27}$$

where $\Psi(x(T), u) = E(x(T), u)$. Under Theorem 4.1, we can obtain the first-order necessary conditions as

$$\frac{\partial J}{\partial \theta_0} = \int_0^{2\pi} \xi_\theta^\top(0, u)du = 0, \tag{4.28}$$

where $\xi = [\xi_x^\top, \xi_\theta^\top]^\top \in \mathbb{R}^{2N_f \times N_f}$ is the costate associated with the states x_f and θ_f, whose dynamics are given by:

$$\begin{cases} \dot{\xi}(t,u) &= -\dfrac{\partial g_f}{\partial x_f}^\top (x(t),u)\xi(t,u) \qquad (4.29\text{a}) \\ \xi(T,u) &= \dfrac{\partial \Psi}{\partial x_f}^\top (x(T),u) \qquad (4.29\text{b}) \end{cases}$$

More specifically, the costates can be described as:

$$\xi_x(T,\theta_l) = \frac{\partial E}{\partial x_f}^\top (x(T),\theta_l), \qquad \xi_\theta(T,\theta_l) = \frac{\partial E}{\partial \theta_f}^\top (x(T),\theta_l) = 0, \tag{4.30}$$

and

$$\begin{aligned} \dot{\xi}_x &= -\frac{\partial g_x}{\partial x_f}^\top \xi_x - \frac{\partial g_\theta}{\partial x_f}^\top \xi_\theta = \eta \frac{\partial^2 E}{\partial x_f^2} P_1 P_3^\top \xi_x + \eta \frac{\partial^2 E}{\partial x_f^2} P_2 \xi_\theta \\ &= \eta \frac{\partial^2 E}{\partial x_f^2} \left(P_1 P_3^\top \xi_x + P_2 \xi_\theta \right) \qquad (4.31) \\ \dot{\xi}_\theta &= -\frac{\partial g_x}{\partial \theta_f}^\top \xi_x - \frac{\partial g_\theta}{\partial \theta_f}^\top \xi_\theta \\ &= \eta \left[\frac{\partial E}{\partial x_f} \right] \left(P_1 P_2^\top + (P_2(\theta_f) + \epsilon P_2(\theta_f - \phi)) P_1^\top \right) \xi_x - \eta \left[\frac{\partial E}{\partial x_f} \right] P_3 \xi_\theta \\ &= \eta \left[\frac{\partial E}{\partial x_f} \right] \left((P_1 P_2^\top + P_4 P_3^\top) \xi_x - P_3 \xi_\theta \right) \qquad (4.32) \end{aligned}$$

where

$$\begin{aligned} \eta &= \frac{1}{1 + \epsilon \cos(\phi)}, \\ P_4(\theta_f) &= P_2(\theta_f) + \epsilon P_2(\theta_f - \phi), \end{aligned}$$

and

$$\left[\frac{\partial E}{\partial x_f} \right] \triangleq \begin{bmatrix} \frac{\partial E}{\partial x_1} & & & \\ & \frac{\partial E}{\partial x_2} & & \\ & & \ddots & \\ & & & \frac{\partial E}{\partial x_{N_f}} \end{bmatrix}.$$

Now the costates dynamics from Equations (4.30) to (4.32), together with the optimality condition from Eq. (4.28), can be used to solve the optimal control problem given in Eq. (4.27) numerically, based on the following gradient descent principle:

$$\theta_0^{(c+1)} = \theta_0^{(c)} - \eta^{(c)} \frac{\partial J}{\partial \theta_0}^\top, \tag{4.33}$$

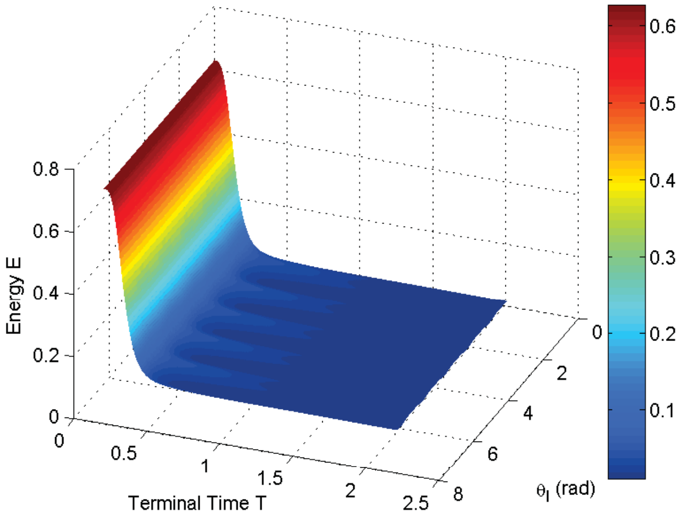


Figure 4.3: Edge-tension energy distribution with optimal initial orientations; θ_l is used in Eq. (4.24) to generate instantaneous movement of the leader, which is also considered as the exogenous input.

where η is the step size and c is the iteration count. We use the steepest descent method with Armijo step size [5] to update η. In the next section, we show the simulation results through the use of MATLAB®.

4.2.4 Simulation Results

In order to perform the gradient descent numerically as in Eq. (4.33), we need to find an initial guess of the orientations θ_0. Although the method of gradient descent does not yield global solution, we notice that through multiple simulations, the "Tangential" setup of the orientations is more likely to be the optimal solution. Therefore, our initial guess is set to be "Tangential", where all follower orientations are tangential to the circle in the same clockwise or counter-clockwise direction. Note that if the initial guess is far from tangential, the result might be some other local minimum, as explained in Section 4.2.2.2.

To illustrate the readiness of the formation system with respect to different initial conditions, we use the edge-tension energy to measure how well the formation shape is maintained.

Figure 4.3 shows the edge-tension energy distribution when initial orientations of the followers are aligned according to the solution to the optimal control problem given in Eq. (4.27). Clearly, the energy E decreases and approaches zero along all directions of the leader's movement. This means that the system with the optimal set of orientations is able to recover the shape against arbitrary perturbation to the

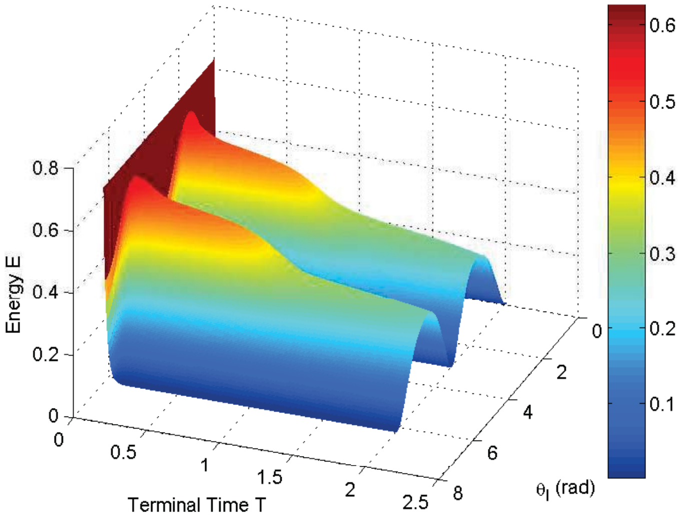


Figure 4.4: Edge-tension energy distribution with uniform orientations.

leader on a plane, and thus having high readiness.

Figure 4.4 shows distribution of the edge-tension energy E when the initial orientations of the followers are uniformly aligned as $\theta_0 = \mathbf{0}$. The local minimum shown in the figure indicate that the formation system is "ready" for the leader's movement in the direction of $\theta_l = 0$ and $\theta_l = \pi$, but is not "ready" when the leader moves in any other direction. Therefore, the overall "readiness" is low for uniform initial orientations in terms of the cost J given in Eq. (4.27) (which is also shown in Figure 4.5).

Figure 4.5 shows the cost J given in Eq. (4.27) under different sets of initial orientations, which also indicates different readiness levels of the system. The curve marked "Optimal" shows the cost J with optimal initial orientations, which are slightly smaller than the one with "Tangential" after approximately 0.2 time units. An interesting observation is that the uniform orientations lead to a very low cost in the beginning of the formation recovering. This comes from the fact that the uniform orientation setup has very low cost along $\theta_l = 0$ and $\theta_l = 2\pi$ direction, as shown in Figure 4.4. The integration of the cost over the input space $[0, 2\pi]$ benefits from this, and thus the cost drops really fast in the beginning. However, the cost slowly approaches zero due to the low readiness in the rest part of the input space. On the other hand, the costs for "Optimal" and "Tangential" orientations approach zero much faster, which indicates higher readiness for these two initial conditions.

The optimal orientations calculated numerically are depicted in Figure 4.6. The difference of this "Optimal" setup from the "Tangential" one is about 2.1 deg to the

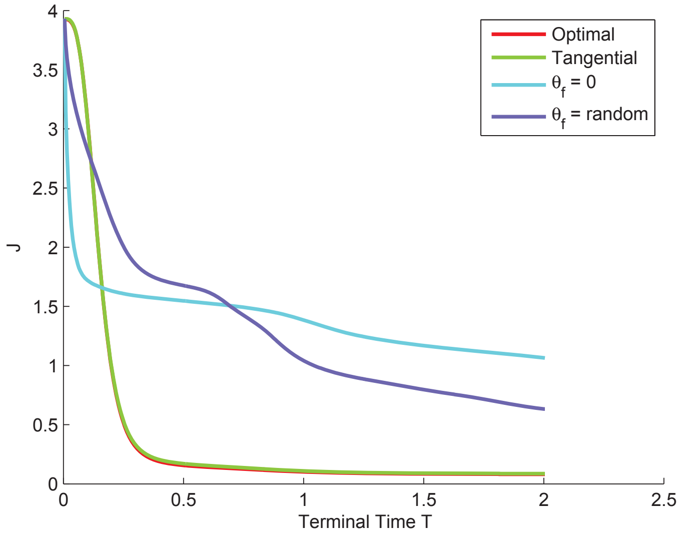


Figure 4.5: Cost comparison; the costs for tangential and optimal setups approach zero much faster than the other two configurations, indicating the faster recovery of the formation shape.

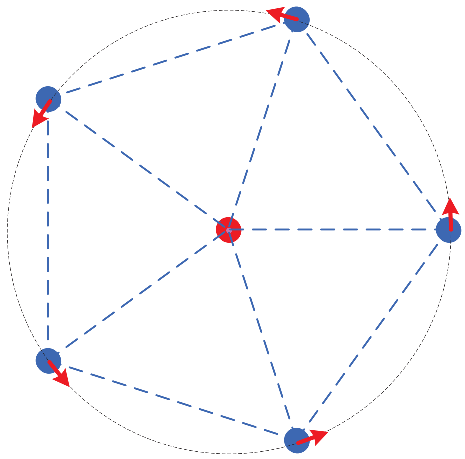


Figure 4.6: Optimal initial orientations of the convoy; the red arrows indicate the orientations, and the blue dashed lines are the communication or sensing links.

inner side of the circle. This also reflects the fact that the tangential initial heading is nearly optimal. In the case where the followers need to be moving or spinning around the leader robot, the tangential heading setup can behave just as well while still keeping the circular shape.

4.3 Summary and Discussion

In this section, we have proposed a novel notion, readiness, to describe the system initial conditions. The general idea of this notion is to characterize the system performance in terms of external disturbances. A system that recovers faster from these disturbances has higher readiness. We have applied optimization techniques of calculus of variations to analyze which initial conditions bring optimal readiness to the system. This concept is illustrated in a multi-agent system with nonholonomic constraints. The initial orientations of the agents are the variables to be optimized in the readiness optimization. The comparative results with other sets of initial orientations show the importance of this notion in interpreting the properties of multi-agent systems with nonholonomic constraints.

Proposition 4.1 shows that the distances between the agents are driven to the desired ones (if initial states are in the neighborhood of the set X^*), although there is no guarantee for the convergence of the orientations of the agents. In fact, we do not require this convergence, because we focus on the optimal initial orientations that lead to the fastest convergence in positions, or rather in the formation shape recovery. And the final orientations of the agents do not affect the instantaneous property of the system.

Other than the orientation optimization, the notion of readiness can also be used to describe the optimal poses of the agents, including both the positions and orientations. This would give more generic results in terms of the initial conditions of the multi-agent system. Meanwhile, the design of the instantaneous and terminal costs in Section 4.2.3 is rather simplified for clarity and illustration purposes, which only considers the final formation shape. In case some other metrics, such as the control effort over a specific period of time, need to be optimized through the performance index, the instantaneous cost can be adapted to punish the difference of these metrics from the expected values.

CHAPTER 5

CONTROLLER PARAMETER OPTIMIZATION

This chapter presents the design of parameterized Model Predictive Control (MPC) in a dynamic convoy protection scenario. Provided that the dynamics of the target of interest are known *a priori*, a centralized control strategy is proposed. This strategy is based on a new Kuramoto-like model for multi-robot coordination, which only takes into account the local information of each agent's neighbors and spreads out the agents equidistantly on a circle. The implementation of this strategy is illustrated on a car-like robot convoy through a locker-room agreement, where the controller parameters are optimized through the parameterized MPC before deployment. In case the dynamics of the protected target are not known *a priori*, a decentralized strategy is performed, which is similar to the centralized strategy except that the convoy controller requires no dynamic dependencies between the agents. Therefore, the parameterized MPC can be executed in a distributed fashion on the robot convoy online during the mission, where the motion of the target is part of the feedback information in the optimization process. The decentralized case is illustrated through both simulations and experiments based on the convoy model of car-like robots.

In the following sections, we first define the convoy protection scenario of interest, and provide an overview on the research work of using optimal control for multi-robot coordination (Sections 5.1 and 5.2). Then we describe the proposed new interaction model, called *Cosine-Kuramoto* model, for the balanced deployment of agents on a circle (Section 5.3). Afterward, we demonstrate the usage of parameterized MPC in the car-like robot convoy through two control strategies, and derive the optimality conditions for solving the optimal convoy protection problem (Section 5.4). The effectiveness of the overall control framework is showcased through simulation tools and hardware experiments (Sections 5.5 and 5.6). The chapter is concluded with a summary and a few discussions (Section 5.7).

5.1 Scenario and Problem Definition

This section introduces the multi-robot coordination scenario of our interest, namely *convoy protection*, and presents the technical problems that we aim to solve.

The phrase *convoy protection* refers to a formation of mobile vehicles coordinated

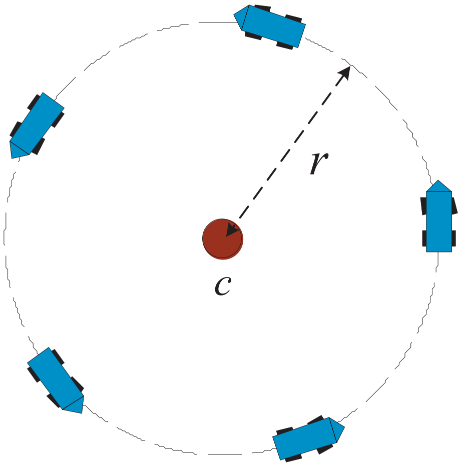


Figure 5.1: Illustration of mobile robots in circular convoy. The moving target is represented as a disc in the center of the circle.

to provide mutual support and protection to the target of interest. It can be considered as a problem of formation control in the context of robotics. Most research on convoy protection application involve heterogeneous systems. For instance, Ding et al. [27] describe a multi-UAV (Unmanned Aerial Vehicle) convoy protection problem, which consists of multiple aerial vehicles with limited range of view to provide surveillance and protection over a group of ground vehicles. However, the homogeneous multi-robot system can also be considered for convoy protection, e.g. Parker and Howard [114].

In our scenario, we consider a group of car-like mobile robots providing circular convoy protection to a moving target on the ground. Figure 5.1 shows the basic setup in the scenario. The robot convoy is lined up on a circle, while trying to keep the target in the center of the circle, in case the target moves in some direction on the plane. The convoy is expected to keep an even distribution on the circle, while avoiding collision with each other during movement. According to the case study in Chapter 4, the robots are initially located with their orientations being tangential to the circle. This setup is close to the optimal readiness, and it enables the formation to spin around the circle, as discussed in Section 4.2.4.

The questions related to this scenario that we are interested in can be described from the following three aspects:

1. How to maintain an even distribution of the vehicles on the circle during the convoy movement in a decentralized fashion?

2. How to spend the least possible control effort and respond reactively to keep

the target at the centroid in case the target starts moving?

3. How to adjust the protection perimeter (considering the constrained movement of car-like mobile robots)?

Remarks:

1. The phrase "decentralized" refers mainly to the scalability of the system under the control law to be designed. This requirement allows the convoy system to be scaled at a large number of agents, which still interact only with the neighboring agents, without knowing the information on a global level. Meanwhile, the even distribution is needed in order to form a balanced protection of the target.

2. It is conceivable that the convoy is able to track the target in any direction if all agents are spinning fast enough. This requires large control efforts, e.g. fuel, battery, etc. However, we would like to investigate how to spend the minimal amount of control effort which still achieves the task.

3. The convoy needs to reduce the perimeter of protection in case the target stays still. This is supposed to be realized without performing back and forth maneuvers, but rather with smooth transitions, such as spiral motion. It also applies to the case where the convoy needs to increase the perimeter during movement.

Therefore, in the rest of this chapter, we propose a synthesized control framework to achieve these control objectives, and present the results through simulation and experimental work.

5.2 Related Work

This section presents an overview on the research work in three aspects of multi-robot coordination, which are closely related to our proposed control framework. First, we discuss the cooperative robotic systems and their application fields in general. The second aspect is a particular multi-agent synchronization protocol, called Kuramoto model. Then the last topic is on the application of MPC for robot formation control and navigation.

5.2.1 Cooperative Systems

Typically, cooperative systems consist of multiple robotic agents performing certain tasks collaboratively to achieve a common goal. This field of research is of particular importance in multi-agent system theory due to its wide applications, such as coverage control, area patrolling, and convoy protection, etc. Spry et al. [130] introduce an organization and coordination strategy for multiple UAVs to provide video and sensor coverage of a moving ground convoy. The hierarchical control architecture has multiple layers, from the top regional layer that monitors the status of the region in the mission for vehicle assignment, and the team control layer that decomposes the mission into multiple subtasks, to the bottom supervisory and executive layers that report vehicle and task status, process sensor information, and finally perform the actual protecting movement. This movement is realized through the flight path generation of both lateral and longitudinal orbits, and trajectory tracking control on the heading. The velocities of the vehicles are assumed to be constant in this case. Hattenberger et al. [52, 53] present an optimal approach in convoy self-configuration according to the changing environment with threats for a fleet of UAVs. The management of the formation flight is centralized on one UAV, while the trajectory tracking control of each vehicle is distributed to get a higher robustness. The stable communication between the vehicles is assumed. They further evaluate the algorithms in a simulated hostile environment with hardware-in-the-loop, and show the ability for real-time implementation. Ding et al. [27] propose a time-optimal method in path planning for multiple UAVs during a convoy protection mission. The optimal coordination guarantees the continuous coverage of the moving ground vehicles with the minimum number of UAVs performing alternating circular-arc motion. They also derive the bound on the convoy speed, under which only one UAV is needed for all-time coverage. One limitation exists on the motion of the ground vehicles, which is assumed to be a straight line.

