Distributed cooperative control for multi-agent systems

Guoguang Wen

N° d'ordre: | 1 | 9 | 5 |

ÉCOLE CENTRALE DE LILLE

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Automatique, Génie Informatique, Traitement du Signal et Image

par

Guoguang WEN

DOCTORAT DELIVRE PAR L'ECOLE CENTRALE DE LILLE

Titre de la thèse :

Distributed Cooperative Control for Multi-Agent Systems

Contrôle Coopératif Distribué pour Systèmes Multi-Agents

Soutenue le 26 Octobre 2012 devant le jury d'examen :

Président	M. Philippe MATHIEU	Professeur, Université de Lille 1
Rapporteur	M. Rachid ALAMI	Directeur de Recherche CNRS, LAAS, Toulouse
Rapporteur	M. Francois CHARPILLET	Directeur de Recherche INRIA, LORIA, Nancy
Rapporteur	M. Zhisheng DUAN	Professeur, Peking University, Chine
Membre	M. Nacer K.M'SIRDI	Professeur, Polytech Marseille
Membre	M. Yongguang YU	Professeur, Beijing Jiaotong University, Chine
Directeur de thèse	M. Ahmed RAHMANI	Maître de Conférences, École Centrale de Lille

Thèse préparée dans le Laboratoire d'Automatique, Génie Informatique et Signal
L.A.G.I.S., CNRS UMR 8219 - École Centrale de Lille
École Doctorale SPI 072
PRES Université Lille Nord-de France

À mes parents,

à toute ma famille,

à mes professeurs,

et à mes chèr(e)s ami(e)s.

Acknowledgements

This research work has been realized in École Centrale de Lille, in "Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS)", with the research team "Méthodes & Outils pour la Conception Intégrée de Systèmes (MOCIS)".

First and foremost I offer my sincerest gratitude to my supervisors, Mr. Ahmed Rahmani and Mr. Yongguang Yu who have provided his supervision, advices, and guidance as well as given me extraordinary experiences throughout the work. With their enormous helps, I have obtained some research experiences not only in multi-agent systems but also Intelligent Robotique control. Mr. Rahmani and Mr. Yu have provided an oasis of ideas for this report by a lot of discussions with me and my colleagues. I am indebted to them more than they know.

Mr. Francois CHARPILLET, Mr.RachidAlami and Mr. Zhisheng Duan used their precious times to review this PhD report before publication and gave various critical comments for shaping the final version of this report. They helped me refine the more delicate issues and present them from a more comprehensive perspective.

I would like to thank Mr. Belkacem Ould Bouamama, the team leader of MOCIS, who organized many scientific workshops helping us to exchange ideas. I also would like to thank Mrs. Genevieve Dauphin-Tanguy . During the three years, she was always close to us and encouraged us with enthusiasm for our research. I am grateful to her noble characters.

For the "Accueil à la française", I would like to thank Christine Yvoz, Daniel RiviÍÍre, Patrick Gallais, Régine Duplouich, Tram Nguyen, Vanessa Fleury and Virginie Leclercq who helped us with the administrative matters. I am also grateful to all my PhD student colleagues: Bo Zhang,Haoping Wang, Huarong Wang, Hui Liu, Jian Liu, Ran Zhao, Jiangfeng Liu, Jin Zhao,Jinlin Gong, Lamine

Chalal, Minzhi Luo, Naima Hadji, Wenhua TANG, Yahong Zheng , Yang Qian, Yang Tian, Yongliang Huang ,Youwei Dong, Sha Wang, Jing Bai, Hu wang, Miao Diao, Xia Meng, Lanrong Li etc.

All my gratitude goes to Ms. Hélène Catsiapis who showed us the French language and culture. She made possible all those interesting and unforgettable voyages in France. My knowledge and interest in the French culture is inspired by her work and enthusiasm, she managed to open our appetite for art, history and cuisine/wine.

My PhD study was supported by the cooperation between China and Scholarship Council(CSC) and Ecoles-Centrales Intergroup. So I would like to thank all of the people who have built this relationship and contributed to select candidates for interesting research projects. I am extremely grateful to CSC who financed me.

In the end, I convey special acknowledgement to my wife, Zhaoxia PENG, my parents Guihua Zhang and Hanli WEN as well as the rest of my family for their love and support. Their encouragement and understanding during the course of my dissertation make me pursue the advanced academic degree.

Villeneuve d'Ascq, France *Guoguang WEN*
October, 2012

Résumé en français

Les systèmes multi-agents sont composés d'un ensemble d'agents autonomes capables d'interagir les uns avec les autres et/ou avec leur environnement. Au cours des 20 dernières années, le contrôle coopératif des systèmes multi-agents a suscité beaucoup d'intérêt en raison de son utilisation dans l'étude et la compréhension des comportements de groupe d'individus comme les oiseaux, les poissons, les bactéries..., et mais aussi en raison à leurs vastes applications : réseaux de capteurs, groupe de robots , groupe de drones, etc... L'idée de base est que la coopération entre un ensemble d'agents permet d'effectuer des tâches que l'ensemble des agents opérant individuellement.

Afin d'assurer la coordination avec les autres agents dans un réseau, chaque agent doit partager l'information avec ses voisins de sorte que tous les agents puissent s'accorder sur un objectif d'intérêt commun, par exemple, la valeur de certaines mesures dans un réseau de capteurs, la direction d'une formation de UAVs, ou la position d'une cible pour un groupe de robots de surveillance. La nature interdisciplinaire du problème de contrôle coopératif en réseau de systèmes multi-agents a attiré une attention croissante de chercheurs dans divers domaines de la physique, des mathématiques, de l'ingénierie, la biologie et la sociologie.

Ce travail de thèse s'inscrit dans le cadre général de la coopération multi agents, et la commande distribuée en particulier.

Cette thèse considère principalement trois questions importantes dans le domaine du contrôle distribué coopératif des systèmes multi-agents: le problème du consensus (conditions d'obtention et du maintien), à la navigation en formation (constitution, planification de chemin et contrôle distribué de l'exécution du chemin par les différents agents constituant la formation) et enfin le problÌÍme du maintien en formation d'un groupe d'agents lorsqu'un'agent disparait.

Les principales contributions de cette thèse sont résumées ci dessous:

(1) Problème du consensus leadeur-suiveur pour les systèmes multi-agents Íď dynamique non-linéaire. Nous proposons trois algorithmes pour résoudre le problème du calcul distribué d'un consensus à partir de l'approche leadeur-suiveur dans le contexte multi-agent pour des systèmes à dynamique non-linéaire. La référence est définie comme un leader virtuel dont on n'obtient -localement- que les données de position et de vitesse. Le problème est abordé dans le cadre de différentes topologies de groupes d'agents (topologie fixe bidirectionnelles, topologie fixe unidirectionnelle et topologies variable avec liaisons unidirectionnelles). Des preuves rigoureuses sont établies, fondées notamment sur la théorie des graphes et celle de Lyapunov.

(2) **Problème du suivi par consensus pour les systèmes multi-agents à dynamique non-linéaire.**

Tout d'abord, nous considérons le suivi par consensus pour système multi-agents de premier ordre dans le cas d'une topologie du réseau fixe bidirectionnelle et de topologie du réseau non-orientée. On propose des résultats permettant aux suiveurs de suivre le leadeur virtuel en temps fini en ne considérant que les positions des agents. Ensuite, nous considérons le suivi par consensus de systèmes multi-agents de second ordre dans les cas de topologies bidirectionnelles fixe et variable. La stratégie proposée se base sur la propagation de l'information sur la position et la vitesse des voisins de chaque agent. Les résultats théoriques obtenus sont démontrés et validés par simulation.

(3) **Planification de trajectoire et la commande du mouvement de la formation multi-agents.**

L'idée est d'amener la formation, dont la dynamique est supposée être en 3 dimensions, d'une configuration initiale vers une configuration finale (trouver un chemin faisable en position et orientation) en maintenant sa forme tout le long du chemin en évitant les obstacles et les interblocages.
La stratégie proposée se décompose en trois étapes :
– Une discrétisation de l'environnement 3D vers une grille d'occupation. Un algorithme est proposé afin d'optimiser la trajectoire générée par un algorithme A*, puis quelques points redondants générés par l'algorithme A* sont éliminés.
– Afin de déplacer une formation persistante, un ensemble de lois de commande

décentralisées sont con?ues pour réaliser le mouvement cohésif dans les trois dimensions de l'espace.
– Basée sur ces lois de commande, la forme de la formation est maintenue pendant le mouvement continu à l'aide d'un ensemble de contraintes sur chaque agent et cela en fonction des contraintes de distance liant l'agent à ces voisins (nombre de degrés de liberté).
Les preuves de stabilité et de convergence de ces lois de commande sont données en s'appuyant sur la théorie de Lyapunov.

(4) Problème du maintien en formation rigide d'un système multi-agents.

Dans cette partie, nous traitons le problème du Closing-Ranks, qui se traduit par la réparation d'une formation rigide multi-agents "endommagée" par la perte de l'un de ses agents, afin de récupérer la rigidité. Tout d'abord, nous prouvons que si un agent est perdu, il suffit d'ajouter un ensemble de nouvelles arêtes sur ses sommets voisins tels que le sous-graphe qui est induit par les sommets voisins du sommet perdu est minimalement rigide, puis le graphe résultant est rigide. En utilisant le résultat, nous proposons deux opérations dąŕautoréparation systématique pour récuplęrer la rigidité en cas de perte d'un agent, appelé "Triangle Patch" et "Vertex Addition Patch". Compte tenu de l'exigence de la mise en ?uvre pratique, ces opérations sont proposées sous forme d'algorithmes. Ces réparations s'effectuent de manière décentralisée et distribuée n'utilisant que des informations de voisinage. Afin de garder un graphe minimal, nous considérons le Closing-Ranks basé sur l'opération de contraction des arcs (Edge-Contraction). Nous proposons une condition nécessaire et suffisante pour le cas où une arête d'un graphe minimalement rigide est contractée, ensuite, le principe du Edge-Remplacement est proposé. Tous nos résultats sont prouvés de façon rigoureuse en se basant sur la théorie des graphes, la théorie des matrices, et la théorie de Lyapunov. Des simulations ont permis d'illustrer l'analyse théorique à chaque fois.

Contents

Table of Contents 8

List of Figures 11

1 Introduction 17

1.1 Background and Motivation 17

1.1.1 Consensus problem 18

1.1.2 Planning and control of multi-agent formation 21

1.1.3 Formation keeping 23

1.2 Contributions of Dissertation 27

1.3 Organization of the Dissertation 28

2 Distributed Leader-Following Consensus for Multi-Agent Systems with Nonlinear Inherent Dynamics 31

2.1 Preliminaries 32

2.2 Problem description 34

2.3 Consensus under fixed undirected topology 36

2.4 Consensus under fixed directed topology 41

2.5 Consensus under switching undirected topology 44

2.6 Simulation examples 47

2.6.1 Case of fixed undirected topology 48

2.6.1.1 Dynamics of agents 48

2.6.1.2 Determination of α 49

2.6.1.3 Simulation results 49

2.6.2 Case of fixed directed topology 51

2.6.2.1 Dynamics of agents 51

2.6.2.2 Determination of α 51

2.6.2.3 Simulation results 51
2.6.3 Case of switching undirected topology 52
2.6.3.1 Dynamics of agents 52
2.6.3.2 Determination of α 52
2.6.3.3 Simulation results 53
2.7 Conclusion 53

3 Distributed Consensus Tracking for Multi-Agent Systems With Nonlinear Inherent Dynamics 57
3.1 Consensus Tracking for First-Order Multi-Agent Systems 58
3.1.1 Problem description 58
3.1.2 Consensus tracking under fixed undirected topology 59
3.1.3 Consensus tracking under switching undirected topology . 62
3.2 Consensus Tracking for Second-Order Multi-Agent Systems 66
3.2.1 Problem description 66
3.2.2 Consensus tracking under fixed undirected topology 67
3.2.3 Consensus tracking under switching undirected topology . 70
3.3 Numerical results 73
3.3.1 Case of consensus tracking for first-order multi-agent systems 73
3.3.1.1 Fixed undirected topology 73
3.3.1.2 Switching undirected topology 77
3.3.2 Case of consensus tracking for second-order multi-agent systems 79
3.3.2.1 Fixed undirected topology 79
3.3.2.2 Switching undirected topology 80
3.4 Conclusion 84

4 Planning and Control of Multi-Agent Formation 87
4.1 Preliminaries and problem statement 88
4.1.1 Preliminaries 88
4.1.1.1 Formation graph theory 88
4.1.1.2 Rigid formations 89
4.1.1.3 Persistent formations 93
4.1.2 Problem statement 98
4.2 Path planning 99

4.2.1 Grid Map Generation 99
4.2.2 Path Generation 101
4.3 Control laws for the formation 102
4.3.1 Control law for the 0-DOF agent 103
4.3.2 Control law for the 1-DOF agent 103
4.3.3 Control law for the 2-DOF agent 104
4.3.4 Control laws for the 3-DOF agents 105
4.4 Stability and convergence 106
4.4.1 Stability and convergence for the 3-DOF agent 106
4.4.2 Stability and convergence for the 2-DOF agent 107
4.4.3 Stability and convergence for the 1-DOF agent 109
4.4.4 Stability and convergence for the 0-DOF agent 111
4.5 Simulations 112
4.5.1 Verification of the effectiveness of control laws 112
4.5.2 Verification of the effectiveness of the framework for planning and control of formation 112
4.6 Conclusion 112

5 Rigid Formation Keeping in Multi-Agent systems 119
5.1 Preliminaries and problem statement 120
5.1.1 Preliminaries 120
5.1.2 Problem statement 121
5.2 Rigidity recovery based on two self-repair operations 122
5.2.1 The self-repair operations 124
5.2.2 Implementation algorithms 126
5.3 Rigidity recovery based on edge contraction operation 129
5.3.1 Edge contraction operation 129
5.3.2 A sufficient and necessary condition for an edge to be contractible 130
5.3.3 Minimal rigidity preserving problem 133
5.4 Conclusion 139

Conclusions and Perspectives 140

References 146

List of Figures

1.1 The intriguing animal group behaviors 18
1.2 A group of autonomous unmanned vehicles 19
1.3 An example of the splitting problem 23
1.4 An example of the merging problem 25
1.5 An example of the closing ranks problem 26

2.1 The fixed directed topology for a group of six followers and one virtual leader. 48
2.2 Chaotic behavior of Chua circuit. 49
2.3 Position states of the followers and virtual leader under a fixed undirected topology of Fig.2.1(a). 50
2.4 Velocity states of the followers and virtual leader under a fixed undirected topology of Fig.2.1(a). 50
2.5 Position states of the followers and virtual leader under a fixed directed topology of Fig.2.1(b). 52
2.6 Velocity states of the followers and virtual leader under a fixed directed topology of Fig.2.1(b). 53
2.7 Position states of the followers and virtual leader. 54
2.8 Velocity states of the followers and virtual leader. 54

3.1 Example scenario of obstacle avoidance for all agents. 74
3.2 Trajectories of the virtual leader and the followers under (3.52) and (3.53). The circle denotes the initial position of the virtual leader, while the squares denote the initial positions of the followers. 75
3.3 Position tracking errors of x positions under (3.52) and (3.53). . . 76
3.4 Position tracking errors of y positions under (3.52) and (3.53). . . 76

3.5 Trajectories of the virtual leader and the followers under (3.55) and (3.56). The circle denotes the initial position of the virtual leader, while the squares denote the initial positions of the followers. 78
3.6 Position tracking errors of x positions under (3.55) and (3.56). . . 78
3.7 Position tracking errors of y positions under (3.55) and (3.56). . . 79
3.8 Position states of the followers and virtual leader under algorithm (3.32). 81
3.9 Position tracking errors in x positions. 81
3.10 Position tracking errors in y positions. 82
3.11 Velocity tracking errors in x velocities. 82
3.12 Velocity tracking errors in y velocities. 83
3.13 Trajectories of the followers and virtual leader under algorithm (3.40). 84
3.14 Position tracking errors in x positions. 85
3.15 Position tracking errors in y positions. 85
3.16 Velocity tracking errors in x velocities. 86
3.17 Velocity tracking errors in y velocities. 86

4.1 Rigid and nonrigid formations in 2-dimensional space. 92
4.2 Rigid and nonrigid formations in 3-dimensional space. 93
4.3 Illustration of a formation that is not constraint consistent. 95
4.4 Persistent and nonpersistent formations in 2-dimensional space. . 95
4.5 Persistent and nonpersistent formations in 3-dimensional space. . 96
4.6 Flow chart of formation planning and control 114
4.7 Twenty-six possible orientations for the next point to move in path sequence for the given formation. 115
4.8 A minimally persistent formation with 5 agents in R^3. 115
4.9 The path for the motion of a formation with 5 agents. 116
4.10 Inner-agent distances during motion. 116
4.11 The path followed the center of mass of the formation when the obstacles are simple. 117
4.12 The motion path of a 3-dimensional Multi-agent Formation when the obstacles are simple. 117
4.13 The path followed the center of mass of the formation when the obstacles are more complex. 118

4.14 The motion path of a 3-dimensional Multi-agent Formation when the obstacles are more complex. 118

5.1 Representation of (a) vertex addition operation and reverse vertex addition operation (b) edge splitting operation and reverse edge splitting operation 121

5.2 Illustration of Triangle Patch: (a) the initial rigid formation, (b) the remaining formation after an agent and its incident edges loss, (c) the recovering formation using Triangle Patch operation. . . . 124

5.3 Illustration of Vertex Addition Patch: (a) the initial rigid formation, (b) the remaining formation after an agent and its incident edges loss, (c) the recovering formation using Vertex Addition Patch operation. 125

5.4 Utilizing existing cover edges with the Triangle Patch: (a) the lost vertex has degree 5, (b) edges exist among neighbor vertices of the lost vertex, (c) coordinators for the Triangle Patch are chosen to utilize 4 existing edges, and 3 new edges are added. 125

5.5 Utilizing existing cover edges with the Vertex Addition Patch: (a) the lost vertex has degree 5, (b) edges exist among neighbor vertices of the lost vertex, (c) coordinators for the Vertex Addition Patch are chosen to utilize 4 existing edges, and 3 new edges are added. 126

5.6 Illustration of algorithm for triangle patch. 127

5.7 Illustration of algorithm for vertex addition patch. 128

5.8 Closing ranks in a rigid formation using the edge contraction operation, where the dashed lines indicate deleted double edges . . . 129

5.9 Illustration of edge-replacement principle 134

5.10 The vertices k_1,k_2 and k_3 are common neighbors of the vertices u and v in a minimally rigid graph G represented by the graph (a). The graph (b) obtained by removing edges (u, k_2), (u, k_3) and (u, k_4), adding the edges (k_1, k_2), (k_2, k_3) and (k_3, k_4), is still minimally rigid. 135

5.11 The graph (a) is a minimally rigid graph. When the vertex v_1 is lost, we first add the edge (v_2, v_7) based on edge contraction operation. Then applying the operation in Theorem 5.10, we remove the edge (v_3, v_4) and add the edge (v_2, v_4). The resulting graph (b) is minimally rigid. 137

5.12 The graph in (a) is minimally rigid. One consider the closing ranks problem with loss of agent v_1. If we choose the edge (v_1, v_6) as the contraction edge, the resulting graph shown in (b) is minimally rigid. But if we contract the edge (v_1, v_3), then resulting graph shown in (c) is not rigid. Dashed lines indicate deleted double edges. 138

Chapter 1

Introduction

Contents

1.1 Background and Motivation **17**
1.1.1 Consensus problem 18
1.1.2 Planning and control of multi-agent formation 21
1.1.3 Formation keeping 23
1.2 Contributions of Dissertation **27**
1.3 Organization of the Dissertation **28**

1.1 Background and Motivation

Multi-agent systems are the systems that are composed of a lot of autonomous agents which are able to interact with each other or with their environments. During the last 20 years, the cooperative control of multi-agent systems has been attracting great attention due to wide interests in biological systems for understanding intriguing animal group behaviors, such as flocking of birds (Fig.1.1a), schooling of fishes(Fig.1.1b), and swarming of bacteria(Fig.1.1c), and also due to their emerging broad applications in sensor networks, UAV (Unmanned Air Vehicles) formations, AUVs (autonomous underwater vehicles), MRS (mobile robots systems), robotic teams, to name just a few (see Fig.1.2)(see Gazi & Fidan (2007);Das & Lewis (2011); Olfati-Saber & Murray (2004);Olfati-saber *et al.* (2007);Ren & Beard (2005); Ren *et al.* (2007);Oh & Ahn (2010)). The advancements are that a multi-agent system can perform tasks more efficiently than a single agent or can accomplish tasks not executable by a single one. Moreover, multi-agent systems have advantages like increasing tolerance to possible vehicle fault, providing flexibility to the task execution or taking advantage of distributed sensing and actuation. To coordinate with other agents in a network, every agent needs to share information with its adjacent peers so that all can agree on common goal of interest, e.g. the value of some measurement in a sensor network, the heading of a UAV formation, or the target position of robotic team. Because of the interdisciplinary nature of the field, the study of cooperative control problem of networked multi-agent systems has attracted increasing attention from researchers in various fields of physics, mathematics, engineering, biology, and sociology.

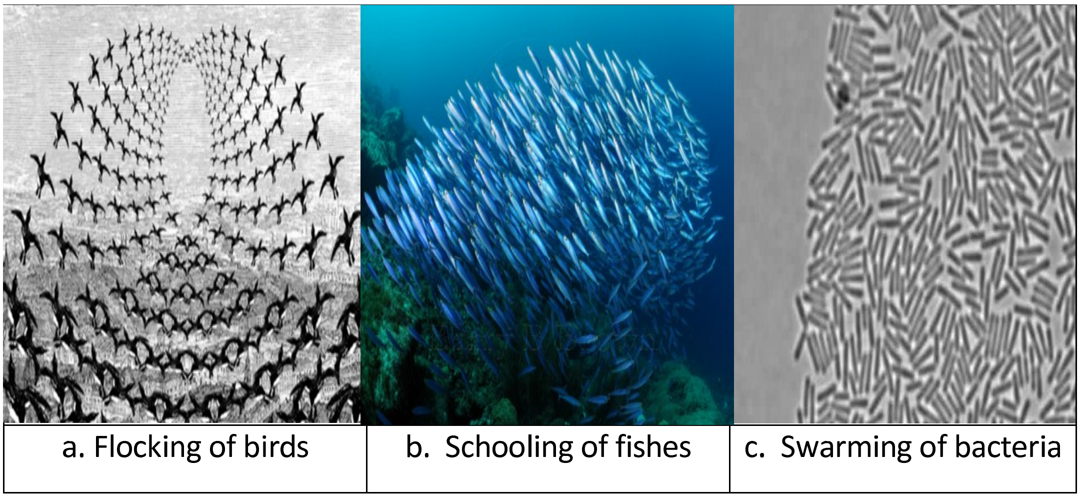

Figure 1.1: The intriguing animal group behaviors

As a consequence to this growing interest, some progress has been made in this area: such as flocking, formation control, formation keeping, and consensus problem. This dissertation mainly considers three important issues in this field of the cooperative control of multi-agent systems: consensus problem for multi-agent systems; planning and control of multi-agent formation; formation keeping.

1.1.1 Consensus problem

Consensus problem, one of the most important and fundamental issues in the cooperative control, has attracted great attention from researchers in recent years

Figure 1.2: A group of autonomous unmanned vehicles

because of broad applications in various real-world multi-agent systems, such as cooperative control of unmanned (air) vehicles, formation control of robots and aircrafts, and design of sensor networks, to name a few (see the survey papers Ren & Beard (2005),Gazi & Fidan (2007), and extensive references therein). The consensus of multi-agent systems means to design a network distributed control policy based on the local information obtained by each agent, such that all agents reach an agreement on certain quantities of interest by negotiating with their neighbors. The quantities of interest might represent attitude, position, velocity, temperature, voltage and so on, depending on different applications.

Numerous interesting results for consensus problem have been obtained in the past decade. Reynolds (1987) first proposed a computer animation model to simulate collective behaviors of multiple agents. Vicsek *et al.* (1995) proposed a discrete-time distributed model to simulate a group of autonomous agents moving in the plane with the same speed but different heading. Jadbabaie *et al.* (2003) gave a theoretical explanation for the consensus behavior of the vicsek's model based on graph and matrix theories. Ren & Beard (2005) extended the result of Jadbabaie *et al.* (2003) and gave some more relaxed conditions . Olfati-Saber & Murray (2004) investigated a systematical framework of consensus problems with undirected interconnection graphs or time-delay by Lyapunov-based approach. Lately, the research of consensus problems for multi-agent systems was also extended to the case of directed topology (Cao *et al.* (2008);Moreau (2005);Olfati-Saber & Murray (2004);Ren & Beard (2005);Jiang & L.Wang (2009);Yu *et al.* (2010);Yu *et al.* (2011)).

Recently, a (virtual) leader-follower approach has been widely used for the consensus problem (Cao & Ren (2012);Peng & Yang (2009);Olfati-Saber (2006);SU *et al.* (2011);Shi *et al.* (2006);Shi *et al.* (2009);Hong *et al.* (2008)). Such a problem is commonly called leader-following consensus problem. In Hong *et al.* (2008), leader-following consensus of multi-agent systems with double-integrator dynamics was studied. However, Hong *et al.* (2008) requires the availability of the acceleration of the virtual leader. Compared with previous works of Hong *et al.* (2008), Cao & Ren (2012) studied a distributed leader-following consensus problem with single-integrator dynamics and double-integrator dynamics under fixed and switching undirected network topologies. It was shown that the velocity measurements of the virtual leader and followers are not required, and the consensus tracking can be achieved in finite time.

The most commonly used models to study the multi-agent systems are single integrator and double integrator models. However, in reality, mobile agents may be governed by more commonly inherent nonlinear dynamics. In recent literatures, Yu *et al.* (2011) studied the leaderless consensus problem for second-order systems with nonlinear inherent dynamics under fixed network topology, in where the general algebraic connectivity in strongly connected networks plays a key role in reaching consensus. In addition, most of aforementioned literature concentrate on studying consensus under fixed communication topology. However, In many applications, the communication topology might be changed. For example, communication links between agents may be unreliable due to communication range limitations. Therefore, the consensus problem under switching topology is also important. In this dissertation, our objective is to investigate the consensus problem for multi-agent systems with nonlinear inherent dynamics under fixed and switching topologies.

1.1.2 Planning and control of multi-agent formation

The formation planning problem is one of the important areas of interest in multi-agent system. As we know, autonomous agents which work without human operators are required in multi-agent formation fields. In order to achieve tasks, agents have to be intelligent and should decide their own action. When the agent decides its action, it is necessary to plan optimally depending on their tasks. Moreover, it is necessary to plan a collision free path that minimizes a cost function such as of time, energy and distance. When an autonomous agent moves from an initial point to a target point in its given environment, it is necessary to plan an optimal or feasible path avoiding obstacles in its way and providing answer to the criterion of autonomy requirements such as : thermal, energy, time, and safety. Therefore, the major work for path planning for autonomous agent is to search a collision free and obstacle avoidance path in an environment with obstacles. Many of the related studies on path planning for the formation are discussed under the condition that the paths are already given as some functions by assumptions (Gilbert & Johnson (1985);Ge & Cui (2000);Lian & Murray (2002)), but in many cases we can only know the initial position and desired position of the formation, the path functions are impossible to get in advance. To investigate the problem, an alternate solution is to use a real-time objective function to determine the motion, but as the function is always complex, and the final trajectory is usually not the optimal one. The other alternative consists of partitioning the environment with a mesh to which a connectivity graph is associated. A heuristic graph search is then used to find a path which is eventually optimized by virtue of a dynamic programming scheme (Hao & Agrawal (2005)). Similar problems are also discussed in (Ngo *et al.* (2005);Shames *et al.* (2007)).

The formation control is particularly active areas of multi-agent systems since many cooperative tasks are performed by controlling the agents to move cohesively in a specified geometrical pattern. That is to say, the formation satisfies some constraints such as the desired distance between two agents, the desired angle between two lines joining both the agents. The examples can be found in a variety of practical tasks, such as collective attack by a group of combat aircraft, search/ rescue/ surveillance/ cooperative transportation by a troop of robots, underwater exploration/ underwater inspection by a group of AUVs, attitude alignment of clusters of satellites, air traffic management system, automated highway system. Correspondingly, many of the related studies on control of formation are also discussed. In Hao & Agrawal (2005), a theoretical framework is constructed, which consists of graph theoretical analysis, the stability analysis based on Lyapunov approaches and distributed control of multi-agent formations, and the formation control of the whole system is implemented via controlling each agent respectively. In Ngo *et al.* (2005), the task of following a prescribed trajectory for formation, without breaking the topology of control interactions between agents, is discussed, where the shape of the formation is flexible, i.e., no constraint is introduced on the inter-agent distances. In the recent literature, the formation control problem is considered under a recently developed theoretical framework of graph rigidity and persistence. Eren *et al.* (2004) proposed the use of graph rigidity theory for modeling information architectures of formations. The information structure arising from such a formation system can be efficiently modeled by a graph, where agents are abstracted by vertices and actively constrained inter-agent distances by edges. Moreover, Laman (1970) gave a necessary and sufficient condition for a graph to rigid in 2-dimensional space. Tay & Whiteley (1985) provided an extensive review of the state of the art regarding rigid graph. Henneberg (1911) proposed some rules for building rigid graph and proved that every minimally rigid graph can be obtained as the result of vertex addition operation or edge splitting operation. In recently, motion control strategy of rigid and persistent is shown as significant interest. in Sandeep *et al.* (2006), a set of decentralized control laws for the cohesive motion of 2-dimensional multi-agent formation have been presented (The aim is to move a given persistent formation with specified initial position and orientation to a new desired position and orientation cohesively, i.e., without violating the persistence of the formation during the motion). Based on Sandeep *et al.* (2006), two strategies for obstacle avoidance by the entire formation have also been studied further in Shames *et al.* (2007).

In the above literatures, most studies have been focused on either the agent motion planning or the control problem. They are usually treated as two separate problems. Moreover, most of the above works are only based on the 2-dimensional space. However, in the real world, the applications of multi-agent formations are often in a 3-dimensional space as opposed to the plane. Furthermore, in the 3-dimensional space, there are many characteristics which are different from that in 2-dimensional space. Multi-agent formations problem is much more complicated than that in 3-dimensional space. Therefore, it is necessary to study the planning and control problem for multi-agent formation in 3-dimensional space. Toward this objective, Wang *et al.* (2007) discussed the problems of formation flying of multiple UAVs navigating through an obstacle-laden environment. A dual-model approach was presented for formation flying in safe and danger modes, and a technique for obstacle/collision avoidance through a modified Grossberg neural network was developed. Zhao & Tiauw (2011) studied a 3D formation flight control problem. In this paper (Zhao & Tiauw (2011)), based on multiplexed model predictive control, the formation flight with collision free and obstacle avoidance is achieved.

1.1.3 Formation keeping

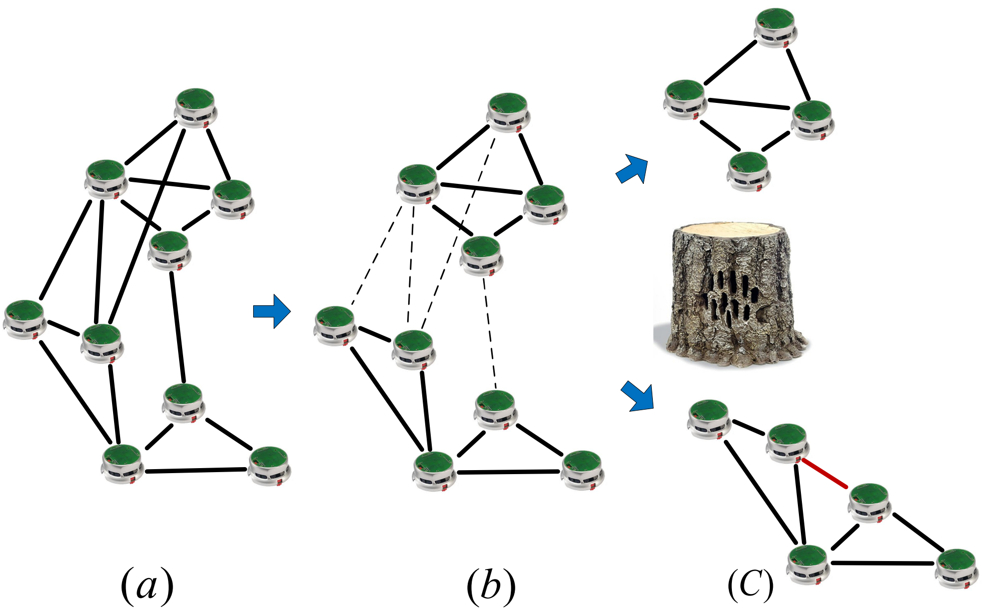

Figure 1.3: An example of the splitting problem

The study of formation keeping is motivated by the obvious advantages achieved by using formations of multiple agents to accomplish an objective. These include increased feasibility, accuracy, robustness, flexibility, cost and energy efficiency, and probability of success. For example, the probability of success will be improved if multiple vehicles are used to carry out a mission in a coordinated manner, e.g. multiple UAVs are assigned to a certain target or multiple AUVs are used to search for an underwater object. Also, cost and energy efficiency may be maximized if multiple agents coordinate their movements in a certain way, e.g. multiple aircraft flying in a V-shape formation to maximize fuel efficiency. Furthermore, spacecraft interferometer applications in deep space, using formations of multiple microspacecraft instead of a monolithic spacecraft can reduce the mission cost and improve system robustness and accuracy. In the recent literature (Eren *et al.* (2004);Anderson *et al.* (2006)), there exist three key operations for the formation keeping. They are named as merging, splitting and closing ranks. The splitting operation may occur because of a change of objective, or to avoid an obstacle etc. Then the agents of a rigid formation are divided into two subsets, and the distance constraints between agents in different subsets are suppressed. In graph theory terms, after the split, there are two separate (sub-)graphs, neither of which may be rigid. The task is to add new distance constraint in the separate subformations to ensure rigidity of both them . The problem is illustrated in Fig. 1.3. Some distance constraints in a rigid formation as shown in Fig.1.3(a) are removed to separate the agents into two independent subsets as shown in Fig.1.3(b). Note that the two independent subsets are not rigid. To regain the rigidity of two subformations, new links are added within each formation of agents, as in the original formation, and are incident to the removed edges (Fig.1.3(c)). In merging operation, two given rigid formations are required to be merged in a single formation via adding some extra links such that the resultant single formation is also rigid. More specifically, the task is to determine the additional distance constraints, with one agent in each formation, such that the union of the agents of the two formations, and the union of the distance constraints in the original formations and the new distance constraints, will describe a single rigid formation. The problem is illustrated in Fig.1.4. Two distinct rigid formation are shown in Fig.1.4(a). They are merged into the large rigid formation as shown in Fig.1.4(b) by inserting three interconnecting edges. The closing ranks problem deals with the addition of links to a rigid formation that is “damaged" by losing one of its agents, in order to recover its rigidity. This problem is motivated by the fact that a single agent or a small number of agents could be lost in formations in real world applications because of the following reasons: enemy attack or jamming, random mechanical or electrical failure, and intentionally deploying an agent for a separate task. Consequently, this leads to the failure of

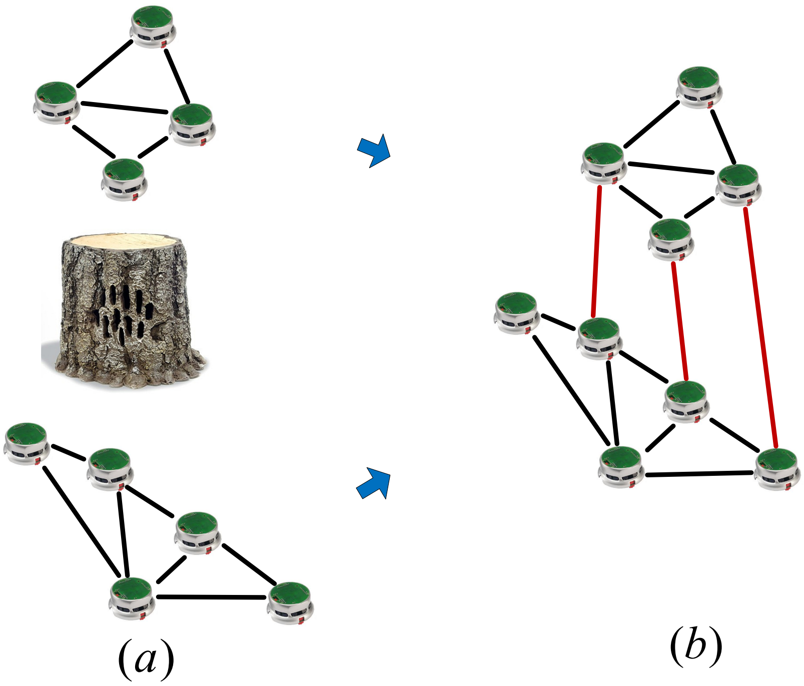

Figure 1.4: An example of the merging problem

some performances of tasks such as formation shape control and self-localization. So it is very important to take the 'self-repair' operation in the formation shape maintenance task. To illustrate the closing ranks problem intuitively, an example is given in Fig.1.5. It is shown that in an initial rigid formation with minimum communication links (Fig.1.5 (a)), when an agent is removed (Fig.1(b)), then the remaining formation is not rigid(Fig.1.5(b)). In order to recover the rigidity of the formation, just one new link created in the "damaged" formation would be sufficient as depicted in Fig.1.5(c). From this example, it is clear to see that the main goal of closing ranks problem is to give a systematic way for creating the minimum number of new links to maintain the rigidity of a formation when a agent is lost. The terms "rigid" will be introduced in Chapter 4. From the above discussion, we can note that the splitting problem is actually a particular case of the closing ranks problem. In this dissertation, we will only consider closing ranks problem.

There are numerous studies on the formation keeping. Also, various strategies and approaches have been proposed. Based on a graph theoretic theorem, Tay & Whiteley (1985) proved that there exist new links among the neighbors of lost agent can be added to recover rigidity of 'damaged' formation. Then based

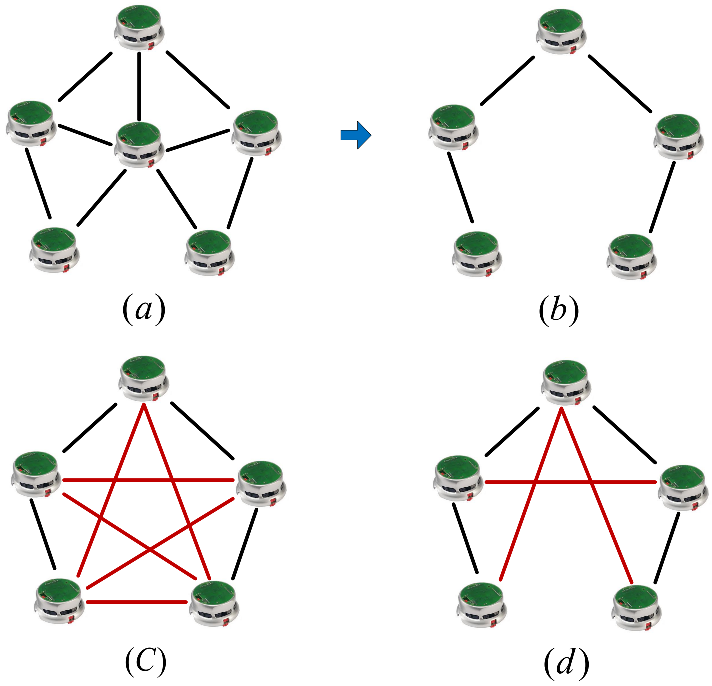

Figure 1.5: An example of the closing ranks problem

on this result, Eren *et al.* (2002) presented a systematic method to solve the problem restoring rigidity with local repair, but it cannot always be implemented using only local information. Summers *et al.* (2008) and Summers *et al.* (2009) introduced two closing ranks algorithms, named double patch and wheel patch, which can be implemented using only local information. However, in wheel patch algorithm, the number of edges added is more than 2k-3 which is the number of edges to generate minimally rigid graph. Recently, Fidan *et al.* (2010) proposed a new operation method named Edge Contraction to the problem of restoring rigidity after loss of an agent. It is shown that, if an edge corresponding to an information link incident to lost agent, is contractible, then it is simple to find the edges which need to be added for recovering rigidity of 'damaged' formation. However, this paper (Fidan *et al.* (2010)) doesn't give a sufficient and necessity condition for an edge to be contractible.