Apart from using the UAVs, ground convoy unit for active protection also poses interesting perspectives in different research fields. Pasqualetti et al. [115] consider the problem of cooperative patrolling of an area by multiple autonomous ground mobile robots. The area patrol refers to the repeated visits of the predefined viewpoints by the robots. The design of optimal trajectory and distributed control laws minimizes the longest time interval between any two visits of a viewpoint with weighted priorities, and realizes the self-organization of the robots along the optimal trajectory. This resulting optimal trajectory is called *Equal-Time-Spacing* trajectory. It describes a stop-go type of trajectory, where the waiting intervals at each viewpoint are equal among the robots and constant in time. The experiment is performed on 3 robots and a tour with 5 viewpoints, which demonstrates interesting cooperative behaviors. Parker and Howard [114] demonstrate a *formation-in-motion* concept. They simulate a ground robot formation, which is able to hold the desired shape while being teleoperated to move through a cluttered environment. The traditional leader-follower framework is used, such that the position of the leader is given through the operator, and the positions of the followers are assigned based on the environment conditions (e.g. obstacles) and the required shape of formation through a centralized utility function. This proposed framework shows high reactiveness and autonomy of the robotic cooperative system. This formation-in-motion concept is also reflected in our following research work, whereas we do not assign the positions for each robot explicitly in order to implement the control design in a distributed way. Another example of ground convoy is demonstrated in the adaptive coverage control problem addressed by Renzaglia et al. [122]. They present a cognitive adaptive optimization (CAO) algorithm for positioning a team of mobile robots for surveillance in a non-convex environment with unknown obstacles. The proposed algorithm approximates the coverage criterion function, and minimizes the difference between the approximation and the actual coverage measurement, in order to maximize the coverage area. Distributed solution is also obtained in case of limited communication links by minimizing the overlapping of each robot's field of view with the obstacles and with that of other robots. Spletzer [129] deal with the problem of optimal positioning strategy in shape changes of robotic formation, considering the minimization of either the maximum distance that any robot travels, or the total distance traveled by the formation. This optimization design has a direct application in extending the lifetime of the sensor nodes in mobile ad-hoc networks (MANETs). More applications of cooperative systems can be found for area patrolling [48], coverage control [37, 51], circular pursuit [7, 91], and automatic deployment [26].

5.2.2 Kuramoto Model

Kuramoto model was first proposed by Yoshiki Kuramoto in 1975 [75], where the problem of synchronizing coupled nonlinear oscillators was addressed in the context of physics. Later on, his book [76] published in 1984 further elaborated on this problem. Strogatz [133] provides a detailed review of the research before year 2000 on Kuramoto oscillator models, especially the contributions from Crawford [23]. The stability issues related to synchronized nonlinear oscillators based on Kuramoto model are analyzed by Jadbabaie et al. [60]. They derive the necessary and sufficient conditions on the coupling weight in the model. These conditions are later improved by Chopra and Spong [21], where the exponential synchronization of the angular frequencies of oscillators can be guaranteed.

Meanwhile, the extension from the classical Kuramoto models on coupled oscillators to other research fields has been established over the last decade. Florian and Bullo [41] investigate the transient stability issues in power networks in relation to the synchronization of non-uniform Kuramoto models. The resulting characterization based on this relationship improves the understanding of power network with respect to the underlying network parameters and initial conditions. Moreau [100] study the relation between the information flow and stability of a generalized multi-agent system, which encompasses not only Kuramoto synchronization model but also the consensus and swarming models. Acebrón et al. [1] provide further applications of coupled Kuramoto models to various scientific areas, including neural networks, laser arrays, etc. Klein et al. [74] analyze the discrete time Kuramoto model in all-to-all and one-to-all broadcast networks for multi-vehicle coordination. The integration of communication and control helps to design the network routing optimization framework, which deals with energy efficiency and heading alignment issues for multi-agent systems. In the following, we briefly introduce the Kuramoto model in a mathematical formulation.

The classic Kuramoto model consists of N coupled oscillators, whose dynamics are governed by the following equation:

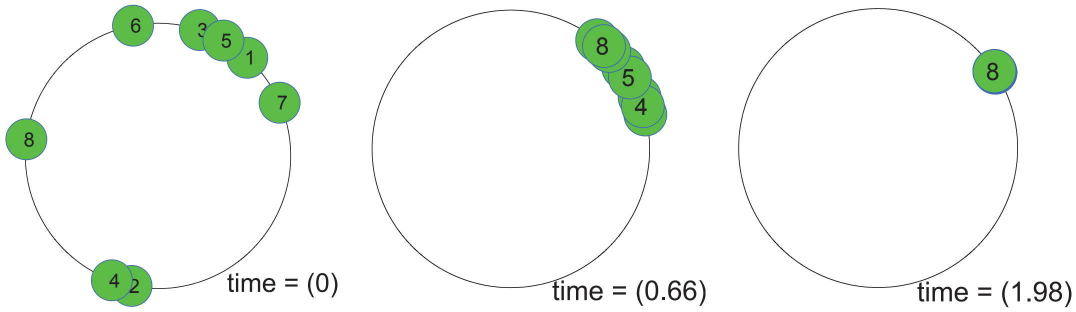
$$\dot{\psi}_i = \omega_i - \frac{K}{N}\sum_{j=1}^{N}\sin(\psi_i - \psi_j), i = 1, ..., N, \tag{5.1}$$

where ψ_i is the phase of oscillator i, ω_i is the oscillator's initial natural frequency, and $K > 0$ is the coupling gain. The oscillators are said to have *frequency synchronization* if:

$$\lim_{t\to\infty} |\dot{\psi}_i(t) - \dot{\psi}_j(t)| = 0, \forall i, j = 1, ..., N. \tag{5.2}$$

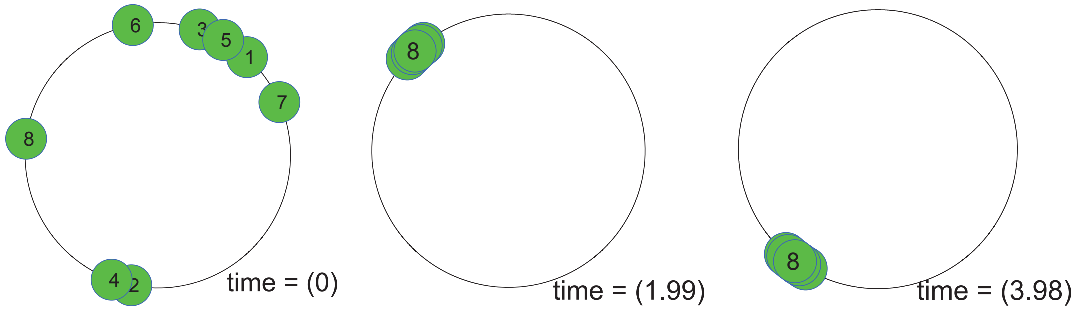
The necessary condition derived by Chopra and Spong [21] states that there exists a lower bound of coupling gain $K = K_c > 0$, below which the oscillators cannot synchronize. Furthermore, if all the natural frequencies (ω_i) are identical, the phases of the oscillators will converge to a common value ψ_∞, which is referred as *phase synchronization*. If the natural frequencies are non-identical, then the phase difference between each pair of oscillators $\psi_i - \psi_j$ will converge to a constant value, which is not necessarily zero.

As an example of this classic Kuramoto model, consider the evolutions of multi-agent system shown in Fig. 5.2 and Fig. 5.3. The original phases of the oscillators are mapped to the positions of the agents on the circle, while the frequencies are mapped to the angular velocities of the agents. In the phase synchronization example, all



(a) Initial **(b)** Synchronizing **(c)** Final

Figure 5.2: Phase synchronization example, where $\omega_i = 0$ in Eq. (5.1)



(a) Initial **(b)** Synchronizing **(c)** Final

Figure 5.3: Frequency synchronization example, where $\omega_i \neq 0$ in Eq. (5.1)

agents end up on top of each other and stay still, if the natural frequencies are assumed to be zero. In the frequency synchronization, the angular velocities of all agents converge to a common value, since the natural frequencies are not all zero.

5.2.3 Parameterized MPC for Multi-Robot Coordination

As introduced in Section 2.2.1, MPC requires an optimal control problem to be solved at each control cycle. With the expense of being computationally extensive, MPC can provide closed-loop optimal solution to nonlinear constrained dynamical systems. This feature has been the major motive for its wide-application in multi-robot systems.

Most of the research work on applying MPC to multi-robot systems have the same principle with some variations in modeling and implementation. Dunbar and Murray [31] analyze the stabilization problem for multi-vehicle formation with MPC. They provide a set of permissible equilibria for the stabilization, rather than the exact position assignment for each vehicle. The result is simulated in the formation maneuvers of 3 vehicles for trajectory tracking under centralized optimization scheme.

Later on, they study the decentralized case of formation stabilization in [32], where the states of the vehicles are coupled in the cost function and the dynamics of the vehicles are decoupled from each other. This modeling structure is also applied in our proposed decentralized strategy discussed in Section 5.4. Another decentralized framework of formation control with obstacle avoidance is simulated by Mejía and Stipanović [94], where the trajectory tracking is realized by setting a series of way points on the desired path. The optimization process forces the vehicles to stay on the trajectory as close as possible, while taking into account the obstacle avoidance function in the overall cost. They also implement a limit cycle method [70] in symmetric cases of the formation movement to avoid infeasible control outputs. The proposed MPC framework is limited in the application on real-world robotic vehicles. Since the computation of optimization cycle needs to take a few seconds, the distances between vehicles should be relatively large enough, or the velocities of the vehicles are low enough, so that no collision would happen during the optimization process. Wesselowski and Fierro [146] perform the optimal formation control from a different perspective. They apply a dual-mode MPC for the robot formation, where the desired terminal constraints are used as the objective function to be minimized in the first mode, and a second separation-bearing controller[1] is designed locally in the terminal region to achieve stable formation control. The first mode is mainly designed to respect the state constraints, the collision avoidance and desired formation shape in a combined structure. Saska [125] solve the problem of optimal trajectory planning for a formation of nonholonomic mobile robots to reach the target by using decentralized MPC. The proposed method is divided into two parts. The first part covers the control inputs for the virtual leader that generates the desired trajectories for the whole formation based on a constant sampling rate, and in the second part the prediction horizon is also considered as a variable in the optimization so as to release the computation burden in each control cycle. More examples can be found in [6, 97, 142, 145].

It can be observed that most research work on MPC are restrained to simulations due to the computational burden in the nature of optimization process. However, as introduced in Section 2.2.2, parameterized MPC is considered as one remedy to alleviate the computational complexity. This method utilizes the feedback control law to simulate the state trajectories, and further optimizes over the feedback parameters to obtain optimal solution. Droge et al. [30] specifically investigate the problem of

[1] This is the same controller used in Section 3.3.

applying parameterized MPC in single unicycle mobile robot navigation. The use of parameterized feedback law transforms the classic MPC problem into a parameter optimization problem, where the system states are decoupled from the costates. Consequently, this transformation reduces the complexity in simulating the gradients of the system states forward in time. Meanwhile, the desired behavior of the mobile robot is represented in the performance index to be minimized, which allows for modularized behavior design in complex environment. Our proposed method is largely based on this work, and further extended to multi-robot case with different feedback control laws.

5.3 Model Design for Balanced Deployment

In this section, we present the model design for the convoy protection scenario given in Section 5.1 in terms of the balanced deployment on a circle. By extending the Kuramoto synchronization model in Eq. (5.1) to the multi-robot coordination case, we propose a decentralized cosine-Kuramoto model for automatic distribution of the robots. This new model is illustrated through two simple examples based on single-integrator dynamics.

5.3.1 Cosine-Kuramoto Model

Consider N agents on a cycle graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathsf{v}_1, ..., \mathsf{v}_N\}$ is the set of nodes, $\mathcal{E} = \{(\mathsf{v}_i, \mathsf{v}_{i+1})\}, i = 1, ..., N$ is the set of edges, and $\mathsf{v}_0 \triangleq \mathsf{v}_N, \mathsf{v}_{N+1} \triangleq \mathsf{v}_1$. The incidence matrix $D \in \mathbb{R}^{N\times N}$ of a cycle graph is given in Eq. (2.14), which we rewrite as follows:

$$D = \begin{pmatrix} -1 & 0 & \cdots & +1 \\ +1 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & +1 & -1 \end{pmatrix}. \tag{5.3}$$

The graph *Laplacian* of such a cycle graph can be represented as:

$$L = DD^\top = D^\top D = -D - D^\top, \tag{5.4}$$

Hence, the Kuramoto model given in Eq. (5.1) can be rewritten in stack form:

$$\dot{\psi} = \omega - K \cdot D \sin(D^\top \psi), \tag{5.5}$$

where $\psi = [\psi_1, ..., \psi_N]^\top$ are the phases of the oscillators; $\omega = [\omega_1, ..., \omega_N]^\top$ are the natural frequencies; D is given in Eq. (5.3); $\sin(D^\top \psi) = [\sin(\psi_2 - \psi_1), ..., \sin(\psi_1 - \psi_N)]^\top$.

Suppose we have N homogeneous vehicles, whose dynamics are described in polar coordinates:

$$\dot{s} = u, \tag{5.6}$$

where $s = [\rho, \phi]^\top$ are the radial and angular coordinates (corresponding to the orientations) of the vehicles, $\rho \in \mathbb{R}^N, \phi = [\phi_1, \ldots, \phi_N]^\top \in [0, 2\pi)^N$; $u = [u_1, u_2]^\top$ is the control force, $u_1, u_2 \in \mathbb{R}^N$. The new Kuramoto-like model for coordinating the orientations of the vehicles is given as follows:

$$\dot{\phi} = K \cdot D \cos(D^\top \phi), \tag{5.7}$$

where D is the incidence matrix given in Eq. (5.3); $K > 0$ is the coupling gain. Due to the difference of applying the cosine term from the original Kuramoto model, we call Eq. (5.7) *Cosine-Kuramoto model.*

Note that in Eq. (5.6), we only control the orientation or the angular coordinate of the vehicle but not the radial coordinate, which means that $u_1 = 0$ in this case. The following theorem states the capability of the proposed Cosine-Kuramoto model in multi-agent coordination.

Theorem 5.1 (Balanced Deployment). *In the cycle graph $\mathcal{G}_c$, assume N vehicles' initial orientations satisfy $0 \leq \phi_1(0) < \phi_2(0) < ... < \phi_N(0) < 2\pi$, $N > 2$, under the dynamics of the Cosine-Kuramoto model given in Eq.* (5.7), *all orientations will asymptotically converge to a balanced distribution of equilibrium on the circle, i.e. $\phi_{i+1} - \phi_i = \frac{2\pi}{N}, \forall i \in \{1, 2, ..., N\}$, where $\phi_{N+1} \triangleq \phi_1$.*

Proof. For simplicity purpose and assuming no confusion on the notation arises, we let $x \triangleq D^\top \phi \bmod 2\pi$, which is the orientation difference vector, where $x = [x_1, \ldots, x_N]^\top \in [0, 2\pi)^N, \sum_{i=1}^N x_i = 2\pi$. We rewrite Eq. (5.7) in terms of x :

$$\dot{x} = K D^\top D \cos(x).$$

Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}\|x\|^2 \geq 0 \tag{5.8}$$

Correspondingly, the time derivative of V is

$$\begin{aligned} \dot{V} &= x^\top \dot{x} = K x^\top D^\top D \cos(x) \\ &= -K x^\top D \cos(x) - K x^\top D^\top \cos(x), \end{aligned} \tag{5.9}$$

where

$$x^\top D\cos(x) = \sum_{i=1}^{N}(x_{i+1}-x_i)\cos x_i, \quad x_{N+1} \triangleq x_1. \tag{5.10}$$

According to Proposition A.1 in Appendix A, we know that $x^\top D\cos(x) \geq 0$ (The second term $x^\top D^\top\cos(x) \geq 0$ can be proved analogously). Therefore, we have $\dot{V} \leq 0$. Now using LaSalle's invariance principle [69], we first define the set $\Omega_c = \{x \in [0,2\pi)^N | \sum_{i=1}^N x_i = 2\pi, V(x) \leq c, c \in \mathbb{R}^+\}$. Since $\dot{V} \leq 0$ and V is also quadratic, Ω_c is then a compact and positively invariant set. Define

$$V_o = \{x \in \Omega_c | \dot{V} = 0\}. \tag{5.11}$$

Based on Proposition A.1, $\dot{V} = 0 \Leftrightarrow x_{i+1} = x_i, \forall i \in \{1,2,...,N\}$. Considering the condition $\sum_{i=1}^N x_i = 2\pi$, we get:

$$V_o = \left\{x \in \Omega_c | x_i = \frac{2\pi}{N}, \forall i \in \{1,2,...,N\}\right\}. \tag{5.12}$$

Now let $M = V_o$, which is (in terms of ϕ):

$$M = \left\{\phi | \phi_{i+1} - \phi_i = \frac{2\pi}{N} \bmod 2\pi, \forall i \in \{1,2,...,N\}\right\}, \tag{5.13}$$

then every trajectory of ϕ will approach M as $t \to \infty$.

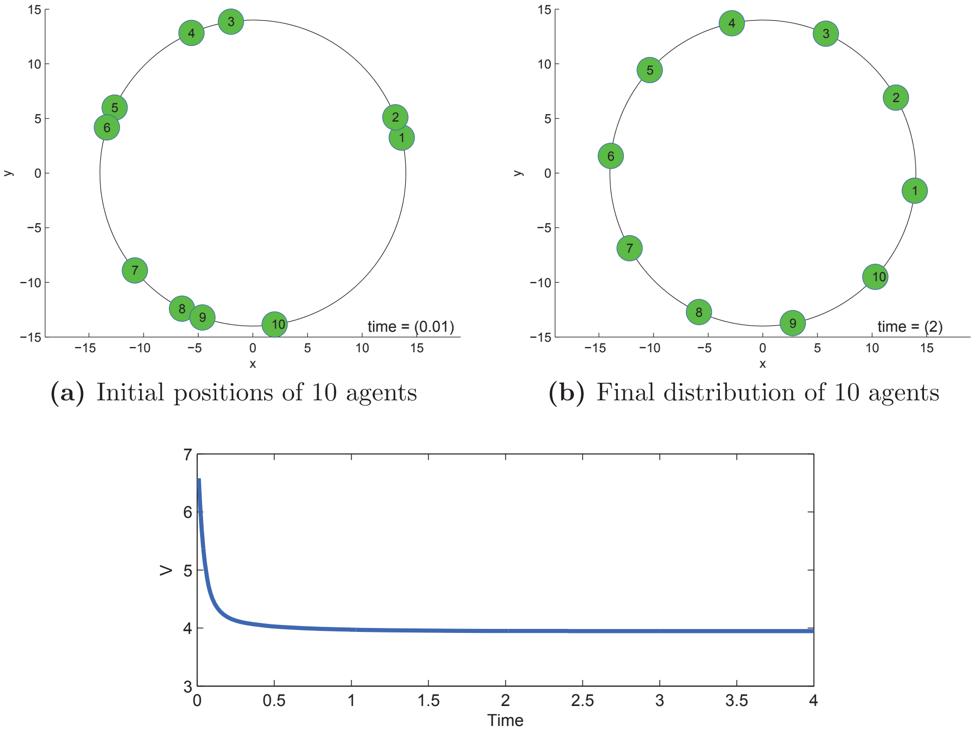
This completes the proof. □

Theorem 5.1 shows that the convergence of the cosine-Kuramoto model for multi-agent coordination is guaranteed as long as the initial positions of the agents are not identical. On a circular formation, this is reflected by the non-overlapping angular coordinates of the agents. The decentralized nature of this interaction model is illustrated through the follow-up examples.

5.3.2 Examples

To better demonstrate the capability of the cosine-Kuramoto model, and to illustrate the main ideas as clearly as possible, we present two simple simulations in this section. First example shows the basic principle of the proposed model by automatically deploying the agents on a circle, given random initial angular coordinates. The second one shows the decentralized property of the model by self-organizing the distribution pattern once new agents are added into the network.

Note that the cosine-Kuramoto model also applies to other closed-form shapes, since the only conditions required in Theorem 5.1 are that the initial orientations are sequentially ordered and not all the same. For the purpose of being clear and illustrative, the following two examples are only visualized on a circular form.



(a) Initial positions of 10 agents

(b) Final distribution of 10 agents

(c) Lyapunov function from Eq. (5.8)

Figure 5.4: Balanced distribution of 10 agents

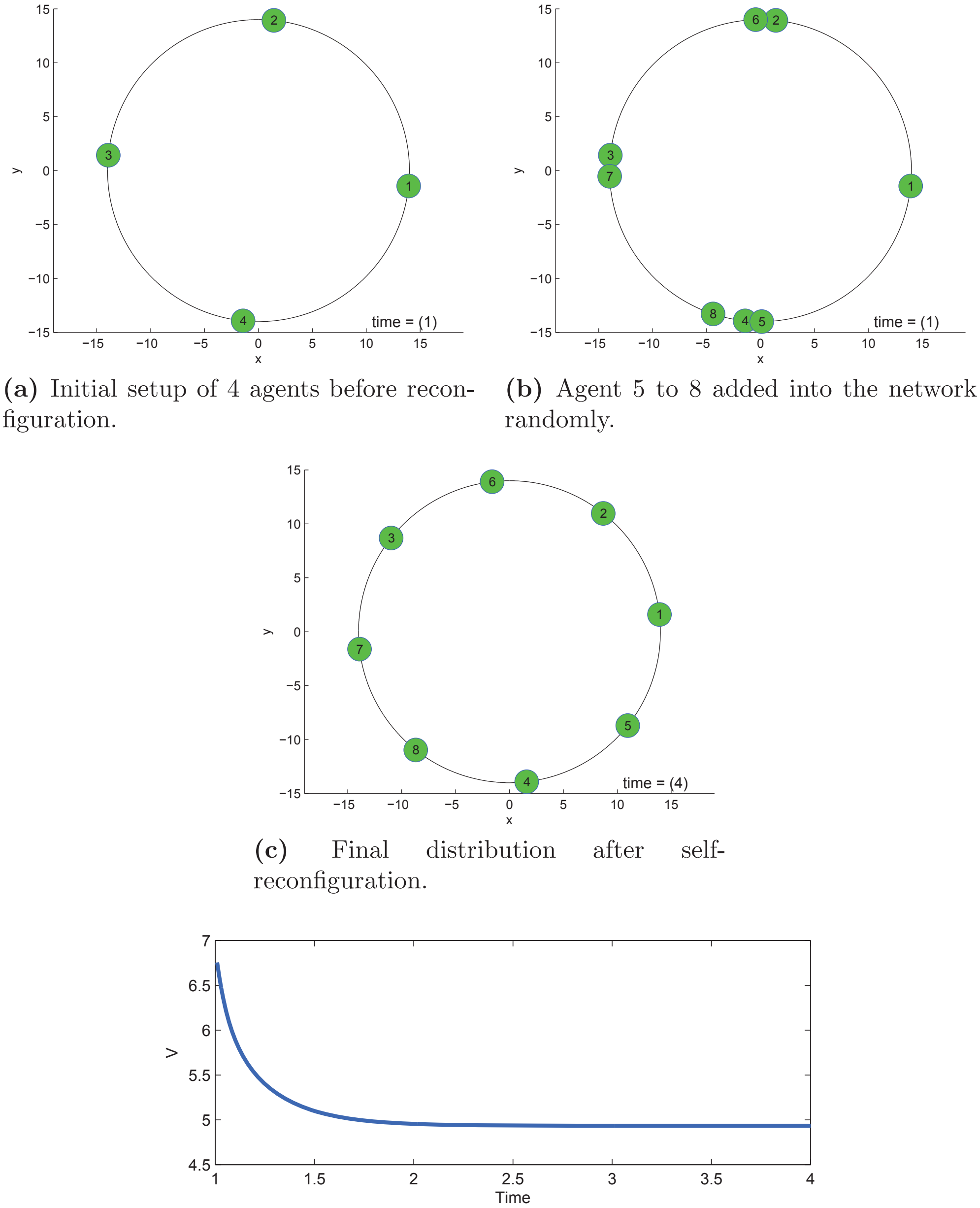
5.3.2.1 Self-deployment example

Given $N = 10$ agents with the dynamics of (5.6) under the network topology of an undirected cycle graph described in Section 5.3.1. Assume all agents initially line up on a circle of radius $\rho_i = 14, \forall i = 1, 2, ..., N$, and ϕ_i is randomly generated, which is shown in Figure 5.4a. Using the new Kuramoto-like model described in Eq. (5.7), we have:

$$u_2 = \dot{\phi} = 10 \cdot D \cos(D^\top \phi), \tag{5.14}$$

where we set $K = 10$, and incidence matrix D is given in Eq. (5.3). After 200 simulation steps, these 10 agents (almost) end up in a balanced distribution on the circle[2], as shown in Figure 5.4b, i.e. $\phi_{i+1} - \phi_i = \frac{2\pi}{10}$. The corresponding Lyapunov function from Eq. (5.8) is given in Figure 5.4c, which is strictly decreasing as expected.

[2] Since the convergence of the agents is an asymptotical property, which can only be achieved exactly when $t \to \infty$, we therefore consider the distribution being balanced if the angular difference $\|x\|$ is small enough.



(a) Initial setup of 4 agents before reconfiguration.

(b) Agent 5 to 8 added into the network randomly.

(c) Final distribution after self-reconfiguration.

(d) Lyapunov function after agents are added.

Figure 5.5: Self-organizing behavior in balanced distribution with added agents.

5.3.2.2 Self-organizing example

This example shows that the positions of agents can be automatically reorganized after adding a number of agents into the network, and the system still ends up with a balanced distribution.

The initial setup of the scene consists of 4 agents on a circle, shown in Figure 5.5a, which have already settled in a balanced distribution under cosine-Kuramoto dynamics. In Figure 5.5b, 4 more agents are added into the network randomly without overlapping with the existing ones, where the cycle topology holds for the new system. This means the agents can still obtain the relative phase difference from their direct neighbors. And Figure 5.5c shows the final distribution of the agents after they reconfigure themselves under the same cosine-Kuramoto model. Figure 5.5d gives the Lyapunov function from the time when 4 extra agents are added into the network.

5.4 Optimal Convoy Control

In this section, we first formulate the design requirements in terms of the information flow in a general optimal control problem for a multi-agent system, and the technical assumptions we use for our intended convoy protection scenario given in Section 5.1. Under these conditions, the model and control laws for this dynamic convoy protection problem are designed. Two distinct strategies are discussed using different control laws. We also introduce the parameterized optimal control scheme under these two control strategies, based on the behavior-dependent performance index to optimize the control inputs, so that the overall convoy protection task can be performed in a synthesized framework.

5.4.1 Design Requirements and Technical Assumptions

In order to design the optimal controller for the convoy protection task, we need to evaluate the feasibility of the optimization in terms of the information that is needed by each agent in the convoy. In particular, following [29], we address the design requirements for the agents in terms of the dynamic dependencies and performance (or cost) dependencies .

Droge and Egerstedt [29] derive the theorem on the feasibility of distributed parameterized optimization in multi-agent systems. The theorem states that the dynamic dependencies between the agents cause the induced communication graph to explode. For instance, if agent i dynamically depends on agent j, and agent j dynamically depends on agent k, then agent i will eventually need information from agent k to perform the optimization. If the graph for dynamic dependency is a cycle graph, this means each agent will need to know the information from the rest of the agents, which violates the principle of distributed optimization. On the other hand, if the system has only cost dependencies among the agents, the induced information graph is the undirected cost graph. This allows that the information needed by each agent can be described by a function of the separability of the collective cost designed for the complete system.

In the follow-up design for car-like robot convoy, we present two types of control laws, which are considered centralized and decentralized strategies considering the different induced information flow. We state the corresponding technical assumptions as follows:

- In the centralized strategy, we assume that the dynamics of the target to protect are given *a prior*, and thus we can use this information to perform the optimization on a central computing unit, as in a locker-room agreement. We also assume that the agents can obtain the location of their neighbors during execution. In this case, we can assign the agents with both dynamic dependencies and cost dependencies, since the optimization is not run in a distributed way, i.e. no optimization is needed during the task. This brings low computational demands for the agents.

- In the decentralized strategy, we assume the motion of the target is unknown beforehand, but the position of the target is acquired by all the agents during each control step. Meanwhile, the optimization needs to be performed on every agent in each control step on-the-fly. We also assume that the agents can obtain the information from their neighbors during the task. In this case, the dependencies between the agents only appear in the cost function, while the dynamics of each agent requires no dependencies from the other agents.

5.4.2 Model and Control of Car-like Robot Convoy

As introduced in Section 5.1, we take the same kinematic equations from Eq. (2.20) for each agent in the convoy, depicted in Figure 5.6. Now consider the collective state q of the N-agent system defined by:

$$q \triangleq [x^\top, y^\top, \theta^\top, c^\top, r]^\top \in \mathbb{R}^{N\times N\times N\times 2\times 1},$$

where $x = [x_1, \ldots, x_N]^\top$, $y = [y_1, \ldots, y_N]^\top$ are the positions of the robots, $\theta = [\theta_1, \ldots, \theta_N]^\top$, are the orientations, $c = [c_1, c_2]^\top \in \mathbb{R}^2$ is the coordinate of the central target, and $r \in \mathbb{R}$ is the desired protection radius of the convoy circle. Following

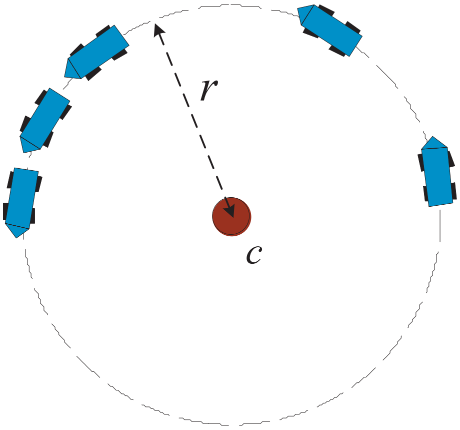


Figure 5.6: Convoy Protection with Car-like Robots; the protected target is denoted by the brown disc in the center, and the dashed circle indicates the desired convoy shape with the protection perimeter being r.

Eq. (2.20), the collective state dynamics of the convoy are given by[3]:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{c} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} v * \cos\theta \\ v * \sin\theta \\ v * [\tan\beta]/l \\ f_c(t) \\ f_r(t) \end{bmatrix} \tag{5.15}$$

where $f_c(t) : \mathbb{R} \mapsto \mathbb{R}^2$ and $f_r(t) : \mathbb{R} \mapsto \mathbb{R}$ are the functions describing the motion of the target and the change in convoy radius; velocity controls $v \triangleq [v_1, \dots v_N]^\top$ and steering controls $\beta \triangleq [\beta_1, \dots \beta_N]^\top$ are the control inputs to be designed.

In the following, we present two types of control law, which are considered as the centralized and decentralized strategies, respectively. In the centralized strategy, we use the cosine-Kuramoto model to control velocity, and a PD controller for steering angle, while in the decentralized case, we simply use the set-point method to directly control the velocity and steering angle.

5.4.2.1 Centralized Control Strategy

In case the motion of the target is known *a prior*, i.e. $f_c(t)$ and $f_r(t)$ are given, the control inputs for the robots can be computed with a locker-room agreement. This means that a central computing unit optimizes the set of parameters before the protection task starts based on the specific mission requirements. Then during

[3] operator $*$ indicates element-wise multiplication.

the task, each agent is able to execute the control law on its own using the same optimized parameters, while only considering the information of its neighboring agents and the target. The design of the control law is presented in this section, while the performance index for optimization is introduced in Section 5.4.3.1.

We design the velocity controls v and steering controls β in stack forms to be:

$$\begin{cases} v = \omega * d & (5.16a) \\ \beta = \arctan\left(\mathbf{1}\frac{l}{r} + k_p e + k_d \dot{e}\right) & (5.16b) \end{cases}$$

where we incorporate the cosine-Kuramoto model through the angular velocity[4]

$$\omega = \omega_0 + KD\cos(D^\top \theta) \tag{5.17}$$

and distance error $e = [e_1, \ldots, e_N]^\top = d - \mathbf{1}r$, distance $d = [d_1, \ldots, d_N]^\top = \|p - c \otimes \mathbf{1}\|^r$, position $p = [p_1, \ldots, p_N]^\top, p_i = [x_i, y_i]^\top$, initial angular velocity $\omega_0 = [\omega_{01}, \ldots, \omega_{0N}]^\top, i = 1, 2, \ldots, N$. K is the coupling factor for the cosine-Kuramoto model. Due to the symmetric structure of the desired convoy, we can assign the same initial value to ω_0, i.e.,

$$\omega_{0i} = u_0, \quad i = 1, \ldots, N.$$

Note that we can express the time derivative of the distance error e_i in Eq. (5.16b) as:

$$\begin{aligned} \dot{e}_i &= \frac{d\|p_i - c\|}{dt} - f_r(t) \\ &= \frac{(p_i - c)^\top}{\|p_i - c\|}(\dot{p}_i - f_c(t)) - f_r(t) \\ &= \frac{(p_i - c)^\top}{d_i}\begin{pmatrix} v_i\cos\theta_i - f_{c1}(t) \\ v_i\sin\theta_i - f_{c2}(t) \end{pmatrix} - f_r(t) \end{aligned} \tag{5.18}$$

5.4.2.2 Decentralized Control Strategy

The control law presented in Eq. (5.16) and (5.17) introduces dynamic coupling of the agents into the optimization framework. As pointed out in Section 5.4.1, this control law will cause the information flow to explode due to the dynamic dependencies if we perform distributed optimization. Therefore, in the decentralized strategy, we use the set-point method to avoid dynamic dependencies. It has the following simple form:

$$\begin{cases} v_i = v_{id} & (5.19a) \\ \beta_i = \beta_{id}, & (5.19b) \end{cases}$$

[4] operator $\|\cdot\|^r$ indicates row-wise norm.

where v_{id} and β_{id} are the set-point inputs for agent i, which are also considered as system parameters to be optimized.

5.4.3 Parameterized Model Predictive Control

With the convoy control design outlined in Section 5.4.2, we need to assign values to the tunable gains in the controllers. As discussed in Section 5.2.3, parameterized MPC is a technique suitable for this gain tuning, since it takes the system dynamics into account, and computes an optimal control action in a closed loop based on some pre-defined performance index, or cost. It also adapts the control action according to the changes in the dynamics. The computation of the optimal solution is significantly reduced, since the state trajectory generation is based on the parameter optimization using the parameterized control laws as a base, as pointed out by Howard [56]. Thus, instead of optimizing over the control input for some time horizon as in a traditional MPC approach, we optimize over the relatively low-dimensional parameter space to reduce the computational complexity.

In the rest of this section, we first consider the general case for parameterized MPC [30] approach and show the optimality conditions, and then identify the performance index based on the desired behaviors for the convoy protection problem. The corresponding update rules for control signals using the optimality conditions are discussed in the end.

Given the convoy system with dynamics of Eq. (5.15), we define:

$$\dot{q}(t) \triangleq f(q(t), u(t), t). \tag{5.20}$$

In general, the state feedback control law takes the form:

$$u(t) = \kappa(q(t), \gamma)$$

where γ is the parameter vector. This allows us to simplify the system as:

$$\dot{q} = f(q(t), \kappa(q(t), \gamma), t) = f(q(t), \gamma, t). \tag{5.21}$$

In case of the centralized strategy, the control law $\kappa(q(t), \gamma)$ is given in Eq. (5.16) and (5.17), where

$$\gamma = [k_d, k_p, K, u_0]^\top \in \mathbb{R}^4. \tag{5.22}$$

For the decentralized case, $\kappa(q(t), \gamma)$ is given in Eq. (5.19), where

$$\gamma = [\gamma_1^\top, \ldots, \gamma_N^\top]^\top \in \mathbb{R}^{2N}, \quad \gamma_i = [v_{id}, \beta_{id}]^\top, i = 1, \ldots, N. \tag{5.23}$$

The optimal parameter vector γ^* is the obtained through the minimization of the following cost functional:

$$\gamma^* = \arg\min_{\gamma} J(\gamma), \tag{5.24}$$

where

$$J(\gamma) = \int_{t_0}^{t_f} L(q(t), \gamma)dt + \Psi(q(t_f), \gamma), \quad \text{s.t. Eq. (5.21)}. \tag{5.25}$$

This cost functional takes the standard Bolza form as in Eq. (2.17).

The following theorem states the first order necessary conditions of optimality, which can be used in the update rule discussed in Sections 5.4.3.2 and 5.4.3.3 to find the optimal set of parameters.

Theorem 5.2. *The first order necessary conditions of optimality for minimizing the cost given in Eq.* (5.25) *with respect to the parameter vector, γ, is given by*

$$\frac{\partial J}{\partial \gamma} = \xi(t_0)^\top = 0, \tag{5.26}$$

where the costate ξ associated with the parameter vector γ is defined by:

$$\dot{\xi} = -\frac{\partial L}{\partial \gamma}^\top - \frac{\partial f}{\partial \gamma}^\top \lambda; \quad \xi(t_f) = \frac{\partial \Psi}{\partial \gamma}^\top (q(t_f), \gamma); \tag{5.27}$$

and the dynamics of the second costate λ associated with the state vector q is given by:

$$\dot{\lambda} = -\frac{\partial L}{\partial q}^\top - \frac{\partial f}{\partial q}^\top \lambda; \quad \lambda(t_f) = \frac{\partial \Psi}{\partial q}^\top (q(t_f), \gamma). \tag{5.28}$$

Proof. The proof follows standard variational methods, which is a simplification of the problem considered in [30], where the switching time horizon is replaced with a single fixed-length horizon. See Appendix C for details. □

Theorem 5.2 provides analytic form of the cost gradients, which removes the cross-dependency between the state q and the costates λ, ξ. This result largely alleviates the main difficulty in solving the two-point boundary value problem, where the system states depend on the costates. The cost gradient from Theorem 5.2 can be used in combination with the parameter update rules (introduced in Sections 5.4.3.2 and 5.4.3.3) to tune the feedback controller.

5.4.3.1 Performance Index

For both centralized and decentralized strategies, we can design the performance index in Eq. (5.25) based on desired behaviors or criteria of the convoy. These criteria are explained below[5].

1. Spinning velocity ω

   To ensure that the agents expend as little energy as possible during the process while approaching the desired convoy trajectory, we define an instantaneous cost for the spinning velocity ω as follows:

   $$< cen > \qquad L_1(q,\gamma) = \frac{\rho_1}{2}\|\omega\|^2 \tag{5.29a}$$

   $$< dec > \qquad L_{i1}(q,\gamma_i) = \frac{\rho_1}{2}\left(\frac{v_i}{d_i}\right)^2 \tag{5.29b}$$

   where ω is given in Eq. (5.17) for centralized strategy, and $\frac{v_i}{d_i}$ represents the spinning velocity for agent i in decentralized strategy.