1.2 Contributions of Dissertation

This dissertation considers the cooperative control of multi-agent systems. Several important and related problems in this field are studied such as: consensus problem for multi-agent systems; planning and control of multi-agent formation; formation keeping in multi-agent systems. The main contributions of this dissertation are summarized as follows:

(1) To solve the leader-following consensus problem for multi-agent systems with nonlinear inherent dynamics, we propose three consensus algorithms under fixed undirected communication topology, fixed directed communication topology and switching undirected communication topology, respectively, in which each follower only needs the position and velocity information of its neighbors. It is shown that leader-following consensus for multi-agent systems are well achieved with the proposed protocols.

(2) To study the consensus tracking problem for multi-agent systems with nonlinear inherent dynamics, some distributed consensus algorithms according to the fixed and switching undirected communication topologies are developed. In first-order multi-agent systems each follower only needs the position information of its neighbors, and in second-order multi-agent systems each follower only needs the position and velocity information of its neighbors. It is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus tracking for first-order multi-agent systems can be achieved in finite time,and the consensus tracking for second-order multi-agent systems can be achieved at least globally exponentially.

(3) To study the path planning and motion control of multi-agent formation, a practicable framework is provided. In order to find a collision-free and deadlock-free feasible path for the whole formation, an optimizing algorithm is given to optimize the path generated by using A^* search algorithm. In order to move a persistent formation as a large entity, a set of decentralized control laws are designed for realizing the cohesive motion in 3-dimensional space. Furthermore, the rigorous proofs regarding the stability and convergence of proposed control laws are also given.

(4) To consider the formation keeping problem in multi-agent systems. We mainly focus on the closing ranks problem which deals with the addition of links to a rigid multi-agent formation that is "damaged" by losing one of its agents, in

order to recover rigidity. Firstly, we prove that if an agent is lost, we only need to add a set of new edges on its neighbor vertices such that the subgraph which is induced by neighbor vertices of the lost vertex is minimally rigid, and then the resulting graph is rigid. Utilizing the result, we propose two systematic 'self-repair' operations to recover the rigidity in case of agent removals, named Triangle Patch and Vertex Addition Patch. Considering the requirement of practical implementation, we give two algorithms corresponding the above two operations. As these two operations cannot guarantee that the result graph is minimally rigid. Next, we consider the closing ranks based on another graph operation named edge contraction. First, we propose a sufficient and necessary condition for the case that an edge of a minimally rigid graph is contractible: an edge in a minimally rigid graph is contractible if and only if there are no two minimally rigid subgraphs such that their intersection consists of the contraction edge and its two incident vertices. Later, an Edge-Replacement Principle is proposed. Based on Edge-Replacement Principle and edge contraction operation, the minimally rigidity preserving problem is studied. A set of graph theoretical results operations are established to solve the corresponding closing ranks problems.

1.3 Organization of the Dissertation

The remainder of the dissertation is organized as follows.

In Chapter 2, the distributed leader-following consensus for multi-agent systems with nonlinear inherent dynamics is investigated. The consensus reference is taken as a virtual leader, whose output are only its position and velocity information that is available to only a set of a group of followers. In this chapter, the consensus tracking problem is considered under the following three topologies: 1) the fixed undirected network topology; 2) the fixed directed network topology; 3) the switching undirected network topology. In the fixed or switching undirected network topology case, it is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus can also be achieved globally exponentially with the proposed protocol. In the fixed directed network topology case, it is shown that if the directed graph associated with the virtual leader and followers contains a directed spanning tree, the consensus can also be achieved globally exponentially with the proposed protocol.

Rigorous proofs are given by using graph theory, matrix theory, and Lyapunov theory. Some sample simulations are presented to illustrate the theoretical analysis.

In Chapter 3, we investigate the consensus tracking problem for multi-agent systems with nonlinear dynamics. First, we consider the consensus tracking for first-order multi-agent systems under fixed undirected network topology and switching undirected network topology. It is shown that if the undirected graph associated with the virtual leader and followers is connected, all followers can track the virtual leader in finite time with the proposed protocol under only the position measurements of the followers and the virtual leader. Next, we consider the consensus tracking for second-order multi-agent systems under fixed undirected network topology and switching undirected network topology. The proposed consensus control strategy is implemented based on only the position and velocity information of each agent's neighbors. It is assumed that only one follower knows the position and velocity information of virtual leader, and it is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus tracking can be achieved at least globally exponentially. Rigorous proofs are given by using graph theory, matrix theory, and Lyapunov theory. Finally, simulation examples are given to verify the theoretical analysis.

In Chapter 4, we provide a practicable framework for path planning and motion control of 3-dimensional multi-agent formation. The formation path planning is performed by using the A^* search algorithm coupled with an optimizing algorithm to generate a collision-free and deadlock-free feasible path. For the formation motion control, we propose a set of decentralized control laws for the cohesive motion of 3-dimensional multi-agent formation with a point-agent system model. Based on these control laws, the shape of the formation is maintained during continuous motion, which can efficiently avoid inter-agent collision. Rigorous proofs regarding the stability and convergence of proposed control laws are given by using Lyapunove theory. The effectiveness of the proposed framework is demonstrated by simulation.

In Chapter 5, we consider the closing ranks problem which deals with the addition of links to a rigid multi-agent formation that is "damaged" by losing one of its agents, in order to recover rigidity. We model the information architecture of formation as a graph, where each vertex represents an agent in formation, and each edge represents a communication link between a pair of agents. Firstly, we prove that if an agent is lost, we only need to add a set of new edges on its neighbor vertices such that the subgraph which is induced by neighbor vertices of the lost vertex is minimally rigid, and then the resulting graph is rigid. Utilizing the result, we propose two systematic 'self-repair' operations to recover the rigidity in case of agent removals, named Triangle Patch and Vertex Addition Patch. Considering the requirement of practical implementation, we give two algorithms corresponding the above two operations. As these two operations cannot guarantee that the result graph is minimally rigid. Next, we consider the closing ranks based on another graph operation named edge contraction. First, we propose a sufficient and necessary condition for the case that an edge of a minimally rigid graph is contractible: an edge in a minimally rigid graph is contractible if and only if there are no two minimally rigid subgraphs such that their intersection consists of the contraction edge and its two incident vertices. Later, an Edge-replacement Principle is proposed. Based on Edge-replacement Principle and edge contraction operation, the minimally rigidity preserving problem is studied. A set of graph theoretical results operations are established to solve the corresponding closing ranks problems.

In chapter 6, we summarize the results in this dissertation and identifies several possible directions for future research.

Chapter 2

Distributed Leader-Following Consensus for Multi-Agent Systems with Nonlinear Inherent Dynamics

Contents

2.1 Preliminaries **32**

2.2 Problem description **34**

2.3 Consensus under fixed undirected topology **36**

2.4 Consensus under fixed directed topology **41**

2.5 Consensus under switching undirected topology . . . **44**

2.6 Simulation examples **47**

2.6.1 Case of fixed undirected topology 48

2.6.2 Case of fixed directed topology 51

2.6.3 Case of switching undirected topology 52

2.7 Conclusion **53**

In this chapter, the distributed leader-following consensus for second-order multi-agent systems with nonlinear inherent dynamics is investigated. The consensus reference is taken as a virtual leader, whose output are only its position and velocity information that is available to only a set of a group of followers. In this chapter, the consensus tracking problem is considered under the following three topologies: 1) the fixed undirected network topology; 2) the fixed directed network topology; 3) the switching undirected network topology. In the fixed or switching undirected network topology case, it is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus can also be achieved globally exponentially with the proposed protocol. In the fixed directed network topology case, it is shown that if the directed graph associated with the virtual leader and followers contains a directed spanning tree, the consensus can also be achieved globally exponentially with the proposed protocol. Rigorous proofs are given by using graph theory, matrix theory, and Lyapunov theory. Some sample simulations are presented to illustrate the theoretical analysis.

2.1 Preliminaries

In this chapter, a graph will be used to model communication topology among n agents. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ be a weighted graph of order n with the finite nonempty set of nodes $\mathcal{V} = \{v_1, ..., v_n\}$, the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and a weighted adjacency matrix $\mathcal{A} = (a_{ij})_{n\times n}$. Here, each node v_i in $\mathcal{V}$ corresponds to an agent i, and each edge $(v_i, v_j) \in \mathcal{E}$ in a weighted directed graph corresponds to an information link from agent j to agent i, which means that agent i can receive information from agent j. In contrast, the pairs of nodes in weighted undirected graph are unordered, where an edge $(v_j, v_i) \in \mathcal{E}$ denotes that agent i and j can receive information from each other.

The weighted adjacency matrix $\mathcal{A}$ of a digraph $\mathcal{G}$ is represented as

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \in \mathcal{R}^{n\times n}$$

where a_{ji} is the weight of the link (v_i, v_j) and $a_{jj} = 0$ for any $\nu_j \in \mathcal{V}, a_{ji} > 0$ if $(\nu_i, \nu_j) \in \mathcal{E}$, and $a_{ji} = 0$ otherwise. The weighted adjacency matrix $\mathcal{A}$ of a weighted undirected graph is defined analogously except that $a_{ij} = a_{ji}, \forall i \neq j$, since $(v_i, v_j) \in \mathcal{E}$ implies $(v_j, v_i) \in \mathcal{E}$. We can say v_i is a neighbor vertex of v_j, if $(v_i, v_j) \in \mathcal{E}$.

The Laplacian matrix $L = (l_{ij})_{n\times n}$ of graph $\mathcal{G}$ is defined by

$$l_{ij} = -a_{ij}$$

$$l_{ii} = \sum\nolimits_{j=1, j\neq i}^{n} a_{ij}, i \neq j; i, j \in \{1, ..., n\}.$$

For an undirected graph, L is symmetric positive semi-definite. However, L is not necessarily symmetric for a directed graph.

Definition 2.1 *(Ren & Beard (2005)): A directed path from node v_i to v_j is a sequence of edges $(v_i, v_{j1}), (v_{j1}, v_{j2}), ..., (v_{jl}, v_j)$ in a directed graph $\mathcal{G}$ with distinct nodes $v_{jk}, k = 1, ..., l$.*

Definition 2.2 *(Ren & Beard (2005)): The directed graph $\mathcal{G}$ is said to have a directed spanning tree if there is a node (called root node) that can reach all the other nodes following a directed path in graph $\mathcal{G}$.*

Lemma 2.3 *(Chung (1997)): A weighted undirected graph $\mathcal{G}$ is connected if and only if its Laplacian matrix L has a eigenvalue zero with multiplicity 1 and corresponding eigenvector **1**, and all other eigenvalues have positive real parts.*

Lemma 2.4 *(Ren & Beard (2005)): A weighted directed graph $\mathcal{G}$ has a directed spanning tree if and only if its Laplacian matrix L has a eigenvalue zero with multiplicity 1 and corresponding eigenvector **1**, and all other eigenvalues have positive real parts.*

Definition 2.5 *(Horn & Johnson (1985)) Let $A \in R^{m\times n}$, $B \in R^{p\times q}$. Then the Kronecker product $\otimes$ (or tensor product) of A and B is defined as the matrix*

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathcal{R}^{mp\times nq}.$$

More explicitly, we have

$$\mathbf{A}\otimes\mathbf{B}=\begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}.$$

Example:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\otimes\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix}=\begin{bmatrix} 1\cdot 0 & 1\cdot 5 & 2\cdot 0 & 2\cdot 5 \\ 1\cdot 6 & 1\cdot 7 & 2\cdot 6 & 2\cdot 7 \\ 3\cdot 0 & 3\cdot 5 & 4\cdot 0 & 4\cdot 5 \\ 3\cdot 6 & 3\cdot 7 & 4\cdot 6 & 4\cdot 7 \end{bmatrix}=\begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

Lemma 2.6 *(Horn & Johnson (1985)) Kronnecker product $\otimes$ have the following properties: for matrices A, B, C and D with appropriate dimensions,*
(1) $(A+B)\otimes C = A\otimes C + B\otimes C$;
(2) $(A\otimes B)(C\otimes D) = AC\otimes BD$;
(3) $(A\otimes B)^T = A^T\otimes B^T$;
(4) $(\xi A)\otimes B = A\otimes(\xi B)$,*where ξ is a constant.*

Notations: Some mathematical notations are used throughout this chapter. Let I_n denote the $n\times n$ identity matrix, $0_{m\times n}$ denote the $m\times n$ zero matrix, and $\mathbf{1}_n = [1,1,\ldots,1]^T \in R^n$ ($\mathbf{1}$ for short, when there is no confusion). $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the smallest and the largest eigenvalues of the matrix A respectively.

2.2 Problem description

Consider a multi-agent system that is made up of one virtual leader (labeled as 0) and n followers (labeled as 1 to n). Let the graph $\mathcal{G}$ represent the communication topology of all followers.

The dynamics of each follower i $(i = 1, ..., n)$ is given by

$$\begin{aligned} \dot{\xi}_i(t) &= v_i(t), \\ \dot{v}_i &= f(t, \xi_i(t), v_i(t)) + u_i(t), \end{aligned} \tag{2.1}$$

where $\xi_i(t) \in R^m$ is the position vector, $f(t, \xi_i(t), v_i(t)) \in R^m$ is its nonlinear inherent dynamics, and $u_i(t)$ is its control input. When $f(t, \xi_i(t), v_i(t)) \equiv 0$, the multi-agent system has double-integrator dynamics.

The dynamics of the virtual leader 0 is given by

$$\begin{aligned} \dot{\xi}_0(t) &= v_0(t), \\ \dot{v}_0(t) &= f(t, \xi_0(t), v_0(t)), \end{aligned} \tag{2.2}$$

where $\xi_0(t) \in R^m$ and $f(t, \xi_0(t), v_0(t)) \in R^m$ are, respectively, the position states and a nonlinear vector-valued continuous function to describe the dynamics of virtual leader.

The consensus problem of the multi-agent system (2.1) is to design the control inputs $u_i(t), i = \{1, ..., n\}$, such that the below equation (2.3) is satisfied

$$\begin{aligned} \lim_{t\to\infty} \left\|\xi_i(t) - \xi_0(t)\right\|_2 = 0, \\ \lim_{t\to\infty} \left\|v_i(t) - v_0(t)\right\|_2 = 0, \end{aligned} \tag{2.3}$$

for any i and for any arbitrary initial position states.

We suppose that the virtual leader share the same nonlinear inherent dynamics with all followers, and these nonlinear inherent dynamics satisfy a Lipchitz-type condition given by Definition 2.7 as follows, which is satisfied in many well-known systems.

Definition 2.7 $\forall \xi, v, \zeta, \gamma \in R^m; \forall t \geq 0$, *there exist two nonnegative constants* l_1 *and* l_2 *such that*

$$\left\|f(t, \xi, v) - f(t, \zeta, \gamma)\right\|_2 \leq l_1 \left\|\xi - \zeta\right\|_2 + l_2 \left\|v - \gamma\right\|_2. \tag{2.4}$$

Remark 1. Note from Definition 2.7 that the constants l_1 or l_2 might be zero. In this chapter, in order to satisfy the requirements of designing consensus protocols, we assume that $r_1 = \lfloor l_1 \rfloor + 1$ and $r_2 = \lfloor l_2 \rfloor + 1$, where $\lfloor \cdot \rfloor$ denotes floor function (i.e., $\lfloor x \rfloor = \max\{m \in \mathbb{Z} \,|m \leq x\}$, where $\mathbb{Z}$ is the set of integers.).

In this chapter, the consensus problem is considered under fixed undirected topology, fixed directed topology, and switching undirected topology.

2.3 Consensus under fixed undirected topology

In this section, we consider the fixed undirected topology case. To satisfy the equation (2.3), we consider the following control input (2.5),

$$u_i(t) = -\alpha \sum_{j=0}^{n} a_{ij}[r_1(\xi_i(t) - \xi_j(t)) + r_2(v_i(t) - v_j(t))], \tag{2.5}$$

where α is a nonnegative constant and $a_{ij}, i, j = 1, ...n$, is the (i, j)th entry of the adjacency matrix $\mathcal{A}$ associated to $\mathcal{G}$. Note that $a_{i0} > 0$ $(i = 1, ..., n)$ if the virtual leader's position and velocity information are available to follower i, and $a_{i0} = 0$ otherwise.

Substituting (2.5) to (2.1) gives

$$\begin{aligned} \dot{\xi}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= f(t, \xi_i(t), v_i(t)) - \alpha \sum_{j=0}^{n} a_{ij}[r_1(\xi_i(t) - \xi_j(t)) \\ &\quad + r_2(v_i(t) - v_j(t))]. \end{aligned} \tag{2.6}$$

Let $M = L + diag(a_{10}, ..., a_{n0})$, where L is the Laplacian matrix of $\mathcal{G}$.

Lemma 2.8 *(Wen et al. (2012 b)) Suppose that the fixed undirected graph $\mathcal{G}$ is connected and at least one a_{i0} is positive. Let $H = \begin{bmatrix} \frac{1}{2}\alpha\frac{r_2}{r_1}M & \frac{1}{2r_2}I_n \\ \frac{1}{2r_2}I_n & \frac{1}{2r_2}I_n \end{bmatrix}$ and $Q = \begin{bmatrix} \alpha M & \frac{1}{2}\alpha M \\ \frac{1}{2}\alpha M & \alpha M - \frac{r_1}{r_2^2}I_n \end{bmatrix}$. If $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\lambda_{\min}(M)}$, then the matrices H and Q are symmetric positive definite, and $\lambda_{\min}(Q) > 2$.*

Proof: Since the fixed undirected graph $\mathcal{G}$ is connected and at least one a_{i0} is positive, the matrix M is symmetric positive definite. Therefore M can be diagonalized as $M = \mathrm{T}^{-1}\Delta\mathrm{T}$, where $\Delta = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ with λ_i being the ith eigenvalue of M. Then we can obtain a matrix Π such that

$$\begin{aligned}\Pi &= \begin{bmatrix} \frac{1}{2}\alpha\frac{r_2}{r_1}\Delta & \frac{1}{2r_2}I_n \\ \frac{1}{2r_2}I_n & \frac{1}{2r_2}I_n \end{bmatrix} \\ &= \begin{bmatrix} T & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & T \end{bmatrix} H \begin{bmatrix} T^{-1} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & T^{-1} \end{bmatrix},\end{aligned} \tag{2.7}$$

where $\mathbf{0}_{n\times n}$ is the $n \times n$ zero matrix. It is easy to see that Π is symmetric and has the same eigenvalues as H.

Let δ be an eigenvalue of Π. Because Δ is a diagonal matrix, it follows from (2.7) that δ satisfies

$$(\delta - \frac{\alpha r_2}{2r_1}\lambda_i)(\delta - \frac{1}{2r_2}) - \frac{1}{4r_2^2} = 0,$$

which can be simplified as

$$\delta^2 - \frac{\alpha r_2^2\lambda_i + r_1}{2r_1 r_2}\delta + \frac{\alpha r_2^2\lambda_i - r_1}{4r_1 r_2^2} = 0. \tag{2.8}$$

Since the matrix Π is symmetric, all roots of (2.8) are real. Therefore, all roots of (2.8) are positive if and only if

$$\frac{\alpha r_2^2\lambda_i + r_1}{2r_1 r_2} > 0$$

and

$$\frac{\alpha r_2^2\lambda_i - r_1}{4r_1 r_2^2} > 0.$$

Since the matrix M is symmetric positive definite, $\lambda_i > 0$. Then we can obtain that H is positive definite if

$$\alpha > \frac{r_1}{r_2^2\lambda_{\min}(M)}. \tag{2.9}$$

Next, we consider the matrix Q. Similar to the analysis of the matrix H, there exist a matrix J such that

$$\begin{aligned} J &= \begin{bmatrix} \alpha\Delta & \frac{1}{2}\alpha\Delta \\ \frac{1}{2}\alpha\Delta & \alpha\Delta - \frac{r_1}{r_2^2} I_n \end{bmatrix} \\ &= \begin{bmatrix} T & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & T \end{bmatrix} Q \begin{bmatrix} T^{-1} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & T^{-1} \end{bmatrix} \end{aligned} \tag{2.10}$$

where $\mathbf{0}_{n\times n}$ is the $n \times n$ zero matrix. J is symmetric and has the same eigenvalues as Q.

Let ε be an eigenvalue of J. Because Δ is a diagonal matrix, it follows from (2.10) that ε satisfies

$$(\varepsilon - \alpha\lambda_i)(\varepsilon - \alpha\lambda_i + \frac{r_1}{r_2^2}) - \frac{1}{4}\alpha^2\lambda_i^2 = 0,$$

which can be simplified as

$$\varepsilon^2 - (2\alpha\lambda_i - \frac{r_1}{r_2^2})\varepsilon + \alpha\lambda_i(\alpha\lambda_i - \frac{r_1}{r_2^2}) - \frac{1}{4}\alpha^2\lambda_i^2 = 0. \tag{2.11}$$

Therefore, $\lambda_{\min}(Q) > 2$ if and only if the following two inequations (2.12) and (2.13) are satisfied:

$$\frac{(2\alpha\lambda_i - \frac{r_1}{r_2^2})}{2} > 2, \tag{2.12}$$

$$4 - 2(2\alpha\lambda_i - \frac{r_1}{r_2^2}) + \alpha\lambda_i(\alpha\lambda_i - \frac{r_1}{r_2^2}) - \frac{1}{4}\alpha^2\lambda_i^2 > 0. \tag{2.13}$$

From (2.12), we can get that

$$\alpha > \frac{r_1 + 4r_2^2}{2r_2^2\lambda_i}. \tag{2.14}$$

From (2.13), we can get that

$$\alpha > \frac{(8r_2^2 + 2r_1) + 2\sqrt{(r_1 + r_2^2)^2 + 3r_2^4}}{3r_2^2\lambda_i} \tag{2.15}$$

or

$$\alpha < \frac{(8r_2^2 + 2r_1) - 2\sqrt{(r_1 + r_2^2)^2 + 3r_2^4}}{3r_2^2\lambda_i}. \tag{2.16}$$

Therefore, we can obtain from (2.14) (2.15) and (2.16) that Q is positive definite and $\lambda_{\min}(Q) > 2$ if

$$\alpha > \frac{(8r_2^2 + 2r_1) + 2\sqrt{(r_1 + r_2^2)^2 + 3r_2^4}}{3r_2^2\lambda_{\min}(M)}. \tag{2.17}$$

Combining the results (2.9) and (2.17), the proof is completed.

Theorem 2.9 *(Wen et al. (2012 b)) Suppose that the undirected graph $\mathcal{G}$ is connected and at least one $a_{i0} > 0$. If $\alpha > \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\lambda_{\min}(M)}$, then the leader-following consensus in system (2.6) is achieved.*

Proof: Let $\tilde{\xi}_i(t) = \xi_i(t) - \xi_0(t), \tilde{v}_i(t) = v_i(t) - v_0(t), i = \{1, ..., n\}$. Form (2.2) and (2.6),

$$\begin{aligned} \dot{\tilde{\xi}}_i(t) =& \tilde{v}_i(t), \\ \dot{\tilde{v}}_i(t) =& f(t, \xi_i(t), v_i(t)) - f(t, \xi_0(t), v_0(t)) \\ &- \alpha \sum_{j=0}^{n} a_{ij}[r_1(\tilde{\xi}_i(t) - \tilde{\xi}_j(t) + r_2(\tilde{v}_i(t) - \tilde{v}_j(t))], \end{aligned} \tag{2.18}$$

where $\tilde{\xi}_i(t) - \tilde{\xi}_j(t) = (\xi_i(t) - \xi_0(t)) - (\xi_j(t) - \xi_0(t)) = \xi_i(t) - \xi_j(t)$ has been used.

$$\text{Let } \tilde{\xi}(t) = [\tilde{\xi}_1^T(t), ..., \tilde{\xi}_n^T(t)]^T, \tilde{v}(t) = [\tilde{v}_1^T(t), ..., \tilde{v}_n^T(t)]^T,$$

and

$$\begin{aligned} F(t, \tilde{\xi}(t), \tilde{v}(t)) = [&(f(t, \xi_1(t), v_1(t)) - f(t, \xi_0(t), v_0(t)))^T, ..., \\ &(f(t, \xi_n(t), v_n(t)) - f(t, \xi_0(t), v_0(t)))^T]^T. \end{aligned}$$

Rewrite (2.18) in the matrix form as

$$\begin{aligned} &\dot{\tilde{\xi}}(t) = \tilde{v}(t), \\ &\dot{\tilde{v}} = F(t, \tilde{\xi}(t), \tilde{v}(t)) - \alpha(M \otimes I_m)(r_1\tilde{\xi}(t) + r_2\tilde{v}(t)) \end{aligned} \tag{2.19}$$

Consider the Lyapunov function candidate

$$V(t) = \left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{array} \right] (H \otimes I_m) \left[\begin{array}{c} r_1\tilde{\xi} \\ r_2\tilde{v} \end{array} \right]. \tag{2.20}$$

Tacking the time derivative of $V(t)$ along the trajectory of (2.19) gives

$$\begin{aligned}
\dot{V} =&\tilde{v}^T \frac{1}{2}\alpha r_1 r_2 (M\otimes I_m)\tilde{\xi} + \tilde{\xi}^T \frac{1}{2}\alpha r_1 r_2 (M\otimes I_m)\tilde{v}\\
&+ \tilde{v}^T r_1 (I_n\otimes I_m)\tilde{v}\\
&+ \dot{\tilde{v}}^T [\frac{1}{2} r_1 (I_n\otimes I_m)\tilde{\xi} + \frac{1}{2} r_2 (I_n\otimes I_m)\tilde{v}]\\
&+ [\tilde{\xi}^T \frac{1}{2} r_1 (I_n\otimes I_m) + \tilde{v}^T \frac{1}{2} r_2 (I_n\otimes I_m)]\dot{\tilde{v}}\\
=&-\tilde{\xi}^T \left[\alpha r_1^2 (M\otimes I_m)\right]\tilde{\xi} - \tilde{\xi}^T \left[\frac{1}{2}\alpha r_1 r_2 (M\otimes I_m)\right]\tilde{v}\\
&-\tilde{v}^T \left[\frac{1}{2}\alpha r_1 r_2 (M\otimes I_m)\right]\tilde{\xi}\\
&-\tilde{v}^T \left[\alpha r_2^2 (M\otimes I_m) - r_1 (I_n\otimes I_m)\right]\tilde{v}\\
&+(r_1\tilde{\xi}^T + r_2\tilde{v}^T)(I_n\otimes I_m)F(t,\tilde{\xi},\tilde{v})\\
=&-\left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{array}\right](Q\otimes I_m)\left[\begin{array}{c} r_1\tilde{\xi}\\ r_2\tilde{v}\end{array}\right]\\
&+(r_1\tilde{\xi}^T + r_2\tilde{v}^T)(I_n\otimes I_m)F(t,\tilde{\xi},\tilde{v})\\
\leq&-\lambda_{\min}(Q)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2) + (\left\|r_1\tilde{\xi}\right\|_2 + \|r_2\tilde{v}\|_2)^2\\
\leq&-\lambda_{\min}(Q)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2) + 2(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2)\\
=&-(\lambda_{\min}(Q)-2)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2).
\end{aligned} \tag{2.21}$$

Since α satisfies that

$$\alpha > \frac{(8r_2^2 + 2r_1) + 2\sqrt{(r_1+r_2^2)^2 + 3r_2^4}}{3r_2^2\lambda_{\min}(M)},$$

it follows from Lemma 2.8 that $\lambda_{\min}(Q) > 2$. Furthermore, $\dot{V}$ is negative definite. Therefore, it follows that $\tilde{\xi}_i(t) \to \mathbf{0}_n$ and $\tilde{v}_i(t) \to \mathbf{0}_n$ as $t\to\infty$, where $\mathbf{0}_n$ is $n\times 1$ zero vector. Equivalently, it follows that $\xi_i(t)\to\xi_0(t)$ and $v_i(t)\to v_0(t)$ as $t\to\infty$.

Next, we prove that the consensus can be achieved globally exponentially. Note that

$$\begin{aligned}
V =& \left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{array}\right](H\otimes I_m)\left[\begin{array}{c} r_1\tilde{\xi}\\ r_2\tilde{v}\end{array}\right]\\
\leq& \lambda_{\max}(H)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2).
\end{aligned}$$

It then follow from (2.21) that

$$\begin{aligned}\dot{V}(t) \leq & -(\lambda_{\min}(Q)-2)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2) \\ \leq & -\frac{(\lambda_{\min}(Q)-2)}{\lambda_{\max}(H)}V(t).\end{aligned}$$

Therefore, $\dot{V}(t) \leq V(0)e^{-\frac{(\lambda_{\min}(Q)-2)}{\lambda_{\max}(H)}t}$. The proof is completed.

2.4 Consensus under fixed directed topology

In this section, leader-following consensus for second-order multi-agent systems with nonlinear inherent dynamics under fixed directed topology is investigated. Here, we still consider the control input (2.5).

Since the graph $\mathcal{G}$ is a fixed directed graph, generally speaking, the matrix M may be asymmetric. Let the graph $\tilde{\mathcal{G}}$ represent the communication topology of all followers and the virtual leader. Assume that the virtual leader has no information about followers, and has independent motion. It implies that if the fixed directed graph $\tilde{\mathcal{G}}$ contains a directed spanning tree, then the node corresponding to the virtual leader 0 is the root node.

Lemma 2.10 *(Wen et al. (2012 b)) Suppose the fixed directed graph $\tilde{\mathcal{G}}$ contains a directed spanning tree, then there exists a symmetric positive definite matrix S such that the matrix $SM + M^TS$ is also symmetric positive definite.*

Proof: Consider the following Laplacian matrix of $\tilde{G}$,

$$\tilde{L} = \begin{bmatrix} M & -\mathbf{a}_0 \\ \mathbf{0}_n^T & 0 \end{bmatrix}, \tag{2.22}$$

where $\mathbf{a}_0 = [a_{10}, ..., a_{n0}]^T$.

Since the fixed directed graph $\tilde{\mathcal{G}}$ has a directed spanning tree, by Lemma 2.4, $\tilde{L}$ has a simple eigenvalue zero with an associated eigenvector $\mathbf{1}$, and all other eigenvalues have positive real parts. Note from (2.22) that, each element in the row $n+1$ is zero, which implies that all eigenvalues of the sub-matrix M are not zero. Moreover, note that each row sum of the matrix $\tilde{L}$ is zero. Using the elementary transformation to (2.22), we can find a matrix $P = \begin{bmatrix} I_n & \mathbf{1}_n \\ \mathbf{0}_n^T & 1 \end{bmatrix}$, such that

$$P\tilde{L}P^{-1} = \begin{bmatrix} M & \mathbf{0}_n \\ \mathbf{0}_n^T & 0 \end{bmatrix}. \tag{2.23}$$

It implies that all eigenvalues of M are the eigenvalues of $\tilde{L}$. Hence all eigenvalues of M have positive real parts. Using Theorem 1.2 in Zoran Gajic (1995), there exists a symmetric positive definite matrix S such that $SM + M^TS$ is also symmetric positive definite. The proof is completed.

In this chapter, in order to simplify the process of calculating α, we choose $M^TS+SM = I_n$.

Lemma 2.11 *(Wen et al. (2012 b)) Suppose that the fixed directed graph $\tilde{\mathcal{G}}$ contains a directed spanning tree. Let* $\Theta = \begin{bmatrix} \frac{\alpha r_2}{r_1} I_n & \frac{1}{r_2} S \\ \frac{1}{r_2} S & \frac{1}{r_2} S \end{bmatrix}$ *and* $\Xi = \begin{bmatrix} \alpha I_n & \boldsymbol{0}_{n\times n} \\ \boldsymbol{0}_{n\times n} & \alpha I_n - \frac{2r_1}{r_2^2} S \end{bmatrix}$. *If* $\alpha > (\frac{2r_1}{r_2^2}+4)\lambda_{\max}(S)$*, then the matrices Θ and Ξ are symmetric positive definite, and* $\lambda_{\min}(\Xi) > 4\lambda_{\max}(S)$.

Proof: the proof is similar to that of Lemma 2.8 and is therefore omitted here.

Theorem 2.12 *(Wen et al. (2012 b)) Suppose that the fixed directed graph $\tilde{\mathcal{G}}$ contains a directed spanning tree. Using (2.5) for (2.1), if* $\alpha > (\frac{2r_1}{r_2^2} + 4)\lambda_{\max}(S)$*, then the leader-following consensus in system (2.6) can be achieved globally exponentially.*

Proof: Using the same operation as in the proof of Theorem 2.9, we can still obtain the equation (2.19).

Consider the following Lyapunov function candidate:

$$V(t) = \begin{bmatrix} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{bmatrix} (\Theta \otimes I_m) \begin{bmatrix} r_1\tilde{\xi} \\ r_2\tilde{v} \end{bmatrix}. \tag{2.24}$$

Tracking the time derivative of $V(t)$ along the trajectory of (2.19) gives

$$\begin{aligned}
\dot{V} =&\tilde{v}^T\alpha r_1 r_2(I_n\otimes I_m)\tilde{\xi}+\tilde{\xi}^T\alpha r_1 r_2(I_n\otimes I_m)\tilde{v}\\
&+2\tilde{v}^T r_1(S\otimes I_m)\tilde{v}+\dot{\tilde{v}}^T(S\otimes I_m)(r_1\tilde{\xi}+r_2\tilde{v})\\
&+(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(S\otimes I_m)\dot{\tilde{v}}\\
=&-\tilde{\xi}^T\left[\alpha r_1^2\left((M^TS+SM)\otimes I_m\right)\right]\tilde{\xi}\\
&+\tilde{\xi}^T\left[(\alpha r_1 r_2(M^TS+SM)\ -\alpha r_1 r_2 I_n)\otimes I_m\right]\tilde{v}\\
&+\tilde{v}^T\left[(\alpha r_1 r_2(M^TS+SM)\ -\alpha r_1 r_2 I_n)\otimes I_m\right]\tilde{\xi}\\
&+\tilde{v}^T\left[(\alpha r_2^2(M^TS+SM)-2r_1S)\otimes I_m\right]\tilde{v}\\
&+F^T(t,\tilde{\xi},\tilde{v})(S\otimes I_m)(r_1\tilde{\xi}+r_2\tilde{v})\\
&+(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(S\otimes I_m)F(t,\tilde{\xi},\tilde{v}),
\end{aligned}$$

Note that we choose $M^TS+SM=I_n$, by the Definition 2.7, one can obtains

$$\begin{aligned}
\dot{V}=&-\left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T\end{array}\right](\Xi\otimes I_m)\left[\begin{array}{c} r_1\tilde{\xi}\\ r_2\tilde{v}\end{array}\right]\\
&+2(\tilde{\xi}^T r_1+\tilde{v}^T r_2)(S\otimes I_m)F(t,\tilde{\xi},\tilde{v})\\
\leq&-\lambda_{\min}(\Xi)(\left\|r_1\tilde{\xi}\right\|_2^2+\|r_2\tilde{v}\|_2^2)\\
&+2\lambda_{\max}(S)(\left\|r_1\tilde{\xi}\right\|_2+\|r_2\tilde{v}\|_2)^2\\
\leq&-\lambda_{\min}(\Xi)(\left\|r_1\tilde{\xi}\right\|_2^2+\|r_2\tilde{v}\|_2^2)\\
&+4\lambda_{\max}(S)(\left\|r_1\tilde{\xi}\right\|_2^2+\|r_2\tilde{v}\|_2^2)\\
=&-\left[\lambda_{\min}(\Xi)-4\lambda_{\max}(S)\right](\left\|r_1\tilde{\xi}\right\|_2^2+\|r_2\tilde{v}\|_2^2).
\end{aligned}\tag{2.25}$$

Since $\alpha>(\frac{2r_1}{r_2^2}+4)\lambda_{\max}(S)$, it follows from Lemma 2.11 that $\lambda_{\min}(\Xi)>4\lambda_{\max}(S)$. Furthermore, $\dot{V}$ is negative definite. Hence, $\tilde{\xi}_i(t)\to\mathbf{0}_n$ and $\tilde{v}_i(t)\to\mathbf{0}_n$ as $t\to\infty$, where $\mathbf{0}_n$ is $n\times 1$ zero vector. Equivalently, it follows that $\xi_i(t)\to\xi_0(t)$ and $v_i(t)\to v_0(t)$ as $t\to\infty$.

Next, we prove that the consensus can be achieved globally exponentially. Note that

$$\begin{aligned}
V=&\left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T\end{array}\right](\Theta\otimes I_m)\left[\begin{array}{c} r_1\tilde{\xi}\\ r_2\tilde{v}\end{array}\right]\\
\leq&\lambda_{\max}(\Theta)(\left\|r_1\tilde{\xi}\right\|_2^2+\|r_2\tilde{v}\|_2^2).
\end{aligned}$$

From (2.25), $\dot{V}(t)$ satisfies that

$$\begin{aligned}\dot{V}(t) \leq & -(\lambda_{\min}(\Xi) - 4\lambda_{\max}(S))(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2) \\ \leq & -\frac{(\lambda_{\min}(\Xi) - 4\lambda_{\max}(S))}{\lambda_{\max}(\Theta)}V(t)\end{aligned}$$

Therefore, $\dot{V}(t) \leq V(0)e^{-\frac{(\lambda_{\min}(\Xi)-4\lambda_{\max}(S))}{\lambda_{\max}(\Theta)}t}$. The proof is completed.

2.5 Consensus under switching undirected topology

In this section, we consider the leader-following consensus problem under switching undirected topology.

Let $\bar{N}_i \subseteq \{0, ..., n\}$ denote the neighbor set of follower $i \subseteq \{1, ..., n\}$ in the team consisting of n followers and the virtual leader. Assume that if $\|\xi_i(t) - \xi_j(t)\|_2 \leq R$ at time t, then $j \in \bar{N}_i(t)$, else $j \notin \bar{N}_i(t)$, where R denotes the communication radius of the agents.

Let us consider the following control input,

$$u_i(t) = -\alpha \sum_{j\in\bar{N}_i(t)}^{n} b_{ij}[r_1(\xi_i(t) - \xi_j(t)) + r_2(v_i(t) - v_j(t))], \tag{2.26}$$

where α is a nonnegative constant, and $b_{ij}, i = 1, ..., n, j = 0, ..., n$, is positive constant. Substituting (2.26) to (2.1) gives

$$\begin{aligned}\dot{\xi}_i(t) =& v_i(t), \\ \dot{v}_i(t) =& f(t, \xi_i(t), v_i(t)) - \alpha \sum_{j\in\bar{N}_i(t)}^{n} b_{ij}[r_1(\xi_i(t) - \xi_j(t)) \\ & + r_2(v_i(t) - v_j(t))].\end{aligned} \tag{2.27}$$

Let $\bar{M}(t) = (\bar{m}_{ij}(t)) \in R^{n\times n}$, and

$$\bar{m}_{ij}(t) = \begin{cases} -b_{ij}, & j \in \bar{N}_i(t), j \neq i, \\ 0, & j \notin \bar{N}_i(t), j \neq i, \\ \sum_{k\in\bar{N}_i(t)} b_{ik}, & j = i. \end{cases} \tag{2.28}$$

It is easy to verify that $\bar{M}(t)$ is a symmetric positive definite matrix at each time instant.

Lemma 2.13 *(Wen et al. (2012 b)) Suppose that the switching undirected graph $\mathcal{G}$ is connected and at least one a_{i0} is positive at each time instant. Let* $\tilde{H}(t) = \begin{bmatrix} \frac{1}{2}\alpha\frac{r_2}{r_1}\bar{M}(t) & \frac{1}{2r_2}I_n \\ \frac{1}{2r_2}I_n & \frac{1}{2r_2}I_n \end{bmatrix}$ *and* $\tilde{Q}(t) = \begin{bmatrix} \alpha\bar{M}(t) & \frac{1}{2}\alpha\bar{M}(t) \\ \frac{1}{2}\alpha\bar{M}(t) & \alpha\bar{M}(t) - \frac{r_1}{r_2^2}I_n \end{bmatrix}$. *If* $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$, *then the matrices $\tilde{H}(t)$ and $\tilde{Q}(t)$ are symmetric positive definite at each time instant, and $\chi_{\min} > 2$, where $\gamma_{\min} = \min_t\{\lambda_{\min}(\bar{M}(t))\}$, $\chi_{\min} = \min_t\{\lambda_{\min}(\tilde{Q}(t))\}$*
.

Proof: the proof is similar to that of Lemma 2.8 and is therefore omitted here.