2. Orientation difference $\theta_i - \theta_{id}$

   To avoid undesired oscillations of the agents during movement, we need to penalize the orientation difference of each agent from its nominal orientation on the circle, which results in the following instantaneous cost:

   $$< cen > \qquad L_2(q,\gamma) = -\frac{\rho_2}{2}\sum_{i=1}^{N}\cos(\theta_i - \theta_{id}) \tag{5.30a}$$

   $$< dec > \qquad L_{i2}(q,\gamma_i) = -\frac{\rho_2}{2}\cos(\theta_i - \theta_{id}) \tag{5.30b}$$

   where $\theta_{id} = \left(\frac{\pi}{2} + \text{atan2}(y_i - c_2, x_i - c_1)\right) \bmod 2\pi$ is the desired orientation of agent i at position $p_i = (x_i, y_i)$.

3. Inter-vehicle distance $\|p_i - p_j\|$

   To avoid collision between the agents, we penalize the inter-vehicle distances, such that if two agents are closing to each other, the cost increases. This is realized through the following exponential function of the distance between two

[5] We use the notations of $< cen >$ and $< dec >$ in front of the costs to represent whether the cost is for centralized or decentralized control strategy, respectively. The subscript i represents the ith agent in the formation.

neighboring agents for the instantaneous cost:

$$< cen > \qquad L_3(q,\gamma) = \frac{\rho_3}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \exp\left(-\|p_i - p_j\|^2\right) \tag{5.31a}$$

$$< dec > \qquad L_{i3}(q,\gamma_i) = \frac{\rho_3}{2} \sum_{j \in \mathcal{N}_i} \exp\left(-\|p_i - p_j\|^2\right) \tag{5.31b}$$

4. Radial error e

   In order to keep track of the moving target and maintain the circular shape, we penalize the radial distance error of each agent to the target. This part is considered in the terminal cost, since the focus is to have the convoy shape toward the end of each optimization process without enforcing it at each time instant.

$$< cen > \qquad \Psi_1(q,\gamma) = \frac{\rho_4}{2}\|e\|^2 \tag{5.32a}$$

$$< dec > \qquad \Psi_{i1}(q,\gamma_i) = \frac{\rho_4}{2} e_i^2 \tag{5.32b}$$

   where e is defined in Eq. (5.16b).

5. Balanced deployment $\cos(\theta_i - \theta_j)$

   Apart from keeping the desired distance to the target, the agents are expected to be evenly distributed on the circle toward the end of each optimization process, which yields the following terminal cost:

$$< cen > \qquad \Psi_2(q,\gamma) = \frac{\rho_5}{2} \sum_{i=1}^{N} \cos(\theta_{i+1} - \theta_i) \tag{5.33a}$$

$$< dec > \qquad \Psi_{i2}(q,\gamma_i) = \frac{\rho_5}{2} \sum_{j \in \mathcal{N}_i} \cos(\theta_i - \theta_j) \tag{5.33b}$$

Note that $\rho_m > 0, m = 1, \ldots, 5$ are the cost weights. Depending on the required tasks, we can manually tune these weights to influence the performance accordingly. In the follow-up simulations and experiment, these weights are to be empirically determined and set for the complete task. Section 5.7 provides further discussions on the choices of these weights.

Now the overall cost to be minimized can be written as the sum of the costs designed above:

$$
\begin{aligned}
< cen > \qquad J(\gamma) &= \int_{t_0}^{t_f} L_1(q(t),\gamma) + L_2(q(t),\gamma) + L_3(q(t),\gamma) dt \\
&\quad + \Psi_1(q(t_f),\gamma) + \Psi_2(q(t_f),\gamma), \qquad (5.34) \\
&\text{s.t.} \quad \dot{q} = f(q(t),\gamma,t). \\
< dec > \qquad J(\gamma) &= \sum_{i=1}^{N} J_i(\gamma_i) = \sum_{i=1}^{N} \Bigg[\int_{t_0}^{t_f} \Big(L_{i1}(q(t),\gamma_i) + L_{i2}(q(t),\gamma_i) \\
&\quad + L_{i3}(q(t),\gamma_i) \Big) dt + \Psi_{i1}(q(t_f),\gamma_i) + \Psi_{i2}(q(t_f),\gamma_i) \Bigg], \qquad (5.35) \\
&\text{s.t.} \quad \dot{q} = f(q(t),\gamma,t).
\end{aligned}
$$

Note that cost $J(\gamma)$ in Eq. (5.34) is minimized on a central computing unit before the task starts, and the solution γ^* is used throughout the whole task. On the other hand, the cost $J_i(\gamma_i)$ in Eq. (5.35) is processed on each individual agent's processing unit during the task.

As explained in Section 5.4.1, the dynamic dependency of the centralized control law restricts the optimization process to be performed beforehand, as in a locker-room agreement. But the decentralized strategy allows the agents to execute the optimization online in a distributed fashion. In the following, we present the different numerical update rules of the parameter set γ in these two strategies.

5.4.3.2 Centralized Update Rule

In centralized strategy, the solution to the parameter set γ is given based on the numerical steepest decent method, i.e.,

$$
\gamma^{(c+1)} = \gamma^{(c)} - \eta^{(c)} \frac{\partial J}{\partial \gamma}^{\top}, \qquad (5.36)
$$

where $\eta^{(c)}$ is the Armijo step size; c indicates the iteration count; J is given in Eq. (5.34), and $\frac{\partial J}{\partial \gamma}^{\top}$ is calculated according to Theorem 5.2. The parameter optimization algorithm with the centralized strategy is summarized in Algorithm 2.

5.4.3.3 Decentralized Update Rule

In decentralized strategy, we adopt the consensus-based distributed optimization method, as introduced in [77, 105], to solve for γ^*. The idea is for each agent to

Algorithm 2 Centralized Optimization with Parameterized MPC

1. Simulate the state trajectories $q(t), t \in [0, t_f]$ of all agents at time t_0 with the initial parameter set γ_0.
2. Calculate the total cost $J(\gamma_0)$ given in Eq. (5.34) with $q(0)$.
3. Iterate with Armijo algorithm until step size $\eta^{(c)}$ converges to zero or the maximum iteration count is exceeded.
4. Apply the corresponding γ^* to each agent, and execute the control law during the task.

combine the gradient-descent step with a step in the direction of consensus on the variables being optimized. This algorithm can be written as:

$$\gamma_i^{(k+1)} = \gamma_i^{(k)} - \mu \frac{\partial J_i}{\partial \gamma_i}(\gamma_i) - \sum_{j \in \mathcal{N}_i} \alpha_{ij}(\gamma_i^{(k)} - \gamma_j^{(k)}), \tag{5.37}$$

where μ is a constant step size along the gradient toward optimality, and α_{ij} is the consensus weight between the agents i and j. For simplicity, we assume equal weights, i.e. $\alpha_{ij} = \alpha_0, \forall(\mathsf{v}_i, \mathsf{v}_j) \in \mathcal{E}$. The convergence proof of this algorithm can be found in [77], where a diminishing step size μ is needed as time goes to infinity. However, Nedić and Ozdaglar [106] have shown that desirable performance can still be achieved by using constant step size in the gradient-descent. Moreover, the deviation from convergence can be arbitrarily small by increasing the consensus weight. The overall distributed optimization algorithm with the decentralized strategy for finding the optimal parameter set γ^* is summarized in Algorithm 3[6]. This algorithm for the distributed optimization is further illustrated through the block diagram given in Figure 5.7. In principle, agent i exchanges information of state q_i and parameter vector γ_i with all the neighbors, so that it can perform the trajectory generation for both states and costates to update the parameter vector for the next control loop.

5.5 Case Studies

In this section, we demonstrate the two control strategies discussed in Section 5.4 for the convoy protection problem through simulations in MATLAB®. These simulations show the ability of the synthesized optimization framework in dynamically adjusting the convoy shape and spinning velocity, while keeping track of the target to protect.

[6] Initial time t_0 in continuous time corresponds to k in discrete time. Thus $t_0 + \Delta t$ corresponds to $k + 1$.

Algorithm 3 Consensus-based MPC with distributed optimization

1. At time t_0, agent i simulates its own state trajectory $q_i(t)$ and its neighbors' state trajectories $q_j(t), t \in (t_0, t_f]$ forward in time based on Eq. (5.15), using the current parameter set $\gamma_i(t_0), \gamma_j(t_0)$ and the current states $q_i(t_0), q_j(t_0), j \in \mathcal{N}_i$.
2. Agent i simulates the trajectory of the costates λ_i, λ_j and ξ_i, ξ_j backward in time based on Eq. (5.27) and (5.28) to obtain $\xi_i(t_0)$, i.e. the gradient towards optimality $\frac{\partial J_i}{\partial \gamma_i}(\gamma_i)$.
3. Perform the update rule given in Eq. (5.37), and apply the next step parameter set $\gamma_i^{(k+1)}$ to the system, which leads to the next step state $q_i^{(k+1)}$.
4. Agent i communicates with its neighbors on $\gamma_i^{(k+1)}$ and $q_i^{(k+1)}$.
5. Repeat from Step 1 in the following control loop.

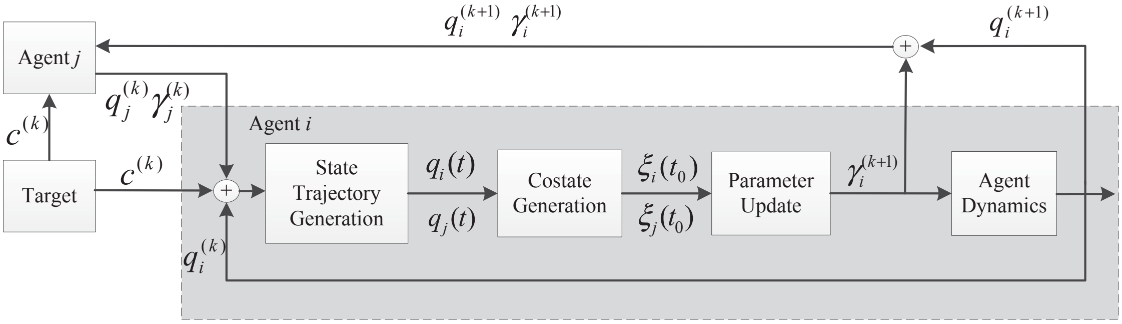
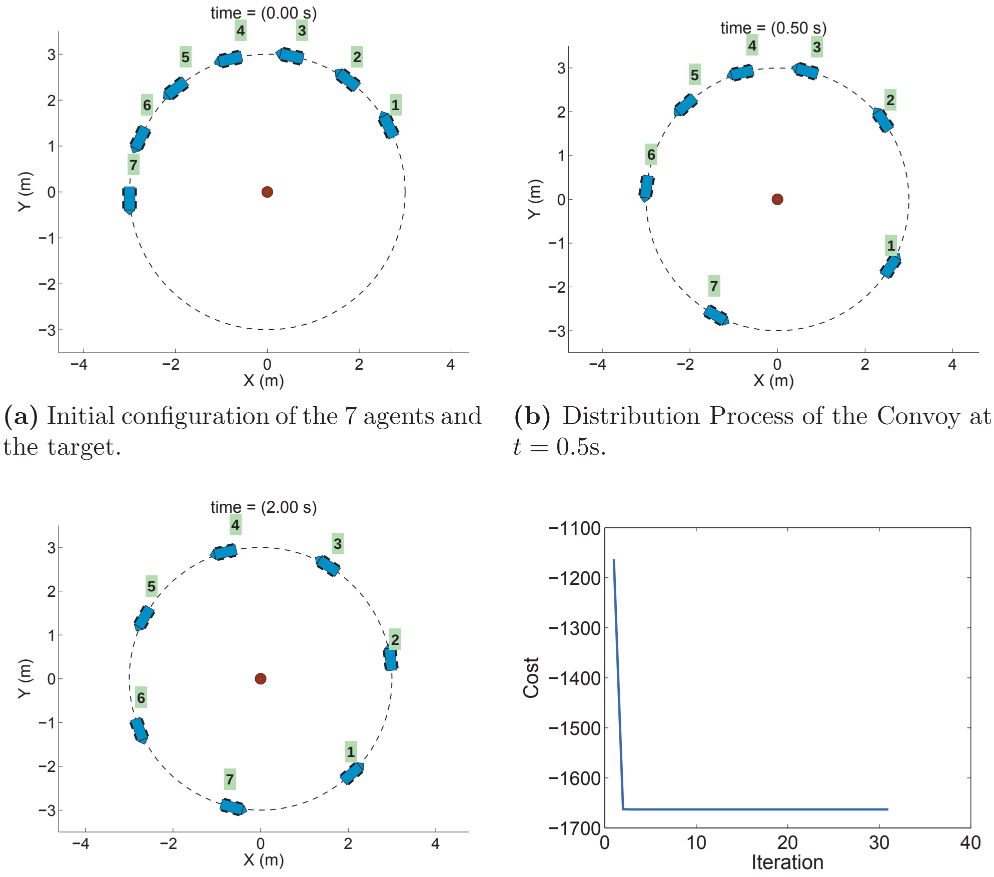


Figure 5.7: Block diagram of the consensus-based parameterized MPC in convoy protection scenario, where the operations in the gray area are performed on each individual agent. $j \in \mathcal{N}_i$ represents one of agent i's neighbors, and $t \in (t_0, t_f]$ indicates the time horizon of the optimization. $c^{(k)}$ is the position of the target at time k.

5.5.1 Case 1: Centralized

In the following scenarios with centralized control law, we assume that the trajectory of the target is given, i.e. both $f_c(t)$ and $f_r(t)$ in Eq. (5.15) are known. Once the optimization is done in the beginning, the agents only need to execute the control law with the optimized parameter vector to perform convoy protection. Optimization during the mission is not needed, which alleviates the burden on the computation of each agent.

The first scenario is to verify the capability of the control strategy in distributing the agents evenly on the circle, when the target is static. We expect the agents



(a) Initial configuration of the 7 agents and the target.

(b) Distribution Process of the Convoy at $t = 0.5$s.

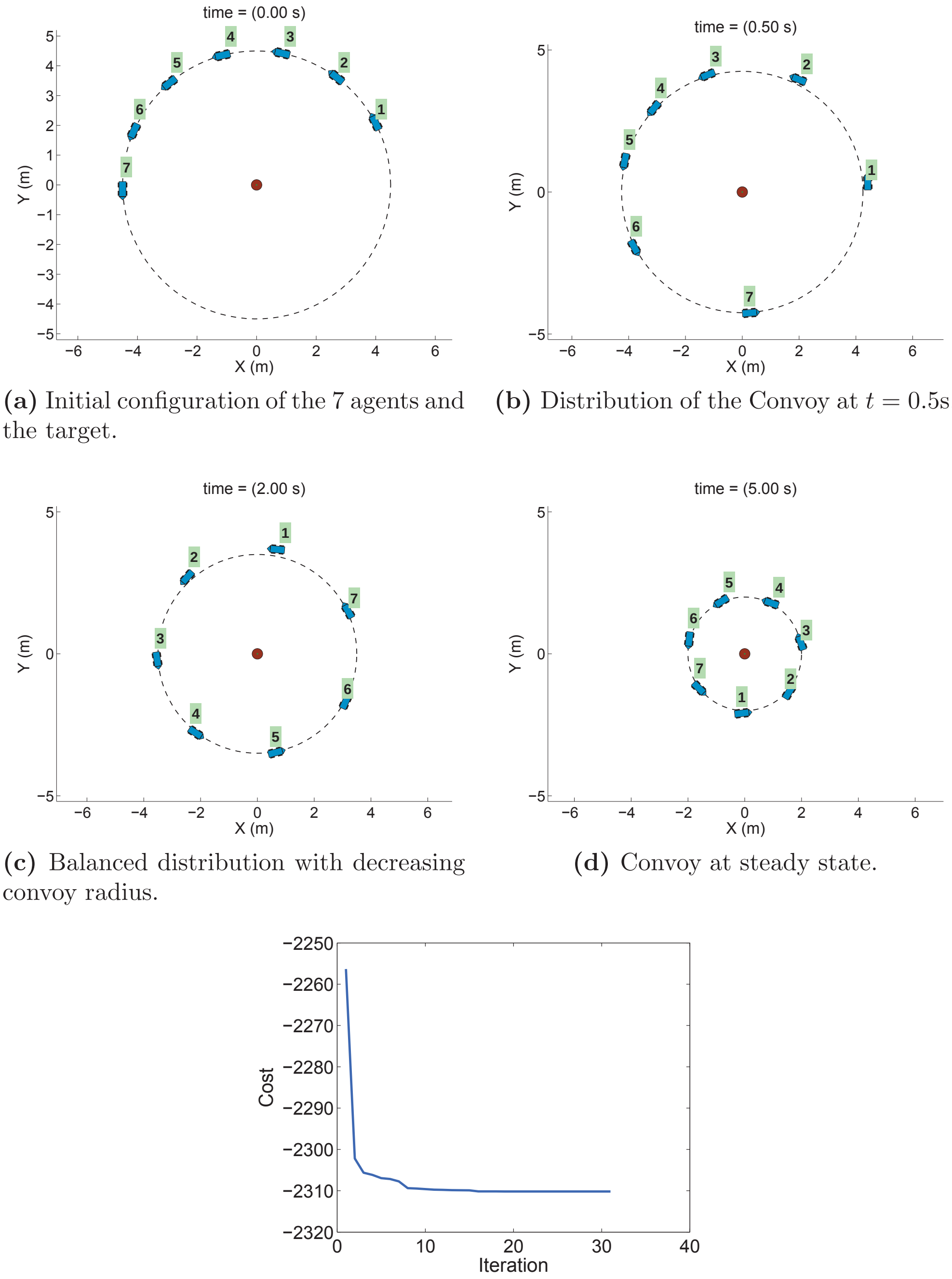
(c) Steady State of the Convoy at $t = 2$s.

(d) Cost Profile in the optimization.

Figure 5.8: Convoy Protection with Static Target (scenario 1 in centralized case).

to stay static on the circle as well, after they form the balanced deployment. The initial parameter vector $\gamma_0 = [k_d, k_p, K, u]^\top = [0.6, 0.1, 2, 2]^\top$, and the simulation step is $\Delta t = 0.001$s. Figure 5.8 shows the motion of the agents, as well as the associated cost. The agents spread out first and stop at the desired formation pattern with balanced distribution. The optimized parameter vector is found to be $\gamma^* = [0.617, 0.04, 1.971, 0]^\top$. With the initial angular velocity set as 2 rad/s, the optimization process finds the optimal base angular velocity to be zero, since the target is static. The cost profile shows the cost versus iteration steps of the optimization. In this simple scenario, the optimum is obtained after only one iteration step. This scenario demonstrates the feasibility of the optimization framework that achieves the desired convoy with the minimal cost.

Figure 5.9 shows the second scenario with the centralized control strategy. The convoy radius is decreasing over time, which translates to a tighter protection of the target. The initial radius is 4.5 m, and $\gamma_0 = [0.6, 0.1, 2, 2]^\top$, with $f_r(t) = -0.5$



(a) Initial configuration of the 7 agents and the target.

(b) Distribution of the Convoy at $t = 0.5$s.

(c) Balanced distribution with decreasing convoy radius.

(d) Convoy at steady state.

(e) Cost profile in the optimization.