Theorem 2.14 *Suppose that the switching undirected graph $\mathcal{G}$ is connected and at least one a_{i0} is positive at each time instant. The leader-following consensus for second-order multi-agent systems (2.27) is achieved if* $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$, *where $\gamma_{\min}$ is defined in Lemma 2.13.*

Proof: Let $\tilde{\xi}_i(t) = \xi_i(t) - \xi_0(t), \tilde{v}_i(t) = v_i(t) - v_0(t), i = \{1, ..., n\}$. Form (2.2) and (2.27), we have

$$\begin{aligned} \dot{\tilde{\xi}}_i(t) &= \tilde{v}_i(t), \\ \dot{\tilde{v}}_i(t) &= -\alpha\sum_{j\in\bar{N}_i(t)}^{n} b_{ij}[r_1(\tilde{\xi}_i(t) - \tilde{\xi}_j(t)) + r_2(\tilde{v}_i(t) - \tilde{v}_j(t))] \\ &\quad + f(t, \xi_i(t), v_i(t)) - f(t, \xi_0(t), v_0(t)). \end{aligned} \tag{2.29}$$

where $\tilde{\xi}_i(t) - \tilde{\xi}_j(t) = (\xi_i(t) - \xi_0(t)) - (\xi_j(t) - \xi_0(t)) = \xi_i(t) - \xi_j(t)$ has been used.

$$\text{Let } \tilde{\xi}(t) = [\tilde{\xi}_1^T(t), ..., \tilde{\xi}_n^T(t)]^T, \tilde{v}(t) = [\tilde{v}_1^T(t), ..., \tilde{v}_n^T(t)]^T,$$

and

$$\begin{aligned} F(t, \tilde{\xi}(t), \tilde{v}(t)) = [&(f(t, \xi_1(t), v_1(t)) - f(t, \xi_0(t), v_0(t)))^T, ..., \\ &(f(t, \xi_n(t), v_n(t)) - f(t, \xi_0(t), v_0(t)))^T]^T. \end{aligned}$$

Rewrite (2.27) in the matrix form as

$$\begin{aligned} \dot{\tilde{\xi}}(t) &= \tilde{v}(t), \\ \dot{\tilde{v}}(t) &= F(t, \tilde{\xi}(t), \tilde{v}(t)) - \alpha(\bar{M}(t) \otimes I_m)(r_1\tilde{\xi}(t) + r_2\tilde{v}(t)). \end{aligned} \tag{2.30}$$

Consider the lyapunov function candidate

$$\begin{aligned} V(t) =& \frac{1}{2}\alpha r_1 r_2 \sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij}(t) + \alpha r_1 r_2 \sum_{i=1}^{n} V_{i0}(t) \\ &+ r_1 \tilde{v}^T(t)\tilde{\xi}(t) + \frac{1}{2} r_2 \tilde{v}^T(t)\tilde{v}(t), \end{aligned} \tag{2.31}$$

where

$$V_{ij}(t) = \begin{cases} \frac{1}{2}b_{ij}(\xi_i(t) - \xi_j(t))^2, & \|\xi_i(t) - \xi_j(t)\|_2 \le R, \\ \frac{1}{2}b_{ij}R^2, & \|\xi_i(t) - \xi_j(t)\|_2 > R, \end{cases} \quad i, j = 1, 2, ..., n, \tag{2.32}$$

$$V_{i0}(t) = \begin{cases} \frac{1}{2}b_{i0}(\xi_i(t) - \xi_0(t))^2, & \|\xi_i(t) - \xi_0(t)\|_2 \le R, \\ \frac{1}{2}b_{i0}R^2, & \|\xi_i(t) - \xi_0(t)\|_2 > R, \end{cases} \quad i = 1, 2, ..., n. \tag{2.33}$$

In what follows, the notation $\xi_i(t)$ and $v_i(t)$ are also written as ξ_i and v_i respectively for simplicity.

Note from (2.31), (2.32) and (2.33) that V is regular but not smooth. We analyze the stability of the system (2.27) using differential inclusions (Paden & Sastry (1987);Clarke (1983)) and nonsmooth analysis (Shevitz & Paden (1994);Tanner *et al.* (2007)). Then system (2.30) can be written as

$$\begin{aligned} \dot{\tilde{\xi}}(t) &\in^{a.e.} K[\tilde{v}(t)] \\ \dot{\tilde{v}}(t) &\in^{a.e.} -K[\alpha(\bar{M}(t) \otimes I_m)(r_1\tilde{\xi}(t) + r_2\tilde{v}(t)) \\ &\qquad - F(t, \tilde{\xi}(t), \tilde{v}(t))], \end{aligned} \tag{2.34}$$

where $K[\cdot]$ is the differential inclusion and $a.e.$ stands for "almost everywhere" (Clarke (1983)).

After some manipulation, $V(t)$ can be written as

$$\begin{aligned} V(t) =&\frac{1}{4}\alpha r_1 r_2 \sum_{i=1}^{n} \sum_{j\neq 0, j\in \bar{N}_i(t)}^{n} b_{ij}R^2 + \frac{1}{2}\alpha r_1 r_2 \sum_{0\notin \bar{N}_i(t)}^{n} b_{i0}R^2 \\ &+ \left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{array}\right] (\tilde{H}(t)\otimes I_m) \left[\begin{array}{c} r_1\tilde{\xi} \\ r_2\tilde{v} \end{array}\right]. \end{aligned} \tag{2.35}$$

By Lemma 2.13, $\tilde{H}(t)$ is symmetric positive definite if $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$ is satisfied.

The generalized time derivative of $V(t)$ is

$$\begin{aligned} \overset{o}{V}(t) =&\tilde{v}^T\frac{1}{2}\alpha r_1 r_2 (\bar{M}\otimes I_m)\tilde{\xi} + \tilde{\xi}^T\frac{1}{2}\alpha r_1 r_2(\bar{M}\otimes I_m)\tilde{v} \\ &+ \tilde{v}^T r_1 (I_n\otimes I_m)\tilde{v} \\ &+ \dot{\tilde{v}}^T[\frac{1}{2}r_1(I_n\otimes I_m)\tilde{\xi} + \frac{1}{2}r_2(I_n\otimes I_m)\tilde{v}] \\ &+ [\tilde{\xi}^T\frac{1}{2}r_1(I_n\otimes I_m) + \tilde{v}^T\frac{1}{2}r_2(I_n\otimes I_m)]\dot{\tilde{v}}. \end{aligned} \tag{2.36}$$

Substituting the trajectory (2.34) to (2.36) and using the similar proof of Theorem 2.9, we can obtain that the generalized derivative of $V(t)$ satisfies

$$\overset{o}{V}(t) \leq -(\chi_{\min} - 2)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2), \tag{2.37}$$

where $\chi_{\min}$ is defined in Lemma 2.13. Because $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$, it follows from Lemma 2.13 that $\chi_{\min} > 2$. Furthermore, the generalized derivative of $V(t)$ is negative definite. Hence, it follows from Theorem 3.1 in Shevitz & Paden (1994) that $\xi_i(t) \to \xi_0(t)$ and $v_i(t) \to v_0(t)$ as $t \to \infty$.

2.6 Simulation examples

In this section, Simulation results are worked out to demonstrate the effectiveness of the proposed the theoretic results. Consider a second-order multi-agent system consisting of one virtual leader indexed by 0 and n followers indexed by 1 to n, respectively.

For simplicity, we suppose that $a_{ij} = 1$ if agent i can receive information from

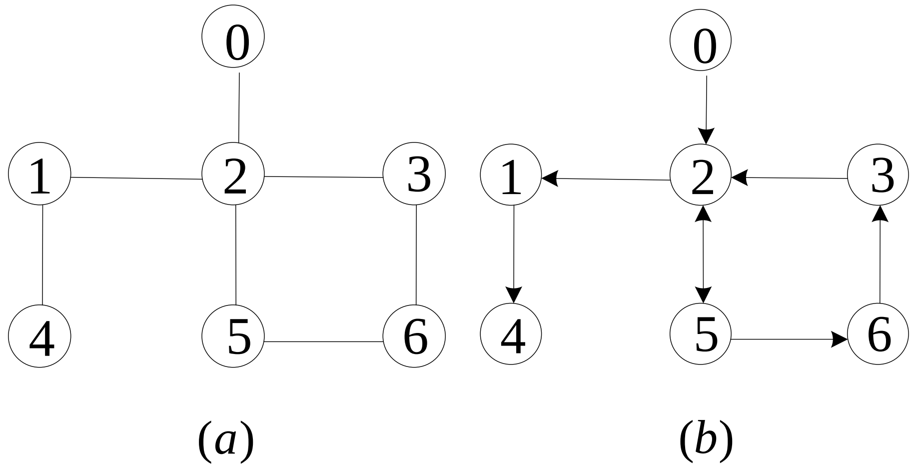

Figure 2.1: The fixed directed topology for a group of six followers and one virtual leader.

agent j, $a_{ij} = 0$ otherwise, $i \in \{1, ..., n\}$ and $j \in \{0, 1, ..., n\}$.

2.6.1 Case of fixed undirected topology

2.6.1.1 Dynamics of agents

The simulation is performed with six followers and one virtual leader. The communication topology is given in Fig. 2.1(a). The nonlinear inherent dynamics of each agent is described by chaotic Chua's circuit (see Fig. 2.2) as follows (Lu & Xi (2003);Matsumoto (1984)),

$$f(t, \xi_i, v_i) = \begin{bmatrix} p(v_{iy} - v_{ix} - h(v_{ix})) \\ v_{ix} - v_{iy} + v_{iz} \\ -qv_{iy} \end{bmatrix} \in R^3, \tag{2.38}$$

where $h(v_{ix}) = bv_{ix} + 0.5(a - b)(|v_{ix} + 1| - |v_{ix} - 1|)$, $p = 10$, $q = 18$, $a = -4/3$ and $b = -3/4$. It is easy to verify that $f(t, \xi_i, v_i)$ satisfies Definition 2.7. Here, the Lipschitz constants are chosen as $l_1 = 0$ and $l_2 = 4.3871$. Hence, $r_1 = 1$ and $r_2 = 5$

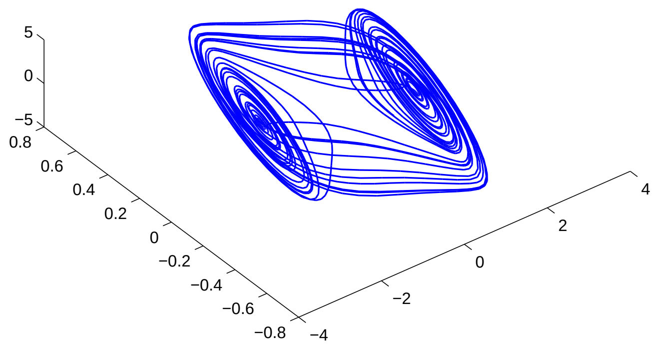

Figure 2.2: Chaotic behavior of Chua circuit.

2.6.1.2 Determination of α

The matrix M can be derived from the topology given in Fig.2.1(a).

$$M = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

It is easy to obtain that $\lambda_{\min}(M) = 0.1284$. By Theorem 2.9, when $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\lambda_{\min}(M)} = 31.4657$, the leader following consensus can be achieved. Here we choose $\alpha = 32$.

2.6.1.3 Simulation results

Figs.2.3 and 2.4 show the position states and velocity states of the virtual leader and followers. The initial position and velocity states of followers are randomly chosen from the cubes $[0, 10] \times [-6, 6] \times [-10, 0]$ and $[-4, 4] \times [-4, 4] \times [-4, 4]$ respectively. The initial position and velocity states of the virtual leader are $\xi_0(0) = [1, 2, 3]^T$ and $v_0(0) = [0.6, -0.2, 0.3]^T$. Simulation results verify the theoretical analysis very well.

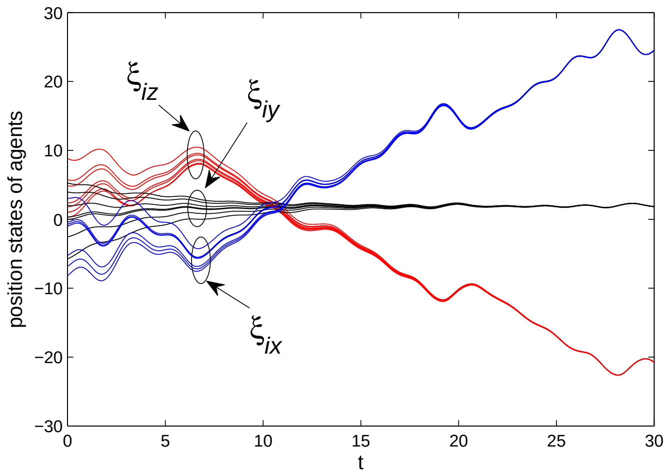

Figure 2.3: Position states of the followers and virtual leader under a fixed undirected topology of Fig.2.1(a).

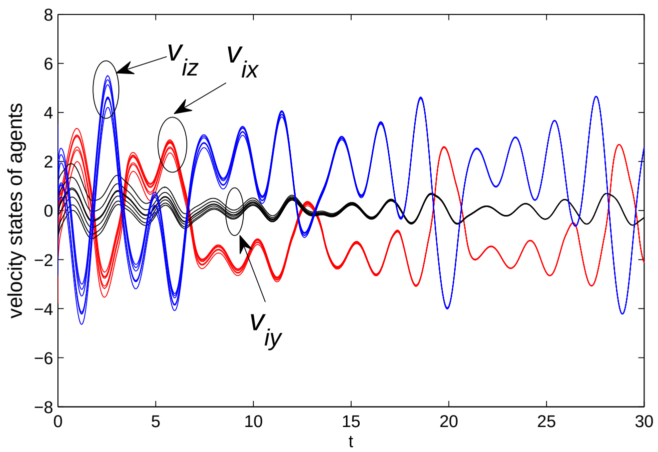

Figure 2.4: Velocity states of the followers and virtual leader under a fixed undirected topology of Fig.2.1(a).

2.6.2 Case of fixed directed topology

2.6.2.1 Dynamics of agents

The simulation is performed with six followers and one virtual leader. The communication topology is given in Fig. 2.1(b). The nonlinear inherent dynamics of each agent is described as follows,

$$f(t,\xi_i,v_i)=\begin{bmatrix}\sin(\xi_{ix})+\cos(v_{ix})\\ 3\cos(\xi_{iy})+3\sin(v_{iy})\end{bmatrix}\in R^2. \tag{2.39}$$

It is easy to verify that $f(t,\xi_i,v_i)$ satisfies Definition 2.7. Here, the Lipschitz constants are chosen as $r_1=4$ and $r_2=4$.

2.6.2.2 Determination of α

Based on the Fig.2.1, the M matrix is obtained

$$\mathrm{M}=\begin{bmatrix}1&-1&0&0&0&0\\0&3&-1&0&-1&0\\0&0&1&0&0&-1\\-1&0&0&1&0&0\\0&-1&0&0&1&0\\0&0&0&0&-1&1\end{bmatrix}$$

Let $SM+M^TS=I_n$. Then we obtain the S matrix

$$S=\begin{bmatrix}0.7500&0.2538&0.1454&0.2500&0.1910&0.0820\\0.2538&0.6121&0.4766&0.0741&1.0827&0.5007\\0.1454&0.4766&0.9766&0.0370&1.1490&0.7387\\0.2500&0.0741&0.0370&0.5000&0.0463&0.0185\\0.1910&1.0827&1.1490&0.0463&3.0268&1.4442\\0.0820&0.5007&0.7387&0.0185&1.4442&1.2387\end{bmatrix}$$

Using Theorem 2.12, $\alpha>(\frac{2r_1}{r_2^2}+4)\lambda_{\max}(S)=21.5928$. Here we choose $\alpha=22$.

2.6.2.3 Simulation results

Figs.2.5 and 2.6 show the position states and velocity states of the virtual leader and followers. The initial position and velocity states of followers are randomly chosen from the square box $[10,30]\times[-5,5]$ and $[0,3]\times[0,5]$ respectively. The

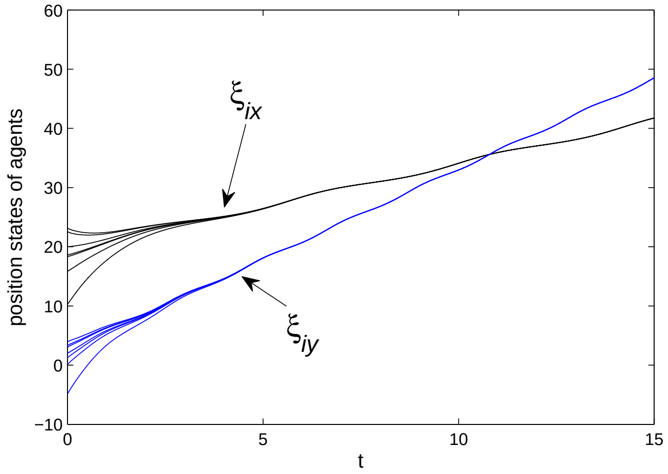

Figure 2.5: Position states of the followers and virtual leader under a fixed directed topology of Fig.2.1(b).

initial position and velocity states of the virtual leader are $\xi_0(0) = [20, 2]^T$ and $v_0(0) = [0.6, 4]^T$. Simulation results verify the theoretical analysis very well.

2.6.3 Case of switching undirected topology

2.6.3.1 Dynamics of agents

In this section, we consider a multi-agent system consisting of one virtual leader indexed by 0 and four followers indexed by 1 to 4. the communication radius of agents is $R = 5$. the nonlinear inherent dynamics of each followers and virtual leader is given by

$$f(t, \xi_i, v_i) = \begin{bmatrix} \sin(\xi_{ix})\cos(t) + \cos(v_{ix})\sin(t) \\ \cos(\xi_{ix})\sin(t) + \sin(v_{ix})\cos(t) \end{bmatrix} \in R^2. \tag{2.40}$$

It is easy to verify that $f(t, \xi_i, v_i)$ satisfies Definition 2.7. Here, the Lipschitz constants are chosen as $l_1 = 1$ and $l_2 = 1$. Hence, $r_1 = 2$ and $r_2 = 2$

2.6.3.2 Determination of α

Note that the matrix $\bar{M}(t)$ is time-varying in the switching topology case, and it is not easy to calculate the values of $\gamma_{\min}$. However, because the number of agents

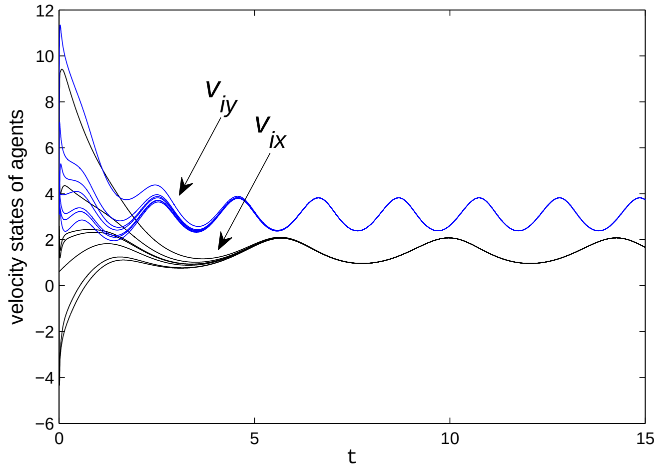

Figure 2.6: Velocity states of the followers and virtual leader under a fixed directed topology of Fig.2.1(b).

is given, we can get all potential communication topologies among agents, and then we can calculate the smallest eigenvalues of the matrix M in all potential topologies cases, which can be used to take the place of $\gamma_{\min}$. Therefore, it is easy to obtain $\alpha > 33.5008$. Let's take $\alpha = 34$.

2.6.3.3 Simulation results

Figs.2.7 and 2.8 show the position states and velocity states of the virtual leader and followers. The initial position and velocity states of followers are randomly chosen from the square box $[3, 11] \times [-4, 4]$ and $[-2, 2] \times [-2, 2]$ respectively. The initial position and velocity states of the virtual leader are $\xi_0(0) = [6, 0]^T$ and $v_0(0) = [1, 1]^T$. Simulation results verify the theoretical analysis very well.

2.7 Conclusion

In this chapter, we studied distributed leader-following consensus problem for second-order multi-agent systems with nonlinear inherent dynamics. The consensus problem is considered under three communication topologies: fixed undirected topology, fixed directed topology and switching undirected topology. According to the three topologies and Lipschitz-type conditions that are satisfied

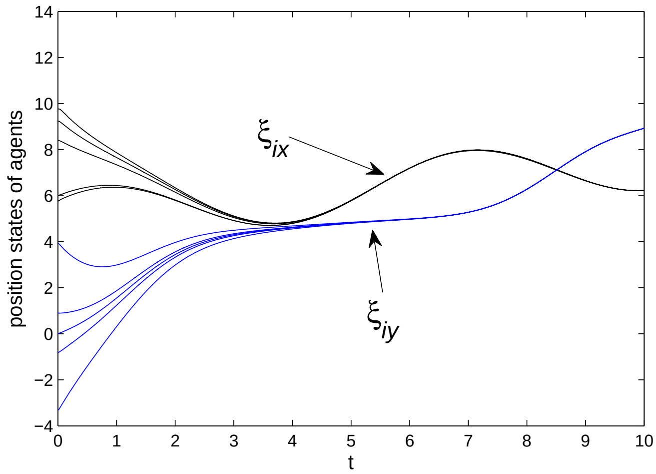

Figure 2.7: Position states of the followers and virtual leader.

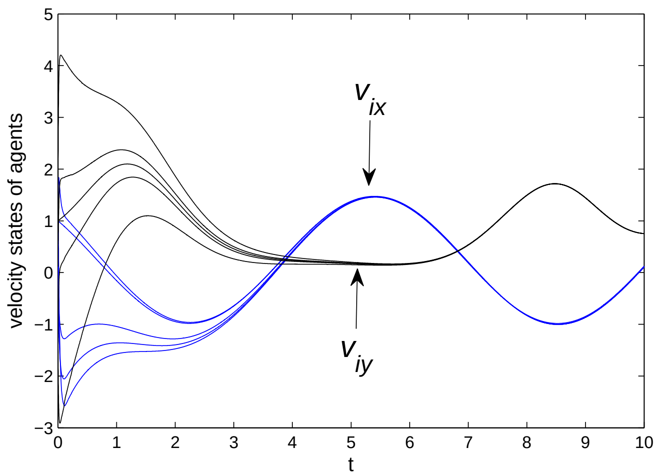

Figure 2.8: Velocity states of the followers and virtual leader.

by the nonlinear inherent dynamics of multi-agent systems, we proposed leader-following consensus algorithms respectively. In the fixed or switching undirected network topology case, it is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus can be achieved globally exponentially with the proposed protocol. In the fixed directed network topology case, it is shown that if the directed graph associated with the virtual leader and followers contains a directed spanning tree, the consensus can also be achieved globally exponentially with the proposed protocol. Rigorous proofs are given by using graph theory, matrix theory, and Lyapunov theory. Finally, some simulation examples are provided for illustration.

Chapter 3

Distributed Consensus Tracking for Multi-Agent Systems With Nonlinear Inherent Dynamics

Contents

3.1 Consensus Tracking for First-Order Multi-Agent Systems . . . 58

3.1.1 Problem description . . . 58

3.1.2 Consensus tracking under fixed undirected topology . . . 59

3.1.3 Consensus tracking under switching undirected topology 62

3.2 Consensus Tracking for Second-Order Multi-Agent Systems . . . 66

3.2.1 Problem description . . . 66

3.2.2 Consensus tracking under fixed undirected topology . . . 67

3.2.3 Consensus tracking under switching undirected topology 70

3.3 Numerical results . . . 73

3.3.1 Case of consensus tracking for first-order multi-agent systems . . . 73

3.3.2 Case of consensus tracking for second-order multi-agent systems . . . 79

3.4 Conclusion . . . 84

In Chapter 2, we have studied leader-following consensus for multi-agent systems with nonlinear inherent dynamics. It is easy to see from Chapter 1 that the trajectory of virtual leader is only determined by its nonlinear inherent dynamics. In fact, it is necessary to choose the trajectory for the virtual leader in some real-world applications, for example, obstacles avoidance, such that all followers can track the leader's trajectory. In this chapter, we investigate the consensus tracking problem for multi-agent systems with nonlinear dynamics. The consensus reference is taken as a virtual leader, whose dynamics is consisted of two terms: nonlinear inherent dynamics and another function that responsible to controlling the trajectory of the virtual leader. The dynamics of each follower is also consists of two terms: nonlinear inherent dynamics and a simple communication protocol relying only on the position of its neighbors. In this chapter, the consensus tracking problem is considered under fixed undirected network topology and switching undirected network topology. It is shown that if the undirected graph associated with the virtual leader and followers is connected, all followers can track the virtual leader in finite time with the proposed protocol under only the position measurements of the followers and the virtual leader. Rigorous proofs are given by using graph theory, matrix theory, and Lyapunov theory. Some sample simulations are presented to illustrate the theoretical analysis.

3.1 Consensus Tracking for First-Order Multi-Agent Systems

3.1.1 Problem description

In this chapter, we consider a multi-agent system that is made up of one virtual leader (labeled as 0) and n agents (labeled as agent 1 to n and called followers hereafter). Let the graph G represent the interaction topology of all followers.

The dynamics of each follower i $(i = 1, ..., n)$ is given by

$$\dot{\xi}_i(t) = f(t, \xi_i(t)) + u_i(t) \tag{3.1}$$

where $\xi_i(t) \in R^m$ is the position vector, $f(t, \xi_i(t)) \in R^m$ is its inherent nonlinear dynamics, and $u_i(t)$ is the control input. When $f(t, \xi_i(t)) \equiv 0$, the multi-agent system has single-integrator dynamics.

The dynamics of the virtual leader 0 is given by

$$\dot{\xi}_0(t) = f(t, \xi_0(t)) + g(t, \xi_0(t)), \tag{3.2}$$

where $\xi_0(t) \in R^m$ is the position vector, $f(t, \xi_0(t)) \in R^m$ describe the inherent nonlinear dynamics of virtual leader, and $g(t, \xi_0(t)) \in R^m$ is responsible for controlling trajectory of virtual leader. We assume that $|g(t, \xi_0(t))| \leq C_0$, where C_0 is a positive constant.

The consensus tracking problem of the multi-agent system is to design a control inputs $u_i(t), i = \{1, ..., n\}$ such that equation (3.3) is verified

$$\lim_{t \to \infty} \|\xi_i(t) - \xi_0(t)\|_2 = 0 \tag{3.3}$$

for any i and for any arbitrary initial position states.

We suppose that the virtual leader share the same nonlinear inherent dynamics with all followers, and these nonlinear inherent dynamics satisfy a Lipchitz-type condition given by Definition 3.1 as follows, which is satisfied in many well-known systems.

Definition 3.1 $\forall \xi, \zeta \in R^m; \forall t \geq 0$, *there exists a nonnegative constant* l *such that*

$$\|f(t, \xi) - f(t, \zeta)\|_2 \leq l \, \|\xi - \zeta\|_2. \tag{3.4}$$

The consensus tracking problem is considered under fixed and switching undirected network topologies.

3.1.2 Consensus tracking under fixed undirected topology

In this section, we consider the fixed undirected topology case. To satisfy the equation (3.3), we consider the following control input (3.5) which has been used in Cao & Ren (2012) to solve consensus problem for single-integrator dynamics case,

$$u_i(t) = -\alpha \sum_{j=0}^{n} a_{ij}(\xi_i(t) - \xi_j(t)) - \beta \text{sgn}\Big(\sum_{j=0}^{n} a_{ij}(\xi_i(t) - \xi_j(t))\Big) \tag{3.5}$$

where α is a nonnegative constant, β is a positive constant, sgn$(\cdot)$ is the signum function, and $a_{ij}, i, j = 1, ...n$, is the (i, j)th entry of the adjacency matrix $\mathcal{A}$ associated to G. Note that $a_{i0} > 0 (i = 1, ..., n)$ if the virtual leader's position is available to follower i, and $a_{i0} = 0$ otherwise.

Using (3.5), (3.1) can be rewritten as

$$\dot{\xi}_i(t) = f(t, \xi_i(t)) - \alpha \sum_{j=0}^{n} a_{ij}(\xi_i(t) - \xi_j(t)) - \beta \text{sgn}(\sum_{j=0}^{n} a_{ij}(\xi_i(t) - \xi_j(t))) \quad (3.6)$$

Let $M = L + diag(a_{10}, ..., a_{n0})$, where L is the Laplacian matrix of $\mathcal{G}$.

Theorem 3.2 *Suppose that the fixed undirected graph $\mathcal{G}$ is connected and at least one $a_{i0} > 0$. If $\alpha > \frac{l\lambda_{\max}(M)}{\lambda^2_{\min}(M)}$ and $\beta > C_0$, then the system (3.6) satisfies $\|\xi_i(t) - \xi_0(t)\|_2 = 0$ in finite time.*

Proof: Let $\tilde{\xi}_i(t) = \xi_i(t) - \xi_0(t), i = \{1, ..., n\}$. Form (3.2) and (3.6),

$$\begin{aligned}\dot{\tilde{\xi}}_i(t) = & -\alpha \sum_{j=0}^{n} a_{ij}(\tilde{\xi}_i(t) - \tilde{\xi}_j(t)) - \beta \text{sgn}(\sum_{j=0}^{n} a_{ij}(\tilde{\xi}_i(t) - \tilde{\xi}_j(t))) \\ & + f(t, \xi_i(t)) - f(t, \xi_0(t)) - g(t, \xi_0(t)).\end{aligned} \quad (3.7)$$

Let

$$\tilde{\xi}(t) = [\tilde{\xi}_1^T(t), \tilde{\xi}_2^T(t), ..., \tilde{\xi}_n^T(t)]^T,$$

$$F(t, \tilde{\xi}(t)) = [(f(t, \xi_1(t)) - f(t, \xi_0(t)))^T, ..., (f(t, \xi_n(t)) - f(t, \xi_0(t)))^T]^T.$$

Rewrite (3.7) in the matrix form as

$$\dot{\tilde{\xi}}(t) = -\alpha(M \otimes I_m)\tilde{\xi}(t) - \beta \text{sgn}((M \otimes I_m)\tilde{\xi}(t)) + F(t, \tilde{\xi}(t)) - \mathbf{1}g(t, \xi_0(t)), \quad (3.8)$$

where $\otimes$ stands for Kronecker product.

Because the fixed undirected graph $\mathcal{G}$ is connected and at least one of a_{i0} is positive, M is symmetric positive definite.

Consider the following Lyapunov function candidate

$$V(t) = \frac{1}{2}\tilde{\xi}^T(t)(M \otimes I_m)\tilde{\xi}(t). \quad (3.9)$$

Tracking the time derivative of $V(t)$ along the trajectory (3.8) yields

$$\begin{aligned}\dot{V}(t) =&\frac{1}{2}\tilde{\xi}^T(t)(M\otimes I_m)\dot{\tilde{\xi}}(t)+\frac{1}{2}\dot{\tilde{\xi}}^T(t)(M\otimes I_m)\tilde{\xi}(t)\\ =&\frac{1}{2}\tilde{\xi}^T(t)(M\otimes I_m)[-\alpha(M\otimes I_m)\tilde{\xi}(t)-\beta\text{sgn}((M\otimes I_m)\tilde{\xi}(t))\\ &+F(t,\tilde{\xi}(t))-\mathbf{1}g(t,\xi_0(t))]\\ &+\frac{1}{2}[-\alpha\tilde{\xi}^T(t)(M\otimes I_m)-\beta\text{sgn}(\tilde{\xi}^T(t)(M\otimes I_m))\\ &+F^T(t,\tilde{\xi}(t))-\mathbf{1}^Tg(t,\xi_0(t))](M\otimes I_m)\tilde{\xi}(t)\\ \leq&-\alpha\tilde{\xi}^T(t)(M\otimes I_m)^2\tilde{\xi}(t)-\beta\left\|(M\otimes I_m)\tilde{\xi}(t)\right\|_1\\ &+l\lambda_{\max}(M)\left\|\tilde{\xi}(t)\right\|_2^2+C_0\left\|(M\otimes I_m)\tilde{\xi}(t)\right\|_1\\ =&-(\alpha-\frac{l\lambda_{\max}(M)}{\lambda_{\min}^2(M)})\lambda_{\min}^2(M)\left\|\tilde{\xi}(t)\right\|_2^2-(\beta-C_0)\left\|(M\otimes I_m)\tilde{\xi}(t)\right\|_1.\end{aligned}\tag{3.10}$$

Note that if $\alpha>\frac{l\lambda_{\max}(M)}{\lambda_{\min}^2(M)}$ and $\beta>C_0$, $\dot{V}(t)<0$.

Next, we prove that $V(t)$ will decrease to zero in finite time. From (3.10), when $\alpha>\frac{l\lambda_{\max}(M)}{\lambda_{\min}^2(M)}$ and $\beta>C_0$, $\dot{V}(t)$ will also satisfy that

$$\begin{aligned}\dot{V}(t)&\leq-(\beta-C_0)\left\|(M\otimes I_m)\tilde{\xi}(t)\right\|_1\\ &\leq-(\beta-C_0)\lambda_{\min}(M)\left\|\tilde{\xi}(t)\right\|_2\\ &\leq-(\beta-C_0)\frac{\sqrt{2}\lambda_{\min}(M)\sqrt{V(t)}}{\sqrt{\lambda_{\max}(M)}}.\end{aligned}\tag{3.11}$$

From (3.11),

$$\sqrt{V(t)}\leq\sqrt{V(0)}-\frac{\sqrt{2}}{2}(\beta-C_0)\frac{\lambda_{\min}(M)}{\sqrt{\lambda_{\max}(M)}}t.\tag{3.12}$$

Let $\sqrt{V(0)}-\frac{\sqrt{2}}{2}(\beta-C_0)\frac{\lambda_{\min}(M)}{\sqrt{\lambda_{\max}(M)}}t^*=0$, then

$$t^*=\frac{\sqrt{2\tilde{\xi}^T(0)(M\otimes I_m)\tilde{\xi}(0)}\sqrt{\lambda_{\max}(M)}}{(\beta-C_0)\lambda_{\min}(M)}.\tag{3.13}$$

Therefore, when $t>t^*$, we have $V(t)=0$. The proof is completed.

Remark 3.3 *α is chosen such that it satisfies the condition of Theorem 3.2. Indeed, the calculation of the expression $\frac{l\lambda_{\max}(M)}{\lambda_{\min}^2(M)}$ is used to select α. β is chosen* *according to the bound of time t^*, see (3.13). Indeed, first choosing the time t^*, β can be determined using the following expression*

$$\beta = \frac{\sqrt{2\tilde{\xi}^T(0)(M \otimes I_m)\tilde{\xi}(0)}\sqrt{\lambda_{\max}(M)}}{t^*\lambda_{\min}(M)} + C_0. \tag{3.14}$$

3.1.3 Consensus tracking under switching undirected topology

In this section, we consider the consensus tracking problem under switching undirected topology.

Let $\bar{N}_i \subseteq \{0, ..., n\}$ denote the neighbor set of follower $i \subseteq \{1, ..., n\}$ in the team consisting of n followers and the virtual leader. Assume that if $\|\xi_i(t) - \xi_j(t)\|_2 \leq R$ at time t, then $j \in \bar{N}_i(t)$, else $j \notin \bar{N}_i(t)$, where R denotes the communication radius of the agents.

Let us consider the following control input, which has been used in Cao & Ren (2012) to solve consensus problem for single-integrator dynamics case,

$$u_i(t) = -\alpha \sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i(t) - \xi_j(t)) - \beta\text{sgn}(\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i(t) - \xi_j(t))), \tag{3.15}$$

where α is a nonnegative constant, β is a positive constant, $sgn(\cdot)$ is the signum function, and $b_{ij}, i = 1, ..., n, j = 0, ..., n$, are positive constants. Substitute (3.1) in (3.15) as

$$\dot{\xi}_i(t) = f(t, \xi_i(t)) - \alpha \sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i(t) - \xi_j(t)) - \beta\text{sgn}(\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i(t) - \xi_j(t))). \tag{3.16}$$

Let $\bar{M}(t) = (\bar{m}_{ij}(t)) \in R^{n\times n}$, and

$$\bar{m}_{ij}(t) = \begin{cases} -b_{ij}, & j \in \bar{N}_i(t), j \neq i, \\ 0, & j \notin \bar{N}_i(t), j \neq i, \\ \sum_{k\in\bar{N}_i(t)} b_{ik}, & j = i. \end{cases} \tag{3.17}$$

It is easy to verify that $\bar{M}(t)$ is a symmetric positive definite matrix at each time instant.

Let $\tilde{\mathcal{G}}$ represent the interaction topology of all agents and the virtual leader.

Theorem 3.4 *Suppose that the switching undirected graph $\tilde{\mathcal{G}}$ is connected at each time instant. For system (3.16), if $\alpha > \frac{l\gamma_{\max}}{\gamma_{\min}^2}$ and $\beta > C_0$, then $\|\xi_i(t) - \xi_0(t)\|_2 = 0$ in finite time, where $\gamma_{\min} = \min_t\{\lambda_{\min}(\bar{M}(t))\}$, $\gamma_{\max} = \max_t\{\lambda_{\max}(\bar{M}(t))\}$.*

Proof: Consider the Lyapunov function candidate

$$V(t) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij}(t) + \sum_{i=1}^{n} V_{i0}(t), \tag{3.18}$$

where

$$V_{ij}(t) = \begin{cases} \frac{1}{2}b_{ij}(\xi_i(t) - \xi_j(t))^2, & \|\xi_i(t) - \xi_j(t)\|_2 \le R, \\ \frac{1}{2}b_{ij}R^2, & \|\xi_i(t) - \xi_j(t)\|_2 > R, \end{cases} \quad i, j = 1, 2, ..., n, \tag{3.19}$$

$$V_{i0}(t) = \begin{cases} \frac{1}{2}b_{i0}(\xi_i(t) - \xi_0(t))^2, & \|\xi_i(t) - \xi_0(t)\|_2 \le R, \\ \frac{1}{2}b_{i0}R^2, & \|\xi_i(t) - \xi_0(t)\|_2 > R, \end{cases} \quad i = 1, 2, ..., n. \tag{3.20}$$

In what follows, the notation $\xi_i(t)$ is written as ξ_i for simplicity.