Figure 5.9: Convoy protection with decreasing convoy radius (scenario 2 in centralized case).

m/s. The optimization in this case finds the optimal parameter vector to be $\gamma^* = [0.021, 0.075, 2.03, 0.987]^\top$. This optimum is obtained after approximately 16 iteration steps, as can be observed from the cost profile. This scenario shows the capability of the same control framework in adapting the convoy to the changing environment.

A third scenario of centralized control strategy is illustrated in Figure 5.10, where the target is moving at a constant rate of $f_c(t) = [2, 2]^\top$ m/s. The initial parameter set is $\gamma_0 = [0.6, 0.2, 3, 10]^\top$, and the convoy radius is 3 m. The optimization generates the optimal parameter vector as $\gamma^* = [0.801, 0.885, 2.497, 4.539]^\top$. The cost profile shows that the optimization has not yet converged to a stable value after 30 steps with decreasing cost. However, the obtained parameter set is already applicable in the scenario, considering the low convergence rate in cost after a few iteration steps. This simulation provides an outlook on how the convoy control framework is able to react to the fast-changing environment, provided that the agents are capable of performing the necessary motion under the mechanical limits.

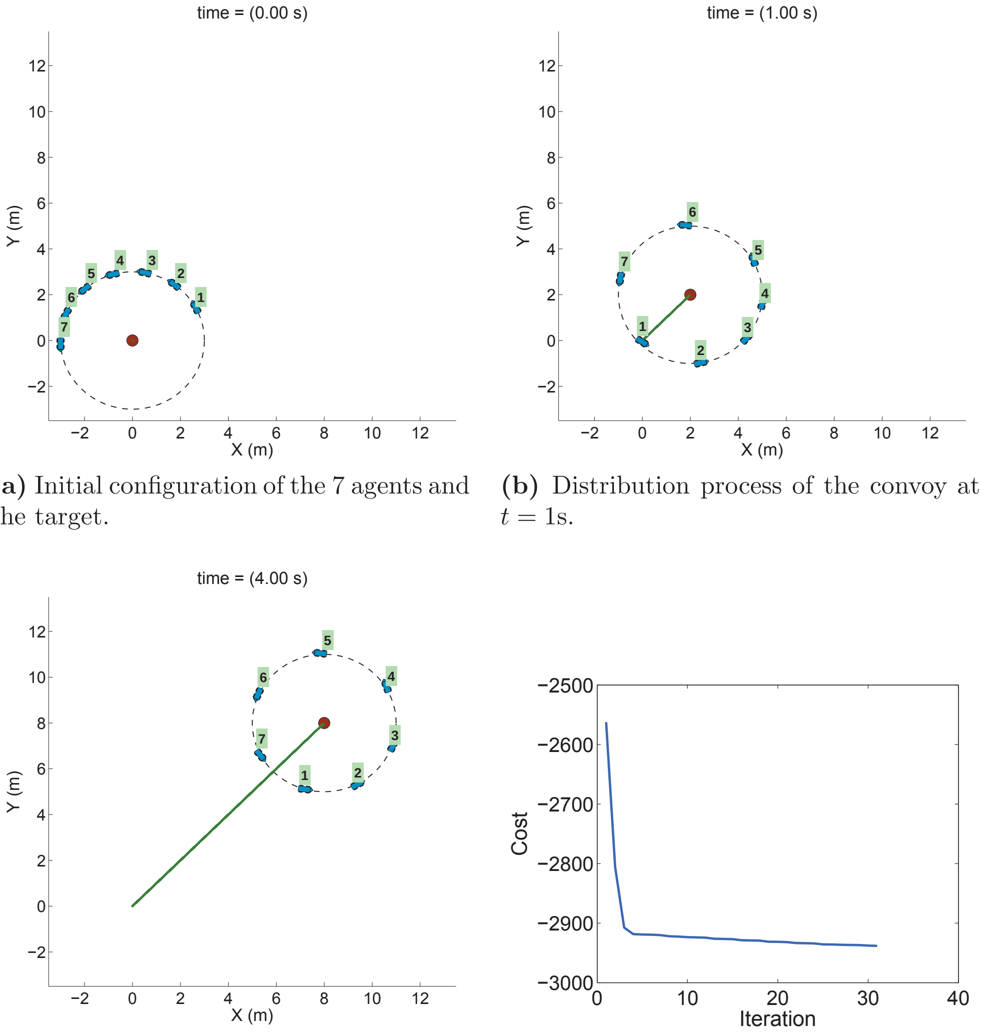
The examples shown in this section have a limitation in real-world implementation, since it requires the motion of the target to be given in advance. Any change in the target motion during the task makes the convoy movement non-optimal. For instance, if the target were to stop in scenario 3, the convoy would be spinning around the target on the circle at some angular velocity without coming to a stop, since the optimal parameters are chosen initially for the moving target. This drawback is overcome by the decentralized control strategy, which is illustrated in the next section.

5.5.2 Case 2: Decentralized

This section demonstrates the capability of the decentralized control strategy in performing convoy protection. Different from the centralized case, the motion of the target to protect is not known *a prior*, although it is assumed that each agent is able to obtain the position of the target on-the-fly. We illustrate the decentralized concept of the optimization framework in one combined example.

In every control cycle, the optimization runs according to Algorithm 3, where we set the prediction horizon $t_f = 2$ s, with the simulation step of $\Delta t = 0.01$ s. The constant gradient step size is $\mu = 0.001$, while the consensus weight a_0 is computed according to the Metropolis-based algorithm [147] to ensure distributed average consensus. The overall flow of the simulation can be divided into the following four phases.

- *Phase 1, $t \in [0, 10)s$:* The agents are deployed automatically on the circular convoy, when the target is static;
- *Phase 2, $t \in [10, 20)s$:* The convoy starts moving along the direction of the target's movement, and the convoy radius is enlarged in the meantime (from



(a) Initial configuration of the 7 agents and the target.

(b) Distribution process of the convoy at $t = 1$s.

(c) Convoy in motion with desired shape at $t = 4$s.

(d) Cost profile in the optimization.

Figure 5.10: Convoy protection with moving target (scenario 3 in centralized case).

$r = 2.5m$ to $r = 3.5m$);

- *Phase 3,* $t \in [20, 50)s$ *:* The target changes the direction of movement (at $t = 20s, 30s, 40s$). Consequently, the convoy also changes the motion to keep track of the target while maintaining the enlarged radius as the perimeter for protection;

- *Phase 4,* $t \in [50, 75]s$ *:* The motion of the target comes to a pause. Therefore, the convoy decreases the perimeter to have a tighter protection (from $r = 3.5m$ at $t = 50s$ to $r = 2.5m$ at $t = 60s$), and remains still afterwards (from $t = 60s$ to $t = 75s$).

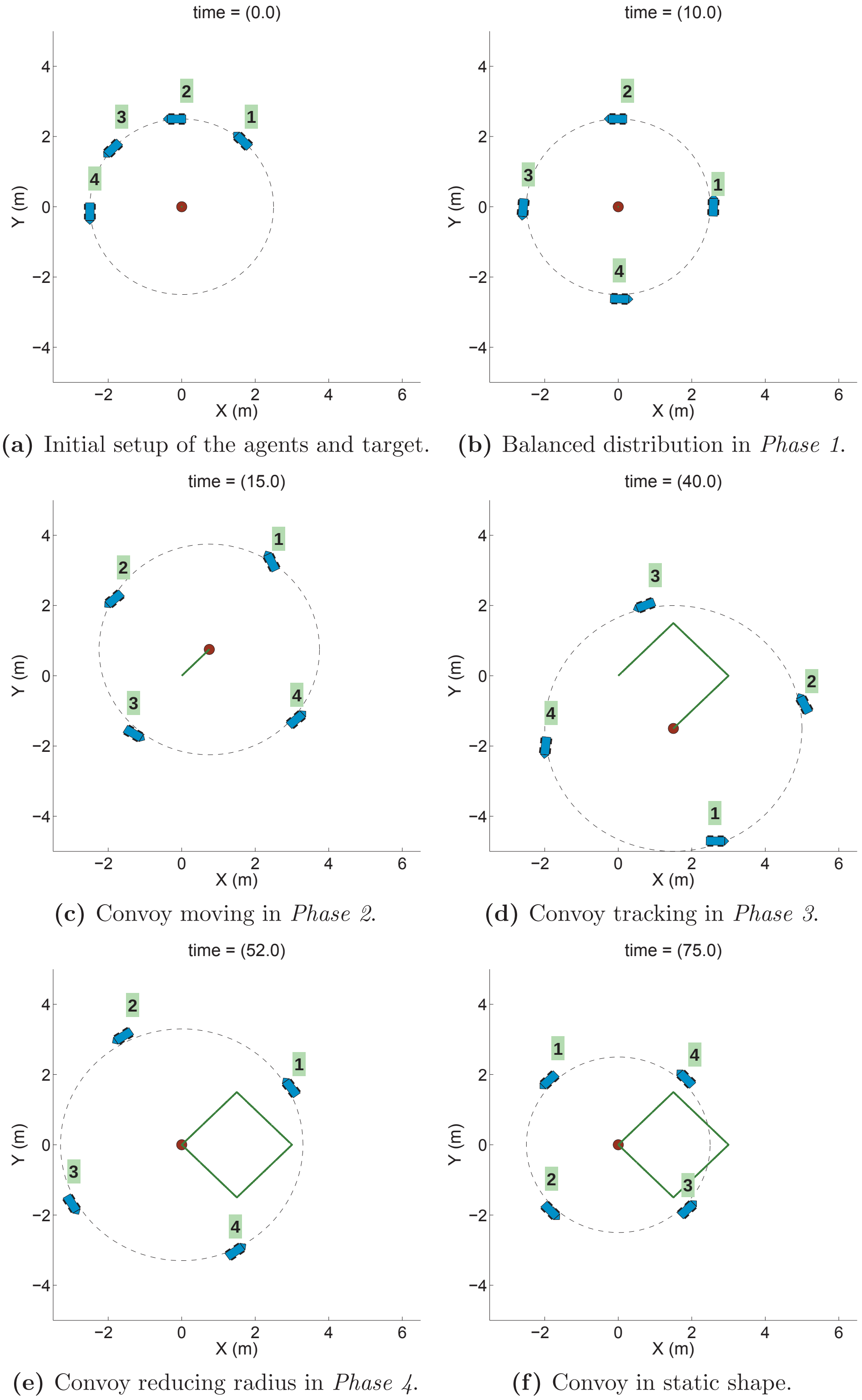
Figures 5.11a to 5.11f illustrate the complete process of the phases mentioned above. The simulation includes four agents, which are expected to keep track of the moving target and perform convoy protection task. The motion of the target is reflected as feedback information in the optimization framework, as modeled in Eq. (5.15). The two optimal control signals are shown in Figure 5.12. The initial controls are $\gamma_0 = [v_0, \beta_0]^\top = [0.4, 0]^\top$. The initial convoy radius is 2.5 m.

The optimization algorithm is implemented in C/C++, with MEX wrappers[7] to interface with MATLAB®. It computes the optimal parameter vector γ, which, in this case, is the set of control variables for the car-like mobile robots. The average computation time for running this algorithm throughout the complete task is 0.01s in each control loop[8]. This is a significant improvement compared to a conventional MPC implementation. Using `fmincon` function from optimization toolbox in MATLAB® takes up to a few seconds for this nonlinear MPC problem, which is not suitable for hardware implementation in the multi-agent coordination problem [95].

Meanwhile, compared to the developed centralized case before, the decentralized control strategy is well-suited for online implementation, since it captures the unexpected motion of the target during the mission with only the computation time for parameter optimization. Therefore, in the next section, we demonstrate how this framework works out on our real-world mobile robots.

[7] Refer to the online documentation for "C/C++ Source Mex-Files" at `http://www.mathworks.com/help/matlab/matlab_external/c-c-source-mex-files.html`.

[8] The supporting hardware and software in running this simulation is mainly the quad-core Intel(R) i5-2400 at 3.10 GHz under the MATLAB® version 2011b.



(a) Initial setup of the agents and target.

(b) Balanced distribution in *Phase 1*.

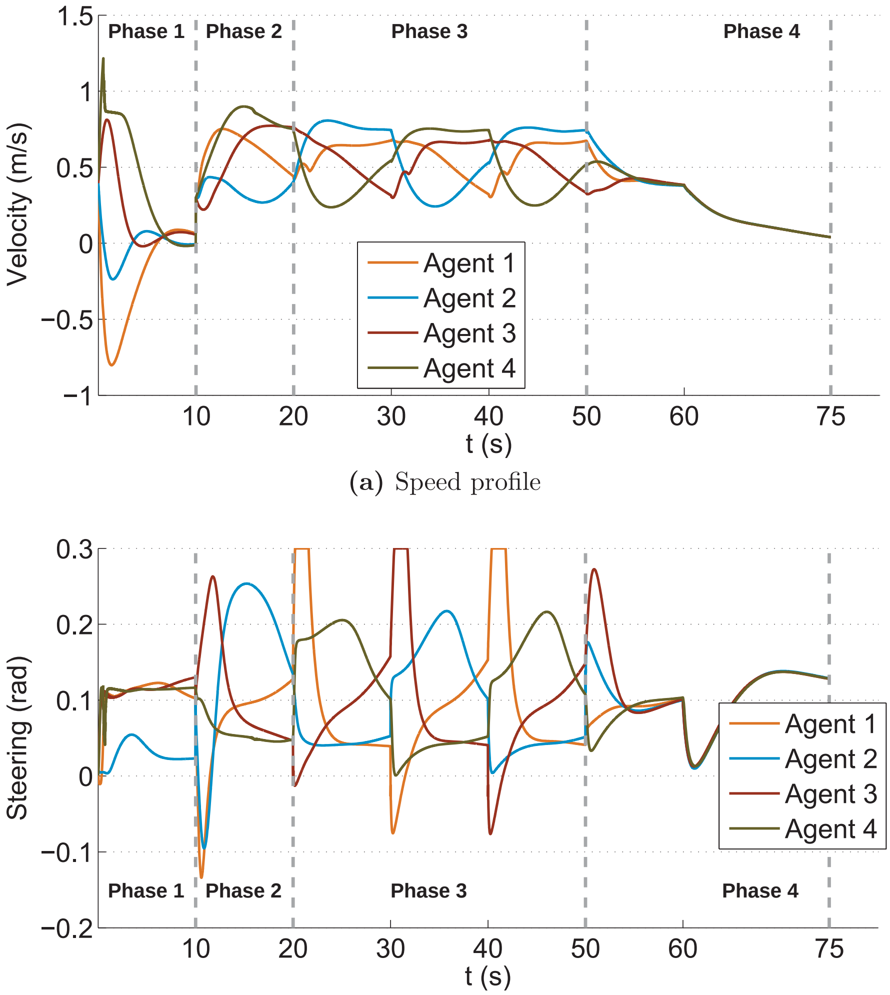
(c) Convoy moving in *Phase 2*.

(d) Convoy tracking in *Phase 3*.

(e) Convoy reducing radius in *Phase 4*.

(f) Convoy in static shape.

Figure 5.11: Decentralized optimization control in convoy protection; the dashed line indicates the desired convoy location; the green square is the path traveled by the target in red disc. 4 mobile robots are protecting the target on the desired circle.



(a) Speed profile

(b) Steering profile

Figure 5.12: Optimal control signals in the decentralized control strategy; The gray dashed lines indicate the transition between different phases of motion; Note that the steering angles are bounded by ±0.3 rad due to mechanical limits.

5.6 Experimental Validation

To showcase the proposed decentralized optimal control strategy for convoy protection, we implement the algorithm on three of our MERLIN robots, which are introduced in Section 2.3. The test field has an area of 6m x 6m. The initial convoy radius is set to be 2.5 m, and the initial positions are shown in Figure 5.13a. The target in the experiment is a virtual point indicated with the blue disc. The desired convoy circle is denoted by the dashed line in green. The poses of the robots are based on the odometry data, which is explained in Section 2.3.

The experiment is structured in the following three steps:

1. *Step 1:* The mobile robots distribute themselves automatically on the circular convoy, keeping the target at the centroid of the circle.

2. *Step 2:* Since the target is still static, the robots start to shrink the circle with the decreasing perimeter for protection.

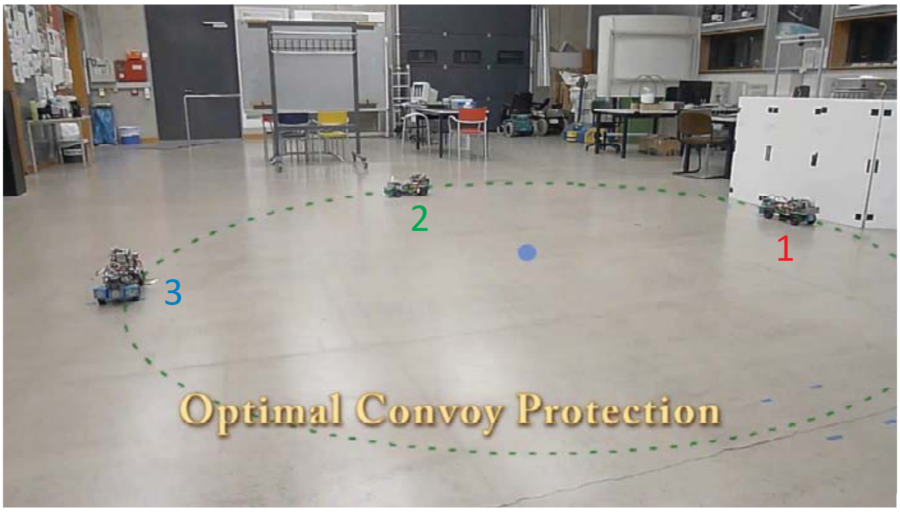
3. *Step 3:* Once the target moves, the robots gradually increase the convoy radius and move along with the target while keeping the balanced distribution and circular shape. And finally the target stops moving, which makes the convoy also come to a slow balanced stop on a larger reference circle.

Figures 5.13b to 5.13d are the snapshots of the experiment for the three corresponding steps. In this case, each robot communicates with its two neighbors about the current states and control inputs at a frequency of 5 Hz, while the computation for optimization algorithm takes approx. 35 ~ 40 ms (c.a. 28 Hz)[9]. The position information of the target point is simulated on a separate computer and transmitted to each vehicle.