From (3.18), (3.19) and (3.20), note that V is regular but not smooth. We analyze the stability of the system (3.16) using differential inclusions (Filippov & Arscott (1988);Paden & Sastry (1987)) and nonsmooth analysis (Clarke (1983);Tanner *et al.* (2007)). Then system (3.16) can be written as

$$\dot{\xi}_i \in^{a.e.} -K[\alpha \sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i - \xi_j) + \beta\text{sgn}(\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i - \xi_j)) - f(t, \xi_i)], \tag{3.21}$$

where $K[\cdot]$ is the differential inclusion and $a.e.$ stands for "almost everywhere" (Paden & Sastry (1987))

Tacking the generalized time derivative of $V(t)$ along the trajectory (3.21) yields

$$
\begin{aligned}
\overset{\circ}{V}(t) =& \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\frac{\partial V_{ij}}{\partial \xi_i}\dot{\xi}_i + \frac{\partial V_{ij}}{\partial \xi_j}\dot{\xi}_j] + \sum_{i=1}^{n}[\frac{\partial V_{i0}}{\partial \xi_i}\dot{\xi}_i + \frac{\partial V_{i0}}{\partial \xi_0}\dot{\xi}_0] \\
=& \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\bar{N}_i, j\neq 0} b_{ij}[(\xi_i-\xi_j)\dot{\xi}_i + (\xi_j-\xi_i)\dot{\xi}_j] + \sum_{0\in\bar{N}_i} b_{i0}[(\xi_i-\xi_0)\dot{\xi}_i - (\xi_i-\xi_0)\dot{\xi}_0] \\
=& \sum_{i=1}^{n}\sum_{j\in\bar{N}_i} b_{ij}(\xi_i-\xi_j)\dot{\xi}_i - \sum_{0\in\bar{N}_i} b_{i0}(\xi_i-\xi_0)\dot{\xi}_0 \\
=& \sum_{i=1}^{n}\sum_{j\in\bar{N}_i} b_{ij}(\xi_i-\xi_j)\{-\alpha\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i-\xi_j) \\
& - \beta\mathrm{sgn}(\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i-\xi_j)) + f(t,\xi_i) - f(t,\xi_0) + f(t,\xi_0)\} \\
& - \sum_{0\in\bar{N}_i} b_{i0}(\xi_i-\xi_0)(f(t,\xi_0)+g(t,\xi_0)) \\
=& -\alpha\sum_{i=1}^{n}\left[\sum_{j\in\bar{N}_i} b_{ij}(\xi_i-\xi_j)\right]^2 - \beta\sum_{i=1}^{n}\left|\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i-\xi_j)\right| \\
& + \sum_{i=1}^{n}\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i-\xi_j)(f(t,\xi_i)-f(t,\xi_0)) - \sum_{i=1}^{n}\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i-\xi_j)f(t,\xi_0) \\
& - \sum_{i=1}^{n}\sum_{j\in\bar{N}_i(t), j\neq 0} b_{ij}(\xi_i-\xi_j)g(t,\xi_0) - \sum_{0\in\bar{N}_i} b_{i0}(\xi_i-\xi_0)f(t,\xi_0) \\
& - \sum_{0\in\bar{N}_i} b_{i0}(\xi_i-\xi_0)g(t,\xi_0) \\
=& -\alpha\tilde{\xi}^T\left[(\bar{M}(t)\otimes I_m)\right]^2\tilde{\xi} - \beta\left\|(\bar{M}(t)\otimes I_m)\tilde{\xi}\right\|_1 \\
& + \tilde{\xi}^T(\bar{M}(t)\otimes I_m)F(t,\tilde{\xi}) - g(t,\xi_0)\sum_{i=1}^{n}\sum_{j\in\bar{N}_i(t)} b_{ij}(\xi_i-\xi_j) \\
=& -(\alpha - \frac{l\gamma_{\max}}{\gamma_{\min}^2})\gamma_{\min}^2(\bar{M}(t))\left\|\tilde{\xi}\right\|_2^2 - (\beta - C_0)\gamma_{\min}\left\|\tilde{\xi}\right\|_1,
\end{aligned}
\tag{3.22}
$$

where $\tilde{\xi} = [\tilde{\xi}_1^T, \tilde{\xi}_2^T, ..., \tilde{\xi}_n^T]^T$ with $\tilde{\xi}_i = \xi_i - \xi_0$, and

$$
F(t,\tilde{\xi}) = [(f(t,\xi_1) - f(t,\xi_0))^T, ..., (f(t,\xi_n) - f(t,\xi_0))^T]^T, i = \{1, ..., n\}.
$$

Moreover, we have used the fact

$$\sum_{i=1}^{n}\sum_{j\in\bar{N}_i,j\neq 0} b_{ij}(\xi_i-\xi_j)f(t,\xi_0)=0$$

and

$$\sum_{i=1}^{n}\sum_{j\in\bar{N}_i,j\neq 0} b_{ij}(\xi_i-\xi_j)g(t,\xi_0)=0$$

in the calculation of (3.22). From (3.22), if $\alpha > \frac{l\gamma_{\max}}{\gamma_{\min}^2}$ and $\beta > C_0$, then $\overset{\circ}{V}(t) < 0$.

Next, we prove $V(t)$ will decrease to zero in finite time. From (3.22), when $\alpha > \frac{l\gamma_{\max}}{\gamma_{\min}^2}$ and $\beta > C_0$, $\overset{\circ}{V}(t)$ will also satisfy that

$$\begin{aligned}\overset{\circ}{V}(t) \leq & -(\beta - C_0)\gamma_{\min}\left\|\tilde{\xi}\right\|_1 \\ \leq & -(\beta - C_0)\gamma_{\min}\left\|\tilde{\xi}\right\|_2 .\end{aligned} \tag{3.23}$$

From (3.18),(3.19),and (3.20), it is easy to obtain

$$\begin{aligned}V(t) =& \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}V_{ij}(t)+\sum_{i=1}^{n}V_{i0}(t) \\ \leq& \frac{1}{2}\gamma_{\max}\left\|\tilde{\xi}\right\|_2^2 .\end{aligned} \tag{3.24}$$

Then, rewrite (3.23) by (3.24) as

$$\overset{\circ}{V}(t) \leq -\frac{\sqrt{2}(\beta - C_0)\gamma_{\min}\sqrt{V(t)}}{\sqrt{\gamma_{\max}}}. \tag{3.25}$$

From (3.25)

$$\sqrt{V(t)} \leq \sqrt{V(0)} - \frac{\sqrt{2}}{2}(\beta - C_0)\frac{\gamma_{\min}}{\sqrt{\gamma_{\max}}}t. \tag{3.26}$$

Let $\sqrt{V(0)} - \frac{\sqrt{2}}{2}(\beta - C_0)\frac{\gamma_{\min}}{\sqrt{\gamma_{\max}}}t^* = 0$, then

$$t^* = \frac{\sqrt{2\gamma_{\max}V(0)}}{(\beta - C_0)\gamma_{\min}}. \tag{3.27}$$

Therefore, when $t > t^*$, we have $V(t) = 0$. The proof is completed.

Remark 3.5 *Based on the Theorem 3.4, α can be determined using the calculation of the expression $\frac{l\gamma_{\max}}{\gamma_{\min}^2}$. From equation (3.27), β can be chosen according to the bound of time t^*. Choosing t^*, the parameter β is determined using the following expression*

$$\beta = \frac{\sqrt{2\gamma_{\max} V(0)}}{t^* \gamma_{\min}} + C_0. \tag{3.28}$$

3.2 Consensus Tracking for Second-Order Multi-Agent Systems

3.2.1 Problem description

Suppose that the multi-agent system studied in this chapter consists of one virtual leader (labeled as 0) and n agents (labeled as agent 1 to n and called followers hereafter). Let the graph G represent the interaction topology of all followers.

The dynamics of each follower i $(i = 1, ..., n)$ is given by

$$\begin{aligned} \dot{\xi}_i &= v_i, \\ \dot{v}_i &= f(t, \xi_i, v_i) + u_i, \end{aligned} \tag{3.29}$$

where $\xi_i \in R^m$ is the position vector, $f(t, \xi_i) \in R^m$ is its inherent nonlinear dynamics, and $u_i(t)$ is the control input. When $f(t, \xi_i) \equiv 0$, the multi-agent system has double-integrator dynamics.

The dynamics of the virtual leader 0 is given by

$$\begin{aligned} \dot{\xi}_0 &= v_0, \\ \dot{v}_0 &= f(t, \xi_0, v_0) + g(t, \xi_0, v_0), \end{aligned} \tag{3.30}$$

where $\xi_0 \in R^m$ and $f(t, \xi_0) \in R^m$ are, respectively, the position states and a nonlinear vector-valued continuous function to describe the dynamics of virtual leader.

The consensus problem of the multi-agent system (3.29) is to design a control inputs $u_i, i = \{1, ..., n\}$ such that equation (3.31) is verified

$$\begin{aligned} \lim_{t \to \infty} \|\xi_i - \xi_0\|_2 &= 0, \\ \lim_{t \to \infty} \|v_i - v_0\|_2 &= 0, \end{aligned} \tag{3.31}$$

for any i and for any arbitrary initial position states.

We suppose that the virtual leader share the same nonlinear inherent dynamics with all followers, and these nonlinear inherent dynamics satisfy a Lipchitz-type condition given by Definition 2.7, which is satisfied in many well-known systems.

In this chapter, the consensus tracking problem is considered under fixed undirected communication topology and switching undirected communication topology.

3.2.2 Consensus tracking under fixed undirected topology

In this section, the consensus tracking problem is considered under the fixed undirected topology case. To satisfy the equation (3.31), the following control input (3.32) is proposed for each follower,

$$\begin{aligned} u_i(t) = & -\alpha \sum_{j=0}^{n} a_{ij}[r_1(\xi_i - \xi_j) + r_2(v_i - v_j)] \\ & -\beta \text{sgn}\{\sum_{j=0}^{n} a_{ij}[r_1(\xi_i - \xi_j) + r_2(v_i - v_j)]\}, \end{aligned} \tag{3.32}$$

where α is a nonnegative constant, β is a positive constant, $a_{ij}, i, j = 1, ...n$, is the (i, j)th entry of the adjacency matrix $\mathcal{A}$ associated to G, $\text{sgn}(\cdot)$ is the signum function, and r_1, r_2 is defined as in Definition 2.7. In addition, $a_{i0} > 0 (i = 1, ..., n)$ if the virtual leader's position is available to follower i, and $a_{i0} = 0$ otherwise.

Substituting (3.32) to (3.29) gives

$$\begin{aligned} \dot{\xi}_i = & v_i, \\ \dot{v}_i = & f(t, \xi_i, v_i) - \alpha \sum_{j=0}^{n} a_{ij}[r_1(\xi_i - \xi_j) + r_2(v_i - v_j)] \\ & -\beta \text{sgn}\{\sum_{j=0}^{n} a_{ij}[r_1(\xi_i - \xi_j) + r_2(v_i - v_j)]\}. \end{aligned} \tag{3.33}$$

Let $M = L + diag(a_{10}, ..., a_{n0})$, where L is the Laplacian matrix of $\mathcal{G}$.

Theorem 3.6 *Suppose that the undirected graph $\mathcal{G}$ is connected and at least one $a_{i0} > 0$. If $\alpha > \frac{(8r_2^2+2r_1+2\sqrt{\hat{\Omega}})\lambda_{\max}(M)}{3r_2^2\lambda_{\min}^2(M)}$, and $\beta > C_0$, then second-order consensus tracking in system (3.33) is achieved in finite time, where $\hat{\Omega} = 4r_2^4+2r_1r_2^2+r_1^2$.*

Proof: Let $\tilde{\xi}_i = \xi_i - \xi_0, \tilde{v}_i = v_i - v_0, i = \{1, ..., n\}$. Form (3.30) and (3.33),

$$\begin{aligned}\dot{\tilde{\xi}}_i &= \tilde{v}_i,\\ \dot{\tilde{v}}_i &= -\alpha \sum_{j=0}^{n} a_{ij}[r_1(\tilde{\xi}_i - \tilde{\xi}_j) + r_2(\tilde{v}_i - \tilde{v}_j)]\\ &\quad - \beta \mathrm{sgn}\{\sum_{j=0}^{n} a_{ij}[r_1(\tilde{\xi}_i - \tilde{\xi}_j) + r_2(\tilde{v}_i - \tilde{v}_j)]\}\\ &\quad + f(t, \xi_i, v_i) - f(t, \xi_0, v_0) + g(t, \xi_0, v_0),\end{aligned} \tag{3.34}$$

where $\tilde{\xi}_i - \tilde{\xi}_j = (\xi_i - \xi_0) - (\xi_j - \xi_0) = \xi_i - \xi_j$ has been used.

Let $\tilde{\xi} = [\tilde{\xi}_1^T, \tilde{\xi}_2^T, ..., \tilde{\xi}_n^T]^T, \tilde{v} = [\tilde{v}_1^T, \tilde{v}_2^T, ..., \tilde{v}_n^T]^T$, and $F(t, \tilde{\xi}, \tilde{v}) = [(f(t, \xi_1, v_1) - f(t, \xi_0, v_0))^T, ..., (f(t, \xi_n, v_n) - f(t, \xi_0, v_0))^T]^T$. Rewrite (3.34) in the matrix form as

$$\begin{aligned}\dot{\tilde{\xi}} &= \tilde{v},\\ \dot{\tilde{v}} &= F(t, \tilde{\xi}, \tilde{v}) - \alpha(M \otimes I_m)(r_1\tilde{\xi} + r_2\tilde{v})\\ &\quad - \beta \mathrm{sgn}((M \otimes I_m)(r_1\tilde{\xi} + r_2\tilde{v})) + \mathbf{1}g(t, \xi_0, v_0).\end{aligned} \tag{3.35}$$

Consider the Lyapunov function candidate

$$V(t) = \begin{bmatrix} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{bmatrix} (H \otimes I_m) \begin{bmatrix} r_1\tilde{\xi} \\ r_2\tilde{v} \end{bmatrix}. \tag{3.36}$$

The time derivative of $V(t)$ is

$$\begin{aligned}\dot{V} =& \tilde{v}^T \frac{1}{2}\alpha r_1 r_2 (M \otimes I_m)\tilde{\xi} + \tilde{\xi}^T \frac{1}{2}\alpha r_1 r_2 (M \otimes I_m)\tilde{v}\\ &+ \tilde{v}^T r_1 (I_n \otimes I_m)\tilde{v} + \dot{\tilde{v}}^T (M \otimes I_m) \left[\frac{1}{2} r_1 \tilde{\xi} + \frac{1}{2} r_2 \tilde{v}\right]\\ &+ (\frac{1}{2} r_1 \tilde{\xi}^T + \frac{1}{2} r_2 \tilde{v}^T)(M \otimes I_m)\dot{\tilde{v}}\end{aligned} \tag{3.37}$$

Substituting the trajectory of (3.35) to (3.37) gives

$$
\begin{aligned}
\dot{V} =& -\tilde{\xi}^T\left[\alpha r_1^2(M^2\otimes I_m)\right]\tilde{\xi}-\tilde{\xi}^T\left[\frac{1}{2}\alpha r_1 r_2(M^2\otimes I_m)\right]\tilde{v}\\
&-\tilde{v}^T\left[\frac{1}{2}\alpha r_1 r_2(M^2\otimes I_m)\right]\tilde{\xi}\\
&-\tilde{v}^T\left[\alpha r_2^2(M^2\otimes I_m)-r_1(M\otimes I_m)\right]\tilde{v}\\
&+\beta(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\mathrm{sgn}((M\otimes I_m)(r_1\tilde{\xi}+r_2\tilde{v}))\\
&+(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)F(t,\tilde{x},\tilde{v})\\
&+(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\mathbf{1}g(t,\xi_0,v_0)\\
=& -\left[\begin{array}{cc} r_1\tilde{x}^T & r_2\tilde{v}^T\end{array}\right](Q\otimes I_m)\left[\begin{array}{c} r_1\tilde{x}\\ r_2\tilde{v}\end{array}\right]\\
&-\beta\left\|(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\right\|_1\\
&+(\tilde{x}^T r_1+\tilde{v}^T r_2)(M\otimes I_m)F(t,\tilde{x},\tilde{v})\\
&+(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\mathbf{1}g(t,\xi_0,v_0)
\end{aligned}
\tag{3.38}
$$

Since

$$
\left\|F(t,\tilde{\xi},\tilde{v})\right\|_2 \le \left\|r_1\tilde{\xi}\right\|_1+\|r_2\tilde{v}\|_2,
$$

$$
|g(t,\xi_0,v_0)|<C_0,
$$

$$
\left\|(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\right\|_1 > \left\|(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\right\|_2,
$$

it follows that

$$
\begin{aligned}
\dot{V}\le& -\lambda_{\min}(Q)(\|r_1\tilde{x}\|_2^2+\|r_2\tilde{v}\|_2^2)\\
&+\lambda_{\max}(M)(\|r_1\tilde{x}\|_2+\|r_2\tilde{v}\|_2)^2\\
&-\beta\left\|(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\right\|_2\\
&+C_0\left\|(r_1\tilde{\xi}^T+r_2\tilde{v}^T)(M\otimes I_m)\right\|_2\\
\le& -\left[\lambda_{\min}(Q)-2\lambda_{\max}(M)\right](\|r_1\tilde{x}\|_2^2+\|r_2\tilde{v}\|_2^2)\\
&-(\beta-C_0)\left\|(r_1\tilde{\xi}^T+r_2\tilde{v}^T)\right\|_2
\end{aligned}
\tag{3.39}
$$

Since $\alpha > \frac{(8r_2^2+2r_1+2\sqrt{\hat{\Omega}})\lambda_{\max}(M)}{3r_2^2\lambda_{\min}^2(M)}$ and $\beta > C_0$, it follows from Lemma 2.8 that $\dot{V}$ is negative definite. Therefore, it follows that $\tilde{\xi}_i(t)\to\mathbf{0}_n$ and $\tilde{v}_i(t)\to\mathbf{0}_n$ as $t\to\infty$, where $\mathbf{0}_n$ is $n\times 1$ zero vector. Equivalently, it follows that $\xi_i(t)\to\xi_0(t)$ and $v_i(t)\to v_0(t)$ as $t\to\infty$.

Next, we prove that the consensus can be achieved at least globally exponentially. Note that

$$\begin{aligned} V =& \begin{bmatrix} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{bmatrix} (H \otimes I_m) \begin{bmatrix} r_1\tilde{\xi} \\ r_2\tilde{v} \end{bmatrix} \\ \leq& \lambda_{\max}(H)(\left\| r_1\tilde{\xi} \right\|_2^2 + \|r_2\tilde{v}\|_2^2). \end{aligned}$$

From (3.39), $\dot{V}(t)$ satisfies that

$$\begin{aligned} \dot{V}(t) \leq& -(\lambda_{\min}(Q) - 2\lambda_{\max}(M))(\left\| r_1\tilde{\xi} \right\|_2^2 + \|r_2\tilde{v}\|_2^2) \\ \leq& -\frac{(\lambda_{\min}(Q) - 2\lambda_{\max}(M))}{\lambda_{\max}(H)} V(t) \end{aligned}$$

Therefore, $\dot{V}(t) \leq V(0)e^{-\frac{(\lambda_{\min}(Q)-2\lambda_{\max}(M))}{\lambda_{\max}(H)}t}$. The proof is completed.

3.2.3 Consensus tracking under switching undirected topology

This section deals with the second-order consensus tracking problem under switching undirected topology.

Let $\bar{N}_i \subseteq \{0, ..., n\}$ denote the neighbor set of follower $i \subseteq \{1, ..., n\}$ in the team consisting of n followers and the virtual leader. Assume that if $\|\xi_i(t) - \xi_j(t)\|_2 \leq R$ at time t, then $j \in \bar{N}_i(t)$, else $j \notin \bar{N}_i(t)$, where R denotes the communication radius of the agents.

Let us consider the following control input,

$$\begin{aligned} u_i =& -\alpha \sum_{j\in\bar{N}_i(t)} b_{ij}[r_1(\xi_i - \xi_j) + r_2(v_i - v_j)] \\ &- \beta\mathrm{sgn}\{\sum_{j\in\bar{N}_i(t)} b_{ij}[r_1(\xi_i - \xi_j) + r_2(v_i - v_j)]\}, \end{aligned} \tag{3.40}$$

where $b_{ij}, i = 1, ..., n, j = 0, ..., n$, is positive constant, and α, β , r_1, r_2, $\mathrm{sgn}(\cdot)$ are defined as in (3.32).

Substituting (3.40) to (3.29) gives

$$\begin{aligned}\dot{\xi}_i &= v_i,\\ \dot{v}_i &= f(t,\xi_i,v_i) - \alpha \sum_{j\in\bar{N}_i(t)} b_{ij}[r_1(\xi_i-\xi_j)+r_2(v_i-v_j)]\\ &\quad - \beta \mathrm{sgn}\{\sum_{j\in\bar{N}_i(t)} b_{ij}[r_1(\xi_i-\xi_j)+r_2(v_i-v_j)]\}.\end{aligned} \tag{3.41}$$

Let $\bar{M}(t) = (\bar{m}_{ij}(t)) \in R^{n\times n}$, where

$$\bar{m}_{ij}(t) = \begin{cases} -b_{ij}, & j \in \bar{N}_i(t), j \neq i,\\ 0, & j \notin \bar{N}_i(t), j \neq i,\\ \sum_{k\in\bar{N}_i(t)} b_{ik}, & j = i.\end{cases} \tag{3.42}$$

Based on the condition of the theorem, $\bar{M}(t)$ is a symmetric positive definite matrix at each time instant.

Theorem 3.7 *Suppose that the switching undirected graph $\mathcal{G}$ is connected and at least one a_{i0} is positive at each time instant. The consensus tracking for the second-order multi-agent systems (3.41) is achieved if* $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$, *where $\gamma_{\min}$ is defined in Lemma 2.13.*

Proof: Let $\tilde{\xi}_i(t) = \xi_i(t) - \xi_0(t), \tilde{v}_i(t) = v_i(t) - v_0(t), i = \{1,...,n\}$. Form (3.30) and (3.41), we have

$$\begin{aligned}\dot{\tilde{\xi}}_i &= \tilde{v}_i,\\ \dot{\tilde{v}}_i &= -\alpha \sum_{j\in\bar{N}_i(t)} b_{ij}[r_1(\xi_i-\xi_j)+r_2(v_i-v_j)]\\ &\quad - \beta \mathrm{sgn}\{\sum_{j\in\bar{N}_i(t)} b_{ij}[r_1(\xi_i-\xi_j)+r_2(v_i-v_j)]\}\\ &\quad + f(t,\xi_i,v_i) - f(t,\xi_0,v_0) + g(t,\xi_0,v_0).\end{aligned} \tag{3.43}$$

where $\tilde{\xi}_i - \tilde{\xi}_j = (\xi_i - \xi_0) - (\xi_j - \xi_0) = \xi_i - \xi_j$ has been used.

Let $\tilde{\xi} = [\tilde{\xi}_1^T, \tilde{\xi}_2^T, ..., \tilde{\xi}_n^T]^T, \tilde{v} = [\tilde{v}_1^T, \tilde{v}_2^T, ..., \tilde{v}_n^T]^T$, and $F(t,\tilde{\xi},\tilde{v}) = [(f(t,\xi_1,v_1) - f(t,\xi_0,v_0))^T, ..., (f(t,\xi_n,v_n) - f(t,\xi_0,v_0))^T]^T$. Rewrite (3.43) in the matrix form as

$$\begin{aligned}\dot{\tilde{\xi}} &= \tilde{v},\\ \dot{\tilde{v}} &= F(t,\tilde{\xi},\tilde{v}) - \alpha(M\otimes I_m)(r_1\tilde{\xi}+r_2\tilde{v})\\ &\quad - \beta \mathrm{sgn}((M\otimes I_m)(r_1\tilde{\xi}+r_2\tilde{v})) + \mathbf{1}g(t,\xi_0,v_0).\end{aligned} \tag{3.44}$$

Consider the lyapunov function candidate

$$\begin{aligned} V(t) =& \frac{1}{2}\alpha r_1 r_2 \sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij}(t) + \alpha r_1 r_2 \sum_{i=1}^{n} V_{i0}(t) \\ &+ r_1 \tilde{v}^T(t)\tilde{\xi}(t) + \frac{1}{2} r_2 \tilde{v}^T(t)\tilde{v}(t), \end{aligned} \tag{3.45}$$

where

$$V_{ij}(t) = \begin{cases} \frac{1}{2} b_{ij}(\xi_i(t) - \xi_j(t))^2, & \|\xi_i(t) - \xi_j(t)\|_2 \leq R, \\ \frac{1}{2} b_{ij} R^2, & \|\xi_i(t) - \xi_j(t)\|_2 > R, \end{cases} \\ i, j = 1, 2, ..., n, \tag{3.46}$$

$$V_{i0}(t) = \begin{cases} \frac{1}{2} b_{i0}(\xi_i(t) - \xi_0(t))^2, & \|\xi_i(t) - \xi_0(t)\|_2 \leq R, \\ \frac{1}{2} b_{i0} R^2, & \|\xi_i(t) - \xi_0(t)\|_2 > R, \end{cases} \\ i = 1, 2, ..., n. \tag{3.47}$$

Note from (3.45), (3.46) and (3.47) that V is regular but not smooth. We analyze the stability of the system (3.41) using differential inclusions (Paden & Sastry (1987);Clarke (1983)) and nonsmooth analysis (Shevitz & Paden (1994);Tanner *et al.* (2007)). Then system (3.44) can be written as

$$\begin{aligned} \dot{\tilde{\xi}} &\in^{a.e.} K\left[\tilde{v}\right], \\ \dot{\tilde{v}} &\in^{a.e.} -K[\alpha(M \otimes I_m)(r_1\tilde{\xi} + r_2\tilde{v}) \\ &\qquad + \beta \mathrm{sgn}((M \otimes I_m)(r_1\tilde{\xi} + r_2\tilde{v})) \\ &\qquad - F(t, \tilde{\xi}, \tilde{v}) - 1g(t, \xi_0, v_0)], \end{aligned} \tag{3.48}$$

where $K[\cdot]$ is the differential inclusion and $a.e.$ stands for "almost everywhere" (Clarke (1983)).

After some manipulation, $V(t)$ can be written as

$$\begin{aligned} V(t) =& \frac{1}{4}\alpha r_1 r_2 \sum_{i=1}^{n} \sum_{j\neq 0, j\in \bar{N}_i(t)}^{n} b_{ij}R^2 + \frac{1}{2}\alpha r_1 r_2 \sum_{0\notin \bar{N}_i(t)}^{n} b_{i0}R^2 \\ &+ \left[\begin{array}{cc} r_1\tilde{\xi}^T & r_2\tilde{v}^T \end{array}\right] (\tilde{H}(t) \otimes I_m) \left[\begin{array}{c} r_1\tilde{\xi} \\ r_2\tilde{v} \end{array}\right]. \end{aligned} \tag{3.49}$$

By Lemma 2.13, $\tilde{H}(t)$ is symmetric positive definite if $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$ is satisfied.

The generalized time derivative of $V(t)$ is

$$\begin{aligned}\overset{o}{V}(t) =& \tilde{v}^T \frac{1}{2}\alpha r_1 r_2(\bar{M} \otimes I_m)\tilde{\xi} + \tilde{\xi}^T \frac{1}{2}\alpha r_1 r_2(\bar{M} \otimes I_m)\tilde{v} \\ &+ \tilde{v}^T r_1(I_n \otimes I_m)\tilde{v} \\ &+ \dot{\tilde{v}}^T[\frac{1}{2}r_1(I_n \otimes I_m)\tilde{\xi} + \frac{1}{2}r_2(I_n \otimes I_m)\tilde{v}] \\ &+ [\tilde{\xi}^T \frac{1}{2}r_1(I_n \otimes I_m) + \tilde{v}^T \frac{1}{2}r_2(I_n \otimes I_m)]\dot{\tilde{v}}.\end{aligned} \tag{3.50}$$

Substituting the trajectory (3.48) to (3.50) and using the similar proof of Theorem 3.6, we can obtain that the generalized derivative of $V(t)$ satisfies

$$\begin{aligned}\overset{o}{V}(t) =& -(\chi_{\min} - 2)(\left\|r_1\tilde{\xi}\right\|_2^2 + \|r_2\tilde{v}\|_2^2) \\ &-(\beta - C_0)\left\|(r_1\tilde{\xi}^T + r_2\tilde{v}^T)\right\|_2,\end{aligned} \tag{3.51}$$

where $\chi_{\min}$ is defined in Lemma 2.13. Because $\alpha \geq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\gamma_{\min}}$, it follows from Lemma 2.13 that $\chi_{\min} > 2$. Furthermore, under the condition of the theorem $\beta > C_0$, the generalized derivative of $V(t)$ is negative fefinite. Hence, it follows from Theorem 3.1 in Shevitz & Paden (1994) that $\xi_i(t) \to \xi_0(t)$ and $v_i(t) \to v_0(t)$ as $t \to \infty$.

3.3 Numerical results

In this section, some simulations are given to verify our theoretical results. Let us consider a multi-agent system consisting of one virtual leader indexed by 0 and n followers indexed by 1 to n, respectively.

For simplicity, we consider that $a_{ij} = 1$ if agent i can receive information from agent j, $a_{ij} = 0$ otherwise, $i \in \{1, ..., n\}$ and $j \in \{0, 1, ..., n\}$.

3.3.1 Case of consensus tracking for first-order multi-agent systems

3.3.1.1 Fixed undirected topology

The communication topology is given in Fig.2.1(a). Suppose that the nonlinear inherent dynamics of the follower i ($i = 1, ..., 6$) is described by the equa-

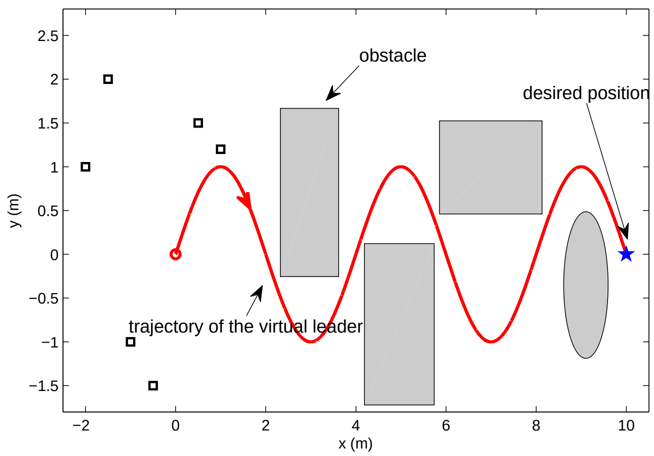

Figure 3.1: Example scenario of obstacle avoidance for all agents.

tion (3.52). Their initial x positions and y positions are given by $\xi_x(0) = [\ -1 \quad -0.5 \quad -2 \quad -1.5 \quad 1 \quad 0.5\ ]$ and $\xi_y(0) = [\ -1 \quad -1.5 \quad 1 \quad 2 \quad 1.2 \quad 1.5]$ respectively.

$$f(t, \xi_i(t)) = [\frac{1}{10}\sin(\xi_{ix}(t) + \frac{\pi}{3}), \frac{1}{10}\xi_{iy}(t)\cos(t)]^T \in R^2. \tag{3.52}$$

The nonlinear inherent dynamics of the virtual leader 0 is given by equation (3.53) and its initial and desired positions are given in Fig.3.1.

$$f(t, \xi_0(t)) = [\frac{1}{10}\sin(\xi_{0x}(t)+\frac{\pi}{3}), \frac{1}{10}\xi_{0y}(t)\cos(t)]^T \in R^2. \tag{3.53}$$

In addition, there are obstacles between the initial position and desired position of the virtual leader. It is necessary to choose the trajectory of the virtual leader, so that it can reach the desired position by avoiding obstacles (Fig.3.1). Here, $g(t, \xi_0(t))$ is chosen by equation (3.54). It follows that $C_0 = 1.5$.

$$g(t, \xi_0(t)) = [1, \frac{3}{2}\cos(\frac{1}{2}\pi t)]^T \in R^2. \tag{3.54}$$

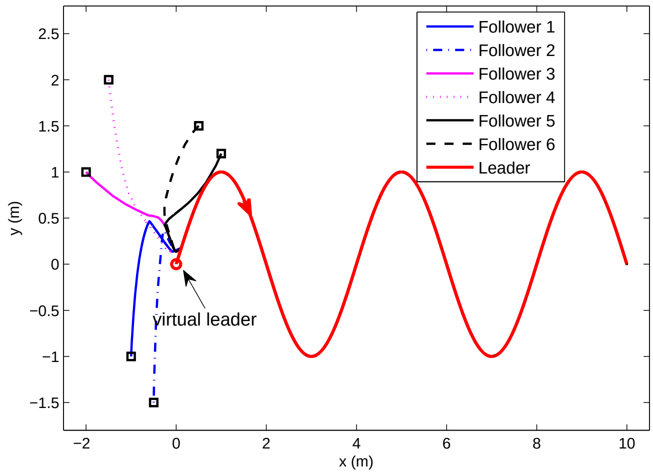

Figure 3.2: Trajectories of the virtual leader and the followers under (3.52) and (3.53). The circle denotes the initial position of the virtual leader, while the squares denote the initial positions of the followers.

The matrix M can be derived from the topology given in Fig.2.1(a).

$$M = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Using Theorem 3.2, it is easy to verify that when $\alpha > \frac{l\lambda_{\max}(M)}{\lambda^2_{\min}(M)} = 31.19329$, the consensus tracking can be achieved. Here we choose $\alpha = 31.2$.

Suppose that the consensus tracking has to be achieved in at most $t^* = 100$s, we have to choose β to be bigger than 2.7583. Let's take $\beta = 3$.

Fig.3.2 shows the trajectories of the virtual leader and the followers. Figs. 3.3 and 3.4 show the position tracking errors of x and y positions respectively. It is easy to see from Figs. 3.2, 3.3 and 3.4 that the consensus tracking can be achieved in about 0.25 seconds. This value is remarkably less than the desired bound 100 seconds.

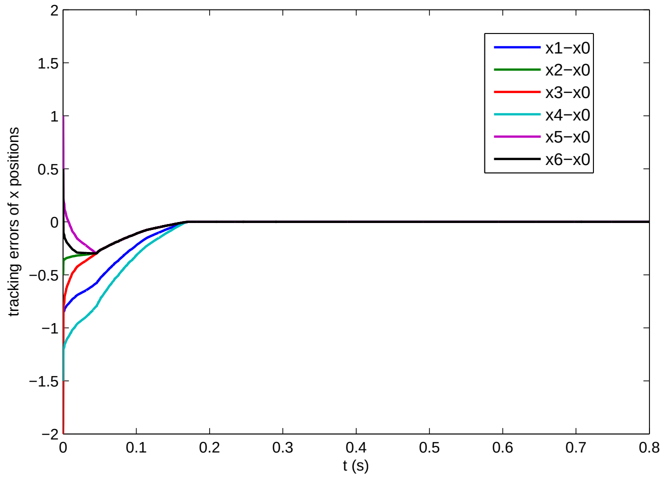

Figure 3.3: Position tracking errors of x positions under (3.52) and (3.53).

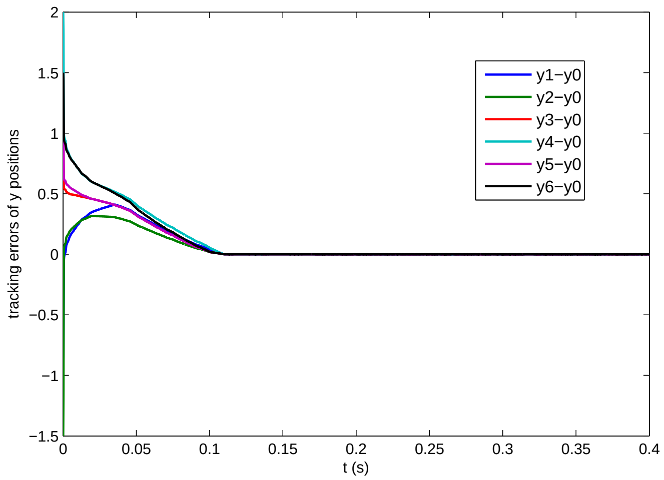

Figure 3.4: Position tracking errors of y positions under (3.52) and (3.53).

3.3.1.2 Switching undirected topology

Let us consider a multi-agent system consisting of one virtual leader indexed by 0 and four followers indexed by 1 to 4. The communication radius of agents is $R = 1.5$. Suppose that the nonlinear inherent dynamics of each follower is described by equation (3.55)

$$f(t, \xi_i(t)) = \begin{bmatrix} \frac{1}{10}\sin(\xi_{ix}(t) + \frac{\pi}{4})\sin(t) \\ \frac{1}{10}\cos(\xi_{iy}(t) + \frac{\pi}{4})\cos(t) \end{bmatrix} \in R^2. \tag{3.55}$$

Their initial x and y positions are given by $\xi_x(0) = \begin{bmatrix} -1.5 & -2 & -1.8 & -1.2 \end{bmatrix}$ and $\xi_y(0) = \begin{bmatrix} 0.5 & 1.2 & -0.5 & -0.2 \end{bmatrix}$ respectively (Fig.3.5).

The nonlinear inherent dynamics of the virtual leader is given by

$$f(t, \xi_0(t)) = \begin{bmatrix} \frac{1}{10}\sin(\xi_{0x}(t) + \frac{\pi}{4})\sin(t) \\ \frac{1}{10}\cos(\xi_{0y}(t) + \frac{\pi}{4})\cos(t) \end{bmatrix} \in R^2, \tag{3.56}$$

and its initial position is given in Fig.3.5.

Similar to the purpose of obstacles avoidance, the trajectory of the virtual leader is chosen to be $\xi_0(t) = \left[t, \frac{1}{3}\sin(2\pi t)\right]^T$. Therefore $g(t, \xi_0(t))$ is chosen by

$$g(t, \xi_0(t)) = \begin{bmatrix} -\frac{1}{10}\sin(\xi_{0x}(t) + \frac{\pi}{4})\sin(t) + 1 \\ -\frac{1}{10}\cos(\xi_{0y}(t) + \frac{\pi}{4})\cos(t) + \frac{2\pi}{3}\cos(2\pi t) \end{bmatrix} \in R^2. \tag{3.57}$$

From (3.57), let's take $C_0 = \frac{1}{10} + \frac{2\pi}{3} = 2.1944$.

Note that the matrix $\bar{M}(t)$ is time-varying in the switching topology case, and it is not easy to calculate the values of $\gamma_{\max}$ and $\gamma_{\min}$. However, because the number of agents is given, we can get all the potential communication topologies among agents, and then calculate the largest and the smallest eigenvalues of the matrix M in all potential topologies cases, which can be used to take the place of $\gamma_{\max}$ and $\gamma_{\min}$. Therefore, it is easy to obtain $\alpha > 24.2850$. Let's take $\alpha = 24.3$.

In order to achieve the consensus tracking in at most $t^* = 10$s, we have to choose β to be bigger than 8.3475. Let's take $\beta = 8.4$.

Fig.3.5 shows the trajectories of the virtual leader and the followers. Figs. 3.6 and 3.7 show the position tracking errors of x and y positions respectively. It is easy to see from Figs. 3.5, 3.6 and 3.7 that the consensus tracking can be achieved in about 0.05 seconds. This value is remarkably less than the desired bound 10 seconds.

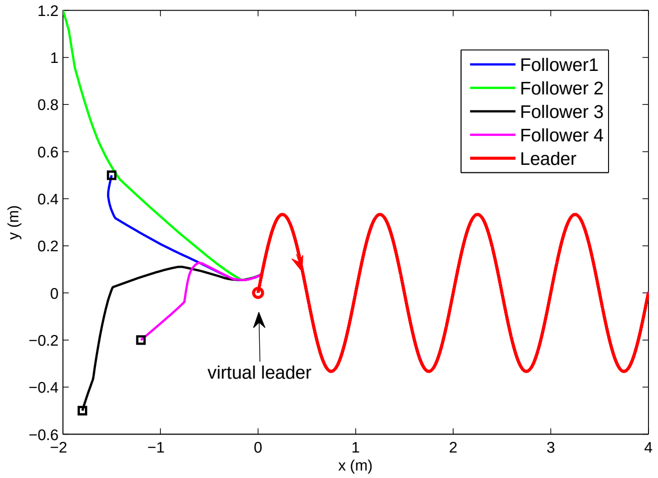

Figure 3.5: Trajectories of the virtual leader and the followers under (3.55) and (3.56). The circle denotes the initial position of the virtual leader, while the squares denote the initial positions of the followers.

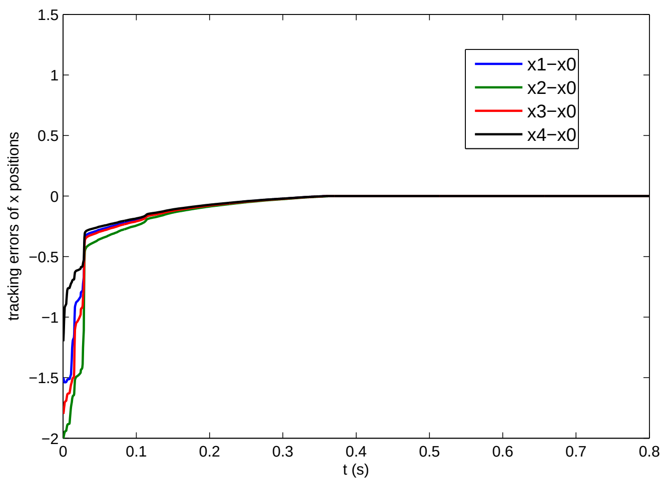

Figure 3.6: Position tracking errors of x positions under (3.55) and (3.56).

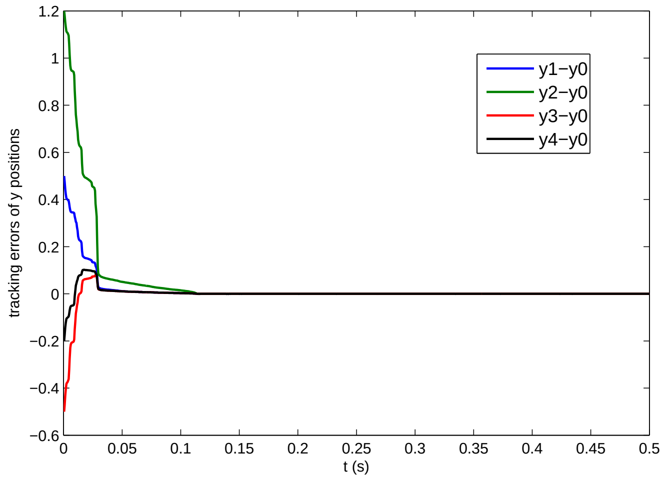

Figure 3.7: Position tracking errors of y positions under (3.55) and (3.56).

3.3.2 Case of consensus tracking for second-order multi-agent systems

3.3.2.1 Fixed undirected topology

Consider a second-order multi-agent system consisting of one virtual leader indexed by 0 and six followers indexed by 1 to 6, respectively. The communication topology is given in Fig. 2.1(a). The nonlinear inherent dynamics of each follower is given as follows,

$$f(t, \xi_i, v_i) = \begin{bmatrix} \sin(\xi_{ix})\cos(t) + \cos(v_{ix})\sin(t) \\ \cos(\xi_{iy})\sin(t) + \sin(v_{iy})\cos(t) \end{bmatrix} \in R^2. \tag{3.58}$$

It is easy to verify that $f(t, \xi_i, v_i)$ satisfies Definition 2.3. Here, the Lipschitz constants are chosen as $r_1 = 3$ and $r_2 = 3$. The trajectory of virtual leader is chosen as $\xi_0(t) = [t, \sin(t)]^T \in R^2$. It follows that the dynamics of virtual leader is given as

$$f(t, \xi_0, v_0) + g(t, \xi_0, v_0) = \begin{bmatrix} 0 \\ -\sin(t) \end{bmatrix} \in R^2. \tag{3.59}$$

Furthermore,

$$g(t, \xi_0, v_0) = \begin{bmatrix} -\sin(\xi_{ix})\cos(t) - \cos(v_{ix})\sin(t) \\ -\cos(\xi_{iy})\sin(t) - \sin(v_{iy})\cos(t) - \sin(t) \end{bmatrix} \tag{3.60}$$

From (3.63), let's take $C_0 = 3$.

In this simulation, we suppose that $a_{ij} = 1$ if agent i can receive information from agent j, $a_{ij} = 0$ otherwise, $i \in \{1, ..., n\}$ and $j \in \{0, 1, ..., n\}$. Therefore, the matrix M can be derived from the topology given in Fig.2.1(a).

$$M = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

It is easy to obtain that $\lambda_{\min}(M) = 0.1284$. By Theorem 3.6, when $\alpha \leq \frac{(8r_2^2+2r_1)+2\sqrt{(r_1+r_2^2)^2+3r_2^4}}{3r_2^2\lambda_{\min}(M)} = 31.4657$, and $\beta > C_0 = 3$, the consensus tracking can be achieved. Here we choose $\alpha = 32$, $\beta = 4$.

Fig.3.8 shows the position states of the virtual leader and followers. Figs.3.9 and 3.10 show the the position tracking errors of x or y positions respectively. Figs. 3.11 and 3.12 show the the position tracking errors of x or y positions respectively. Here, the initial position and velocity states of followers are randomly chosen from the cubes $[-3, 3] \times [-3, 3]$ and $[-2, 2] \times [-2, 2]$ respectively, and the initial position and velocity states of the virtual leader are $\xi_0(0) = [0, 0]^T$ and $v_0(0) = [1, 1]^T$. It can be seen all followers ultimately track the virtual leader. Simulation results verify the theoretical analysis very well.

3.3.2.2 Switching undirected topology

Consider a multi-agent system consisting of one virtual leader indexed by 0 and four followers indexed by 1 to 4. The communication radius of agents is $R = 5$. The nonlinear inherent dynamics of each followers and virtual leader is given by

$$f(t, \xi_i, v_i) = \begin{bmatrix} \sin(\xi_{ix}) + \cos(v_{ix}) \\ \cos(\xi_{ix}) + \sin(v_{ix}) \end{bmatrix} \in R^2. \tag{3.61}$$

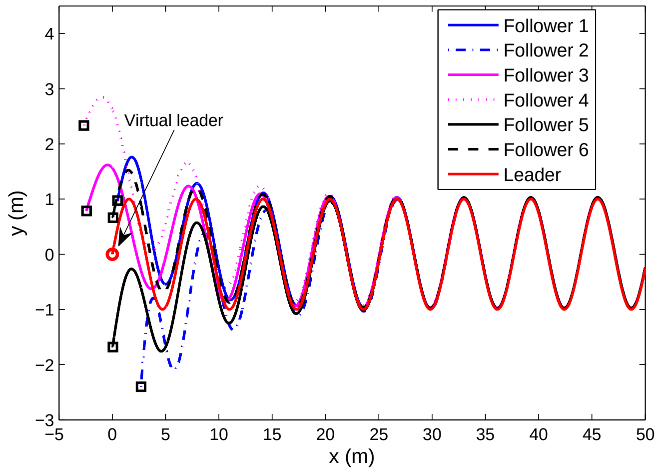

Figure 3.8: Position states of the followers and virtual leader under algorithm (3.32).

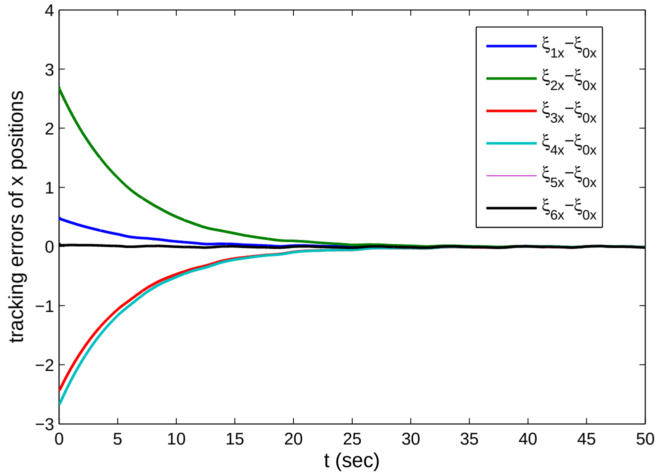

Figure 3.9: Position tracking errors in x positions.

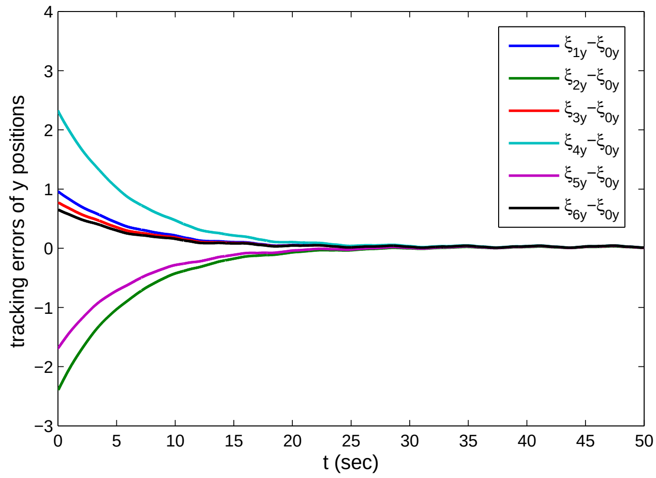

Figure 3.10: Position tracking errors in y positions.

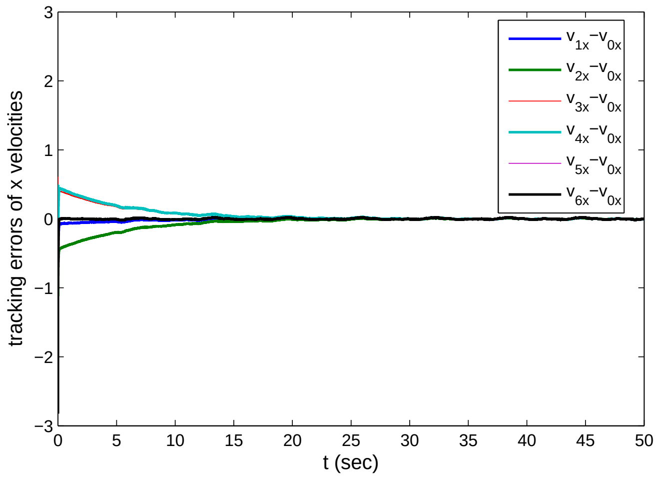

Figure 3.11: Velocity tracking errors in x velocities.

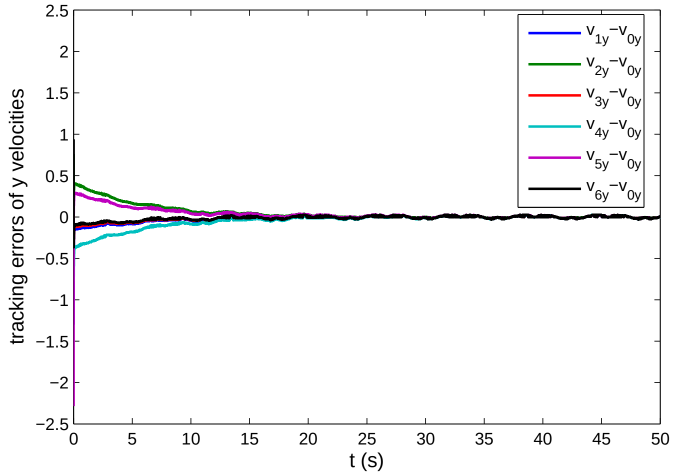

Figure 3.12: Velocity tracking errors in y velocities.

It is easy to verify that $f(t, \xi_i, v_i)$ satisfies Definition 2.3. Here, the Lipschitz constants are chosen as $r_1 = 3$ and $r_2 = 3$. The trajectory of virtual leader is chosen as $\xi_0(t) = [\cos(t), \sin(t)]^T \in R^2$. It follows that the dynamics of virtual leader is given as

$$f(t, \xi_0, v_0) + g(t, \xi_0, v_0) = \begin{bmatrix} -\cos(t) \\ -\sin(t) \end{bmatrix} \in R^2. \tag{3.62}$$

Furthermore,

$$g(t, \xi_0, v_0) = \begin{bmatrix} -\sin(\xi_{ix}) - \cos(v_{ix}) - \cos(t) \\ -\cos(\xi_{iy}) - \sin(v_{iy}) - \sin(t) \end{bmatrix} \tag{3.63}$$

From (3.63), let's take $C_0 = 3$.

Note that the matrix $\bar{M}(t)$ is time-varying in the switching topology case, and it is not easy to calculate the values of $\gamma_{\min}$. However, because the number of agents is given, we can get all potential communication topologies among agents, and then we can calculate the smallest eigenvalues of the matrix M in all potential topologies cases, which can be used to take the place of $\gamma_{\min}$. Therefore, it is easy to obtain $\alpha > 33.5008$. Let's take $\alpha = 34$.By Theorem 3.7, $\beta > C_0 = 3$. Here, we chose $\beta = 4$.

Fig. 3.13 shows the position states of the virtual leader and followers. Figs. 3.14, 3.15, 3.16, and 3.17 show the tracking errors of the x and y positions and

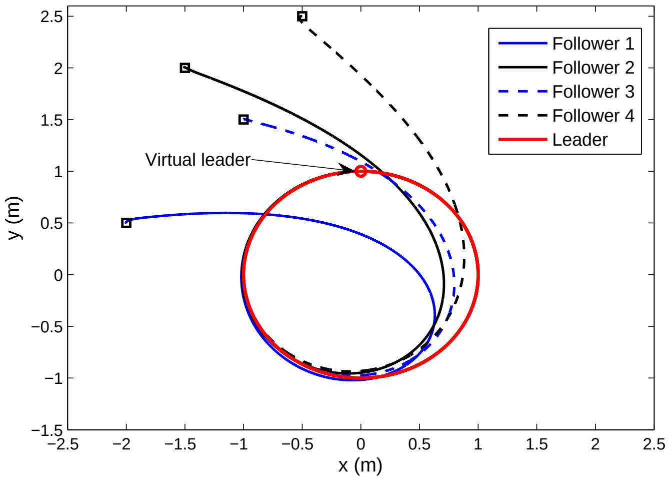

Figure 3.13: Trajectories of the followers and virtual leader under algorithm (3.40).

velocities respectively.The initial position states of followers are chosen as $\xi_1(0) = [-2, 0.5]^T$, $\xi_2(0) = [-1.5, 2]^T$, $\xi_3(0) = [-1, 1.5]^T$, $\xi_4(0) = [-0.5, 2.5]^T$, and the initial velocity states are randomly chosen from the square box $[-2, 2] \times [-2, 2]$. The initial position and velocity states of followers are chosen as $\xi_0(0) = [0, 1]^T$ and $v_0(0) = [1, 0]^T$. It can be seen from Figs. 3.14, 3.15, 3.16, and 3.17 that the tracking errors ultimately converge to zero. That is all followers ultimately track the virtual leader as also shown in Fig. 3.13. Simulation results verify the theoretical analysis very well.

3.4 Conclusion

In this chapter, we have studied the consensus tracking problems for first-order and second-order multi-agent systems with nonlinear inherent dynamics under the fixed undirected topology and the switching undirected topology. It is shown that followers can track the virtual leader even when only one follower has just access to the information of the virtual leader. Finally, several examples were provided to demonstrate the effectiveness of our theoretical results.

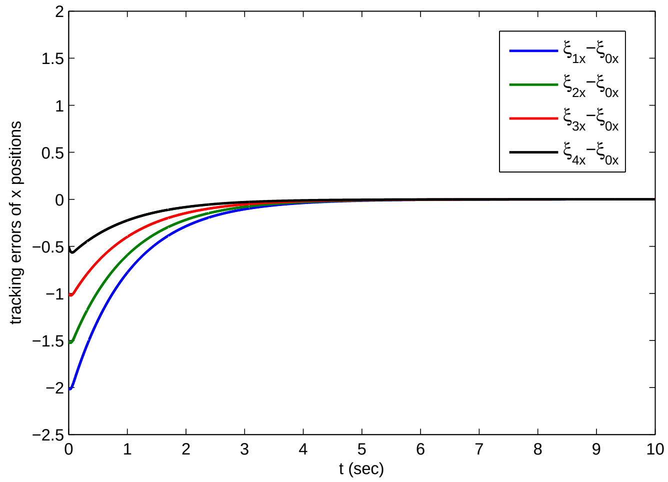

Figure 3.14: Position tracking errors in x positions.

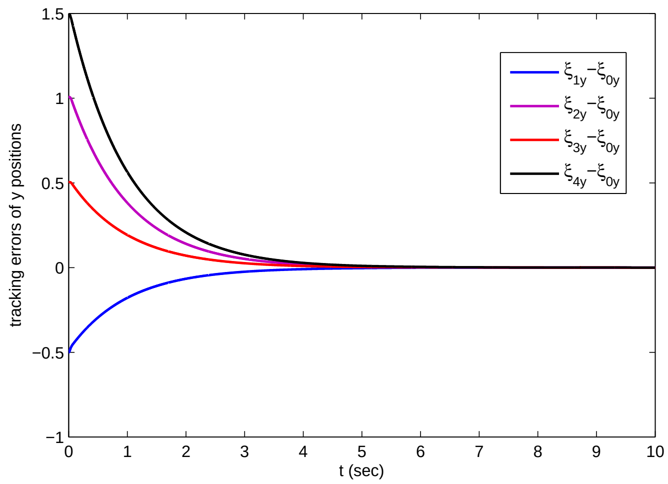

Figure 3.15: Position tracking errors in y positions.

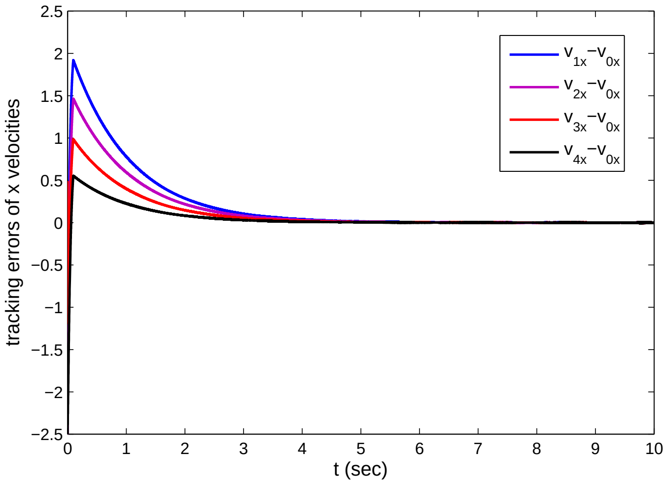

Figure 3.16: Velocity tracking errors in x velocities.

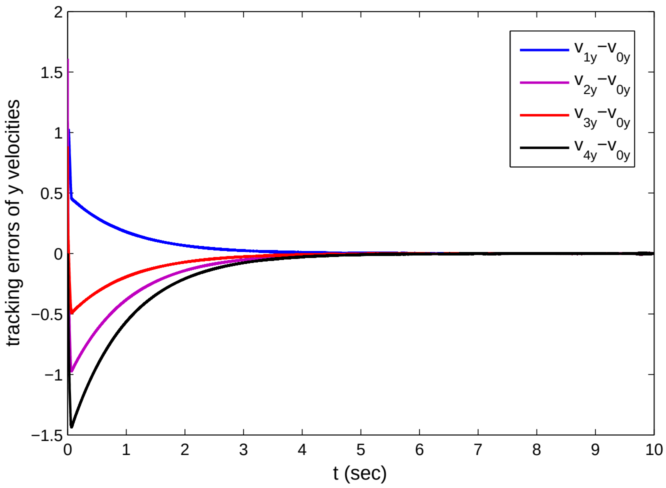

Figure 3.17: Velocity tracking errors in y velocities.

Chapter 4

Planning and Control of Multi-Agent Formation

Contents

4.1 Preliminaries and problem statement 88
4.1.1 Preliminaries 88
4.1.2 Problem statement 98
4.2 Path planning 99
4.2.1 Grid Map Generation 99
4.2.2 Path Generation 101
4.3 Control laws for the formation 102
4.3.1 Control law for the 0-DOF agent 103
4.3.2 Control law for the 1-DOF agent 103
4.3.3 Control law for the 2-DOF agent 104
4.3.4 Control laws for the 3-DOF agents 105
4.4 Stability and convergence 106
4.4.1 Stability and convergence for the 3-DOF agent 106
4.4.2 Stability and convergence for the 2-DOF agent 107
4.4.3 Stability and convergence for the 1-DOF agent 109
4.4.4 Stability and convergence for the 0-DOF agent 111
4.5 Simulations 112

4.5.1 Verification of the effectiveness of control laws 112
4.5.2 Verification of the effectiveness of the framework for planning and control of formation 112
4.6 Conclusion 112

In this chapter, we provide a practicable framework for path planning and motion control of 3-dimensional multi-agent formation. The formation path planning is performed by using the A* search algorithm coupled with an optimizing algorithm to generate a collision-free and deadlock-free feasible path. For the formation motion control, we propose a set of decentralized control laws for the cohesive motion of 3-dimensional multi-agent formation with a point-agent system model. Based on these control laws, the shape of the formation is maintained during continuous motion, which can efficiently avoid inter-agent collision. Rigorous proofs regarding the stability and convergence of proposed control laws are given by using Lyapunov theory. The effectiveness of the proposed framework is shown by simulation.

4.1 Preliminaries and problem statement

In this section, we first give a brief introduction to some important notions which will be used through this dissertation, and then state our problem.

4.1.1 Preliminaries

4.1.1.1 Formation graph theory

A formation F is represented by a graph $G = (V, E)$, with a vertex set $V = \{1, 2, ..., N\}$ and an edge set $E \subset V \times V$, where each vertex in V corresponds to an agent A_i in the formation F, and each edge $(i, j) \in E$ corresponds to an information link between a pair (A_i, A_j) of agents. Here, G is called the underlying graph of the formation F. The graph G can be directed or undirected depending on the properties of information links of the formation F. In a formation represented by a directed graph, only one of the agents in each pair, for example, agent A_i, actively maintains its distance to agent A_j at the desired distance d_{ij}, that is to say, only agent A_i has to receive information from agent A_j. Then in the directed graph, there exists a directed edge $\overrightarrow{(i, j)} \in E$ from vertex i to vertex

j. In this case, we can say the agent A_i follows the agent A_j or A_i is a follower of A_j. In an undirected graph, if there exists an edge $(i, j) \in E$, it means that $\overrightarrow{(i,j)} \in E$ and $\overrightarrow{(j,i)} \in E$, and then we can say that i and j are adjacent or that j is a neighbor vertex of i. The number of neighbor vertices of a vertex i (the number of edges to which it is incident) is called the degree of vertex i.

A graph on n vertices is called complete and denoted by K_n if $E = V \times V$, i.e., every pair of vertices is connnected by an edge. The complete graph contains $\frac{1}{2}n(n-1)$ edges. Graphs can sometimes be defined with self-edges, where $(i,i) \in E$, or with multiple edges between two vertices. A simple graph is one without self-edges and multiple edges. This dissertation exclusively uses simple graphs.

A path is a sequence of vertices $v_1, ..., v_m$ with $(i, i+1) \in E \; \forall i \in \{1, ..., m-1\}$. Vertices v_1 and v_m are said to be connected by a path. A graph is called connected if every pair of vertices is connected by a path. A graph is called k-connected if it remains connected after removing any $k-1$ edges.

A subgraph of a graph $G = (V, E)$ is a graph $G' = (V', E')$ with $V^{'} \subseteq V$ and $E^{'} \subseteq E$. The subgraph induced by a set of vertices V' is obtained by removing all vertices in $V \backslash V'$ (where $V \backslash V' = \{v \in V | v \in V and \; v \notin V'\}$) and all edges incident to them.

4.1.1.2 Rigid formations

A formation is named rigid if, during the continuous motion, it satisfies that the distance of each agent pairs explicitly remains constant. A rigid formation is further called minimally rigid if no single inter-agent distance constraint can be removed without losing rigidity. As mentioned above, a formation can be represented by a graph. In what follows we briefly introduce the theory of graph rigidity. For the details, the interested readers can refer to references Yu *et al.* (2007), Fidan *et al.* (2007) and Wen *et al.* (2010)).

In $R^n (n \in \{2, 3\})$, a representation of an undirected graph $G = (V, E)$ with a vertex set $V = \{1, 2, ..., N\}$ and an edge set $E \subset V \times V$ is a function $p : V \to R^n$. We say that $p(i) \in R^n$ is the position of the vertex i, and define the distance between two representations p_1 and p_2 of the same graph by

$$\delta(p_1, p_2) = \max_{i \in V} \|p_1(i) - p_2(i)\|$$

A distance set $\bar{\delta}$ for G is a set of distances $\delta_{ij} > 0$, defined for all edges $(i,j) \in E$. A distance set is realizable if there exists a representation p of the graph for which $\|p(i) - p(j)\| = \delta_{ij}$ for all $(i,j) \in E$. Such a representation is then called a realization. Note that each representation p of a graph induces a realizable distance set (defined by $\delta_{ij} = \|p(i) - p(j)\|$ for all $(i,j) \in E$), of which it is a realization.

A representation p is rigid if there exists $\varepsilon > 0$ such that for all realizations p' of the distance set induced by p and satisfying $\delta(p,p') < \varepsilon$, there holds $\|p'(i) - p'(j)\| = \|p(i) - p(j)\|$ for all $i,j \in V$. (We say in this case that p and p' are congruent)(Tay & Whiteley (1985)). A graph is said to be generically rigid if almost all its representations are rigid. Some discussions on the need for using 'generic' and 'almost all' can be found in previous works (see for examples Eren *et al.* (2004) and Yu *et al.* (2007)). One reason for using these terms is to avoid the problems arising from having three or more collinear agents in R^2, or four or more coplanar agents in R^3.

A widely approach in the analysis of rigidity is the use of linear algebraic tools such as the rigidity matrix (Tay & Whiteley (1985)). For a graph $G = (V,E)$ in R^d, the rigidity matrix $R(G)$ of G is defined as the $|E| \times d\,|V|$ matrix

$$\begin{bmatrix} \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & p^T(i) - p^T(j) & 0 & \cdots & 0 & p^T(j) - p^T(i) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

where each row $\begin{bmatrix} 0 & \cdots & 0 & p^T(i) - p^T(j) & 0 & \cdots & 0 & p^T(j) - p^T(i) & 0 & \cdots & 0 \end{bmatrix}$ corresponds to an edge $(i,j) \in E$, $p^T(i) - p^T(j)$ is a row d–vector in the d vertices is rigid if and only if for almost all representations, $R(G)$ has rank $d\,|V| - d(d+1)/2$, which is the maximum rank of $R(G)$ (Tay & Whiteley (1985)). Another notion that is widely used in rigidity analysis is minimal rigidity. A graph is called minimally rigid if it is rigid and if there exists no rigid graph with the same number of vertices and a smaller number of edges. Equivalently, a graph is minimally rigid if it is rigid and if no single edge can be removed without losing rigidity.

In the following, we give some theorems which can be used to test the rigidity (or minimal rigidity) of a graph.

Theorem 4.1 *(Laman (1970);Whiteley (1997)) A graph $G = (V,E)$ with $|V| \geq 2$ vertices and $|E|$ edges, is rigid in R^2 if and only if there is a subset $E' \subseteq E$ such that* *(i)* $|E'| \leq 2|V| - 3$,

(ii) For all non-empty $E'' \subseteq E'$ *there holds* $|E''| \leq 2|V''| - 3$, *where* V'' *is the set of vertices incident to the edges of* E''.

The first condition gives the minimum number of edges required for a rigid graph: given $|V|$ vertices, one must have at least $2|V| - 3$ edges. The second condition gives the manner in which a minimum set of edges must be distributed amongst the vertices to ensure rigidity.

Theorem 4.2 *(Laman (1970);Whiteley (1997)) A graph* $G = (V, E)$ *with* $|V| \geq 2$ *vertices and* $|E|$ *edges, is minimally rigid in* R^2, *if and only if it is rigid and contains* $2|V| - 3$ *edges, or equivalently if and only if*

(i) $|E| = 2|V| - 3$,

(ii) For all non-empty $E' \subseteq E$ *there holds* $|E'| \leq 2|V'| - 3$, *where* V' *is the set of vertices incident to the edges of* E'.

From condition (i), the minimally rigid graph has exactly $|E| = 2|V| - 3$ edges. The number of the required edges is linear. In contrast, in a complete graph, the number of required edges is quadratic in the number of vertices. This difference can be quite significant for the large formations. For example, a minimally rigid formation with 100 agents requires 197 distances to be maintained whereas a complete architecture would require $4,950$ links.

An example to illustrate the rigid formation in 2-dimensional space has been given in Fig.4.1. The example shows that the formation in Fig.4.1 (a) is not rigid, since it can be deformed by a smooth motion without affecting the distance between the agents connected by edges, as shown in Fig.4.1 (b). The formation represented in Fig.4.1 (c) and Fig.4.1 (d) are rigid, since they cannot be deformed. The formation represented in Fig.4.1 (c) is minimally rigid, because the removal of any edge renders it nonrigid. Finally, Fig.4.1 (d) is not minimally rigid, since any single edge can be removed without losing rigidity.

Unfortunately, the above analogous criterion in R^3 is only necessary.

Theorem 4.3 *(Hendrickx et al. (2007)) If a graph* $G = (V, E)$ *with* $|V| \geq 3$ *vertices and* $|E|$ *edges is rigid in* R^3, *then there exists* $E' \subseteq E$ *such that*

(i) $|E'| = 3|V| - 6$,

(ii) For all non-empty $E' \subseteq E$ *there holds* $|E''| \leq 3|V''| - 6$, *where* V'' *is the set of vertices incident to edges of* E''.

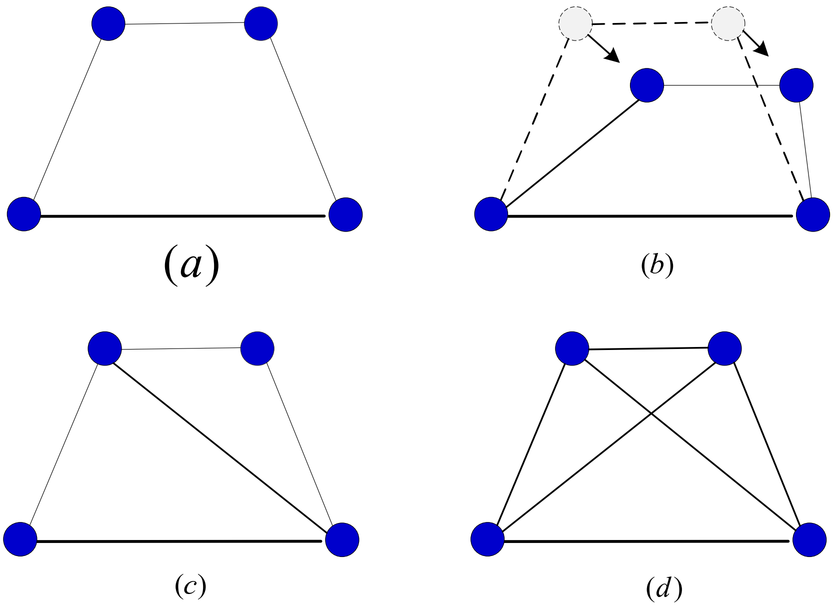

Figure 4.1: Rigid and nonrigid formations in 2-dimensional space.

(iii) The graph $G = (V', E')$ is 3-connected (i.e. remains connected after removal of any pair of vertices).

Theorem 4.4 *(Hendrickx et al. (2007)) A graph $G = (V, E)$ with $|V| \geq 3$ vertices and $|E|$ edges, is minimally rigid in R^3 if and only if it is rigid and contains $3|V| - 6$ edges.*

Fig.4.2 represents the 3-dimensional rigid and nonrigid formations. It can be seen that the formation in Fig.4.2 (a) is not rigid. The formation represented in Fig.4.2 (b) is minimally rigid, and that in the Fig.4.2 (c) is a rigid formation but not minimally rigid.

The condition (iii) in Theorem 4.3, which also implies the 3-connectivity of G, is not usually stated but is independently necessary even if the first two conditions are satisfied. Fig.4.2(d) shows an example of a non-rigid graph for which conditions (i) and (ii) in Theorem 4.3 are satisfied, but not condition (iii). Intuitively, the graph G' in the theorem needs to be sufficient to ensure "alone" the rigidity of G. 3-connectivity is then needed as otherwise two or more parts of the graph could rotate around the axis defined by any pair of vertices whose removal would disconnect the graph. Note that such connectivity condition is not necessary in 2-dimensional spaces, as the counting conditions (i) and (ii) of Theorem 4.1 imply 2-connectivity. For more information on necessary conditions for rigidity in three-dimensional spaces, we refer the reader to Mantler & Snoeyink (2004) and Hendrickx *et al.* (2007).

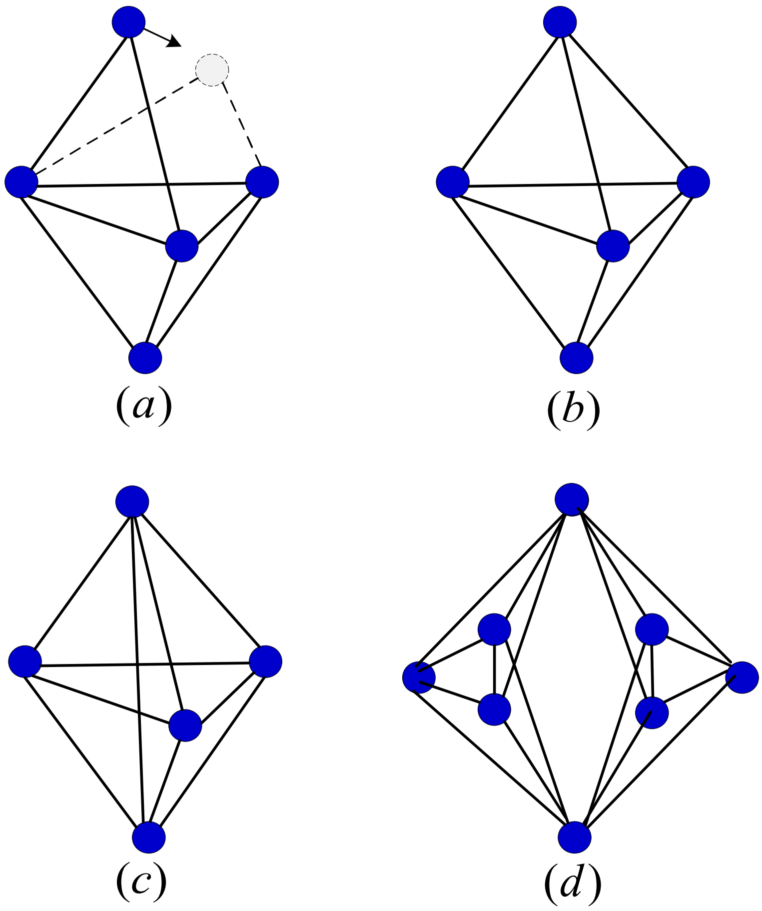

Figure 4.2: Rigid and nonrigid formations in 3-dimensional space.

4.1.1.3 Persistent formations

As mentioned in Subsection 4.1.1.2, rigidity is an undirected notion. Corresponding to the rigidity in the undirected graph, the persistence in the directed graph is defined. We call a directed graph rigid if and only if its underlying undirected graph is rigid. A formation is called persistent if it is rigid and satisfies another condition named constraint consistence, which is equivalent to the requirement that it is possible to maintain the nominated inter-agent distances. Similarly, a formation is called minimally persistent if it is minimally rigid and constraint consistent. In the following, we give the formal definitions on persistence from the graph theory.

Consider a directed graph $G = (V, E)$, a representation $p : V \to R^n$ ($n \in \{2, 3\}$), of G, and a set of desired distances $d_{ij} > 0$, $\forall\overrightarrow{(i,j)} \in E$. Note here that the representation, vertex positions, and distance between two representations corresponding to a directed graph are defined exactly the same as the ones corresponding to undirected graphs. We say that the edges $\overrightarrow{(i,j)} \in E$ is active if $|p(i) - p(j) = d_{ij}|$. We also say that the position of vertex $i \in V$ is fitting for the distance set $\bar{d}$ if it is not possible to increase the set of active edges leaving i by modifying the position of i while keeping the positions of the other vertices unchanged. More formally, given a representation p, the position of vertex i is fitting if there is no $p^* \in R^n$ for which

$$\left\{\overrightarrow{(i,j)} \in E : |p(i) - p(j) = d_{ij}|\right\} \subset \left\{\overrightarrow{(i,j)} \in E : |p^* - p(j) = d_{ij}|\right\}.$$

A representation p of a graph is called fitting for a certain distance set $\bar{d}$ if all the vertices are at fitting positions for $\bar{d}$. Note that any realization is a fitting representation for its distance set. The representation p is called persistent if there exists $\varepsilon > 0$ such that every representation $p^{'}$ fitting for the distance set induced by p and satisfying $\delta(p, p^{'}) < \varepsilon$ is congruent to p. A graph is then generically persistent if almost all its representations are persistent.

Similarly, a representation p is called constraint consistent if there exists $\varepsilon > 0$ such that any representation $p^{'}$ fitting for the distance set $\bar{d}$ induced by p and satisfying $\delta(p, p^{'}) < \varepsilon$ is a realization of $\bar{d}$. Again, we say that a graph is generically constraint consistent if almost all its representations in R^n are constraint consistent.

An example to illustrate the constraint consistence of a formation has been given in Fig. 4.3. Suppose agents 3 and 4 are fixed, and agent 4 satisfies its correct distance constraint from agent 3. Suppose also that agent 2 satisfies its correct distance from agent 3, and agent 1 satisfies its correct distances from 2, 3, and 4. Now let us observe from Fig. 4.3(b) that agent 2 has only one distance constraint from agent 3 , thus it can move on the circle centered at agent 3. As agent 1 has three distance constraints from agents 1, 2 and 3, when agent 2 moves on the circle centered at agent 3, agent 1 is indeed unable to satisfies the three

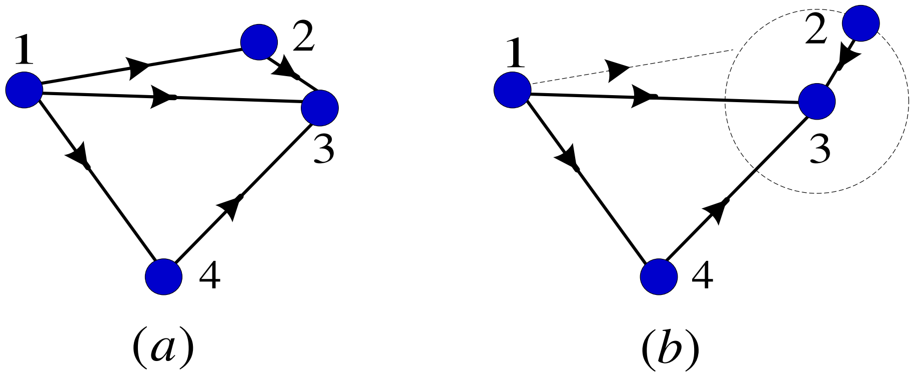

Figure 4.3: Illustration of a formation that is not constraint consistent.

distance constrains from agents 1, 2 and 3 in the same time. that is to say, agent 1 can only satisfy two distance constraints among three. Thus, we can say the formation is not constraint consistent. For more information about constraint consistence, we refer the reader to Anderson *et al.* (2006), Anderson *et al.* (2008) and Hendrickx *et al.* (2006).

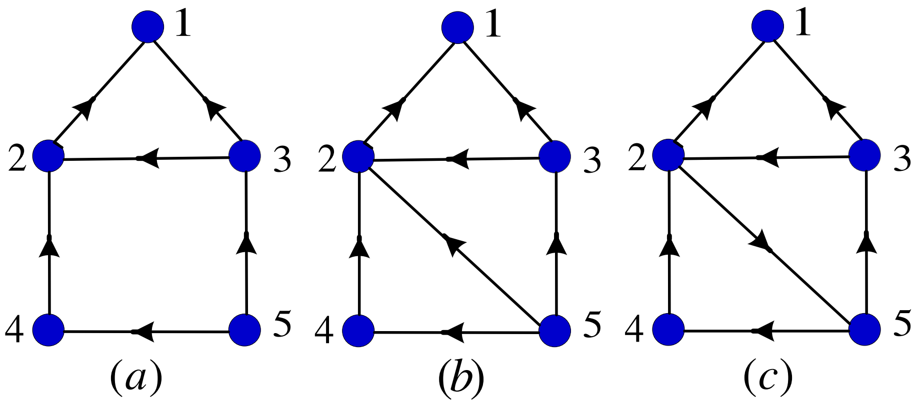

Figure 4.4: Persistent and nonpersistent formations in 2-dimensional space.

In the following, we give the relation among rigidity, persistence, and constraint consistence of a directed graph, which have already been established in (Hendrickx *et al.* (2007);Yu *et al.* (2007);Fidan *et al.* (2007)).

Theorem 4.5 *(Hendrickx* et al. *(2007)) A representation in R^n ($n \in \{2, 3\}$) is persistent if and only if it is rigid and constraint consistent. A graph in R^n ($n \in \{2, 3\}$) is generically persistent if and only if it is generically rigid and generically constraint consistent.*

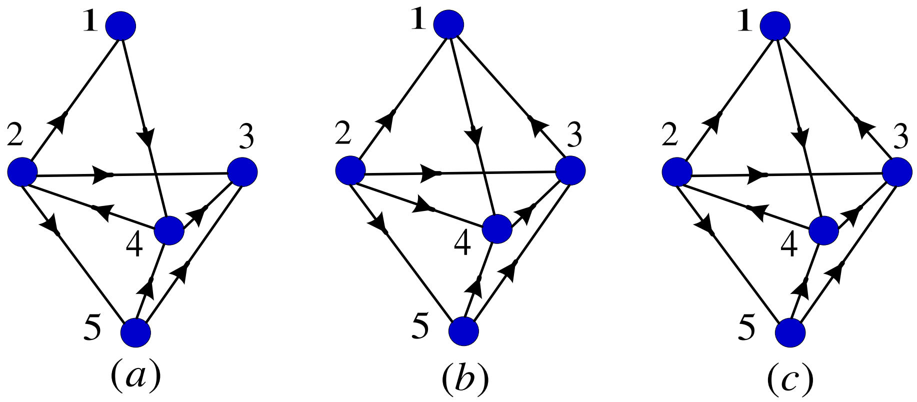

Figure 4.5: Persistent and nonpersistent formations in 3-dimensional space.