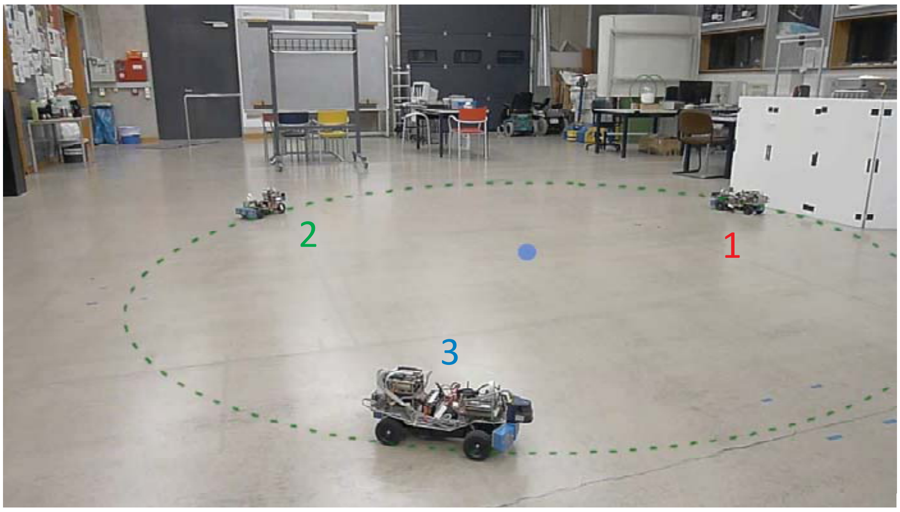
Figure 5.14 shows the velocity and steering control signals of the experiment, which are also the optimized parameters considered in the decentralized strategy. The cost weights discussed in Section 5.4.3.1 are determined through two steps. The relative ratios between the weights are first approximated by the simulation shown in Section 5.5.2, and the final scale of the values is selected through multiple experiments. These weights are finally given as:

$$[\rho_1, \rho_2, \rho_3, \rho_4, \rho_5] = [5, 10, 1, 30, 8].$$

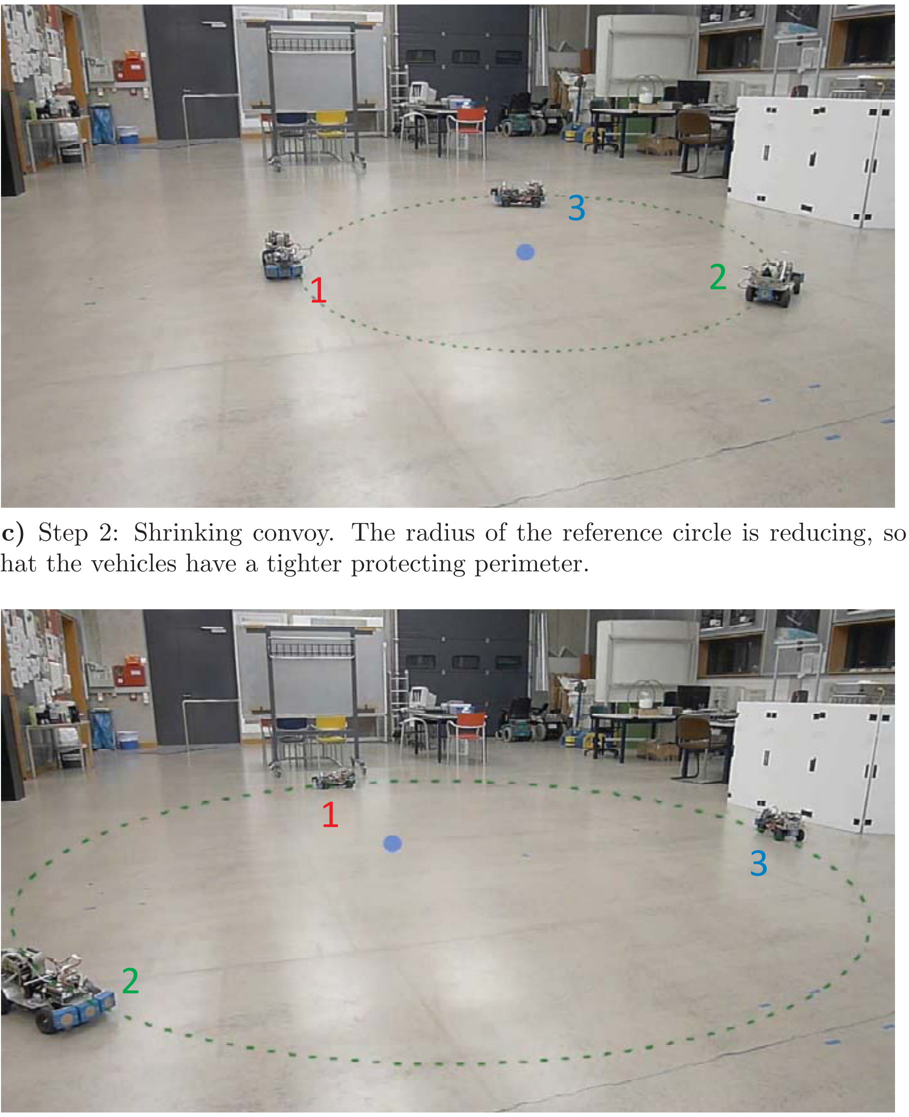
[9] This computation time is achieved under the PC/104 embedded system with Pentium III at 800MHz, which runs Ubuntu 10.04.



(a) Initial setup of the 3 vehicles, indicated by the colored numbers, and the reference circle, which is marked with dashed green line. The target to be protected is marked with the blue disc.



(b) Step 1: Balanced deployment. Three vehicles are deployed equally on the circle, and they are reducing the velocity since the target is not moving.



(c) Step 2: Shrinking convoy. The radius of the reference circle is reducing, so that the vehicles have a tighter protecting perimeter.

(d) Step 3: Expanded convoy. As the target starts moving, the convoy is moving along, keeping the target in the center of the circle. And the radius is increased in order to get a larger protecting perimeter.

Figure 5.13: Snapshots of the optimal convoy protection experiment with three MERLIN vehicles.

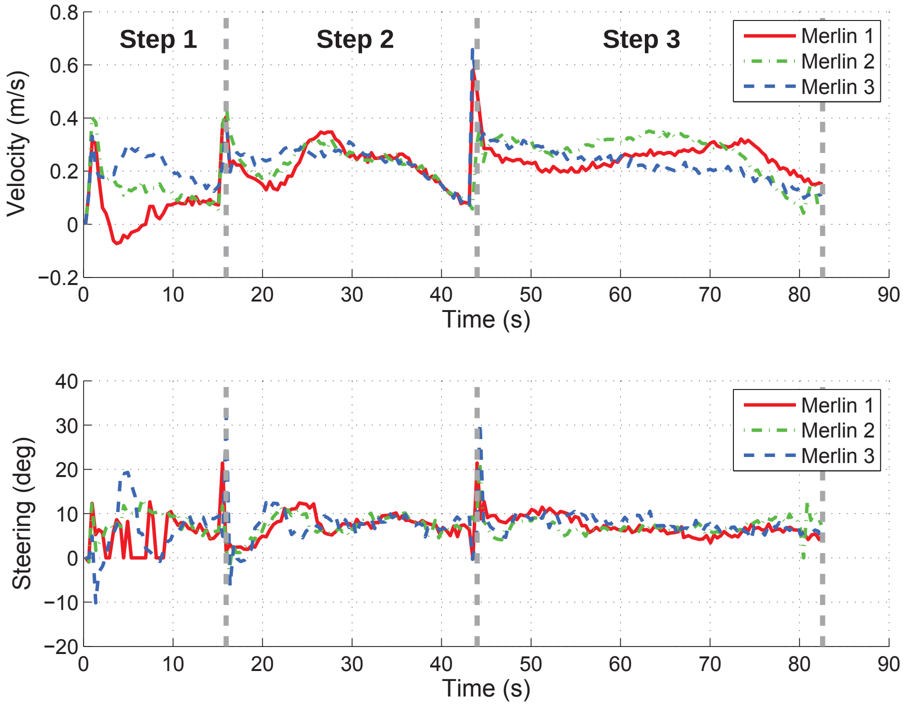


Figure 5.14: Speed and steering profiles in the convoy protection experiment.

During the experiments, we noticed that in order to keep the circular protective shape, it is preferred to set a higher weight for ρ_4, which is the weight for punishing the *radial errors* of the vehicles. This index has higher influence than the others on moving the convoy to keep track of the target. Due to the small number of vehicles and the relatively large space between the vehicles, the weighting factor ρ_3 on *inter-vehicle distance* is set to be small. If there would be more vehicles, this factor needed to be increased to avoid potential collisions between them.

In Figure 5.14, both velocity and steering controls have large disturbances at the start of Step 2 and 3. This is done intentionally to demonstrate the robustness of the MPC framework in the presence of external disturbances, such as in outdoor environment. The uneven ground surface for such vehicles in a convoy protection mission will cause similar interference to the control, and eventually further recovery efforts are needed. However, we observed from this experiment that the proposed MPC framework is quite robust against this type of disturbances, and the convoy settles quickly according to the optimized inputs.

5.7 Summary and Discussion

We have demonstrated the parameterized model predictive control in a convoy protection scenario. This scenario consists of multiple ground mobile robots, which are expected to form a circular convoy to protect the target at the centroid. We have shown both centralized and decentralized optimization for the cases where the motion of the target is given or unknown, respectively. The motion of the convoy is generated through a unified optimal control framework based on the parameterized MPC. The optimal control signals are computed to minimize the behavior-based performance index. We also validated the proposed control scheme on the MERLIN mobile robots to show its feasibility in real-world online implementation.

It should be noted that the MPC framework generates the closed-loop solutions that are capable of handling dynamical changes in the environment, such as the unknown motion of the target to be protected, as well as the varying radius of the protection perimeter. The control signals are considered as the optimized outputs in this framework. The only parameters which need to be tuned initially are the weights on the performance indices or the costs. Different combinations of the weighting factors would result in different performance during the protection task. For instance, lower weights (compared to the rest of the weights in the performance index) on the *spinning velocity* would be suitable for fast-moving targets, since the vehicles are supposed to move fast correspondingly, while the higher weighting factors should be chosen for the slowly-moving targets. On the other hand, a larger weight is necessary on the *inter-vehicle distance* to reduce the risk of collisions, if the number of vehicles in the convoy increases. The online tuning of these weights is one desirable extension to the current platform.

On the other hand, the cosine-Kuramoto model can be considered as a decentralized self-organizing system, where only local communication is needed to achieve the global coordination mission. This modeling concept under Theorem 5.1 can be applied to, for instance, public transit systems, where the transporting vehicles run along fixed routes with closed form. In this case, each vehicle only needs to exchange information with the two neighboring vehicles to maintain the smooth schedule of the complete transit system. Further applications for cosine-Kuramoto model include coverage control, where the agents can be maximally spread out to achieve the optimal coverage, as indicated through the examples in Section 5.3.

CHAPTER 6

CONCLUSIONS

In this dissertation, we presented two methodologies for designing controller parameters in cooperative multi-agent systems. These two approaches reflect general concepts in constructing stable controllers to solve time-delay and optimality problems, and produce particular solutions for the time-delayed formation tracking system and the dynamic convoy protection coordination.

In Chapter 3, based on leader-follower structure, a stable PD-type controller is designed to compensate for the steady-state error and oscillating behaviors in time-delayed formation tracking system. Parameter selection rules are formulated through the analysis on system characteristic equation and its transcendental roots. Meanwhile, the internal dynamics associated with the feedback linearization method are examined, where the conditions for system convergence are provided. In Chapter 4, the initial conditions of a general multi-agent system are characterized in the proposed notion, readiness. It represents the level of how well prepared the system is against external disturbances, especially for a nonholonomic system. This notion improves the understanding of the properties of multi-agent systems, and the meaning of optimal readiness is further studied in a multi-robot scenario through the investigation on initial orientations of the robots. This case study also provides the criteria of initialization in the convoy protection example, which is introduced in the follow-up chapter. In Chapter 5, a new cosine-Kuramoto model, considered as a distributed multi-agent coordination scheme, is proposed to solve the problem of balanced deployment of mobile agents on a circle. The convoy protection scenario is illustrated by both centralized and decentralized control strategies, and the use of distributed optimization technique in real-world implementation is shown in the decentralized case, where parameterized model predictive control provides the corresponding control framework as the basis. Optimality conditions are derived using variational methods, so that the controller parameters for convoy coordination are optimized according to the desired behavioral index.

Future research along the concepts of this dissertation can be performed in the following directions:

- **Formation tracking with obstacle avoidance** The leader-follower formation tracking scenario from this dissertation is to illustrate the controller parameter tuning approach for time-delayed problem. In order to be more applicable in real-world scenario, the obstacle avoidance functionalities need to be integrated. It is conceivable that the delay effect will lead to unexpected collisions during tracking. Therefore, the PD-type controller for delay compensation needs to be combined with the avoidance function for synthesized parameter tuning.

- **Tracking control with blind swarm** The term *blind swarm* refers to the swarm of agents which can only partially acquire the information needed for the coordination, such as self-localization data, position and velocity measurement data, etc. In the tracking control of the V-formation from this dissertation, both planar relative positions and velocities between the leader and followers are required. The behavior of the formation under partial data set is of interest for research, especially under the communication constraints. With a small amount of data needed for control, the agents will reduce the amount of data to be transmitted, thus decreasing the data-jam-induced time delay in communication. This research direction poses fundamental remedies to the problem of time delay in control with communication for large-scale multi-agent systems.

- **Convoy protection with UAVs and human interaction** The methodology developed for the dynamic convoy protection can be readily transferred to the scenario with multiple UAVs, or a combination of UGVs (Unmanned Ground Vehicles) and UAVs. Besides the difference between the dynamics, the problem of coordinating heterogeneous agents becomes more prominent. Moreover, the involvement of human interaction with the convoy is another attractive extension. The question that how to interpret a single human command for a formation of mobile agents is yet to be answered.

- **Multi-target protection with switching topology** In case that multiple dynamic targets need to be protected by a large mobile convoy, the coordination strategy would require the proper split of the agents that perform protection separately, or a jointly protective convoy. The underlying communication topology needs to switch depending on the decisions from the coordination algorithm. The switching topology will affect the performance of distributed optimization which relies on the communication between the neighboring agents. The topics on this dynamic system modeling and switching behavior analysis still remain open.

APPENDIX A

COSINE POSITIVE

The following proofs are provided to support Theorem 5.1 in Section 5.3.1.

Proposition A.1. *Given a set $x = \{x_1, x_2, ..., x_N\}$, where $x_i \in [0, 2\pi)$, $\sum_{i=1}^{N} x_i = 2\pi$ and $N > 2$. Then,*

$$\sum_{i=1}^{N} (x_{i+1} - x_i) \cos x_i \geq 0$$

where $x_{N+1} \triangleq x_1$. Equality holds if and only if $x_i = \frac{2\pi}{N}, \forall i \in \{1, 2, ..., N\}$.

In order to prove this proposition, the following two Lemmas are needed.

Lemma A.1. *Given a set $x = \{x_1, x_2, ..., x_n\}$, where $n \in \mathbb{N}$, $x_i \in [0, 2\pi)$, $\sum_{i=1}^{n} x_i = 2\pi$ and $n > 2$. Then, $\forall k, l \in \{1, 2, ..., n\}$, we have $x_k \leq x_l$ iff. $\cos x_k \geq \cos x_l$.*

Proof. We first show that if $x_k \leq x_l$, then $\cos x_k \geq \cos x_l$ in the following two cases:

- $x_k \leq x_l < \pi$: The monotonicity of cosine function clearly implies $\cos x_k \geq \cos x_l$.
- $\pi \leq x_l$: Consider the following inequality:

$$0 < x_k = 2\pi - \sum_{j=1, j \neq k}^{n} x_j \leq 2\pi - x_l \leq \pi.$$

Hence, in this case,

$$\cos x_k \geq \cos(2\pi - x_l) = \cos x_l.$$

Now we will show if $\cos x_k \geq \cos x_l$, then $x_k \leq x_l$ by contradiction. Suppose we have $\cos x_k \geq \cos x_l$ and $x_k > x_l$. From the analysis above, $x_k > x_l$ implies $\cos x_k < \cos x_l$, which is a contradiction to the assumption. This completes the proof. □

Now we will present a stronger form of Proposition A.1 in the following Lemma.

Lemma A.2. *Suppose that a set $x = \{x_1, x_2, ..., x_n\}$, where $n \in \mathbb{N}$ and $x_i \in [0, 2\pi)$, is such that $\forall k, l \in \{1, 2, ..., n\}$, $x_k \leq x_l$ iff. $\cos x_k \geq \cos x_l$. Then,*

$$f_n(x) \triangleq \sum_{i=1}^{n} (x_{i+1} - x_i) \cos x_i \geq 0, \tag{A.1}$$

where $x_{n+1} \triangleq x_1$.

Proof. We can prove this by induction to n.

- $n = 1$: The proof is trivial.
- $n = 2$: W.l.o.g., we can assume $x_2 \geq x_1$, and therefore, $\cos x_2 \leq \cos x_1$. With these two inequalities, it implies:

$$\begin{aligned} f_n(x) &= (x_2 - x_1)\cos x_1 + (x_1 - x_2)\cos x_2 \\ &\geq (x_2 - x_1)\cos x_2 + (x_1 - x_2)\cos x_2 \\ &= 0. \end{aligned}$$

- $n > 2$: There must be a $j \in \{1, 2, ..., n\}$ s.t. $x_{j-1} \leq x_j$ and $x_{j+1} \leq x_j$. Because, otherwise we would have:

$$x_1 \leq x_2 \leq \cdots \leq x_n \leq x_{n+1} = x_1,$$

or

$$x_1 \geq x_2 \geq \cdots \geq x_n \geq x_{n+1} = x_1,$$

which would imply that all x_i are equal (thus the proof is trivial). Now for such a j, we have $\cos x_{j-1} \geq \cos x_j$ and $\cos x_{j+1} \geq \cos x_j$. Therefore,

$$\begin{aligned} f_n(x) &= \sum_{i=1}^{n} (x_{i+1} - x_i)\cos x_i \\ &= (x_j - x_{j-1})\cos x_{j-1} + (x_{j+1} - x_j)\cos x_j + \sum_{\substack{i=1 \\ i \neq j-1,j}}^{n} (x_{i+1} - x_i)\cos x_i \\ &= (x_j - x_{j+1})\cos x_{j-1} + (x_{j+1} - x_{j-1})\cos x_{j-1} + (x_{j+1} - x_j)\cos x_j \\ &\quad + \sum_{\substack{i=1 \\ i \neq j-1,j}}^{n} (x_{i+1} - x_i)\cos x_i \\ &\geq (x_{j+1} - x_{j-1})\cos x_{j-1} + \sum_{\substack{i=1 \\ i \neq j-1,j}}^{n} (x_{i+1} - x_i)\cos x_i \end{aligned}$$

The last expression has the form of $f_{n-1}(x)$ (It has $n-1$ terms, without the contribution from x_j, not x_n), which implies that $f_n(x) \geq f_{n-1}(x)$. And by the induction hypothesis, $f_n(x) \geq 0$. The equality only holds when $x_j = x_{j-1}, \forall j \in \{1, 2, ..., N\}$. Considering the condition that $\sum_{i=1}^{n} x_i = 2\pi$, the equality indicates $x_i = \dfrac{2\pi}{N}$.

This completes the proof of the Lemma. □

From Lemma A.1 and A.2, we can conclude that Proposition A.1 is proved.

APPENDIX B

PROOF ON OPTIMALITY CONDITIONS FOR OPTIMAL READINESS (THEOREM 4.1)

Proof. Repeat the nonlinear system dynamics analyzed in Section 4.1.2 as follows:

$$\dot{x}(t) = f(x(t), u). \tag{B.1}$$

Since the input u is a constant over a short time interval, we can take integration of u over the entire input space U for the system dynamics given in Eq. (B.1). In order to avoid ambiguity, we denote the system state as $x(t, u)$ in the following derivation to point out that x depends on u implicitly.