To illustrate application of Theorem 4.5, some examples of the persistent and non-persistent formations in R^2 and R^3 are shown in Figs. 4.4 and 4.5, respectively. The graphs in Figs. 4.4 (a) and 4.5(a) are constraint consistent but not rigid, hence they are not persistent. The graphs in Figs. 4.4 (b) and 4.5(b) are rigid but not constraint consistent (Note from Fig.4.4 (b) that, 1, 2, 3 and 4 can move to new positions, without violating the distance constraint on them, for which vertex 5 is unable to satisfy all the three distance constraints on it at the same time. A similar problem at vertex 2 in Fig.4.5(b) implies that it is not constraint consistent.Hence the formation in Fig.4.5(b) is not persistent. The graphs in Figs. 4.4 (c) and 4.5(c) are rigid and constraint consistent, hence they are persistent.

Let $d^-(i)$ and $d^+(i)$ denote the in-and out-degree of the vertex i in the graph G, i.e., the number of edges in G heading to and originating from i, respectively.

In order to check persistence of a directed graph G, the following proposition is given (Yu *et al.* (2007)).

Proposition 4.6 *(Yu* et al. *(2007)) An persistent graph* $(n \in 2, 3)$ *remains persistent after deletion of any edge* $\overrightarrow{(i, j)}$ *for which* $d^+(i) \geq n + 1$*. Similarly, a constraint consistent graph* $(n \in 2, 3)$ *remains constraint consistent after deletion of any edge* $\overrightarrow{(i, j)}$ *for* $d^+(i) \geq n + 1$*.*

Generally, we use the term out-degree of a vertex A_i to denote the number of distance constraints. Associated with the out-degree, the number of degrees of freedom of a vertex i denote the dimension of the set in which the corresponding agent can choose its position(all the other agents being fixed). Usually, we abbreviate degree of freedom as DOF. The number of DOFs of a vertex i in R^n is given by $\max(0, n - d^+(i))$, where $d^+(i)$ represents the out-degree of the vertex i. For example, in R^2, the numbers of DOFs of the vertices with zero and one out-degrees are 2 and 1 respectively, and all the other vertices have 0 DOF; in R^3, the numbers of DOFs of the vertices with zero, one, and two out-degrees are 3, 2, and 1, respectively, and all the other vertices have 0 DOF. In an underlying graph of n-dimensional formation, a vertex with n-DOF in R^n is called a leader. The following corollary provides a natural bound on the total number of degrees of freedom in a persistent graph in R^n $(n \in 2, 3)$, which we also call the total DOF count of that graph in R^n.

Corollary 4.7 *(Yu et al. (2007)) The total DOF count of a persistent graph in R^n $(n \in 2, 3)$ can at most be $n(n+1)/2$, and for a minimally persistent graph in R^n $(n \in 2, 3)$ exactly $n(n+1)/2$.*

Theorem 4.8 *(Fidan et al. (2007)) A directed graph is persistent in $R^n(n \in 2, 3)$ if and only if all those subgraphs are rigid which are obtained by successively removeing outgoing edges from vertices with out-degree larger than n until all such vertices have an out-degree equal to n.*

From the Corollary 4.7, in a 2-dimensional space, for a persistent formation, the sum of DOF of individual agents is at most 3, and for a minimally persistent formation, exactly 3. based on the distribution of these 3 DOFs, the minimally persistent formations can be divided into two categories: $S_1 = \{2, 2, 0, 0, ...\}$, $S_2 = \{1, 1, 1, 0, 0, ...\}$. S_1 is called formation with the leader-follower structure where one agent has 2-DOFs, another has 1-DOF and the rest have 0-DOF. S_2 is called formation with the 3-coleader structure where three agents have 1-DOF and the rest have 0-DOF. Similarly, in a 3-dimensional space, the sum of DOFs of individual agents is at most six and for a minimally persistent formation, exactly six. Based on the distribution of these six DOFs, minimally persistent formations can be divided into the following categories: $S_1 = \{3, 2, 1, 0, 0, ...\}$, $S_2 = \{2, 2, 2, 0, 0, ...\}$, $S_3 = \{3, 1, 1, 1, 0, 0, ...\}$, $S_4 = \{2, 2, 1, 1, 0, 0, ...\}$, $S_5 = \{2, 1, 1, 1, 1, 0, 0, ...\}$, $S_6 = \{1, 1, 1, 1, 1, 1, 0, ...\}$, where in the categories S_1, for example, one agent has three DOFs, one agent has two DOFs, one agent has one DOF, and all the others have zero DOF, Note that in the above categories, one possible category has been skipped, i.e., $S_0 = \{3, 3, 0, 0, ...\}$. This is because it is an undesired category that can be accepted only temporarily during formation changes as a transient state. This is because S_0 apparently allows two leaders simultaneously in a formation, corresponding to failure of structural persistence and we want the DOFs assignment to avoid this state. Further detailed interpretations can be found in Refs. Fidan *et al.* (2007), Yu *et al.* (2007) and Zhao *et al.* (2010).

As mentioned above, provided that the numbers of DOFs of the agents are different, the numbers of the constraints of on them are also different. In R^3, if some agent A_i has 3 DOFs then it can move freely . If the agent A_i has 2 DOFs, then it can move freely on the spherical surface with center A_j and radius d_{ij}, where A_j is the agent which A_i follows and d_{ij} is the distance between them. If the agent A_i has 1 DOF, then it can rotate freely on the circle, which is the intersection of two spherical surfaces with A_j, A_k as the centers, and d_{ij}, d_{ik} as the radius, respectively. If the agent A_i has 0 DOF, then it can move depending on three agents it follows completely.

4.1.2 Problem statement

In this chapter, we consider a persistent formation F with $m \geq 3$ point agents $A_1, \ldots, A_m$ whose initial position and orientation are specified with a set $\bar{d}$ of desired inter-agent distances d_{ij} between neighbor agent pairs (A_i, A_j), where d_{ij} is a scalar. Suppose each agent A_i knows its absolute position, as well as the position of any agent that it follows at any time t. The velocity integrator kinematics for each agent A_i is given by:

$$\dot{p}_i(t) = v_i(t) \tag{4.1}$$

where $p_i(t) = (x_i(t),\, y_i(t),\, z_i(t)) \in R^3$ and $v_i(t) = (v_{xi}(t),\, v_{yi}(t),\, v_{zi}(t)) \in R^3$ $(i \in \{1, \ldots, m\})$ denote the position and velocity of A_i on the 3-dimensional (x, y, z) cartesian coordinate system at time t, respectively. In this chapter, the velocity $v_i(t)$ is considered as the control signal to be generated by the individual controller of agent A_i. It is required that $v_i(t)$ is continuous and satisfies $\mid v_i(t) \mid < \bar{v}$, for some constant maximum speed limit $\bar{v} > 0$ at any $t \geq 0$ for any $i \in \{1, \ldots, m\}$.

Our control task is to move the 3-dimensional "persistent formation" F with specified initial position to a new desired(final) position and orientation in an obstacle rich environment , cohesively. That is , it does not violate the persistence of F during the motion. Also, here the initial and final position $p_i(0)$ and $p_{if}(t)$ of each agent A_i are known and are consistent with a set $\bar{d}$ of desired inter-agent distances d_{ij}.

In order to illustrate the proposed mechanisms for planning and control of the multi-agent formations in 3-dimensional space, a flow chart is shown in Fig. 4.6 (Wen *et al.* (2012 a)). The flow chart can be described as follows. Firstly, a grid map is given based on 3-dimensional environment. For a given formation, we search a preliminary path by using an A^* algorithm. In order to design a better path to follow, a set of waypoints are extracted from this preliminary path. These waypoints are traversed by the center of mass of the formation (i.e. the formation is assumed as a large entity and the center of mass of the formation should follow the designed trajectory). Secondly, according to each agent's position in the formation and the center of mass of the formation, the reference trajectory for each agent is generated. Next, a set of decentralized control laws for the cohesive motion of multi-agent formations are applied to make the formation reach the desired position.

In the following sections, we will organize this chapter under the above framework.

4.2 Path planning

4.2.1 Grid Map Generation

In order to define various subregions in the area of interest, we first introduce the following notation:

$$\Omega(\overline{x},\, \overline{y},\, \overline{z},\, l,\, w,\, h) = \{(x,\, y,\, z)|\ \overline{x} \leq x \leq \overline{x} + l,\, \overline{y} \leq y \leq \overline{y} + w,\, \overline{z} \leq z \leq \overline{z} + h\}$$

which denotes the cubic region on the x-y-z space, where $(\overline{x},\, \overline{y},\, \overline{z})$ are the coordinates of the bottom left corner of the cubic region, l, w, and h are its length, width and height parallel to the x, y, and z-axis, respectively.

The environment map (area of interest) is represented by a cube

$$\Omega_E = \Omega(x_0,\, y_0,\, z_0,\, L,\, W,\, H)$$

with an overlaid grid structure, where the size of each cell grid represents the grid resolution. In this work, the grid resolution is assumed to be $0.1m$, *i.e.* each of x_0, y_0, z_0, L, W, H are assumed to be integer multiples of $0.1m$. The obstacle regions are represented by a sequence of M cubes.

$$\Omega_{oi} = \Omega(x_{oi},\, y_{oi},\, z_{oi},\, l_{oi},\, w_{oi},\, h_{oi}),\;\; i \in \{1,\, ...\,,\, M\}$$

where x_{oi}, y_{oi}, z_{oi}, l_{oi}, w_{oi}, h_{oi} are integer multiples of $0.1m$ for each i. Furthermore, define

$$\Omega_o = \Omega_{o1} \cup \cdots \cup \Omega_{oM}.$$

In addition, $\overline{\Omega}_{0i}$ and $\overline{\Omega}_o$ are represented as

$$\begin{aligned}\overline{\Omega}_{oi} = \Omega(&x_{oi} - \varepsilon,\, y_{oi} - \varepsilon,\, z_{oi} - \varepsilon,\, l_{oi} + 2\varepsilon,\, w_{oi} + 2\varepsilon,\\ &h_{oi} + 2\varepsilon),\;\; i \in \{1,\, 2,\, ...,\, M\}\end{aligned}$$

and

$$\overline{\Omega}_o = \overline{\Omega}_{o1} \cup \cdots \cup \overline{\Omega}_{oM},$$

respectively, where $\varepsilon \geq R_F$, and R_F is the radius of the smallest sphere which can strictly contain all agents. Assume that $\Omega_0 \cup \overline{\Omega}_0 \subset \Omega_E$. The grid-map generation task can be characterized for coloring the $\frac{L}{0.1} \cdot \frac{W}{0.1} \cdot \frac{H}{0.1}$ grids of Ω_E, *i.e.*, each grid can be described

$$\Omega_g[i,\, j,\, k] = \Omega(x_0 + 0.1i,\, y_0 + 0.1j,\, z_0 + 0.1k,\, 0.1,\, 0.1,\, 0.1)$$

where $i \in \{0,\, 1,\, ...,\, (L/0.1)-1\}$, $j \in \{0,\, 1,\, ...,\, (W/0.1)-1\}$, and $k \in \{0,\, 1,\, ...,\, (H/0.1)-1\}$, with one of three colors: white, red, yellow. Each grid $\Omega_g[i,\, j,\, k]$ is colored according to the following rule:

1. If the area $\Omega_g[i,\, j,\, k] \subset \Omega_o$, then the color of the grid $\Omega_g[i,\, j,\, k]$ is red.
2. If the area $\Omega_g[i,\, j,\, k] \subset \overline{\Omega}_o \backslash \Omega_o$, then the color of the grid $\Omega_g[i,\, j,\, k]$ is yellow.
3. Otherwise, $\Omega_g[i,\, j,\, k]$ will be colored white.

The aim of the above colored grids is: the path can only pass through white grid region, the agents can move through yellow and white grid region. Based on the above-mentioned strategy we can guarantee that the formation can not collide with obstacles.

4.2.2 Path Generation

This section aims at finding a path, if it exists, between the initial position and the desired position for a given formation in 3-dimensional space. First, the A^* algorithm (see Nilsson (1980)) is employed to search for an optimal path. We express the optimal path by a sequence of grid squares, which can be called the A^* path sequence denoted by S_{A^*}. Here the terrain in 3-dimensional space can be divided into cubic grids cell, and the maximum number of movements from each grid cube is restricted to 26 directions or less when the adjacent nodes are placed in a yellow region or are the borders of the environment. Fig. 4.7 can show the possible grid cubes to be chosen for being the next element in the path sequence.

It is well known that redundant points will be generated by using the S_{A^*} search algorithm. In order to further optimize the path sequence S_{A^*} , an algorithm similar to the one presented in Shames *et al.* (2007) is given as follows, where we assume that the length of S_{A^*} is N,

1. Construct a reverse waypoint list $W_R[.]$:
   (a) Set $l := 1, k := N$, and consider $n-$th point in the path sequence $\mathrm{S}_{\mathrm{A}^*}[.]$ as the first point in the reverse waypoint list $W_R[.]$, i.e., $W_R[1] := \mathrm{S}_{\mathrm{A}^*}[N]$.
   (b) Do until $k := 1$
      i. Check whether there is a non-obstructed path between the $l-$th point $W_R[l]$ in the reverse waypoint list and $(k-1)-$th point $\mathrm{S}_{\mathrm{A}^*}[k-1]$ in the path sequence.
      ii. If yes, set $k := k-1$ and go to (b).
      iii. If no, take $k-$th point in reverse waypoint list as the next point (i. e. set $W_R[l+1] := \mathrm{S}_{\mathrm{A}^*}[k]$). Set $l := l+1$, and then go to (b).
2. Reverse the order of reverse waypoint list $W_R[.]$ and store it as the waypoint list $W[.]$.

4.3 Control laws for the formation

In this section, we consider the decentralized control laws for the cohesive motion of 3-dimensional multi-agent formations. A preliminary version of this section has appeared in our works Zhao *et al.* (2010);Wen (2009). The aim is to move a given persistent formation with specified initial position and orientation to a new desired position and orientation cohesively, i.e., without violating the persistence of the formation during the motion. According to the previous section, we know, if the DOFs are different, the constraint numbers of the agents are also entirely different. According to the number of DOFs for the agents in the formation, we can label the agents as 0-DOF agents, 1-DOF agents, 2-DOF agents, 3-DOF agents, respectively. Furthermore, we will design the corresponding control laws for different DOF agents, distinctively. For simplicity, in this chapter we consider to control a minimally persistent formation with a leader-follower structure, which consists of one 3-DOF agent, one 2-DOF agent, one 1-DOF agent and $n \in \{0, 1, 2, ..., m\}$ 0-DOF agents. Also, we assume that the 2-DOF agent directly follows the 3-DOF agent, and the 1-DOF agent follows the 3-DOF agent and the 2-DOF agent (see Fig.4.8).

In the following, several other assumptions are given as follows:

1. Each agent A_i knows its own position $p_i(t)$, velocity $v_i(t)$. In addition, for each 0-DOF agent, it knows the position of any agent that it follows at any time $t \geq 0$; for each 1-DOF agent and each 2-DOF agent, it knows its final desired position and the position of any agent that it follows at any time $t \geq 0$; for each 3-DOF agent, it knows its final desired position.

2. The sensing distance range for a neighbor agent pair (A_i, A_j) is sufficiently larger than the desired distance d_{ij} to be maintained.

3. The motion of an agent at any time just relies on the agents it follows and the final position it desires to reach.

4. For each 1-DOF agent or 2-DOF agent, its distance constraints has a higher priority to reach the final desired position, i.e., 1-DOF agents or 2-DOF agent can undergo their DOFs movements, only when their distance constraints are satisfied within a certain error bound.

5. The controller $v_i(t)$ of each agent A_i is continuous and $|v_i(t)| \leq \bar{v}$, $\forall t \geq 0$ for some constant maximum speed limit $\bar{v} > 0$.

Based on the above assumptions and the agent kinematics (4.1), the control laws for the cohesive motion of 3-dimensional multi-agent formations are developed(Wen *et al.* (2012 a)).

4.3.1 Control law for the 0-DOF agent

Consider a 0-DOF agent A_i, and the three agents A_j, A_k and A_l it follows. Due to the distance constraint of keeping $| p_i(t) - p_j(t) |$, $| p_i(t) - p_k(t) |$, $| p_i(t) - p_l(t) |$ at the desired values d_{ij}, d_{ik}, d_{il} respectively, at each time $t \geq 0$, the desired position $p_{id}(t)$ of A_i is the point whose distances to $p_j(t)$, $p_k(t)$, $p_l(t)$ are d_{ij}, d_{ik}, d_{il} respectively. Further, $p_{id}(t)$ must vary continuously. Assuming $| p_i(t) - p_{id}(t) |$ is sufficiently small, $p_{id}(t)$ can be explicitly determined as $p_{id}(t) = \bar{p}_{jkl}(t, p_i(t))$, where $\bar{p}_{jkl}(t, p_0)$ for any $p_0 \in R^3$ denotes the intersection of the spherical surface $O(p_j(t), d_{ij})$, $O(p_k(t), d_{ik})$, and $O(p_l(t), d_{il})$ that is closer to p_0, and in the notion $O(\cdot, \cdot)$ the first argument denotes the center and the second denotes the radius. Based on this observation, the following control law can be designed

$$v_i(t) = \bar{v}\beta_i(t)\delta_{id}(t)/|\delta_{id}(t)| \tag{4.2}$$

where

$$\delta_{id}(t) = p_{id}(t) - p_i(t),$$

$$\beta_i(t) = \begin{cases} 0, & |\delta_{id}(t)| < \varepsilon \\ \frac{|\delta_{id}(t)| - \varepsilon}{\varepsilon} & \varepsilon \leq |\delta_{id}(t)| < 2\varepsilon \\ 1, & |\delta_{id}(t)| \geq 2\varepsilon \,, \end{cases}$$

and $\bar{v} > 0$ represents the maximum speed of all the agents. Here $\varepsilon > 0$ is a small design constant. In (4.2), the switching term $\beta_i(t)$ is introduced to avoid chattering due to small but acceptable errors in the desired inter-agent distances.

4.3.2 Control law for the 1-DOF agent

Consider a 1-DOF agent A_i and the two agents A_j, A_k it follows. A_i requires to maintain its distance to A_j, A_k. It is free to move on the circle which is the intersection of two spherical surfaces with A_j, A_k as the center, and d_{ij}, d_{ik} as the radius, respectively. Also it is not required to use the whole of its velocity capacity to satisfy its distance constraints. Let the circle $C_{jk}(t)$ denote the intersection of the spheres $o(p_j, d_{ij})$ and $o(p_k, d_{ik})$, and $p_{i\hat{d}}(t)$ denote the point on the circle

$C_{jk}(t)$ that is closest to $p_i(t)$. Based on this observation and the assumption, we propose the following control scheme (4.3) for the 1-DOF agent A_i :

$$v_i(t) = \beta_i(t)v_{i1}(t) + \sqrt{1-\beta_i^2(t)}v_{i2}(t) \tag{4.3}$$

where

$$v_{i1}(t) = \overline{v}\delta_{i\hat{d}}(t)/\left|\delta_{i\hat{d}}(t)\right|, \tag{4.4}$$

$$v_{i2}(t) = \overline{v}\bar{\beta}_i\delta^*_{i\hat{d}}(t) \tag{4.5}$$

$$\delta^*_{i\hat{d}}(t) = (\delta_{if}(t) - \frac{\delta_{i\hat{d}}(t)}{|\delta_{i\hat{d}}(t)|}\cdot|\delta_{if}(t)|\cdot\cos\theta)/\left|(\delta_{if}(t) - \frac{\delta_{i\hat{d}}(t)}{|\delta_{i\hat{d}}(t)|}\cdot|\delta_{if}(t)|\cdot\cos\theta)\right|$$

$$\cos\theta = (\delta_{if}(t)\delta_{i\hat{d}}(t))/(|\delta_{if}(t)||\delta_{i\hat{d}}(t)|),$$

$$\delta_{if}(t) = p_{if}(t) - p_i(t),$$

$$\delta_{i\hat{d}}(t) = p_{i\hat{d}}(t) - p_i(t),$$

and

$$\beta_i(t) = \begin{cases} 0, & |\delta_{i\hat{d}}(t)| < \varepsilon \\ \frac{|\delta_{i\hat{d}}(t)|-\varepsilon}{\varepsilon}, & \varepsilon \le |\delta_{i\hat{d}}(t)| < 2\varepsilon \\ 1, & |\delta_{i\hat{d}}(t)| \ge 2\varepsilon \end{cases}$$

$$\bar{\beta}_i(t) = \begin{cases} 0, & |\delta_{if}(t)| < \varepsilon_f \\ \frac{|\delta_{if}(t)|-\varepsilon_f}{\varepsilon_f}, & \varepsilon_f \le |\delta_{if}(t)| < 2\varepsilon_f \\ 1, & |\delta_{if}(t)| \ge 2\varepsilon_f \end{cases}$$

Here ε, $\varepsilon_f > 0$ are small design constants. The switching terms $\beta_i(t)$ and $\overline{\beta}_i(t)$ are introduced to switch translational actions (4.4) to satisfy $|A_iA_j| \cong d_{ij}$, $|A_iA_k| \cong d_{ik}$, and the rotational action (4.5). The term $\overline{\delta}^*_{ji}(t)$ is the unit vector perpendicular to the distance vector $\delta_{i\hat{d}}(t) = p_{i\hat{d}}(t) - p_i(t)$ in the plane formed by vector $\delta_{i\hat{d}}(t)$ and $\delta_{if}(t)$, and we denote the angle between $\delta_{i\hat{d}}(t)$ and $\delta_{if}(t)$ by θ.

4.3.3 Control law for the 2-DOF agent

Consider a 2-DOF agent A_i, which is called the first follower in general, and which is only required to maintain a certain distance to another agent A_j. It is free to move on the spherical surface with the center A_j and radius d_{ij}. Based on this observation, we propose the following control scheme for 2-DOF agent A_i:

$$v_i(t) = \beta_i(t)v_{i1}(t) + \sqrt{1-\beta_i^2(t)}v_{i2}(t) \tag{4.6}$$

where

$$v_{i1}(t) = \bar{v}\bar{\delta}_{ji}(t)/\left|\bar{\delta}_{ji}(t)\right|, \tag{4.7}$$

$$v_{i2}(t) = \bar{v}\bar{\beta}_i(t)\bar{\delta}^*_{ji}(t). \tag{4.8}$$

$$\begin{aligned}
\delta_{ji}(t) &= p_j(t) - p_i(t) = (\delta_{jix}(t),\, \delta_{jiy}(t),\, \delta_{jiz}(t))\,,\\
\bar{\delta}_{ji}(t) &= \delta_{ji}(t) - d_{ij}\delta_{ji}(t)/\left|\delta_{ji}(t)\right|,\\
\delta_{if}(t) &= p_{if}(t) - p_i(t)\,,
\end{aligned}$$

$$\bar{\delta}^*_{ji}(t) = \quad (\delta_{if}(t) - \tfrac{\delta_{ji}(t)}{|\delta_{ji}(t)|}\cdot \mid \delta_{if}(t) \mid \cdot \cos\theta)/|(\delta_{if}(t) - \tfrac{\delta_{ji}(t)}{|\delta_{ji}(t)|}\cdot \mid \delta_{if}(t) \mid \cdot \cos\theta)|\,,$$

$$\cos\theta = (\delta_{if}(t)\delta_{ji}(t))/(|\delta_{if}(t)|\delta_{ji}(t)|)\,,$$

and

$$\beta_i(t) = \begin{cases} 0 & |\bar{\delta}_{ji}(t)| < \varepsilon \\ \frac{|\bar{\delta}_{ji}(t)|-\varepsilon}{\varepsilon} & \varepsilon \le |\bar{\delta}_{ji}(t)| < 2\varepsilon \\ 1 & |\bar{\delta}_{ji}(t)| \ge 2\varepsilon\,, \end{cases}$$

$$\bar{\beta}_i(t) = \begin{cases} 0 & |\delta_{if}(t)| < \varepsilon_f \\ \frac{|\delta_{if}(t)|-\varepsilon_f}{\varepsilon_f} & \varepsilon_f \le |\delta_{if}(t)| < 2\varepsilon_f \\ 1 & |\delta_{if}(t)| \ge 2\varepsilon_f\,, \end{cases}$$

Here $\varepsilon, \varepsilon_f > 0$ are small design constants. In equation (4.6), when $|A_iA_j|$ is sufficiently close to d_{ij}, the switching term $\beta_i(t)$ can make the controller switch between the translational action (4.7) and the rotational action (4.8), in which the action (4.7) satisfies $|A_iA_j| \cong d_{ij}$, and the rotational action (4.8) can make the agent A_i move towards its desired final position p_{if}.

In equation (4.8), $\overline{\delta}^*_{ji}(t)$ is the unit vector perpendicular to the distance vector $\delta_{ji}(t) = p_j(t) - p_i(t)$ in the plane formed by vector $\delta_{ji}(t)$ and $\delta_{if}(t)$, and θ denotes the angle of $\delta_{ji}(t)$ and $\delta_{if}(t)$. The switching term $\overline{\beta}_i(t)$ is designed for avoiding chattering due to small but acceptable errors in the final position of A_i.

4.3.4 Control laws for the 3-DOF agents

Consider a 3-DOF agentA_i, which is called the leader of the formation, since it does not have any constraint to satisfy. It can use its full velocity capacity only to move towards its desired final position p_{if}. Hence the velocity input at each time t can be simply designed as a vector with the constant maximum speed $\bar{v}$ in the direction of $p_{if}(t) - p_i$. The controller law is given below:

$$v_i(t) = \bar{v}\bar{\beta}_i(t)\delta_{if}(t)/|\delta_{if}(t)| \tag{4.9}$$

where

$$\delta_{if}(t) = p_{if}(t) - p_i(t)$$

and

$$\bar{\beta}_i(t) = \begin{cases} 0 & |\delta_{if}(t)| < \varepsilon_f \\ \frac{|\delta_{if}(t)| - \varepsilon_f}{\varepsilon_f} & \varepsilon_f \leq |\delta_{if}(t)| < 2\varepsilon_f \\ 1 & |\delta_{if}(t)| \geq 2\varepsilon_f \,. \end{cases}$$

The switching term $\overline{\beta}_i(t)$ is again chosen for preventing the chatter due to small but acceptable errors in the final position p_{if} of A_i.

4.4 Stability and convergence

In this section, we consider the stability and convergence of the proposed control laws. Consider a persistent formation F with $m \geq 3$ agents denoted by A_1, A_2,...,A_m. Without loss of generality, suppose that A_1 is the 3-DOF agent, A_2 is the 2-DOF agent, A_3 is the 1-DOF agent, and other agents are the 0-DOF agents. Such a formation is depicted in Fig. 4.8 in which A_2 follows A_1, A_3 follows A_1 and A_2 , and A_4 follows A_1, A_2 and A_3.

4.4.1 Stability and convergence for the 3-DOF agent

Note that the 3-DOF agent A_1 uses the control law (4.9). Let us choose the Lyapunov function candidate as follows

$$V_1(t) = \frac{1}{2}\delta_{1f}^T(t)\delta_{1f}(t) \tag{4.10}$$

Taking the derivative of $V_1(t)$ along (4.9) gives

$$\begin{aligned}\dot{V}_1(t) =&\frac{1}{2}\delta_{1_f}^T(t)v_1(t)\\ =&-\bar{v}\bar{\beta}_1(t)\delta_{1_f}^T(t)\delta_{1f}(t)/\left|\delta_{1f}(t)\right|\\ =&-\bar{v}\bar{\beta}_1(t)\left|\delta_{1f}(t)\right|\\ =&\begin{cases}0, & \text{if } \left|\delta_{1f}(t)\right|<\varepsilon_f\\ -\bar{v}\left|\delta_{1f}(t)\right|\frac{\left|\delta_{1f}(t)\right|-\varepsilon_f}{\varepsilon_f}\le -\bar{v}(\left|\delta_{1f}(t)\right|-\varepsilon_f), & \text{if } \varepsilon_f\le\left|\delta_{1f}(t)\right|<2\varepsilon_f\\ -\bar{v}\left|\delta_{1f}(t)\right|\le -2\bar{v}\varepsilon_f, & \text{if } \left|\delta_{1f}(t)\right|\ge 2\varepsilon_f\end{cases}\end{aligned} \tag{4.11}$$

Therefore, $\left|\delta_{1f}(t)\right|\le\left|\delta_{1f}(0)\right|$, $\forall t\ge 0$, and $\lim_{t\to\infty}\left|\delta_{1f}(t)\right|\le\varepsilon_f$, i.e., $p_1(t)$ is always bounded and converges asymptotically to the sphere $o(p_{1f},\varepsilon_f)$ with center p_{1f} and radii ε_f. Indeed, we can also verify from (4.11) that $p_1(t)$ can enter the sphere $o(p_{1f},2\varepsilon_f)$ in finite time and remain there.

4.4.2 Stability and convergence for the 2-DOF agent

The 2-DOF agent A_2 uses the control law (4.6). We aim to prove that: 1) $p_2(t)$ remains bounded, and converges to the finite time-varying sphere of $p_2(t)$ for $t\ge 0$ with certain radii; 2) $p_2(t)$ converges to a fixed sphere $o(p_{2f},\varepsilon_f)$.

First, we prove $p_2(t)$ remains bounded, and converges to finite time-varying sphere of $p_2(t)$ for $t\ge 0$ with certain radii. Consider the following Lyapunov function candidate

$$V_{21}=\frac{1}{2}\bar{\delta}_{12}^T(t)\bar{\delta}_{12}(t). \tag{4.12}$$

Tacking the derivative of $V_{21}(t)$ along (4.6) gives

$$\begin{aligned}\dot{V}_{21}=&\bar{\delta}_{12}^T(t)\dot{\bar{\delta}}_{12}(t)\\ =&\bar{\delta}_{12}^T(t)(v_1(t)-v_2(t))\\ =&\bar{v}\bar{\beta}_1(t)\bar{\delta}_{12}^T(t)\delta_{1f}(t)/\left|\delta_{1f}(t)\right|-\bar{v}\beta_2(t)\bar{\delta}_{12}^T(t)\bar{\delta}_{12}(t)/\left|\bar{\delta}_{12}(t)\right|\\ &-\bar{v}\sqrt{1-\beta_2^2(t)}\bar{\beta}_i(t)\bar{\delta}_{12}^T(t)\bar{\delta}_{12}^*(t)\\ =&\bar{v}\bar{\beta}_1(t)\bar{\delta}_{12}^T(t)\delta_{1f}(t)/\left|\delta_{1f}(t)\right|-\bar{v}\beta_2(t)\left|\bar{\delta}_{12}(t)\right|\end{aligned} \tag{4.13}$$

where we have used the fact that $\bar{\delta}_{12}^T(t)$ is perpendicular to $\bar{\delta}_{12}^*(t)$.

From (4.8) we note that when $\left|\bar{\delta}_{12}^T(t)\right|\ge 2\varepsilon$, $\beta_2=1$. Then we have from (4.13) that

$$\dot{V}_{21} = \begin{cases} -\bar{v}\left|\bar{\delta}_{12}(t)\right| \leq -2\bar{v}\varepsilon, & \text{if } |\delta_{1f}(t)| < \varepsilon_f \\ \bar{\delta}_{12}^T(t)\bar{v}\frac{|\delta_{1f}(t)|-\varepsilon_f}{\varepsilon_f}\frac{\delta_{1f}(t)}{|\delta_{1f}(t)|} - \bar{v}\left|\bar{\delta}_{12}(t)\right| & \\ \qquad \leq -\bar{v}\left|\bar{\delta}_{12}(t)\right|\left(\frac{2\varepsilon_f-|\delta_{1f}(t)|}{\varepsilon_f}\right), & \text{if } \varepsilon_f \leq |\delta_{1f}(t)| < 2\varepsilon_f \\ \bar{\delta}_{12}^T(t)(\bar{v}\delta_{1f}(t)/|\delta_{1f}(t)| - \bar{v}\bar{\delta}_{12}(t)/\left|\bar{\delta}_{12}(t)\right|) \leq 0. & \text{if } |\delta_{1f}(t)| \geq 2\varepsilon_f \end{cases} \tag{4.14}$$

Therefore, when $\left|\bar{\delta}_{12}^T(t)\right| \geq 2\varepsilon$, we obtain $\left|\bar{\delta}_{12}(t)\right| \leq \left|\bar{\delta}_{12}(0)\right|$, $\forall t \geq 0$. Moreover, according to the analysis of stability and convergence for the 3-DOF agent in subsection 4.4.1, $p_1(t)$ asymptotically converges to the sphere $o(p_{1f}, \varepsilon_f)$, i.e. $|\delta_{1f}(t)|$ will satisfy that $|\delta_{1f}(t)| \leq \varepsilon_f$. Hence, it can be obtained from (4.14) that $p_2(t)$ is always bounded and here asymptotically converges to the sphere $o(p_1, d_{12} + 2\varepsilon)$ with center p_1 and radii $d_{12} + 2\varepsilon$.

When $\varepsilon \leq \left|\bar{\delta}_{12}^T(t)\right| < 2\varepsilon$, $\beta_2 = \frac{\left|\bar{\delta}_{12}(t)\right|-\varepsilon}{\varepsilon}$. Then from (4.13) we have

$$\dot{V}_{21} = \begin{cases} -\bar{v}\frac{\left|\bar{\delta}_{12}(t)\right|-\varepsilon}{\varepsilon}\left|\bar{\delta}_{12}(t)\right| \leq -\bar{v}(\left|\bar{\delta}_{12}(t)\right| - \varepsilon), & \text{if } |\delta_{1f}(t)| < \varepsilon_f \\ \bar{\delta}_{12}^T(t)\bar{v}\frac{|\delta_{1f}(t)|-\varepsilon_f}{\varepsilon_f}\frac{\delta_{1f}(t)}{|\delta_{1f}(t)|} - \bar{v}\frac{\left|\bar{\delta}_{12}(t)\right|-\varepsilon}{\varepsilon}\left|\bar{\delta}_{12}(t)\right|, & \text{if } \varepsilon_f \leq |\delta_{1f}(t)| < 2\varepsilon_f \\ \bar{\delta}_{12}^T(t)\bar{v}\frac{\delta_{1f}(t)}{|\delta_{1f}(t)|} - \bar{v}\frac{\left|\bar{\delta}_{12}(t)\right|-\varepsilon}{\varepsilon}\left|\bar{\delta}_{12}(t)\right|. & \text{if } |\delta_{1f}(t)| \geq 2\varepsilon_f \end{cases} \tag{4.15}$$

From (4.15), when $|\delta_{1f}(t)| \geq 2\varepsilon_f$, we can not guarantee that $p_2(t)$ will enter the sphere $o(p_1, d_{12}+\varepsilon)$. But it is guaranteed from (4.14) that $p_2(t)$ is always bounded and remains in the sphere $o(p_1, d_{12} + 2\varepsilon)$. Following the analysis from subsection 4.4.1, once $|\delta_{1f}(t)|$ satisfies the condition $|\delta_{1f}(t)| < \varepsilon_f$, then it can be shown that $p_2(t)$ will enter the sphere $o(p_1, d_{12} + \varepsilon)$ with center p_1 and radii $d_{12} + \varepsilon$.

Next, we prove $p_2(t)$ converges to a fixed sphere $o(p_{2f}, \varepsilon_f)$. Consider the following Lyapunov function candidate

$$V_{22} = \frac{1}{2}\delta_{2f}^T(t)\delta_{2f}(t). \tag{4.16}$$

Taking the derivative of $V_{22}(t)$ along (4.6) gives

$$\begin{aligned} \dot{V}_{22} &= -\delta_{2f}^T(t)v_2(t) \\ &= -\bar{v}\beta_2(t)\delta_{2f}^T(t)\bar{\delta}_{12}(t)/\left|\bar{\delta}_{12}(t)\right| - \bar{v}\sqrt{1-\beta_2^2(t)}\bar{\beta}_2(t)\delta_{2f}^T(t)\bar{\delta}_{12}^*(t) \end{aligned} \tag{4.17}$$

Noting from the foregoing analysis that $p_2(t)$ will enter the sphere $o(p_1, d_{12} + \varepsilon)$, $\beta_2(t) = 0$. Then the equation (4.17) can be written as

$$\begin{aligned}\dot{V}_{22} =& -\bar{v}\bar{\beta}_2(t)\delta_{2f}^T(t)\bar{\delta}_{12}^*(t)\\ =& \begin{cases} 0, & \text{if } |\delta_{2f}(t)| < \varepsilon_f \\ -\bar{v}\frac{|\delta_{2f}(t)|-\varepsilon_f}{\varepsilon_f}\delta_{2f}^T(t)\bar{\delta}_{12}^*(t), & \text{if } \varepsilon_f \le |\delta_{2f}(t)| < 2\varepsilon_f \\ -\bar{v}\delta_{2f}^T(t)\bar{\delta}_{12}^*(t). & \text{if } |\delta_{2f}(t)| \ge 2\varepsilon_f \end{cases}\end{aligned} \tag{4.18}$$

According to the definition of $\bar{\delta}_{12}^*(t)$ in (4.8), it is easy to verify that $\delta_{2f}^T(t)\bar{\delta}_{12}^*(t) > 0$. Hence, from (4.18) and the definition of $\beta_2(t)$, we have $\dot{V}_{22} < 0$. Furthermore, $p_2(t)$ is always bounded and asymptotically converges to the sphere $o(p_{1f}, \varepsilon_f)$ with center p_{1f} and radii ε_f.

4.4.3 Stability and convergence for the 1-DOF agent

The 1-DOF agent A_3 uses the control law (4.6). Similar to the analysis of the stability for 2-DOF agent, the section proves that $p_3(t)$ remains bounded, and converges to $o(p_{3\hat{d}}, \varepsilon)$ with center $p_{3\hat{d}}$ and radii ε, and $p_3(t)$ converges to a fixed sphere $o(p_{3f}, \varepsilon_f)$.

First, let us consider the Lyapunov function candidate as follows

$$V_{31}(t) = \frac{1}{2}\delta_{3\hat{d}}^T(t)\delta_{3\hat{d}}(t). \tag{4.19}$$

Taking the derivative of $V_{31}(t)$ along (4.3) gives

$$\begin{aligned}\dot{V}_{31} =&\delta_{3\hat{d}}^T(t)\dot{\delta}_{3\hat{d}}(t)\\ =&\delta_{3\hat{d}}^T(t)(\dot{p}_{3\hat{d}}(t) - v_3(t))\\ =&\delta_{3\hat{d}}^T(t)(\dot{p}_{3\hat{d}}(t) - \bar{v}\beta_3(t)\delta_{3\hat{d}}(t)/\left|\delta_{3\hat{d}}(t)\right| - \bar{v}\sqrt{1-\beta_3^2(t)}\bar{\beta}_3\delta_{3\hat{d}}^*(t))\\ =&\delta_{3\hat{d}}^T(t)(\dot{p}_{3\hat{d}}(t) - \bar{v}\beta_3(t)\delta_{3\hat{d}}(t)/\left|\delta_{3\hat{d}}(t)\right|)\end{aligned} \tag{4.20}$$

where we have used the fact that $\delta_{3\hat{d}}(t)$ is perpendicular to $\delta_{3\hat{d}}^*(t)$. We note from the assumption 5 and the control laws (4.9) and (4.6) in section 4.3 that, $|\dot{p}_{3\hat{d}}| \le \bar{v}$ when $|\delta_{3\hat{d}}(t)| \ge 2\varepsilon$. Then we have $\dot{V}_{31} \le 0$. Therefore, when $|\delta_{3\hat{d}}(t)| \ge 2\varepsilon$, $\left|\delta_{3\hat{d}}^T(t)\right| \le \delta_{3\hat{d}}^T(0)$, $\forall t \ge 0$, that is, $p_3(t)$ is always bounded. In addition, according to the analysis of stability and convergence for 3-DOF and 2-DOF agents in subsections 4.4.1 and 4.4.2, $p_1(t)$ and $p_2(t)$ are always bounded and asymptotically converge to the spheres $o(p_{1f}, \varepsilon_f)$ and $o(p_{2f}, \varepsilon_f)$ respectively, i.e., $\dot{p}_{3\hat{d}}(t)$ will satisfy $\dot{p}_{3\hat{d}}(t) = 0$. It then follows that

$$\dot{V}_{31}(t) = \begin{cases} 0, & \text{if } \left|\delta_{3\hat{d}}(t)\right| < \varepsilon \\ -\overline{v}\frac{\left|\delta_{3\hat{d}}(t)\right|-\varepsilon}{\varepsilon}\delta_{3\hat{d}}(t)/\left|\delta_{3\hat{d}}(t)\right|, & \text{if } \varepsilon \leq \left|\delta_{3\hat{d}}(t)\right| < 2\varepsilon \\ -\overline{v}\delta_{3\hat{d}}(t)/\left|\delta_{3\hat{d}}(t)\right|. & \text{if } \left|\delta_{3\hat{d}}(t)\right| \geq 2\varepsilon \end{cases} \tag{4.21}$$

Hence, we have $\dot{V}_{31} \leq 0$. Furthermore, $p_3(t)$ is always bounded and asymptotically converges to the sphere $o(p_{3\hat{d}}, \varepsilon)$ with center $p_{3\hat{d}}$ and radii ε.