First, taking the dynamics in Eq. (B.1) into account, we expand the original cost functional to:

$$\hat{J}(x_0) = \int_U \left[\int_0^T \Big\{ \lambda^\top(t, u) \Big(f(x(t, u), u) - \dot{x} \Big) + L(x(t, u), u) \Big\} dt + \Psi(x(T, u), u) \right] du. \tag{B.2}$$

Then, the idea of calculus of variations allows us to perturb the initial condition x_0 with a small change, $x_0 + \epsilon h$. Consequently, this small change causes a variation, $x(t, u) + \epsilon\eta(t, u)$, in the state trajectory, where $\epsilon > 0$. Now based on Eq. (B.2), we have the expanded cost functional after variation:

$$\begin{aligned} \hat{J}(x_0 + \epsilon h) = \int_U \Bigg[\int_0^T & \Big\{ L(x(t, u) + \epsilon\eta(t, u), u) \\ & + \lambda^\top(t, u) \Big(f(x(t, u) + \epsilon\eta(t, u), u) - (\dot{x} + \epsilon\dot{\eta}) \Big) \Big\} dt \\ & + \Psi(x(T, u) + \epsilon\eta(T, u), u) \Bigg] du. \end{aligned} \tag{B.3}$$

Due to the linearity of integration, the following difference can be calculated with the

Taylor expansions of L, f, and Ψ:

$$
\begin{aligned}
\hat{J}(x_0+\epsilon h)-\hat{J}(x_0) &= \int_U \Bigg[\int_0^T \Big\{\frac{\partial L}{\partial x}\epsilon\eta(t,u)+\lambda^\top(t,u)\frac{\partial f}{\partial x}\epsilon\eta(t,u) \\
&\qquad -\epsilon\lambda(t,u)\dot{\eta}\Big\}dt+\frac{\partial \Psi}{\partial x}\epsilon\eta(T,u)\Bigg]du+o(\epsilon) \\
&= \epsilon\int_U \Bigg[\int_0^T \Big\{\frac{\partial L}{\partial x}\eta(t,u)+\lambda^\top(t,u)\frac{\partial f}{\partial x}\eta(t,u) \\
&\qquad +\dot{\lambda}^\top\eta(t,u)\Big\}dt-\lambda^\top(T,u)\eta(T,u)+\lambda^\top(0,u)\eta(0,u) \\
&\qquad +\frac{\partial \Psi}{\partial x(T,u)}\eta(T,u)\Bigg]du+o(\epsilon), \qquad \text{(B.4)}
\end{aligned}
$$

where the last equality above follows from integration by parts

$$
\int_0^T \lambda^\top(t,u)\dot{\eta}dt=[\lambda^\top(t,u)\eta(t,u)]_0^T-\int_0^T\dot{\lambda}^\top\eta(t,u)dt, \qquad \text{(B.5)}
$$

and $o(\epsilon)$ denotes the higher-order terms in the Taylor expansion. By definition of directional derivative, we divide Eq. (B.4) by ϵ and take the limit $\epsilon \to 0$:

$$
\begin{aligned}
\delta\hat{J}(x_0;h) &= \lim_{\epsilon\to 0}\frac{\hat{J}(x_0+\epsilon h)-\hat{J}(x_0)}{\epsilon} \\
&= \int_U \left[\int_0^T \left\{\frac{\partial L}{\partial x}+\lambda^\top\frac{\partial f}{\partial x}+\dot{\lambda}^\top\right\}\eta(t,u)dt\right. \\
&\quad \left.+\left\{\frac{\partial \Psi}{\partial x(T,u)}-\lambda^\top(T,u)\right\}\eta(T,u)\right]du+\int_U\lambda^\top(0,u)\eta(0,u)du. \qquad \text{(B.6)}
\end{aligned}
$$

Now if we design the costate λ to cancel out the first integral based on the following costate dynamics:

$$
\begin{cases}
\dot{\lambda}(t,u) &= -\dfrac{\partial L}{\partial x}^\top(x(t),u)-\dfrac{\partial f}{\partial x}^\top(x(t),u)\lambda(t,u) \\
\lambda(T,u) &= \dfrac{\partial \Psi}{\partial x(T)}^\top(x(T),u),
\end{cases}
\qquad \text{(B.7)}
$$

the resulting directional derivative becomes:

$$
\delta\hat{J}(x_0;h)=\int_U\lambda^\top(0,u)\eta(0,u)du=\left(\int_U\lambda^\top(0,u)du\right)h. \qquad \text{(B.8)}
$$

In Eq. (B.8), we used the fact $\eta(0,u)=h$, i.e., the change of initial value $x_0+\epsilon h$ corresponds to $x(0,u)+\epsilon\eta(0,u)$. Since the functional $\hat{J}$ is differentiable, we can write the directional derivative as $\delta \hat{J}(x_0; h) = \dfrac{\partial \hat{J}}{\partial x_0} h$. Therefore, we obtain the first-order necessary condition for optimal initial condition x_0^* as:

$$\frac{\partial J}{\partial x_0} = \int_U \lambda^\top(0, u) du = 0, \tag{B.9}$$

which is also given in Eq. (4.5). This completes the proof. □

APPENDIX C

PROOF ON OPTIMALITY CONDITION FOR PARAMETERIZED MPC (THEOREM 5.2)

The following proof is based on Droge et al. [30], where parameterized MPC with switching time horizon problem is handled. Here, we provide a simplification of this problem, where only single fixed-length time horizon is considered. The basic approach follows standard variational methods, as shown in Appendix B.

Proof. Repeat the cost functional given in Eq. (5.25) as follows:

$$J(\gamma) = \int_{t_0}^{t_f} L(q(t), \gamma) dt + \Psi(q(t_f), \gamma), \tag{C.1}$$

$$\text{s.t.} \qquad \dot{q} = f(q(t), \gamma, t). \tag{C.2}$$

Now we augment Eq. (C.1) with the dynamics of Eq. (C.2) to:

$$\hat{J}(\gamma) = \int_{t_0}^{t_f} \Big\{ L(q(t), \gamma) + \lambda^\top \Big(f(q(t), \gamma, t) - \dot{q} \Big) \Big\} dt + \Psi(q(t_f), \gamma), \tag{C.3}$$

where λ is the costate associated with the state q. Following the idea of calculus of variations, we perturb the parameter vector with a small change, $\gamma + \epsilon h$, which causes a variation, $q(t) + \epsilon\eta(t)$, in the state trajectory, where $\epsilon > 0$. The augmented cost functional after variation becomes (the dependency of q and η on time t is omitted):

$$\begin{aligned} \hat{J}(\gamma + \epsilon h) = \int_{t_0}^{t_f} \Big\{ & L(q + \epsilon\eta, \gamma + \epsilon h) + \lambda^\top \Big(f(q + \epsilon\eta, \gamma + \epsilon h, t) - (\dot{q} + \epsilon\dot{\eta}) \Big) \Big\} dt \\ & + \Psi(q(t_f) + \epsilon\eta(t_f), \gamma + \epsilon h), \end{aligned} \tag{C.4}$$

We can compute the difference of the cost functionals before and after variation with the Taylor expansion on L, f, and Ψ:

$$\begin{aligned} \hat{J}(\gamma + \epsilon h) - J(\gamma) &= \int_{t_0}^{t_f} \Big\{ \frac{\partial L}{\partial q} \epsilon\eta + \lambda^\top \frac{\partial f}{\partial q} \epsilon\eta - \lambda^\top \epsilon\dot{\eta} \Big\} dt + \int_{t_0}^{t_f} \Big\{ \frac{\partial L}{\partial \gamma} \epsilon h + \lambda^\top \frac{\partial f}{\partial \gamma} \epsilon h \Big\} dt \\ &\quad + \frac{\partial \Psi}{\partial q} \epsilon\eta \Big|_{t_f} + \frac{\partial \Psi}{\partial \gamma} \epsilon h \Big|_{t_f} + o(\epsilon) \\ &= \epsilon \int_{t_0}^{t_f} \Big\{ \frac{\partial L}{\partial q} \eta + \lambda^\top \frac{\partial f}{\partial q} \eta + \dot{\lambda}^\top \eta \Big\} dt + \epsilon \int_{t_0}^{t_f} \Big\{ \frac{\partial L}{\partial \gamma} h + \lambda^\top \frac{\partial f}{\partial \gamma} h \Big\} dt \\ &\quad + \epsilon\, \frac{\partial \Psi}{\partial q} \eta \Big|_{t_f} + \epsilon\, \frac{\partial \Psi}{\partial \gamma} h \Big|_{t_f} - \epsilon\, \lambda^\top \eta \Big|_{t_0}^{t_f} + o(\epsilon), \end{aligned} \tag{C.5}$$

where we invoke the integration by parts on the term $\lambda^\top \dot{\eta}$, i.e.:

$$\int_{t_0}^{t_f} \lambda^\top \dot{\eta} dt = \lambda^\top \eta \Big|_{t_0}^{t_f} - \int_{t_0}^{t_f} \dot{\lambda}^\top \eta dt, \tag{C.6}$$

and $o(\epsilon)$ is the higher-order terms from the Taylor expansion. By definition of directional derivative, we can write:

$$\begin{aligned}\delta \hat{J}(\gamma; h) &= \lim_{\epsilon \to 0} \frac{\hat{J}(\gamma + \epsilon h) - \hat{J}(\gamma)}{\epsilon} \\ &= \int_{t_0}^{t_f} \Big\{ \frac{\partial L}{\partial q} \eta + \lambda^\top \frac{\partial f}{\partial q} \eta + \dot{\lambda}^\top \eta \Big\} dt + \int_{t_0}^{t_f} \Big\{ \frac{\partial L}{\partial \gamma} h + \lambda^\top \frac{\partial f}{\partial \gamma} h \Big\} dt \\ &\quad + \frac{\partial \Psi}{\partial q}(q(t_f), \gamma) \eta(t_f) + \frac{\partial \Psi}{\partial \gamma}(q(t_f), \gamma) h - \lambda^\top(t_f) \eta(t_f) + \lambda^\top(t_0) \eta(t_0). \end{aligned} \tag{C.7}$$

Since $\eta(t_0) = 0$ (initial state $q(t_0)$ has no variation when $\gamma \to \gamma + \epsilon h$), we define the costate λ as:

$$\dot{\lambda} = -\frac{\partial L}{\partial q}^\top - \frac{\partial f}{\partial q}^\top \lambda; \quad \lambda(t_f) = \frac{\partial \Psi}{\partial q}^\top (q(t_f), \gamma), \tag{C.8}$$

and the second costate, ξ associated with the parameter vector γ, is defined by:

$$\xi(t)^\top = \int_t^{t_f} \Big\{ \frac{\partial L}{\partial \gamma} + \lambda^\top \frac{\partial f}{\partial \gamma} \Big\} ds + \frac{\partial \Psi}{\partial \gamma}(q(t_f), \gamma), \tag{C.9}$$

then the directional derivative in Eq. (C.7) becomes:

$$\delta \hat{J}(\gamma; h) = \frac{\partial \hat{J}}{\partial \gamma} h = \xi(t_0)^\top h. \tag{C.10}$$

Therefore, we obtain the first-order necessary condition for optimality on the parameter vector γ as:

$$\frac{\partial J}{\partial \gamma} = \xi(t_0)^\top = 0, \tag{C.11}$$

where

$$\dot{\xi} = -\frac{\partial L}{\partial \gamma}^\top - \frac{\partial f}{\partial \gamma}^\top \lambda; \quad \xi(t_f) = \frac{\partial \Psi}{\partial \gamma}^\top (q(t_f), \gamma). \tag{C.12}$$

Equations (C.8), (C.11) and (C.12) are also given in Theorem 5.2. This completes the proof. □

Bibliography

[1] Acebrón, J. A., Spigler, R., Matematica, D., Tre, R., and Murialdo, L. S. L. (2005). The Kuramoto model : A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, 77(1):137–185.

[2] Ahmadzadeh, A., Buchman, G., Cheng, P., Jadbabaie, A., Keller, J., and Kumar, V. (2006). Cooperative Control of UAVs for Search and Coverage. In *Proceedings of the AUVSI conference on unmanned systems*, pages 1–14. August.

[3] Ahmed, S., Karsiti, M. N., and Hassan, G. M. (2007). Feedback linearized strategies for collaborative nonholonomic robots. In *2007 International Conference on Control, Automation and Systems*, pages 1551–1556. October.

[4] Alamir, M., Murilo, A., Amari, R., Tona, P., Fürhapter, R., and Ortner, P. (2010). On the use of parameterized nmpc in real-time automotive control. In *Automotive Model Predictive Control*, pages 139–149. Springer.

[5] Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3.

[6] Baras, J. S. (2007). MPC based motion control of car-like vehicle swarms. In *Mediterranean Conference on Control and Automation*, pages 1–6. June, IEEE.

[7] Benedettelli, D., Ceccarelli, N., Garulli, A., and Giannitrapani, A. (2010). Experimental validation of collective circular motion for nonholonomic multi-vehicle systems. *Robotics and Autonomous Systems*, 58(8):1028–1036.

[8] Benhimane, S., Malis, E., Rives, P., and Azinheira, J. (2005). Vision-based Control for Car Platooning using Homography Decomposition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2161–2166. April, IEEE.

[9] Berman, S., Lindsey, Q., Sakar, M. S., Kumar, V., and Pratt, S. C. (2011). Experimental Study and Modeling of Group Retrieval in Ants as an Approach to Collective Transport in Swarm Robotic Systems. *Proceedings of the IEEE*, 99(9):1470–1481.

[10] Boutilier, C., Shoham, Y., and Wellman, M. P. (1997). Economic principles of multi-agent systems. *Artificial Intelligence - Special issue on economic principles of multi-agent systems*, 94(1-2):1–6.

[11] Boyd, S., Ghosh, A., Prabbakar, B., and Shah, D. (2005). Gossip algorithms: design, analysis and applications. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1653–1664. IEEE.

[12] Boyd, S. P. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

[13] Breda, D. (2012). On characteristic roots and stability charts of delay differential equations. *International Journal of Robust and Nonlinear Control*, 22(8):892–917.

[14] Brockett, R. and Liberzon, D. (2000). Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 45(7):1279–1289.

[15] Camacho, E. F. and Berenguel, M. (1997). Robust adaptive model predictive control of a solar plant with bounded uncertainties. *International Journal of Adaptive Control and Signal Processing*, 11(4):311–325.

[16] Cannon, M. (2004). Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, 28(2):229–237.

[17] Cao, Y., Yu, W., Ren, W., and Chen, G. (2013). An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438.

[18] Carpin, S. and Parker, L. E. (2002). Cooperative Motion Coordination amidst Dynamic Obstacles. In *Distributed Autonomous Robotic Systems 5*, chapter 5, pages 145–154. Springer Japan, Fukuoka.

[19] Chen, X. and Li, Y. (2006). Smooth formation navigation of multiple mobile robots for avoiding moving obstacles. *International Journal of Control, Automation, and Systems*, 4(4):466–479.

[20] Chen, Y., Xi, N., Li, H., and Wang, Y. (2010). The controllability and observability of the event-based control system. In *2010 8th World Congress on Intelligent Control and Automation*, pages 4834–4838. July.

[21] Chopra, N. and Spong, M. W. (2009). On Exponential Synchronization of Kuramoto Oscillators. *IEEE Transactions on Automatic Control*, 54(2):353–357.

[22] Cortés, J., Martínez, S., and Bullo, F. (2006). Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298.

[23] Crawford, J. D. (1994). Amplitude Expansions for Instabilities in Populations of Globally-Coupled Oscillators. *Journal of Statistical Physics*, 74:1047–1084.

[24] Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., and Taylor, C. (2002). A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825.

[25] Ding, W., Yan, G., Lin, Z., and Lan, Y. (2009). Leader-following formation control based on pursuit strategies. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4825–4830.

[26] Ding, X., Kloetzer, M., Chen, Y., and Belta, C. (2011). Automatic Deployment of Robotic Teams. *IEEE Robotics and Automation Magazine*, 18(3):75–86.

[27] Ding, X., Rahmani, A., and Egerstedt, M. (2010). Multi-UAV Convoy Protection: An Optimal Approach to Path Planning and Coordination. *IEEE Transactions on Robotics*, 26(2):256–268.

[28] Driver, R. D. (1962). Existence and stability of solutions of a delay-differential system. *Archive for Rational Mechanics and Analysis*, 10(1):401–426.

[29] Droge, G. and Egerstedt, M. (2013). Distributed Parameterized Model Predictive Control of Networked Multi-Agent Systems. In *American Control Conference (ACC)*, pages 1332–1337, Washington, DC. June.

[30] Droge, G., Kingston, P., and Egerstedt, M. (2012). Behavior-based switch-time MPC for mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 408–413.

[31] Dunbar, W. and Murray, R. (2002). Model predictive control of coordinated multi-vehicle formations. In *Proceedings of IEEE Conference on Decision and Control*, pages 4631–4636. December.

[32] Dunbar, W. B. and Murray, R. M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558.

[33] Egerstedt, M., Hu, X., and Stotsky, A. (1998). Control of a Car-Like Robot Using a Virtual Vehicle Approach. In *37th IEEE Conference on Decision and Control*, pages 1502–1507. December.

[34] Eghtesad, M. and Necsulescu, D. (2006). Study of the internal dynamics of an autonomous mobile robot. *Robotics and Autonomous Systems*, 54(4):342–349.

[35] Elia, N. and Eisenbeis, J. N. (2011). Limitations of Linear Control Over Packet Drop Networks. *IEEE Transactions on Automatic Control*, 56(4):826–841.

[36] Elsgolts, L. E. and Norkin, S. B. (1973). *Introduction to the theory and application of differential equations with deviating arguments*, volume 105. New York: Academic Press.

[37] Enright, J., Savla, K., and Frazzoli, E. (2008). Coverage control for nonholonomic agents. In *Proceedings of 47th IEEE Conference on Decision and Control*, pages 4250–4256. December.

[38] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2006). A real-time model predictive control approach for autonomous active steering. In *1st IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, Grenoble, France.

[39] Fax, J. and Murray, R. (2004). Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476.

[40] Findeisen, R. and Allgöwer, F. (2004). Computational delay in nonlinear model predictive control. In *Proceedings of International Symposium on Advanced Control of Chemical Processes*, pages 1–6.

[41] Florian, D. and Bullo, F. (2010). Synchronization and Transient Stability in Power Networks and Non-Uniform Kuramoto Oscillators. In *Proceedings of the American Control Conference*, pages 930–937, Baltimore, MD. July.

[42] Franchi, A., Masone, C., Bülthoff, H. H., and Giordano, P. R. (2011). Bilateral teleoperation of multiple UAVs with decentralized bearing-only formation control. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2215–2222. September.

[43] Fridman, E. and Dambrine, M. (2009). Control under quantization, saturation and delay: An LMI approach. *Automatica*, 45(10):2258–2264.

[44] Fridman, E. and Niculescu, S.-i. (2008). On complete Lyapunov-Krasovskii functional techniques for uncertain systems with fast-varying delays. *International Journal of Robust and Nonlinear Control*, 18(3):364–374.

[45] Fuentes, R., Gómez-Sanz, J. J., and Pavón, J. (2004). A Sociological Framework for Multi-agent Systems Validation and Verification. In *Conceptual Modeling for Advanced Application Domains*, pages 458–469. Springer Berlin Heidelberg.

[46] Gamage, G. (2009). Formation control of multiple nonholonomic mobile robots via dynamic feedback linearization. In *International Conference on Advanced Robotics (ICAR)*, pages 1–6. June.

[47] Giannakopoulos, F. and Zapp, A. (2001). Stability and Hopf Bifurcation in Differential Equations with One Delay. *Nonlinear Dynamics and Systems Theory*, 1(2):145–158.

[48] Girard, A., Howell, A., and Hedrick, J. (2004). Border patrol and surveillance missions using multiple unmanned air vehicles. In *Proceedings of 43rd IEEE Conference on Decision and Control*, pages 620–625. December.

[49] Goi, H. K., Giesbrecht, J. L., Barfoot, T. D., and Francis, B. A. (2010). Vision-based autonomous convoying with constant time delay. *Journal of Field Robotics*, 27(4):430–449.

[50] Gu, K., Chen, J., and Kharitonov, V. (2003). *Stability of Time-Delay Systems*. Birkhauser, Boston.

[51] Guo, Y. and Balakrishnan, M. (2006). Complete coverage control for nonholonomic mobile robots in dynamic environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 1704–1709, Orlando, Florida. May.

[52] Hattenberger, G., Alami., R., and Lacroix., S. (2007a). Autonomous configuration control for UAV formation flight in hostile environments. In *Proceedings of the 6th IFAC symposium on Intelligent Autonomous Vehicles (IAV07)*, pages 1–6.

[53] Hattenberger, G., Lacroix, S., and Alami, R. (2007b). Formation flight: evaluation of autonomous configuration control algorithms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2628–2633. October.

[54] Hess, M. (2010). Motion coordination and control in systems of nonholonomic autonomous vehicles. *Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 2. Würzburg: Universität Würzburg.*

[55] Heß, R. and Schilling, K. (2010). GPS / Galileo Testbed Using a High Precision Optical Positioning System. In *The Second International Conference on Simulation, and Programming for Autnonous Robots (SIMPAR)*, pages 87–96. November.

[56] Howard, T. (2009). *Adaptive model-predictive motion planning for navigation in complex environments*. PhD thesis, Robotics Institute, Carnegie Mellon University.

[57] Howard, T., Green, C., and Kelly, A. (2010). Receding horizon model-predictive control for mobile robot navigation of intricate paths. In *Proceedings of the 7th International Conferences on Field and Service Robotics*, pages 69–78. July.

[58] Isidori, A. (1999). *Nonlinear control systems II*. Springer Verlag.

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

[60] Jadbabaie, A., Motee, N., and Barahona, M. (2004). On the Stability of the Kuramoto Model of Coupled Nonlinear Oscillators. In *American Control Conference (ACC)*, pages 4296–4301. July.

[61] Jia, Y. and Xi, N. (2011). Coordinated formation control for multi-robot systems with communication constraints. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 158–163. July.

[62] Kanjanawanishkul, K. (2010). *Coordinated Path Following Control and Formation Control of Mobile Robots*. PhD thesis, University of Tuebingen.

[63] Karsiti, M. N. and Hassan, G. M. (2007). Feedback linearized strategies for collaborative nonholonomic robots. In *2007 International Conference on Control, Automation and Systems*, pages 1551–1556. IEEE.

[64] Kawashima, H. and Egerstedt, M. (2011). Approximate manipulability of leader-follower networks. In *IEEE Conference on Decision and Control and European Control Conference*, pages 6618–6623. December.

[65] Kawashima, H. and Egerstedt, M. (2012). Leader Selection via the Manipulability of Leader-Follower Networks. In *American Control Conference*, pages 6053–6058. June.

[66] Kawashima, H., Zhu, G., Hu, J., and Egerstedt, M. (2012). Responsiveness and manipulability of formations of multi-robot networks. In *51st IEEE Conference on Decision and Control (CDC)*, pages 4622–4628. December.

[67] Keighobadi, J., Menhaj, M. B., and Kabganian, M. (2010). Feedback-linearization and fuzzy controllers for trajectory tracking of wheeled mobile robots. *Kybernetes*, 39(1):83–106.

[68] Kelly, A. and Nagy, B. (2003). Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control. *International Journal of Robotics Research*, 22(7-8):583–601.

[69] Khalil, H. (2002). *Nonlinear systems, 3rd.* New Jewsey, Prentice Hall.

[70] Kim, D.-h. and Kim, J.-h. (2003). A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. *Robotics and Autonomous Systems*, 42(1):17–30.

[71] Kim, D.-H. and Oh, J.-H. (1999). Tracking control of a two-wheeled mobile robot using inputoutput linearization. *Control Engineering Practice*, 7(3):369–373.

[72] Kim, Y., Zhu, G., and Hu, J. (2010). Optimizing formation rigidity under connectivity constraints. In *49th IEEE Conference on Decision and Control (CDC)*, pages 6590–6595. December.

[73] Kirk, D. E. (2004). *Optimal Control Theory: An Introduction.* Dover Publications.

[74] Klein, D. J., Lee, P., Morgansen, K. A., Member, S., and Javidi, T. (2008). Integration of Communication and Control using Discrete Time Kuramoto Models for Multivehicle Coordination over Broadcast Networks. *IEEE Jouranl on Selected Areas in Communications*, 26(4):695–705.

[75] Kuramoto, Y. (1975). Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics*, pages 420–422. Springer.

[76] Kuramoto, Y. (1984). *Chemical oscillations, waves, and turbulence.* Springer Verlag.

[77] Kvaternik, K. and Pavel, L. (2012). A continuous-time decentralized optimization scheme with positivity constraints. In *51st IEEE Conference on Decision and Control (CDC)*, pages 6801–6807. December.

[78] Lee, D., Martinez-palafox, O., and Spong, M. W. (2005a). Bilateral Teleoperation of Multiple Cooperative Robots over Delayed Communication Networks : Application. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 368–373. April.

[79] Lee, D., Martinez-palafox, O., and Spong, M. W. (2005b). Bilateral Teleoperation of Multiple Cooperative Robots over Delayed Communication Networks : Theory. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 362–367. April.

[80] Lee, T. and Dianat, S. (1981). Stability of time-delay systems. *IEEE Transactions on Automatic Control*, 26(4):951–953.

[81] Lefebvre, O., Lamiraux, F., and Bonnafous, D. (2005). Fast Computation of Robot-Obstacle Interactions in Nonholonomic Trajectory Deformation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4612–4617. April.

[82] Li, X. and Xiao, J. (2005). Robot formation control in leader-follower motion using direct Lyapunov method. *International Journal of Intelligent Control and Systems*, 10(3):244–250.

[83] Liberzon, D. (2006). Quantization, Time Delays, and Nonlinear Stabilization. *IEEE Transactions on Automatic Control*, 51(7):1190–1195.

[84] Liberzon, D. (2011). *Calculus of Variations and Optimal Control Theory: A Concise Introduction.* Princeton University Press.

[85] Lin, J., Morse, A., and Anderson, B. (2003). The multi-agent rendezvous problem. In *42nd IEEE International Conference on Decision and Control*, volume 2, pages 1508–1513. IEEE.

[86] Lin, Y., Chen, M., and Mei, R. (2012). Formation control based on second-order feedback linearization. In *31st Chinese Control Conference (CCC)*, pages 5014–5018. July.

[87] Liu, S., Chen, C., Xie, L., and Chang, Y. (2010). Formation control of multi-robot systems. In *11th International Conference on Control Automation Robotics and Vision (ICARCV)*, pages 1057–1062. December.

[88] Luca, A. D., Oriolo, G., and Samson, C. (1998). Feedback Control of a Nonholonomic Car-like Robot. In *Robot Motion Planning and Control*, pages 171–253. Springer Berlin Heidelberg.

[89] Lucena, C., Garcia, A., Romanovsky, A., Castro, J., and Alencar, P. E. (2004). *Software Engineering for Multi-Agent Systems II.* Springer.

[90] Mariottini, G., Morbidi, F., Prattichizzo, D., Vander Valk, N., Michael, N., Pappas, G., and Daniilidis, K. (2009). Vision-Based Localization for Leader-Follower Formation Control. *IEEE Transactions on Robotics*, 25(6):1431–1438.

[91] Marshall, J. a., Fung, T., Broucke, M. E., DEleuterio, G. M., and Francis, B. a. (2006). Experiments in multirobot coordination. *Robotics and Autonomous Systems*, 54(3):265–275.

[92] Mastellone, S., Stipanovic, D. M., Graunke, C. R., Intlekofer, K. a., and Spong, M. W. (2008). Formation Control and Collision Avoidance for Multi-agent Nonholonomic Systems: Theory and Experiments. *The International Journal of Robotics Research*, 27(1):107–126.

[93] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

[94] Mejía, J. S. and Stipanović, D. M. (2008). Computational receding horizon approach to safe trajectory tracking. *Integrated Computer-Aided Engineering*, 15(2):149–161.

[95] Mejia, J. S. and Stipanovic, D. M. (2009). Safe coordination control policy for multiple input constrained nonholonomic vehicles. In *Proceedings of the 48h IEEE*

Conference on Decision and Control (CDC) held jointly with 28th Chinese Control Conference, pages 5679–5684. December, IEEE.

[96] Melchor-Aguilar, D. and Niculescu, S.-I. (2006). Estimates of the attraction region for a class of nonlinear time-delay systems. *IMA Journal of Mathematical Control and Information*, 24(4):523–550.

[97] Menon, P. P., Edwards, C., and Shtessel, Y. B. (2012). Evolving control for preserving connectivity among agents of network with non-cooperative moving agents. In *American Control Conference*, pages 2401–2406. June.

[98] Mesbahi, M. and Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press.

[99] Min, H. J., Drenner, A., and Papanikolopoulos, N. (2009). Vision-based leader-follower formations with limited information. In *2009 IEEE International Conference on Robotics and Automation*, pages 351–356. May, IEEE.

[100] Moreau, L. (2005). Stability of Multiagent Systems With Time-Dependent. *IEEE Transactions on Automatic Control*, 50(2):169–182.

[101] Murray, R. (2007). Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583.

[102] Naghshtabrizi, P., Joa, B., and Xu, Y. (2007). A Survey of Recent Results in Networked Control Systems. *Proceedings of IEEE*, 95(1):138–162.

[103] Naus, G. J. L., Ploeg, J., de Molengraft, M. J. G. V., Heemels, W. P. M. H., and Steinbuch, M. (2010). A Model Predictive Control Approach to Design a Parameterized Adaptive Cruise Control. In *Automotive Model Predictive Control*, volume 402, pages 273–284. Springer.

[104] Necsulescu, D., Pruner, E., Sasiadek, J., and Kim, B. (2010). Control of nonholonomic autonomous vehicles and their formations. In *15th International Conference on Methods and Models in Automation and Robotics*, pages 37–42. IEEE.

[105] Nedic, A. and Ozdaglar, A. (2009). Distributed Subgradient Methods for Multiagent. *IEEE Transactions on Automatic Control*, 54(1):48–61.

[106] Nedić, A. and Ozdaglar, A. (2009). Subgradient Methods for Saddle-Point Problems. *Journal of Optimization Theory and Applications*, 142(1):205–228.

[107] Nelson, D. and Barber, D. (2007). Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529.

[108] Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, 95(1):215–233.

[109] Olfati-Saber, R. and Murray, R. (2004). Consensus Problems in Networks of Agents With Switching Topology and Time-Delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533.

[110] Palafox, O. M. and Spong, M. W. (2009). Bilateral teleoperation of a formation of nonholonomic mobile robots under constant time delay. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2821–2826. October.

[111] Palafox, O. M. and Spong, M. W. (2012). Experiments Of Bilateral Teleoperation Of Nonholonomic Mobile Robots In Formation , Through Constant Delay Communications. In *International Conference on Information and Automation (ICIA)*, pages 814–820. June.

[112] Panimadai Ramaswamy, S. and Balakrishnan, S. (2008). Formation control of car-like mobile robots: A Lyapunov function based approach. In *2008 American Control Conference*, pages 657–662. June.

[113] Papachristodoulou, A., Jadbabaie, A., and Münz, U. (2010). Effects of Delay in Multi-Agent Consensus and Oscillator Synchronization. *IEEE Transactions on Automatic Control*, 55(6):1471–1477.

[114] Parker, L. T. and Howard, A. M. (2009). Assistive Formation Maintenance for Human-Led Multi-Robot Systems. In *IEEE international conference on Systems, Man and Cybernetics*, pages 2350–2355. October.

[115] Pasqualetti, F., Durham, J. W., and Bullo, F. (2012). Cooperative Patrolling via Weighted Tours: Performance Analysis and Distributed Algorithms. *IEEE Transactions on Robotics*, 28(5):1181–1188.

[116] Persis, C. and Mazenc, F. (2010). Stability of quantized time-delay nonlinear systems: a LyapunovKrasowskii-functional approach. *Mathematics of Control, Signals, and Systems*, 21(4):337–370.

[117] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko., E. F. (1964). *The mathematical theory of optimal processes*. Pergamon Press, New York.

[118] Qin, S. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764.

[119] Quevedo, D. E. and Nesic, D. (2011). Input-to-State Stability of Packetized Predictive Control Over Unreliable Networks Affected by Packet-Dropouts. *IEEE Transactions on Automatic Control*, 56(2):370–375.

[120] Raghuwaiya, K. S., Singh, S., and Vanualailai, J. (2011). Formation Control of Mobile Robots. In *World Academy of Science, Engineering and Technology*, pages 762–767.

[121] Rahmani, A., Ji, M., Mesbahi, M., and Egerstedt, M. (2009). Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186.

[122] Renzaglia, A., Doitsidis, L., Martinelli, A., and Kosmatopoulos, E. B. (2011). Adaptive-based Distributed Cooperative Multi-Robot Coverage. In *Proceedings of the American Control Conference*, pages 468–473. June.

[123] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34.

[124] Saerens, B., Diehl, M., and den Bulck, E. V. (2010). Optimal Control Using Pontryagin's Maximum Principle and Dynamic Programming. *Automotive Model Predictive Control, Lecture Notes in Control and Information Sciences*, 402:119–138.

[125] Saska, M. (2011). Trajectory planning and optimal control for formations of autonomous robots. *Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 3. Würzburg: Universität Würzburg.*

[126] Sipahi, R., Niculescu, S., Abdallah, C. T., and Michiels, W. (2011). Stability and Stabilization of Systems with Time Delay. *IEEE Control Systems*, 31(1):38–65.

[127] Slotine, J.-J. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.

[128] Soorki, M., Talebi, H., and Nikravesh, S. (2011). A robust dynamic leader-follower formation control with active obstacle avoidance. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1932–1937. October.

[129] Spletzer, J. R. (2005). Optimal Positioning Strategies for Shape Changes in Robot Teams. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 742–747. April.

[130] Spry, S., Girard, A., and Hedrick, J. (2005). Convoy protection using multiple unmanned aerial vehicles: organization and coordination. In *Proceedings of the American Control Conference*, pages 3524–3529. June.

[131] Stipanović, D. M., Inalhan, G., Teo, R., and Tomlin, C. J. (2004). Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica*, 40(8):1285–1296.

[132] Stone, P. and Veloso, M. (2000). Multiagent Systems : A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3):345–383.

[133] Strogatz, S. H. (2000). From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1-4):1–20.

[134] Sun, K., Heß, R., Xu, Z., and Schilling, K. (2013). Real-Time Robust 6DoF Object Pose Estimation with a ToF Camera and a Color Camera. *Journal of Field Robotics, under review.*

[135] Tanner, H. (2004). On the controllability of nearest neighbor interconnections. In *43rd IEEE Conference on Decision and Control (CDC)*, pages 2467–2472. December.

[136] Tesch, M., Lipkin, K., Brown, I., Hatton, R., Peck, A., Rembisz, J., and Choset, H. (2009). Parameterized and Scripted Gaits for Modular Snake Robots. *Advanced Robotics*, 23(9):1131–1158.

[137] Tigli, J. and Thomas, M. (1994). Use of multi agent systems for mobile robotics control. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 588–592. October, IEEE.

[138] Tipsuwan, Y. and Chow, M.-Y. (2003). Control methodologies in networked control systems. *Control Engineering Practice*, 11(10):1099–1111.

[139] Tomlin, C. and Sastry, S. (1997). Switching through singularities. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 1, pages 1–6. December.

[140] Vasseur, L., Lecointe, O., Dento, J., Cherfaoui, N., Marion, V., and Morillon, J. G. (2004). Leader-follower function for autonomous military convoys. In *Proc. SPIE 5422, Unmanned Ground Vehicle Technology VI*, pages 326–337. September, International Society for Optics and Photonics.

[141] Vinyals, M., Rodriguez-Aguilar, J. A., and Cerquides, J. (2011). A Survey on Sensor Networks from a Multi-Agent perspective. *The Computer Journal*, 54(3):455–470.

[142] Voos, H. (2009). Model predictive collaborative motion planning and control of mobile robots including safety aspects. In *International Conference on Advanced Robotics (ICAR)*, pages 1–6. June.

[143] Široký, J., Oldewurtel, F., Cigler, J., and Prívara, S. (2011). Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy*, 88(9):3079–3087.

[144] Wang, Z., Yang, F., Ho, D., and Liu, X. (2007). Robust H_∞ Control for Networked Systems With Random Packet Losses. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 37(4):916–924.

[145] Wangmanaopituk, S., Voos, H., and Kongprawechnon, W. (2009). Collaborative nonlinear model-predictive collision avoidance and path following of mobile robots. In *ICCAS-SICE*, pages 3205–3210. August.

[146] Wesselowski, K. and Fierro, R. (2003). A Dual-Mode Model Predictive Controller for Robot Formations. In *Proceedings of IEEE Conference on Decision and Control*, pages 3615–3620. December.

[147] Xiao, L., Boyd, S., and Kim, S.-J. (2007). Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46.

[148] Xu, Z., Egerstedt, M., Droge, G., and Schilling, K. (2013a). Balanced Deployment of Multiple Robots Using a Modified Kuramoto Model. In *American Control Conference (ACC)*, pages 6138 – 6144. June.

[149] Xu, Z., Kawashima, H., and Schilling, K. (2013b). Readiness in Formation Control of Multi-Robot System. In *American Control Conference (ACC)*, pages 3032 – 3038. June.

[150] Xu, Z., Ma, L., Wu, Z., Schilling, K., and Necsulescu, D. (2011). Teleoperating a formation of car-like rovers under time delays. In *Proceedings of the 30th Chinese Control Conference*, pages 4095–4101, Yantai. July.

[151] Xu, Z., Schröter, M., Necsulescu, D., Ma, L., and Schilling, K. (2012). Formation Control of Car-like Autonomous Vehicles under Communication Delay. In *Proceedings of the 31st Chinese Control Conference*, pages 6376–6383, Hefei. July.

[152] Xue, Y. and Li, D. (2010). Calculation of PI controller stable region based on D-partition method. In *International Conference on Control Automation and Systems (ICCAS)*, pages 2185–2189. October.

[153] Yang, E., Gu, D., and Hu, H. (2005). Nonsingular formation control of cooperative mobile robots via feedback linearization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 826–831. August.

[154] Zhang, J. and Jayasuriya, S. (2009). Cooperative Control of Multi-robot Nonholonomic Systems with Dynamics Uncertainties and Control Time-Delays. In *Distributed Autonomous Robotic Systems 8*, volume 8, chapter VII, page 477. Springer Berlin Heidelberg.

[155] Zhang, L., Gao, H., Member, S., and Kaynak, O. (2013). Network-Induced Constraints in Networked Control Systems - A Survey. *IEEE Transactions on Industrial Informatics*, 9(1):403–416.

[156] Zhang, W., Branicky, M., and Phillips, S. (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99.

[157] Zhao, Y.-B., Liu, G.-P., and Rees, D. (2010). Actively Compensating for Data Packet Disorder in Networked Control Systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(11):913–917.

[158] Zhu, G. and Hu, J. (2011). Link resource allocation for maximizing the rigidity of multi-agent formations. In *IEEE Conference on Decision and Control and European Control Conference*, pages 2920–2925. December.