Next, we prove that $p_3(t)$ converges to a fixed sphere $o(p_{3f}, \varepsilon_f)$. Consider the Lyapunov function candidate as follows

$$V_{32} = \frac{1}{2}\delta_{3f}^T(t)\delta_{3f}(t). \tag{4.22}$$

Taking the derivative of V_{32} along (4.3) gives that

$$\begin{aligned} \dot{V}_{32} =& \delta_{3f}^T(t)\dot{\delta}_{3f}(t) \\ =& -\delta_{3f}^T(t)v_3(t) \\ =& -\overline{v}\beta_3(t)\delta_{3f}^T(t)\delta_{3\hat{d}}(t)/\left|\delta_{3\hat{d}}(t)\right| - \overline{v}\sqrt{1-\beta_3^2(t)}\bar{\beta}_3\delta_{3f}^T(t)\delta_{3\hat{d}}^*(t) \end{aligned} \tag{4.23}$$

Noting that $p_3(t)$ asymptotically converges to the sphere $o(p_{3\hat{d}}, \varepsilon)$, then $\beta_3(t) = 0$. Hence, the equation (4.23) can be written as

$$\begin{aligned} \dot{V}_{32}(t) =& -\overline{v}\bar{\beta}_3(t)\delta_{3f}^T(t)\delta_{3\hat{d}}^*(t) \\ =& \begin{cases} 0, & \text{if } \left|\delta_{3f}(t)\right| < \varepsilon_f \\ -\overline{v}\frac{\left|\delta_{3f}(t)\right|-\varepsilon_f}{\varepsilon_f}\delta_{3f}^T(t)\delta_{3\hat{d}}^*(t), & \text{if } \varepsilon_f \leq \left|\delta_{3f}(t)\right| < 2\varepsilon_f \\ -\overline{v}\delta_{3f}^T(t)\delta_{3\hat{d}}^*(t). & \text{if } \left|\delta_{3f}(t)\right| \geq 2\varepsilon_f \end{cases} \end{aligned} \tag{4.24}$$

According to the definition of $\delta_{3\hat{d}}^*(t)$ in (4.5), it is easy to verify that $\delta_{3f}^T(t)\delta_{3\hat{d}}^*(t) > 0$. Hence, from (4.24), we have $\dot{V}_{32} \leq 0$. Furthermore, $p_3(t)$ is always bounded and asymptotically converges to the sphere $o(p_{3f}, \varepsilon_f)$ with center p_{3f} and radii ε_f.

4.4.4 Stability and convergence for the 0-DOF agent

The 0-DOF agent A_i uses the control law (4.2), where $i = 4, ..., m$. It can use its full velocity capacity to move towards the point $p_{id}(t)$. In this section we prove that $p_i(t)$ is always bounded and asymptotically converges to the sphere $o(p_{id}(t), \varepsilon)$ with center $p_{id}(t)$ and radii ε. Because the proofs for the stabilities of other 0-DOF agents are similar to that of A_4, we only consider the stability of the agent A_4 here, and the proofs for the stabilities of other 0-DOF agents are omitted. Consider the Lyapunov function candidate as

$$V_{4d} = \frac{1}{2}\delta_{4d}^T(t)\delta_{4d}(t). \tag{4.25}$$

Then the derivative of $V_{4d}(t)$ along (4.2) is given by

$$\begin{aligned} \dot{V}_{4d} =& \delta_{4d}^T(t)\dot{\delta}_{4d}(t) \\ =& \delta_{4d}^T(t)(\dot{p}_{4d} - v_4) \\ =& \delta_{4d}^T(t)(\dot{p}_{4d} - \bar{v}\beta_4(t)\delta_{4d}(t)/\left|\delta_{4d}(t)\right|). \end{aligned} \tag{4.26}$$

Noting from the assumption 5 and the control laws (4.3), (4.6) and (4.9) in section 4.3 that $|\dot{p}_{4d}| \leq \bar{v}$ when $|\delta_{4d}(t)| \geq 2\varepsilon$. Then we have $\dot{V}_{4d} \leq 0$. Therefore, when $|\delta_{4d}(t)| \geq 2\varepsilon$, we have $|\delta_{4d}(t)| \leq |\delta_{4d}(0)|$, that is, $p_4(t)$ is always bounded. In addition, according to the analysis of stability and convergence for 3-DOF, 2-DOF and 1-DOF agents in subsections 4.4.1, 4.4.2 and 4.4.3 , $p_1(t)$, $p_2(t)$ and $p_3(t)$ are always bounded and asymptotically converge to the spheres $o(p_{1f}, \varepsilon_f)$, $o(p_{2f}, \varepsilon_f)$ and $o(p_{3f}, \varepsilon_f)$ respectively, i.e., $\dot{p}_{4d}$ will satisfy $|\dot{p}_{4d}| = 0$. It then follows that

$$\begin{aligned} \dot{V}_{4d} =& -\bar{v}\beta_4(t)\left|\delta_{4d}(t)\right| \\ =& \begin{cases} 0, & \text{if } |\delta_{4d}(t)| < \varepsilon \\ -\bar{v}\frac{|\delta_{4d}(t)|-\varepsilon}{\varepsilon}\left|\delta_{4d}(t)\right| \leq -\bar{v}(|\delta_{4d}(t)| - \varepsilon), & \text{if } \varepsilon \leq |\delta_{4d}(t)| < 2\varepsilon \\ -\bar{v}\left|\delta_{4d}(t)\right|. & \text{if } |\delta_{4d}(t)| \geq 2\varepsilon \end{cases} \end{aligned} \tag{4.27}$$

Hence, we have $\dot{V}_{4d} \leq 0$. Furthermore, $p_4(t)$ is always bounded and asymptotically converges to the sphere $o(p_{id}(t), \varepsilon)$ with center $p_{id}(t)$ and radii ε.

Combining all these results in subsections 4.4.1, 4.4.2, 4.4.3, and 4.4.4, the stability and convergence of the entire formation to the destination can be obtained.

4.5 Simulations

4.5.1 Verification of the effectiveness of control laws

Assume that the desired distance between each neighbor agent pair (A_i, A_j) is $d_{ij} = 1m$ and choose $\bar{v} = 1m/s$, and $\varepsilon_k = 0.02$. Figs.4.9 and 4.10 show the path and inner-agent distances, during the motion of a formation with 5 agents. From Figs. 4.9 and 4.10, the effectiveness of the control law is demonstrated.

4.5.2 Verification of the effectiveness of the framework for planning and control of formation

In the simulation models, each obstacle is considered as a cube or a shape consisting of several cubes. The red area denotes the obstacles, the yellow area denotes the obstacle regions. The path followed by the center of mass of the formation can only be chosen through white grid region. The agents can move through yellow and white grid region. From Fig.4.11, it can be seen that the path followed by the center of mass of the formation between the start position and the goal position is found by using the A^* search algorithm coupled with an optimizing algorithm. From Fig.4.12, it can be seen that a persistent formation can move continuously along the optimum path. Similarly, when the obstacles are very complex, as shown in Figs.4.13 and 4.14, the formation can still find the optimized path to the desired position.

The effectiveness of the proposed mechanisms for planning and control of the 3-dimensional multi-agent formation is demonstrated via the simulation results.

4.6 Conclusion

In this chapter, we have given a practicable framework for planning and control for 3-dimensional multi-agent formations. Firstly, a 3-dimensional space is divided into cubic grid cells, then a grid map is generated. We also use the A* search algorithm to identify the preliminary path for the formation motion. Furthermore, an optimizing algorithm is operated to generate a set of waypoints which were extracted from this preliminary path. Then a collision-free and deadlock-free feasible path is obtained. About the control of formation, we propose a set of decentralized control laws for the cohesive motion of 3-dimensional multi-agent formations based on point-agent system model. It can guarantee that the formation moves as a large entity, i.e. the shape of the formation remains invariable during the continuous motion, which can avoid inter-agent collision. The effectiveness of the proposed framework is demonstrated via simulation results.

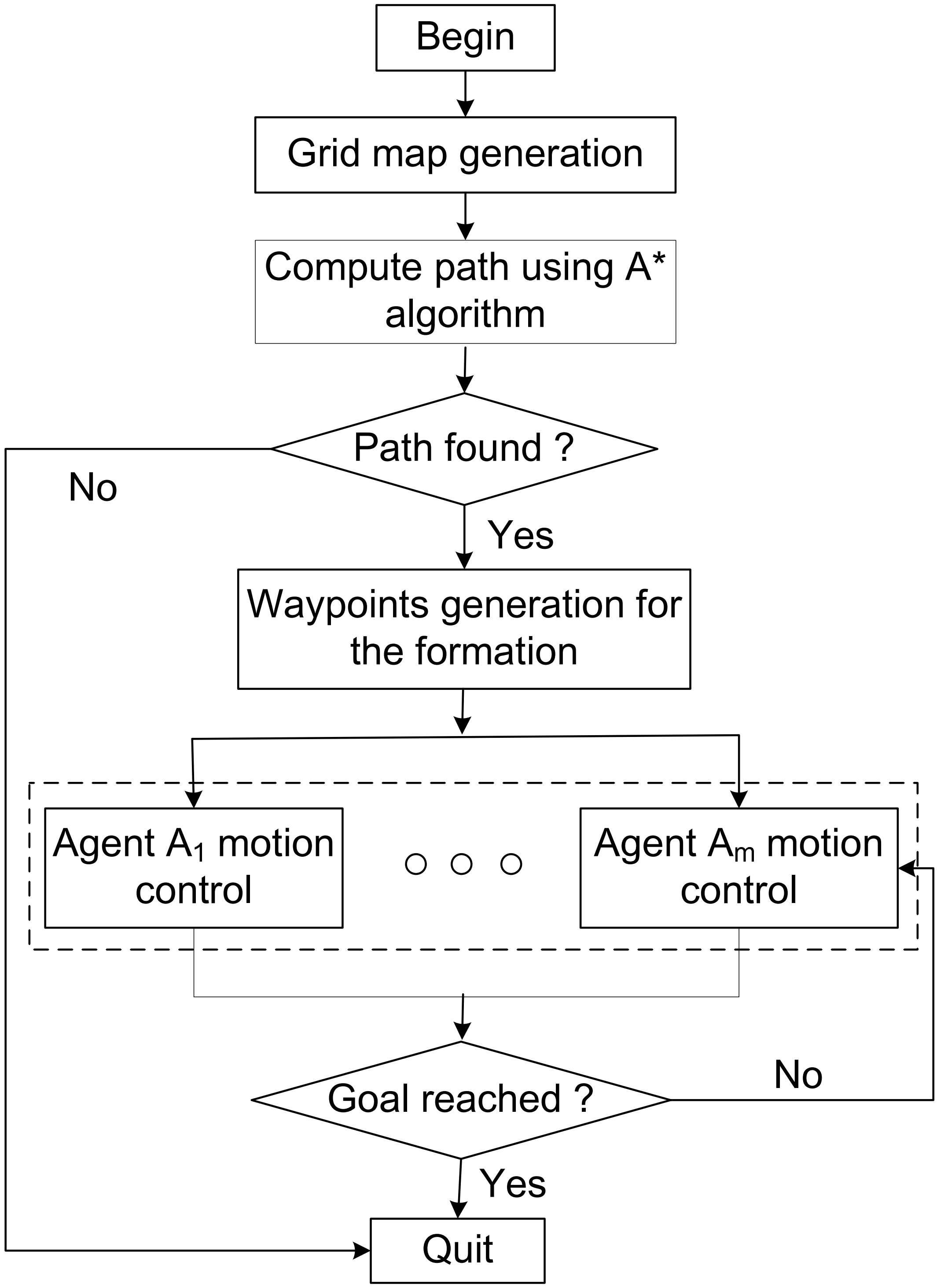

Figure 4.6: Flow chart of formation planning and control

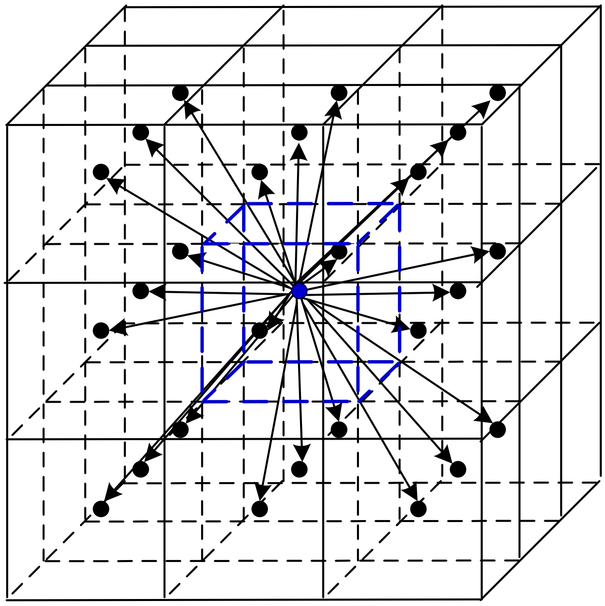

Figure 4.7: Twenty-six possible orientations for the next point to move in path sequence for the given formation.

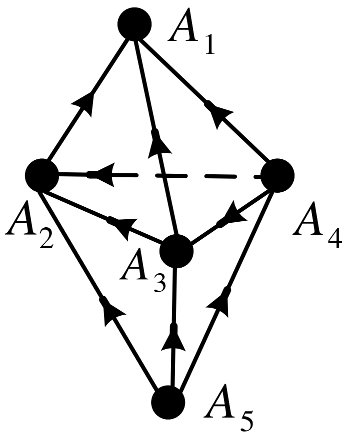

Figure 4.8: A minimally persistent formation with 5 agents in R^3.

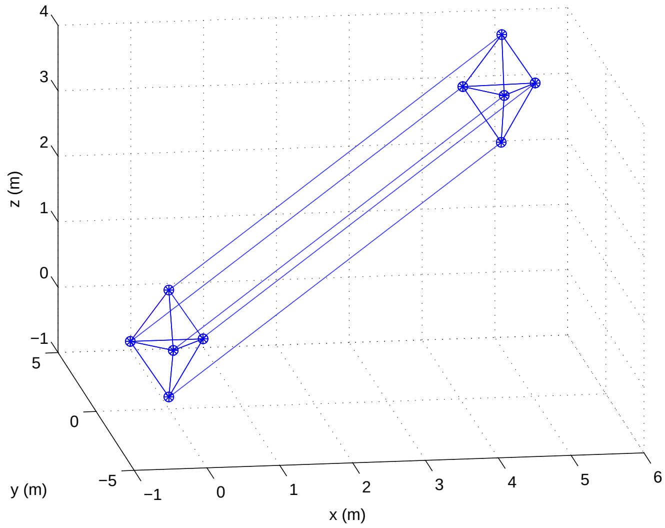

Figure 4.9: The path for the motion of a formation with 5 agents.

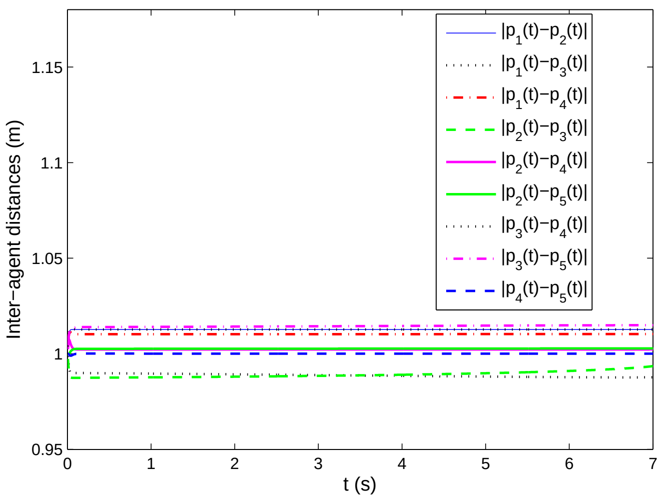

Figure 4.10: Inner-agent distances during motion.

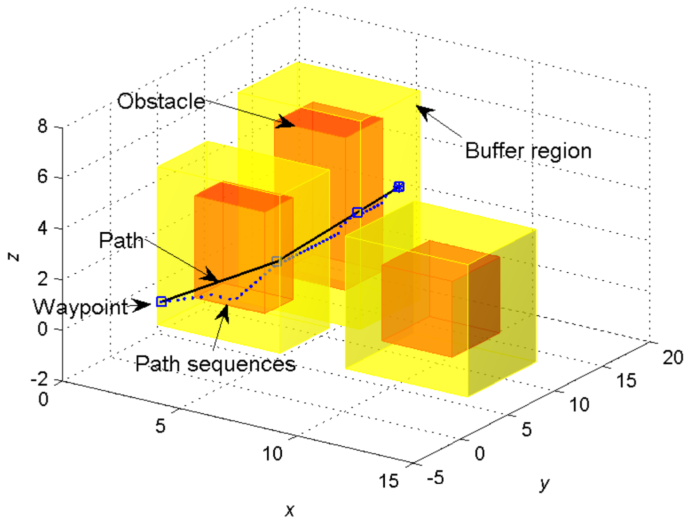

Figure 4.11: The path followed the center of mass of the formation when the obstacles are simple.

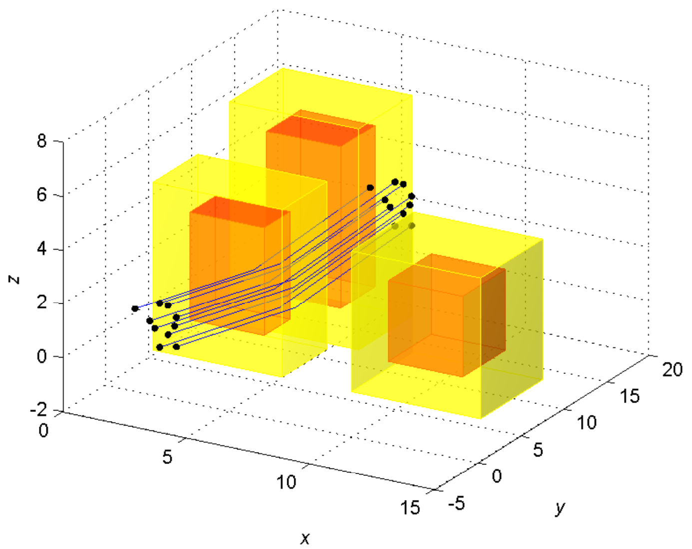

Figure 4.12: The motion path of a 3-dimensional Multi-agent Formation when the obstacles are simple.

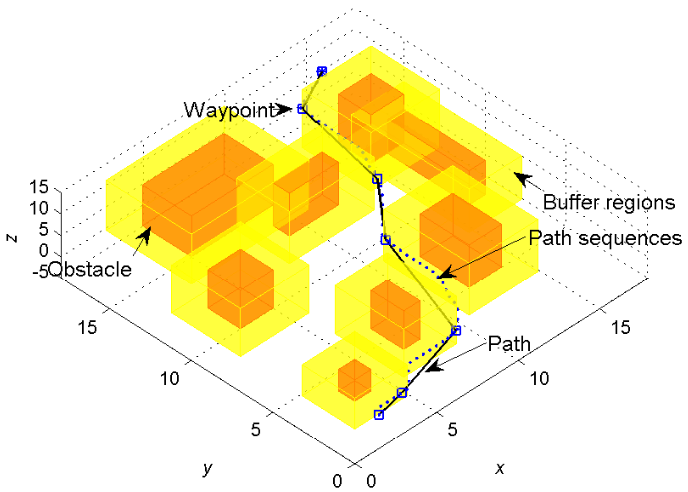

Figure 4.13: The path followed the center of mass of the formation when the obstacles are more complex.

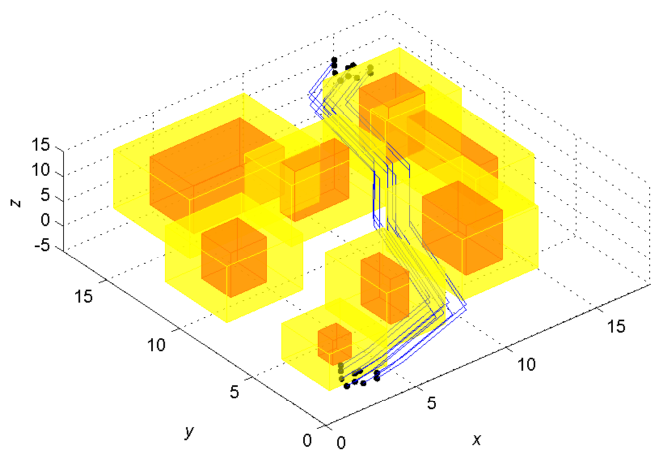

Figure 4.14: The motion path of a 3-dimensional Multi-agent Formation when the obstacles are more complex.

Chapter 5

Rigid Formation Keeping in Multi-Agent systems

Contents

5.1 Preliminaries and problem statement 120

5.1.1 Preliminaries 120

5.1.2 Problem statement 121

5.2 Rigidity recovery based on two self-repair operations 122

5.2.1 The self-repair operations 124

5.2.2 Implementation algorithms 126

5.3 Rigidity recovery based on edge contraction operation 129

5.3.1 Edge contraction operation 129

5.3.2 A sufficient and necessary condition for an edge to be contractible 130

5.3.3 Minimal rigidity preserving problem 133

5.4 Conclusion 139

In this chapter, we consider the formation keeping problem in multi-agent systems. We mainly fucus on the closing ranks problem which deals with the addition of links to a rigid multi-agent formation that is "damaged" by losing one of its agents, in order to recover rigidity. We model the information architecture of formation as a graph, where each vertex represents an agent in formation, and each edge represents a communication link between a pair of agents. Firstly, we prove that if an agent is lost, we only need to add a set of new edges on its neighbor vertices such that the subgraph which is induced by neighbor vertices of the lost vertex is minimally rigid, and then the resulting graph is rigid. Using this result, we propose two systematic 'self-repair' operations to recover the rigidity in case of agent removals, named Triangle Patch and Vertex Addition Patch. Considering the requirement of practical implementation, we give two algorithms corresponding the above two operations. As these two operations cannot guarantee that the result graph is minimally rigid. Next, we consider the closing ranks based on another graph operation named edge contraction. First, we propose a sufficient and necessary condition for the case that an edge of a minimally rigid graph is contractible: an edge in a minimally rigid graph is contractible if and only if there are no two minimally rigid subgraphs such that their intersection consists of the contraction edge and its two incident vertices. Later, an Edge-Replacement Principle is proposed. Based on Edge-Replacement Principle and edge contraction operation, the minimally rigidity preserving problem is studied. A set of graph theoretical results operations are established to solve the corresponding closing ranks problems.

5.1 Preliminaries and problem statement

5.1.1 Preliminaries

The next theorems give an inductive approach that creates (minimal) rigid graphs in two-dimensional space.

Theorem 5.1 *(Vertex Addition Theorem; Tay & Whiteley (1985,?)) Let $G = (V, E)$ in R^2 of $|V|$ vertices and $|E|$ edges be a graph with a vertex i of degree 2; let $G^* = (V^*, E^*)$ denote the subgraph obtained by deleting i and the edges incident with it. Then G is (minimal) rigid if and only if G^* is (minimal) rigid.*

Theorem 5.2 *(Edge Split Theorem ; Tay & Whiteley (1985)). Let $G = (V, E)$ in R^2 of $|V|$ vertices and $|E|$ edges be a graph with a vertex i of degree 3; let V_i be the set of vertices incident to i, and let $G^* = (V^*, E^*)$ denote the subgraph obtained by deleting i and its 3 incident edges. Then G is (minimal) rigid if and* *only if there is a pair j, k of vertices of V_i such that the edge (j, k) is not in E^* and the graph $G' = (V^*, E^* \cup (j, k))$ is (minimal) rigid.*

According to above results, all minimally rigid graphs can be constructed by performing a series of vertex addition or edge splitting operations on K_2 (the complete graph on 2 vertices with only 1 edge). Both operations add a new vertex i to an existing graph. These two operations are defined as follows.

***Vertex Addition Operation*:** Let j, k be two distinct vertices of a minimally rigid graph $G = (V, E)$. A vertex addition operation consists in adding a vertex i , and connecting it to j and k as shown in Fig.5.1(a).

Edge Splitting Operations: Let j, k, l be three vertices of a minimally rigid graph such that there is an edge between j and k. An edge splitting operation consists in removing this edge, adding a vertex i and connecting it to j, k and l, as shown in Fig.5.1(b).

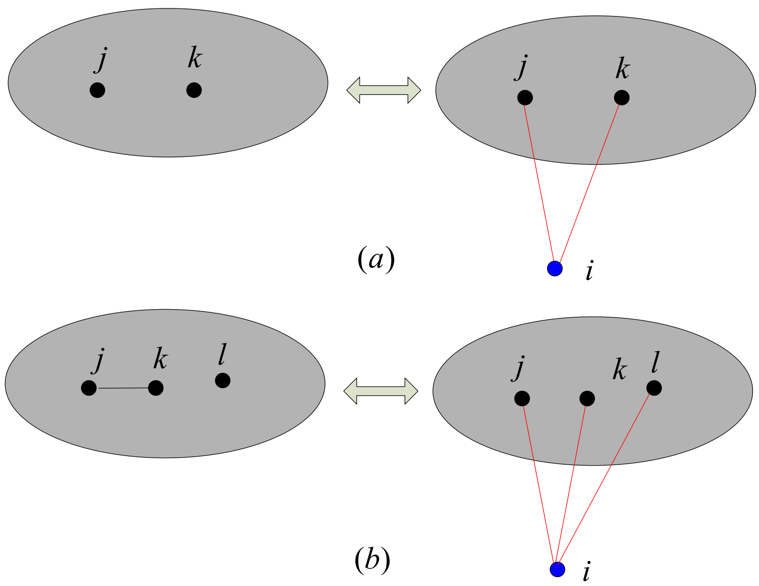

Figure 5.1: Representation of (a) vertex addition operation and reverse vertex addition operation (b) edge splitting operation and reverse edge splitting operation

5.1.2 Problem statement

In this chapter, the closing ranks problem will be discussed, which mainly deals with the addition of links to a rigid formation that is “damaged" by losing one of its agents, in order to recover its rigidity. The formal definition of (minimally) rigidity preserving closing ranks problem can be formulated in terms of its underlying graph G as follows.

Suppose that $G = (V, E)$ is a (minimally) rigid in real 2-dimensional plane. Let i be a vertex in V with a neighbor vertices set V_i (the vertices set incident with i in V) and an incident edges set E_i. Let $G^* = (V^*, E^*)$ be a graph obtained from G by removing vertex i and edges set E_i, i.e., $V^* = V\backslash\{i\}$ and $E^* = E\backslash E_i$. The (minimally) rigidity preserving closing ranks problem is to find the set of new edges E_{new} between the vertices in V_i so that the resulting graph $G' = (V^*, E^* \cup E_{new})$ is (minimally) rigid.

In this chapter, our task is to propose the systematic self-repair operations for creating the minimum number of new links to recover the rigidity(minimally rigidity) of a formation in the case of agent loss.

5.2 Rigidity recovery based on two self-repair operations

In Eren *et al.* (2004), the following existence and necessary lower bound results are established for (number of) new edges to be added in (minimally) rigidity preserving closing ranks problem.

Lemma 5.3 *Let $G = (V, E)$ in R^2 be a minimally rigid graph with a vertex i of degree k. Then there exists at least one set of $k-2$ edges among the neighbors of i such that a minimally rigid graph is obtained by removing i and its incident edges and adding the set of $k-2$ edges.*

Lemma 5.3 provides existence results and the minimal number of edges needed to recover minimal rigidity, and it also tells that it is sufficient to add edges only between neighbors of the lost agent to recover minimal rigidity. However, it is difficult to select the $k-2$ edges, the lemma does not tell which edges are to be added among the neighbors of the removed vertex i. If only based on Lemma 5.3 to design an algorithm for recovering minimal rigidity, it would require a search over all possible sets of $k-2$ edges between neighbors of the removed vertex i for checking minimal rigidity of the graph obtained by insertion of each of these edge sets. If we drop the requirement for minimality, one straightforward solution to the closing ranks problem is a complete cover: simply add every possible edge between neighbor vertices of the lost vertex. Thus, $k(k-1)/2$ edges are required to add.

The following lemma is a substitution principle for rigid graphs(Whiteley (1984)):

Lemma 5.4 *Given a rigid graph $G = (V, E)$ in d-dimensional space, if for any vertex subset $\bar{V}$ the induced subgraph $\bar{G} = (\bar{V}, \bar{E})$ of G is replaced with a minimally rigid graph $G^* = (\bar{V}, E^*)$ on vertices set $\bar{V}$, then the modified graph $G' = (V, E')$ is also a rigid graph in d-dimensional space, where $E' = (E/\bar{E}) \cup E^*$.*

By Lemma 5.3 and Lemma 5.4, the following results can be obtained.

Theorem 5.5 *Let $G = (V, E)$ in R^2 be a (minimally) rigid graph, i be a vertex in V with a neighbor vertices set V_i and an incident edges set E_i, and $G^* = (V^*, E^*)$ denote the subgraph obtained by deleting i and the edges set E_i incident with it. If we can add a new edges set E_{new} on vertices set V_i, such that the induced subgraph $\tilde{G} = (V_i, \tilde{E})$ of G^* is a minimally rigid graph, then the resulting graph $G' = (V^*, E^* \cup E_{new})$ is rigid graph.*

Proof: Let $\bar{G} = (V_i, \bar{E})$ denote the induced subgraph of G^* . Without loss of generality, we suppose the degrees of vertex i and k , and then by Lemma 5.3, there exists at least one set of $k-2$ edges among the neighbors of i such that a minimally rigid graph is obtained by removing i and its incident edges and adding the set of $k-2$ edges. Let the set of $k-2$ edges be $\hat{E}$, the minimally rigid graph be $\hat{G} = (V^*, E^* \cup \hat{E})$, and let the induced subgraph of $\hat{G}$ on vertices set V_i be $G'' = (V_i, \bar{E} \cup \hat{E})$. In graph G^* , we can add a new edge set E_{new} on vertices of V_i, such that the induced subgraph $\tilde{G} = (V_i, \bar{E} \cup E_{new})$ of G^* on vertices set V_i is a minimally rigid graph. By Lemma 5.4**??**, we can replace $G'' = (V_i, \bar{E} \cup \hat{E})$ with a minimally rigid graph $\tilde{G} = (V_i, \bar{E} \cup E_{new})$, and then the resulting graph $G' = (V^*, E^* \cup E_{new})$ is rigid.

Theorem 5.5 shows that, in order to recover rigidity of formation in case of agent loss, we need to only add a set of edges on its neighbor vertices such that the subgraph which is induced by neighbor vertices of the lost vertex is minimally rigid, and then the resulting graph is rigid.

5.2.1 The self-repair operations

Based on Theorem 5.5, we will propose two systematic self-repair operations to recover the rigidity in case of losing one agent for a given rigid formation.

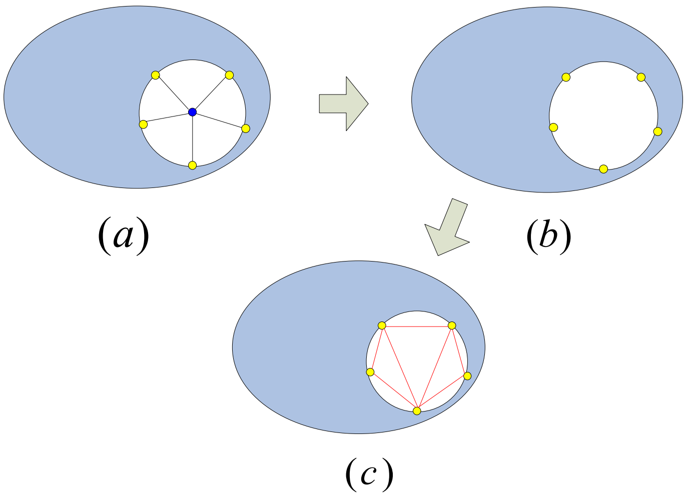

Figure 5.2: Illustration of Triangle Patch: (a) the initial rigid formation, (b) the remaining formation after an agent and its incident edges loss, (c) the recovering formation using Triangle Patch operation.

Triangle Patch: Create a cycle among the neighbors of the lost agent. Then choose arbitrarily some agent of neighbor agents to connect it with each of other neighbors of the lost agent (see Fig.5.2).

Vertex Addition Patch: Choose arbitrarily two agents among the neighbors of the lost agent and connect them, and then for each remaining agent, use the vertex addition operation one by one(see Fig.5.3).

Remark1: In the above operations, when a vertex of degree k is lost, we add $2k-3$ edges using the minimally rigid patch. Although the number of added edges is more than $k-2$, it is still linear in the lost vertex degree, as opposed to quadratic for a complete cover. In fact, one could utilize existing edges among neighbors of the lost vertex, which we call existing cover edges, to minimize the number of new added edges. Simple counting arguments can be used to show that there could be up to $k-1$ existing cover edges. Thus, there are certain scenarios in which one adds $(2k-3)-(k-1) = k-2$ new edges, utilizing the existing cover edges to obtain a minimal cover. This scenarios are illustrated in Fig. 5.4 and Fig. 5.5.

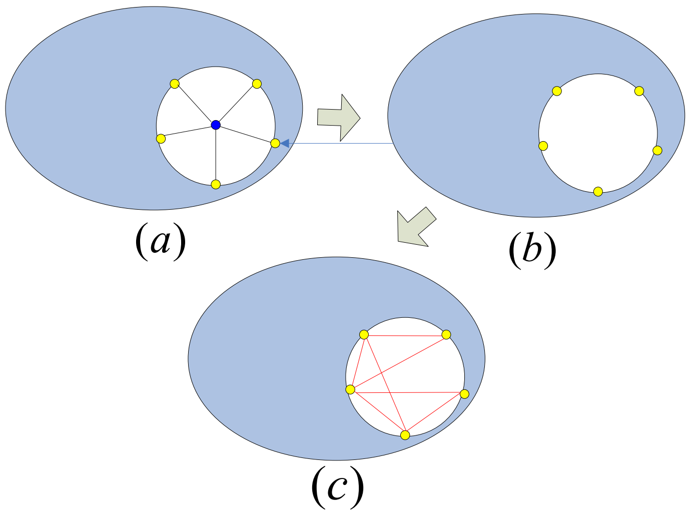

Figure 5.3: Illustration of Vertex Addition Patch: (a) the initial rigid formation, (b) the remaining formation after an agent and its incident edges loss, (c) the recovering formation using Vertex Addition Patch operation.

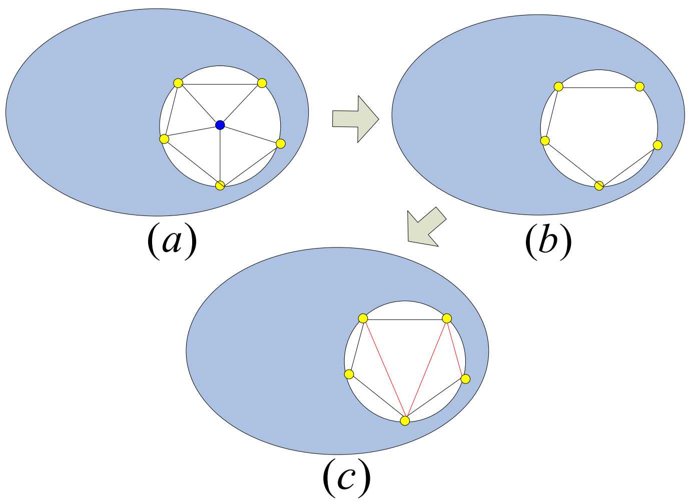

Figure 5.4: Utilizing existing cover edges with the Triangle Patch: (a) the lost vertex has degree 5, (b) edges exist among neighbor vertices of the lost vertex, (c) coordinators for the Triangle Patch are chosen to utilize 4 existing edges, and 3 new edges are added.

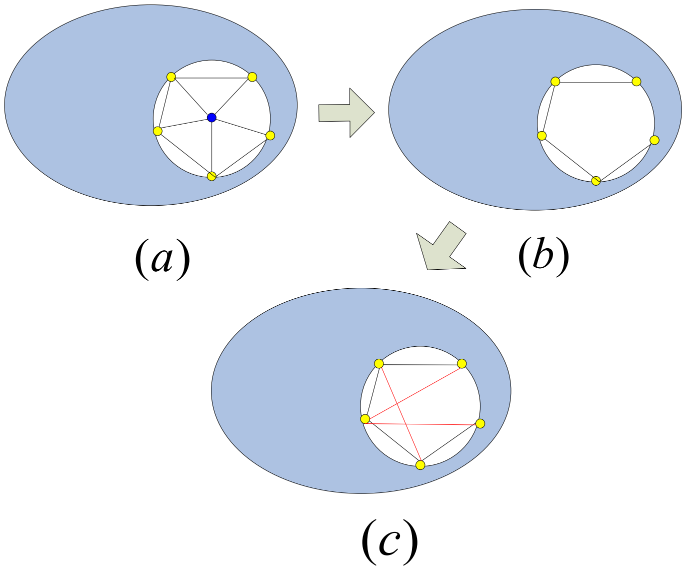

Figure 5.5: Utilizing existing cover edges with the Vertex Addition Patch: (a) the lost vertex has degree 5, (b) edges exist among neighbor vertices of the lost vertex, (c) coordinators for the Vertex Addition Patch are chosen to utilize 4 existing edges, and 3 new edges are added.

5.2.2 Implementation algorithms

Considering the implementation requirement of the operations of Triangle Patch and Vertex Addition Patch, several assumptions are given as followed.

1. Each agent has a unique identification (ID). This is necessary so that the agents can distinguish other when determining which new links to add.

2. Each agent knows all of its neighbor vertices and 2-hop neighbor vertices.

Based on the above assumptions, two algorithms are given to solve the closing ranks problem, in which it is supposed that the degree of lost agent is N. The two algorithms are illustrated in Fig. 5.6 and Fig. 5.7.

Remark In this section, we give two operations. Note that these two operations can guarantee that the resultant graph is minimally rigid. Next, we consider the closing ranks problems based on another graph operation named edge contraction which is proposed by Fidan *et al.* (2010). The term ”edge contraction“ will be introduced in the next section.

Algorithm 1: Triangle Patch

if a agent and its incident edges lost **then**
    for neighbor vertices of lost agent **do**
        give each neighbor vertex a id number ID ($ID \in \{1, 2, 3, \dots, N\}$) ;

Figure 5.6: Illustration of algorithm for triangle patch.

Algorithm 2: Vertex Addition Patch

Figure 5.7: Illustration of algorithm for vertex addition patch.

5.3 Rigidity recovery based on edge contraction operation

5.3.1 Edge contraction operation

Let v_1, v_2 be two adjacent vertices in the graph $G = (V, E)$. The edge contraction operation consists in merging these two vertices v_1, v_2 into one vertex which is adjacent with all such vertices in V that v_1 and v_2 were adjacent to, and deleting the edge $e = (v_1, v_2)$, and finally merging the double edges into one edge, if there exist any.

The graph obtained from G by using edge contraction operation on edge e is denoted by $G_c(e)$, and the edge e is named contraction edge. We say that $e = (v_1, v_2)$ is contractible in a minimally rigid graph G if $G_c(e)$ is rigid. From Fig.5.8(a), we can see that the vertex v is removed in a minimally rigid graph, and then we find new edges $(v_1, v_3), (v_1, v_4), (v_1, v_5)$ using edge contraction operation on the edge (v, v_1). Hence the resulting graph is rigid, i. e. the edge (v, v_1) is contractible. However, it is seen from Fig.5.8(b) that if we choose the edge (v, v_2) as the contraction edge, then the resulting graph is not rigid, i. e. the edge (v, v_2) is not contractible.

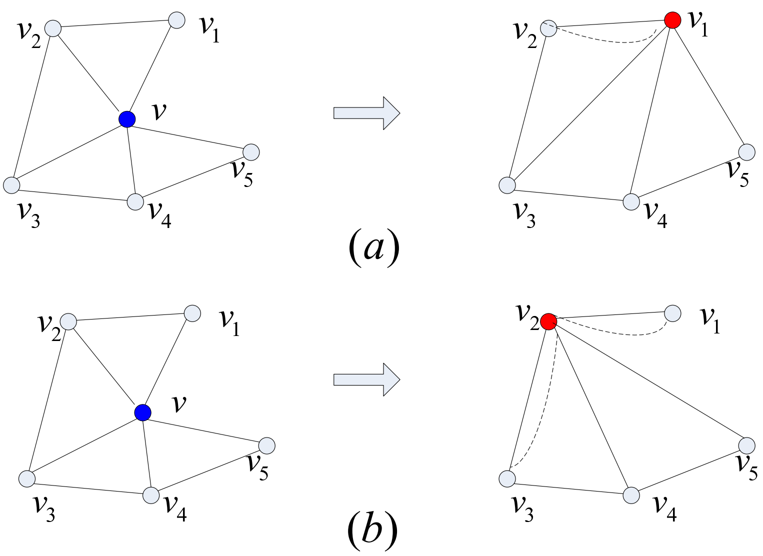

Figure 5.8: Closing ranks in a rigid formation using the edge contraction operation, where the dashed lines indicate deleted double edges

Theorem 5.6 *(Fidan et al. (2010)) Any vertex of a minimally rigid graph with at least three vertices is incident to at least two contractible edges.*

This theorem proposes the existence of contractible edges incident to the lost vertex. However it doesn't tell which edge is contractible, and how to find such a contractible edge. In the following section, we will give a sufficient and necessary condition for an edge of a minimally rigid graph to be contractible.

5.3.2 A sufficient and necessary condition for an edge to be contractible

Theorem 5.7 *Let u be a vertex in a minimally rigid graph $G = (V, E)$, v be a neighbor of u. The edge $e = (u, v)$ is contractible if and only if there don't exist two minimally rigid subgraphs $G^{'} = (V^{'}, E^{'})$ and $G^{''} = (V^{''}, E^{''})$ such that $V^{'} \cap V^{''} = \{u, v\}$, $E^{'} \cap E^{''} = \{e\} = \{(u, v)\}$, $\left|V^{'}\right| \geq 3$ and $\left|V^{''}\right| \geq 3$.*

In order to facilitate the understanding of the readers, the sufficient and necessary directions of Theorem 5.6 are dealt in the next two Propositions 5.8 and 5.9 respectively.

Proposition 5.8 *Let u be a vertex in a minimally rigid graph $G = (V, E)$, v be a neighbor of u. If the edge $e = (u, v)$ is contractible, then there don't exist two minimally rigid subgraphs $G^{'} = (V^{'}, E^{'})$ and $G^{''} = (V^{''}, E^{''})$ such that $V^{'} \cap V^{''} = \{u, v\}$, $E^{'} \cap E^{''} = \{e\} = \{(u, v)\}$, $\left|V^{'}\right| \geq 3$ and $\left|V^{''}\right| \geq 3$.*

Proof: Suppose, to the contrary, there exist two minimally rigid subgraphs $G^{'} = (V^{'}, E^{'})$ and $G^{''} = (V^{''}, E^{''})$ such that $V^{'} \cap V^{''} = \{u, v\}$, $E^{'} \cap E^{''} = \{e\} = \{(u, v)\}$, $\left|V^{'}\right| \geq 3$ and $\left|V^{''}\right| \geq 3$. It follows from Theorem 4.2 that $\left|E^{'}\right| = 2\left|V^{'}\right| - 3$ and $\left|E^{''}\right| = 2\left|V^{''}\right| - 3$. Let n_{uv} be the number of common neighbors of u and v. Using the edge contraction operation on e, the number of edges in G will decrease by $n_{uv} + 1$. Hence in order to prove the sufficient part, we consider the following two cases respectively: (1) there exists no common neighbor of u and v in both $G^{'}$ and $G^{''}$, (2) there exists at least one common neighbor of u and v in either $G^{'}$ or $G^{''}$.

Case (1): there exists no common neighbor of u and v in both $G^{'}$ and $G^{''}$. Since $V^{'} \cap V^{''} = \{u, v\}$, $E^{'} \cap E^{''} = \{e\} = \{(u, v)\}$, by using the edge contraction operation on the edge e, the number of vertices and edges in the graph G decrease by one respectively. Moreover, in subgraphs $G^{'}$ and $G^{''}$, the number of vertices and edges also decrease by one respectively. Let $G_c(e)$ be the graph obtained by using the edge contraction operation on the edge e. Suppose that $G_c(e)$ is rigid. By Theorems 4.1 and 4.2, there exists a minimally rigid subgraph G^* with all vertices of $G_c(e)$, such that any subgraph of G^* satisfies the statement of Theorem 4.1(ii). Since $G^{'}$ and $G^{''}$ are supposed to be minimally rigid in G, we need remove at least two edges from $G^{'}_c(e)$ and $G^{''}_c(e)$ to obtain the minimally rigid subgraph G^*. Thus the subgraph G^* has at most $|E|-3$ edges. But on the other hand, since G^* has $|V|-1$ vertices, by Theorem 4.2, the number of edges of G^* should be $|E|-2$. This gives a contradiction. Thus there exists no such a minimally rigid subgraph G^* with all vertices of $G_c(e)$.

Case (2): there exists at least one common neighbor of u and v in either G' or G''. If there exist two more common neighbors of u and v in either G' or G'', by using edge contraction operation on the edge e, the number of vertices in the graph G decreases by one, and the number of edges in the graph G decreases at least by three. This obviously contradicts to Theorems 4.1 and 4.2.

Without loss of generality, suppose that there exists a unique common neighbor of u and v in G'. By using edge contraction operation on the edge e, the number of vertices in the graph G decreases by one, and the number of edges in the graph G decreases by two. In subgraph G'', the number of vertices and edges decrease by one respectively. If the graph $G_c(e)$ is rigid, by Theorem 4.1, there exists a minimally rigid subgraph G^* with all the vertices of $G_c(e)$, such that any subgraph of G^* satisfies the statement of Theorem 4.1(ii). So at least one edge needs to be removed from $G^{''}_c(e)$. This is impossible, since the number of edges and vertices in G^* don't hold the equality in Theorem 4.2. Thus there does not exists such a minimally rigid subgraph G^* with all the vertices of $G_c(e)$.

Therefore, we conclude that the graph $G_c(e)$ is not rigid, i.e. the edge $e = (u, v)$ is not contractible.

Proposition 5.9 *Let u a vertex in a minimally rigid graph $G = (V, E)$, v be a neighbor of u. Suppose there don't exist two minimally rigid subgraphs $G' = (V', E')$ and $G^{''} = (V^{''}, E^{''})$ such that $V' \cap V^{''} = \{u, v\}$, $E' \cap e^{''} = \{e\} = \{(u, v)\}$, $|V'| \geq 3$ and $|V^{''}| \geq 3$. Then the edge $e = (u, v)$ is contractible.*

Proof: By way of contradiction, we prove if the edge $e = (u, v)$ is not contractible, then there exist two minimally rigid subgraphs $G' = (V', E')$ and $G'' = (V'', E'')$ such that $V' \cap V'' = \{u, v\}$, $E' \cap E'' = \{e\} = \{(u, v)\}$, $|V'| \geq 3$ and $|V''| \geq 3$.

Suppose the edge $e = (u, v)$ is not contractible. Then $G_c(e) = (V_c, E_c)$ is not rigid. So there exists no minimally rigid subgraph containing all vertices of $G_c(e)$. In the following, our proof is divided into three cases.

Case (1). If u and v have at least two common neighbors in G, then obviously there exist two triangles denoted by $G' = (V', E')$ and $G'' = (V'', E'')$, which are minimally rigid and satisfy the conditions $V' \cap V'' = \{u, v\}$, $E' \cap E'' = \{e\} = \{(u, v)\}$, $|V'| = 3$ and $|V''| = 3$.

Case (2). We now consider the case where u and v have the unique common neighbor w in G. Then we can take G' as the triangle with verticesu, v and w. In the following we will find the other rigid graph G'' containing vertices v and u, but not containing their common neighbor w. By the edge contraction operation on the edge e, we have $|E_c| = 2|V_c| - 3$. Since $G_c(e)$ is not rigid, there is a non-empty subgraph $\bar{G} = (\bar{V}, \bar{E})$ in $G_c(e)$ which holds $|\bar{E}| \geq 2|\bar{V}| - 2$, where $\bar{V}$ is the set of vertices incident to edges of $\bar{E}$.

Let v^* denote the vertex of $G_c(e)$ obtained by contracting e. Now we claim that$v^* \in \bar{V}, w \notin \bar{V}$ and $|\bar{V}| \geq 3$. In fact, if $v^* \notin \bar{V}$, then the subgraph $\bar{G} = (\bar{V}, \bar{E})$ is also a subgraph of G, and holds $|\bar{E}| \geq 2|\bar{V}| - 2$. This gives a contradiction to Theorem 4.2(ii). Thus we have $v^* \in \bar{V}$. If $w \in \bar{V}$, then there exist a subgraph in G, constructed as $G'' = (V'', E'') = ((\bar{V} - v^*) \cup \{u, v\}, E'')$ such that $|E''| = |\bar{E}| + 2 \geq 2|\bar{V}| - 2 + 2 = 2|\bar{V}| = 2(|V''| - 1) = 2|V''| - 2$. This also contradicts to Theorem 4.2(ii). So our claim is proved.

Now we construct the subgraph $G'' = (V'', E'') = ((\bar{V} - v^*) \cup \{u, v\}, E'')$ in G. By the above claim, it is easy to verify that $|E''| = 2|V''| - 3$. Hence by Theorem 4.2, the just constructed subgraph G'' is minimally rigid.

Case (3). If there exists no common neighbor of u and v in G, then we will find two minimally rigid graphs $G' = (V', E')$ and $G'' = (V'', E'')$ such that $V' \cap V'' = (u, v)$, $E' \cap E'' = \{e\} = \{(u, v)\}$, $|V'| \geq 4$ and $|V''| \geq 4$. Let v^* denote the vertex of $G_c(e)$ obtained by contracting e, and let e' be an edge incident to the vertex v^* in $G_c(e)$. We denote by $G_c(e) \backslash e'$ the graph obtained by removing the edge e' in $G_c(e)$. Since $G_c(e) \backslash e'$ is not minimally rigid, then there exists one subgraph $\bar{G} = (\bar{V}, \bar{E})$ of $G_c(e)\backslash e'$, such that $|\bar{E}| \geq 2|\bar{V}| - 2$. It is easy to see that $v^* \in \bar{V}$. Otherwise, if $v^* \notin \bar{V}$, then $|\bar{E}| \geq 2|\bar{V}| - 2$ is also a subgraph of G, and satisfies $|\bar{E}| \geq 2|\bar{V}| - 2$. This contradicts to Theorem 4.2(ii). Arguing similarly as in Case (2), we also obtain that the other endpoint of e' doesn't belong to $\bar{V}$. Moreover, there exists a minimally rigid subgraph $\overline{\overline{G}} = ((\bar{V} - v^*) \cup \{u, v\}, \overline{\overline{E}})$ in G. We now take G' as the minimally rigid subgraph in $\overline{\overline{G}}$, which contains the edge e and has a minimum number of vertices. Note that it is possible that G' coincides with $\overline{\overline{G}}$.

Next we will obtain the other minimally rigid subgraph G''. Let e'' be an edge incident to the vertex v^* in $G'_c(e)$. Using the same argument as above, there exists a subgraph $\tilde{G} = (\tilde{V}, \tilde{E})$ in $G_c(e)$, such that $|\tilde{E}| \geq 2|\tilde{V}| - 2$ and $v^* \in \tilde{V}$. Hence, we can obtain a minimally rigid graph $G'' = ((\tilde{V} - v^*) \cup \{u, v\}, E'')$.

In order to finish the proof, it is sufficient to show that $V' \cap V'' = \{u, v\}$, $E' \cap E'' = \{e\} = \{(u, v)\}$. To the contrary suppose that $\{u, v, i_1, i_2, \ldots, i_m\} = V' \cap V''$, $m \geq 1$. Since G' is a minimally rigid subgraph with the minimum number of vertices and containing the edge e, the number of edges incident to the vertex set $\{u, v, i_1, i_2, \ldots, i_m\}$ is strictly less than $2(m+2) - 3 = 2m+1$ by using Theorem 4.2. That is to say, we have at most $2m$ edges incident to the vertex set $\{u, v, i_1, i_2, \ldots, i_m\}$.

Since G is minimally rigid, the union graph $\hat{G} = (\hat{E}, \hat{V})$ of G' and G'' satisfies the equality $|\hat{V}| = |V'| + |V''| - (m + 2)$. It follows that $|\hat{E}| \leq 2|\hat{V}| - 3 = 2(|V'| + |V''| - (m + 2)) - 3$. On the other hand, we calculate the number of edges as follows: $|\hat{E}| \leq |E'| + (|E''| - 2m) = (2|V'| - 3) + (2|V''| - 3 - 2m) = 2(|V'| + |V''| - (m + 2)) - 2$. This gives a contradiction. As a consequence, we have proved that $V' \cap V'' = \{u, v\}$.

Therefore, the proposition has been proved.

5.3.3 Minimal rigidity preserving problem

For some application requirements such as decreasing the number of information links, minimum energy cost, it is necessary to preserve the minimal rigidity of formation. In this section we focus on the minimal rigidity preserving problem after loss of an agent.

For the following situation, the minimal rigidity preserving problem can be always solved systematically .

Theorem 5.10 *Let u be a vertex in a minimally rigid graph $G = (V, E)$, and v be its neighbor. If there exist any common neighbor of the vertices u and v, then the minimal rigidity preserving problem can be always solved by using edge contraction operation and edge-replacement principle after the vertex u lost.*

In order to prove Theorem 5.10, we give some important lemmas. Firstly, based on the above Lemma 5.4, the following Lemma 5.11 (named as Edge-Replacement Principle) can be obtained. The illustration of the Lemma 5.11 is shown in Fig. 5.9.

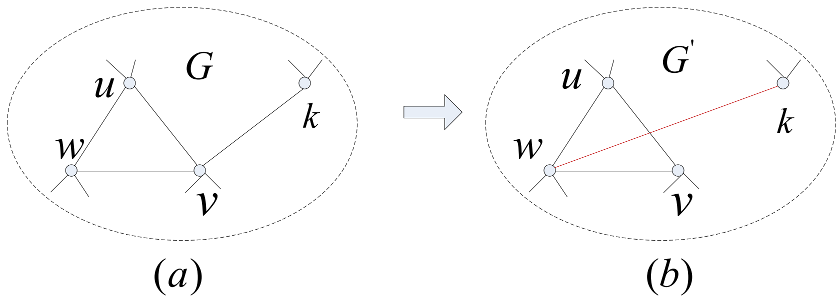

Figure 5.9: Illustration of edge-replacement principle

Lemma 5.11 *(Edge-Replacement Principle) Let u be a vertex in a minimally rigid graph $G = (V, E)$, v be a neighbor of u, w be a common neighbor of u and v, and $e = (v, k)$ be an edge in G. If there exists a minimally rigid subgraph in G containing vertices u, v, k, but w, then the graph $G' = (V', E')$ obtained from G by removing edge e and adding edge (w, k) is minimally rigid.*

Proof: Let $\bar{G} = (\bar{V}, \bar{E})$ be a minimally rigid subgraph in G containing vertices u, v, k, but w. Since $G = (V, E)$ is a minimally rigid graph, it implies the following statements:

(i) There exists no edge between w and k. Otherwise, if there exists an edge (w, k), then the number of vertices and edges of the subgraph $\tilde{G} = (\bar{V} \cup \{w\}, \tilde{E})$ in G will violate the inequality in Theorem 4.2(ii).

(ii) $\tilde{G} = (\bar{V} \cup \{w\}, \tilde{E})$ is also minimally rigid.

Let $\tilde{\tilde{G}} = (\bar{V} \cup \{w\}, \tilde{\tilde{E}})$ be a graph obtained from $\tilde{G}$ by removing edge e and adding edge (w, k). Next we will prove that the graph $\tilde{\tilde{G}}$ is minimally rigid. In fact, since $\bar{G} = (\bar{V}, \bar{E})$ is minimally rigid, $\tilde{\tilde{G}}$ can be obtained from $\bar{G}$ by using edges splitting operations: to add a vertex w, two edges (w, u), (w, v), and to replace e by the edge (w, k). Therefore, by Theorem 5.2 the graph $\tilde{\tilde{G}}$ is also minimally rigid. Furthermore, using Lemma 5.4, we can replace the subgraph $\tilde{G} = (\bar{V} \cup \{w\}, \tilde{E})$ by $\tilde{\tilde{G}} = (\bar{V} \cup \{w\}, \tilde{\tilde{E}})$ such that the graph G' is minimally rigid. The proof is completed.

Lemma 5.12 *Let u be a vertex of a minimally rigid graph $G = (V, E)$ with at least four vertices, v be a neighbor of u, and k_m be common neighbors of u and v, where $m \in \{1, \ldots, N\}, N \geq 2$. Then there is no edge between any two distinct vertices k_i, k_j.*

Proof: Without loss of generality, assume that there exist an edge (k_1, k_2) between k_1 and k_2. Then there exist a subgraph $G' = (V', E')$ of G, where $V' = \{u, v, k_1, k_2\}$, $E' = \{(u, v), (u, k_1), (u, k_2), (v, k_1), (v, k_2), (k_1, k_2)\}$. The number of edges of the subgraph G' is six. However, $6 \geq 2 \times 4 - 3 = 5$, this leads to a contradiction to Theorem 4.2(ii). Hence, there is no edge between any two distinct vertices k_i, k_j. The proof is complete.

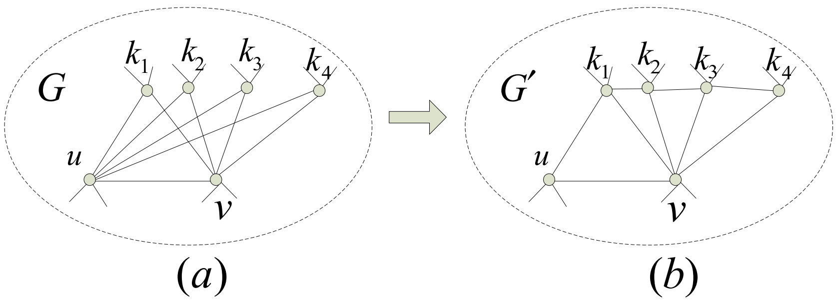

Figure 5.10: The vertices k_1,k_2 and k_3 are common neighbors of the vertices u and v in a minimally rigid graph G represented by the graph (a). The graph (b) obtained by removing edges (u, k_2), (u, k_3) and (u, k_4), adding the edges (k_1, k_2), (k_2, k_3) and (k_3, k_4), is still minimally rigid.

Based on the above Lemmas 5.4, 5.11 and 5.12, next we prove the Theorem 5.10.

Proof of Theorem 5.10:

Let $\bar{G} = (\bar{V}, \bar{E})$ be a subgraph induced by the two vertices u, v, and their common neighbors. By Lemma 5.11, there is no edge among common neighbors of two vertices u, v. In addition, it is easy to know from Theorem 4.2 that the subgraph $\bar{G}$ is minimally rigid. By Lemma 5.12, the subgraph $\bar{G}$ can be replaced by a minimally rigid graph $G^* = (\bar{V}, E^*)$ with the same vertex set $\bar{V}$, which contains unique common neighbor of u and v (To illustrate the scenarios intuitively, an example is given in Fig. 5.10). Through this operation, the number of common neighbors of u and v become to one. Therefore, we only need to prove the case that vertices u and v have unique common neighbor.

Suppose that vertices u and v have unique common neighbor w. To prove this case, we consider the following two cases respectively: (1) If there doesn't exist any other minimally rigid subgraph besides the unique triangle in G, which contains vertices v and u, but w, (2) If there exist another minimally rigid subgraph besides the unique triangle in G, which contains vertices v and u, but w.

Case (1). If there doesn't exist any other minimally rigid subgraph besides the unique triangle in G, which contains vertices v and u but w, we can always find new edges to preserve the minimal rigidity of resulting graph by using the edge contraction operation on the edge (u, v)(see Theorem 5.7).

Case (2). If there exist another minimally rigid subgraph besides the unique triangle in G, which contains vertices v and u, but w, then we can still preserve the resulting graph is minimally rigid based on Lemma 5.11 and edge contraction operation. In fact we can replace an edge which incident to a vertex of contraction edge (u, v) in the minimally rigid subgraph. It will destroy the minimally rigidity of the subgraph. One can use the operation repetitively, until there exists no exist such minimally rigid subgraph besides the unique triangle in G, which contains vertices v and u, but w. Then, based on the Theorem 5.7, we can find the new edges need to be added.

Hence, the proof is completed.

To illustrate the operation intuitively, an example is given in Fig. 5.11. The graph in Fig. 5.11 (a), is minimally rigid. The vertex v_3 is the unique common neighbor of v_1 and v_2. There exist a minimally rigid subgraph induced by the vertices v_1, v_2, v_4, v_5, v_6 and v_7, which contains the vertices v and u, but not w. When the vertex v_1 is lost, one can first add the new edge (v_2, v_7) by applying the edge contraction operation on edge (v_1, v_2), and then remove the edge (v_2, v_4) and add the other edge (v_3, v_4) using on the operation described in Theorem 5.10. Finally, the resulting graph in Fig. 5.11(b) is minimally rigid.

Next we consider a special case for the minimal rigidity preserving problem.

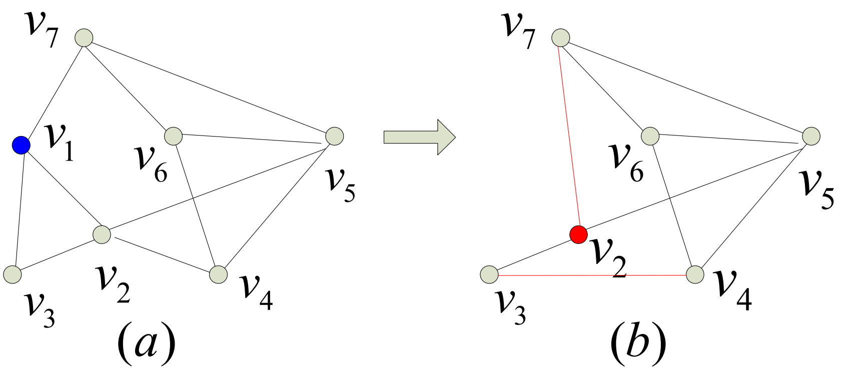

Figure 5.11: The graph (a) is a minimally rigid graph. When the vertex v_1 is lost, we first add the edge (v_2, v_7) based on edge contraction operation. Then applying the operation in Theorem 5.10, we remove the edge (v_3, v_4) and add the edge (v_2, v_4). The resulting graph (b) is minimally rigid.

Theorem 5.13 *Let u be a vertex in a minimally rigid graph $G = (V, E)$, v be a neighbor of u, and $G_n = (V_n, E_n)$ be the subgraph containing only the neighbors of v(but not v) and the edges induced by these neighbor vertices in G. If G_n is a tree (i. e. connected acyclic graph), then the edge (u, v) is contractible if and only if u is a leaf (i. e. vertex of degree 1) of G_n.*

Proof: The sufficient direction follows from Theorem 5.7: Suppose, to the contrary, u is not a leaf of G_n, then there exist two vertices v_1 and v_2 which are neighbors of u in G_n. Therefore there exist two triangles vv_1u and vv_2u in G containing the edge (u, v). Based on the Theorem 5.7, the edge (u, v) is not contractible.

Next, we prove the necessary direction. Since u is a leaf of G_n, there exists a unique common neighbor w of vertices v and u, i.e. there exists a unique triangle G' containing vertices v and u. G' is a minimally rigid subgraph of G. In order to prove the necessary direction, we use the way of contradiction. Suppose that the edge (u, v) is not contractible, then it follows Theorem 5.7 that there exists another one minimally rigid subgraph $G'' = (V'', E'')$ containing vertices v and u, but not containing vertex w.

Let m be the number of vertices of G_n in G'', i.e. $|V_n \cap V''| = m$. Since all neighbors of v are contained in the graph G_n, besides the vertex u, there exist at least another one vertex in G'', i.e. $m > 2$. Let the degree of vertex v be k. Since G_n is a tree in G, $|E_n| = k - 1$. Let $\bar{G}_n = (V_n \cup \{v\}, \bar{E}_n)$ be the subgraph in G

induced by the vertex set $V_n \cup \{v\}$. Hence we have $|\bar{E}_n| = (k-1) + k = 2k - 1$. Let $\tilde{G} = G'' \cup G_n = (\tilde{V}, \tilde{E})$, and the number of vertices of $\tilde{G}$ be n'. Then the number of edges of the graph $\tilde{G}$ will satisfy that

$$\begin{aligned} |\tilde{E}| &= 2k - 1 + 2n' - 3 - m \\ &= (2(k + n' - m) - 3) + m - 1 \\ &\geq 2(k + n' - m) - 3 \end{aligned}$$

From the Theorem 4.2, this gives a contradiction. The proof is completed.

This theorem provides an easy accessible method to check whether an edge is contractible or not, when the stated hypothesis is satisfied. For illustrating the method, an example is shown in Fig.5.12. The graph in Fig.5.12(a) is minimally rigid. We consider the closing ranks problem with loss of the vertex v_1. Observe that the subgraph induced by the neighbors of the vertex v_6 is a tree, and v_1 is a leaf of this tree. Using an edge contraction operation on (v_1, v_6) , the resulting graph in Fig.5.12(b) is minimally rigid. Note that v_2 and v_8 are also vertices satisfying the stated hypothesis of Theorem 5.13, and edges (v_1, v_2) and (v_1, v_8) are also contractible. However, the subgraph induced by the neighbors of the vertex v_3 is a tree, where v_1 is not a leaf of this tree. Applying an edge contraction operation on the edge (v_1, v_3), the resulting graph shown in Fig.5.12(c) is not rigid.

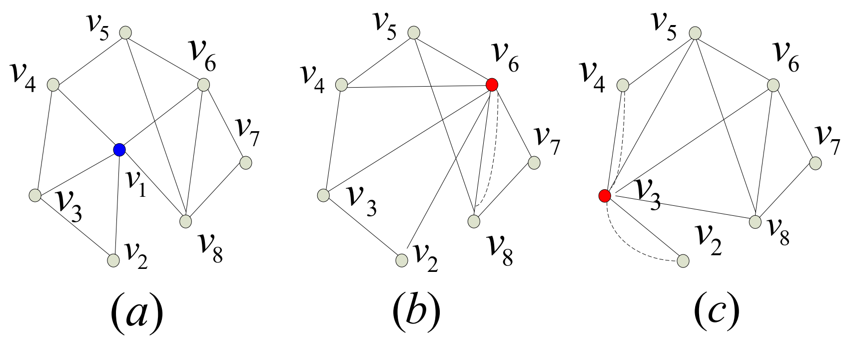

Figure 5.12: The graph in (a) is minimally rigid. One consider the closing ranks problem with loss of agent v_1. If we choose the edge (v_1, v_6) as the contraction edge, the resulting graph shown in (b) is minimally rigid. But if we contract the edge (v_1, v_3), then resulting graph shown in (c) is not rigid. Dashed lines indicate deleted double edges.

5.4 Conclusion

This chapter has considered the decentralized closing ranks problem, dealing with the addition of links to a rigid formation that is “damaged" by losing one of its agents, in order to recover its rigidity. The term “decentralized" encompasses two properties: (1) the formation make a location repair involving only neighbors of a lost agent, and (2) the neighbors perform the repair using only local information (independent of formation size). We proved that if an agent is lost, then we only need to add a set of edges on its neighbor vertices such that the subgraph which is induced by neighbor vertices of the lost vertex is minimally rigid, and then the resulting graph is rigid. Utilizing the result, we proposed two systematic self-repair operations to recover the rigidity in case of agent removals. In order to deal with the minimal rigidity preserving problem, we next consider the closing ranks problem based on edge contraction operation. A sufficient and necessary condition is proposed for the case that an edge of a minimally rigid graph is contractible: there doesn't exist two minimally rigid subgraphs whose intersection consists of the contraction edge and its incident two vertices. Based on the sufficient and necessary condition, we also study the minimally rigidity preserving problem. A set of graph theoretical results operations are established to solve the corresponding closing ranks problems.

Conclusions and Future Work

The purpose of this chapter is to summarize the main results presented in this dissertation and introduce some perspectives of future research to complete and improve this work.

Summary of main results

In this thesis, we have studied the cooperative control for multi-agent systems, and we have mainly focused on the three important topics: planning and control of multi-agent formation; formation keeping; consensus problem for multi-agent systems.

In Chapter 2, the distributed leader-following consensus problem for multi-agent systems with nonlinear inherent dynamics has been investigated. The consensus reference is taken as a virtual leader, whose output are only its position and velocity information that is available to only a set of a group of followers. In this chapter, the consensus tracking problem has been considered under the following three topologies: 1) the fixed undirected network topology; 2) the fixed directed network topology; 3) the switching undirected network topology. In the fixed or switching undirected network topology case, it is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus can also be achieved globally exponentially with the proposed protocol. In the fixed directed network topology case, it is shown that if the directed graph associated with the virtual leader and followers contains a directed spanning tree, the consensus can also be achieved globally exponentially with the proposed protocol. Rigorous proofs are given by using graph theory, matrix theory, and Lyapunov theory. Some sample simulations have been presented to illustrate the theoretical analysis.

In Chapter 3, the consensus tracking problem for multi-agent systems with nonlinear inherent dynamics has been investigated. First, we have considered the consensus tracking for first-order multi-agent systems under fixed undirected network topology and switching undirected network topology. It is shown that if the undirected graph associated with the virtual leader and followers is connected, all followers can track the virtual leader in finite time with the proposed protocol under only the position measurements of the followers and the virtual leader. Next, we have considered the consensus tracking for second-order multi-agent systems under fixed undirected network topology and switching undirected network topology. The proposed consensus control strategy has been implemented based on only the position and velocity information of each agent's neighbors. It is assumed that only one follower knows the position and velocity information of virtual leader, and it is shown that if the undirected graph associated with the virtual leader and followers is connected at each time instant, the consensus tracking can be achieved at least globally exponentially. Rigorous proofs have been given by using graph theory, matrix theory, and Lyapunov theory. Finally, Simulation examples have been given to verify the theoretical analysis.

In Chapter 4, the planning and control problem for multi-agent formation has been considered. A practicable framework for path planning and motion control of 3-dimensional multi-agent formation has been provided. The contributions of the Chapter 2 are threefold. First, in order to find a collision-free and deadlock-free feasible path for the whole formation, an optimizing algorithm has been given to optimize the path generated by A^* search algorithm, and then some redundant points generated by A^* search algorithm are removed. Second, in order to move a persistent formation as a large entity, a set of decentralized control laws have been designed for realizing the cohesive motion in 3-dimensional space. Based on these control laws, the shape of the formation is maintained during continuous motion via a set of constraints on each agent to keep its distance constraints from a prespecified group of other neighboring agents. The rigorous proofs regarding the stability and convergence of proposed control laws have been given by using Lyapunov theory and some geometric relations between agents. Third, it is well known that most of real-world systems, such as biological systems, autonomous vehicles systems and complex systems are time-varying systems. The time-varying multi-agent systems are also very important. Therefore, in this paper, the formation control problem for the time-varying multi-agent systems has been considered, where the individual controller of each agent can adjust its velocity directly according to the control task, that is to say, the velocity of each agent is time-varying. Finally, simulation results on numerical examples are shown to verify the proposed framework.

In Chapter 5, the formation keeping problem has been considered. To address the agent loss problem in rigid formation, two systematic 'self-repair' operations to recover the rigidity in case of agent removals, named Triangle Patch and Vertex Addition Patch, have been proposed. Considering the requirement of practical implementation, two algorithms corresponding the above two operations have been given. As these two operations cannot guarantee that the resultant graph is minimally rigid. Next, the closing ranks problem has been considered based on another graph operation named edge contraction. First, we have proposed a sufficient and necessary condition for the case that an edge of a minimally rigid graph is contractible: an edge in a minimally rigid graph is contractible if and only if there are no two minimally rigid subgraphs such that their intersection consists of the contraction edge and its two incident vertices. Later, an Edge-replacement Principle has been proposed. Based on Edge-replacement Principle and edge contraction operation, the minimally rigidity preserving problem has been studied. A set of graph theoretical results operations have been established to solve the corresponding closing ranks problems.

Future research directions

There are several directions and possible related areas in which we can carry out future work.

In Chapters 2 and 3, the consensus problem for first-order and second-order multi-agent systems with nonlinear inherent dynamics has been studied. The consensus problem for high-order multi-agent systems with nonlinear inherent dynamics will be a future research direction. In this thesis, the consensus tracking problem are considered under undirected communication topologies. In the future, we will consider this problem under directed communication topologies.

In Chapter 4, we have considered the problem of cohesive motion of persistent formation based on the velocity integrator model (4.1) for the agent kinematics.

However, in a real-life formation, kinematic or dynamic model of agents would be more complex than a velocity integrator in general. A more complex agent model than Equation (4.1) would be required in a more practical design and analysis procedure for the cohesive motion problem. Therefore, in the future, we will consider the cohesive motion problem of persistent formation under the following model: (1) double integrator or point mass dynamics model, where the acceleration term $\dot{v}_i = a_i$ can be added to model (4.1) and it is assumed that the control input is the vectorial acceleration a_i; (2) the full actuated uncertain dynamics dynamics model:

$$M_i(p_i)\ddot{p}_i + \eta_i(\dot{p}_i, p_i) = u_i$$

where M_i represents the mass or inertia matrix, η_i represents the centripetal, coriolis, gravitational effects, and other additive disturbances, and u_i is the control signal that can be applied in the fore of a vectorial force; (3) the nonholonomic unicycle dynamics model:

$$(\dot{x}_i, \dot{y}_i) = (v_i \cos\theta_i, v_i \sin\theta_i)$$

$$\dot{\theta}_i = \omega_i$$
$$\dot{v}_i = \frac{1}{m_i} u_{i1}$$
$$\dot{\omega}_i = \frac{1}{\tau_i} u_{i2}$$

where $p_i(t) = (x_i(t), y_i(t))$ denotes the position of A_i as before and $\dot{\theta}_i(t)$ denotes the orientation or steering angle of the agent with respect to a certain fixed axis, v_i is the translational speed, ω_i is the angular velocity, and the control input signals u_{i1} and u_{i2} are the force and torque inputs.

Chapter 5 has addressed rigid-based formation keeping in the event of agent loss. As future extensions of this works, a number of topics are accessible for the future works. First, in this dissertation, we consider the closing ranks problem in undirected graphs. The closing ranks problem in directed graphs is a future work. We will handle the agent loss problem in persistent formation. Second, the current works mainly focuses on the formation keeping problem in 2-dimensional space, The formation keeping problem in 3-dimensional and high-dimensional space will be another future work.

In addition, networks with nonlinear couplings and networks consisting of heterogeneous autonomous agents will be studied in the future. A general stochastic framework may be helpful in bringing more ideas from the study of complex networks into the study of the consensus problem, which is a promising approach deserving further investigation as well.

Since the operations of most of the multi-agent systems are naturally delayed, the study of the multi-agent systems with delay needs more attention.

Bibliography

ANDERSON, B.D.O., YU, C., FIDAN, B. & HENDRICKX, J.M. (2006). Control and information architectures for formations. In *Proc. Of the IEEE conference on Control Applications (in Joint CCA/CACSD/ISIC), pp.1127?138*. 24, 95

ANDERSON, B.D.O., YU, C., FIDAN, B. & HENDRICKX, J.M. (2008). Rigid graph control architectures for autonomous formations. *Control Systems, IEEE*, **28**, 48 –63. 95

CAO, M., MORSE, A.S. & ANDERSON, B.D.O. (2008). Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, **47**, 575–600. 20

CAO, Y. & REN, W. (2012). Distributed coordinated tracking with reduced interaction via a variable structure approach. *Automatic Control, IEEE Transactions on*, **57**, 33 –48. 20, 59, 62

CHUNG, F.R.K. (1997). *Spectral Graph Theory*, vol. 92. American Mathematical Society. 33

CLARKE, F. (1983). *Optimization and nonsmooth analysis*. Canadian Mathematical Society series of monographs and advanced texts, Wiley. 46, 63, 72

DAS, A. & LEWIS, F.L. (2011). Distributed adaptive control for networked multi-robot systems. *Multi-Robot Systems, Trends and Development.*, 34–52, dr. Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2. 17

EREN, T., BELHUMEUR, P. & MORSE, A. (2002). Closing ranks in vehicle formations based on rigidity. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3, 2959 – 2964 vol.3. 26

EREN, T., ANDERSON, B.D.O., MORSE, A.S., WHITELEY, W. & BELHUMEUR, P.N. (2004). Operations on rigid formations of autonomous agents. In *COMMUNICATIONS IN INFORMATION AND SYSTEMS*, 223–258. 22, 24, 90, 122

FIDAN, B., ANDERSON, B., YU, C. & HENDRICKX, J. (2007). Rigidity and persistence for ensuring shape maintenance of multiagent meta formations (ext'd version). *in Modeling and Control of Complex Systems, P. Ioannou and A. Pitsillides (ed.), Taylor & Francis*, **abs/0710.2659**, 247–275. 89, 95, 97, 98

FIDAN, B., HENDRICKX, J.M. & ANDERSON, B.D.O. (2010). Closing ranks in rigid multi-agent formations using edge contraction. *International Journal of Robust and Nonlinear Control*, **20**, 2077–2092. 26, 126, 130

FILIPPOV, A. & ARSCOTT, F. (1988). *Differential Equations With Discontinuous Righthand Sides*. Mathematics and Its Applications, Kluwer Academic Publishers. 63

GAZI, V. & FIDAN, B. (2007). Coordination and control of multi-agent dynamic systems: models and approaches. In *Proceedings of the 2nd international conference on Swarm robotics*, SAB'06, 71–102, Springer-Verlag, Berlin, Heidelberg. 17, 19

GE, S. & CUI, Y. (2000). New potential functions for mobile robot path planning. *Robotics and Automation, IEEE Transactions on*, **16**, 615 –620. 21

GILBERT, E. & JOHNSON, D. (1985). Distance functions and their application to robot path planning in the presence of obstacles. *Robotics and Automation, IEEE Journal of*, **1**, 21 – 30. 21

HAO, Y. & AGRAWAL, S.K. (2005). Formation planning and control of ugvs with trailers. *Auton. Robots*, **19**, 257–270. 21, 22

HENDRICKX, J.M., FIDAN, B., YU, C., ANDERSON, B.D.O. & BLONDEL, V.D. (2006). Primitive operations for the construction and reorganization of minimally persistent formations. *CoRR*, **abs/cs/0609041**. 95

HENDRICKX, J.M., YU, C., FIDAN, B. & ANDERSON, B.D.O. (2007). Rigidity and persistence for ensuring shape maintenance of multiagent meta formations (ext'd version). *CoRR*, **abs/0710.2659**. 91, 92, 93, 95

HENNEBERG, L. (1911). Die graphishche static der starren systeme. *Teubner: Stuttgart*. 22

HONG, Y., CHEN, G. & L.BUSHNELL (2008). Distributed observers design for leader-following control of multi-agent networks. *Automatica*, **44**, 846 – 850. 20

HORN, R.A. & JOHNSON, C.R. (1985). *Matrix analysis / Roger A. Horn, Charles R. Johnson*. Cambridge University Press, Cambridge [Cambridgeshire] ; New York :, repr. with corr. edn. 33, 34

JADBABAIE, A., LIN, J. & MORSE, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, **48**, 988 – 1001. 19

JIANG, F. & L.WANG (2009). Finite-time information consensus for multi-agent systems with fixed and switching topologies. *Physica D: Nonlinear Phenomena*, **238**, 1550 – 1560. 20

LAMAN, G. (1970). On graphs and rigidity of plane skeletal structures. *J. Engrg. Math.*, **4**, 331–340. 22, 90, 91

LIAN, F. & MURRAY, R. (2002). Real-time trajectory generation for the cooperative path planning of multi-vehicle systems. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 4, 3766 – 3769 vol.4. 21

LU, J. & XI, Y. (2003). Linear generalized synchronization of continuous-time chaotic systems. *Chaos, Solitons and Fractals*, **17**, 825 – 831. 48

MANTLER, A. & SNOEYINK, J. (2004). Banana spiders: A study of connectivity in 3d combinatorial rigidity. In *CCCG*, 44–47. 93

MATSUMOTO, T. (1984). A chaotic attractor from chua's circuit. *Circuits and Systems, IEEE Transactions on*, **31**, 1055 –1058. 48

Moreau, L. (2005). Stability of multiagent systems with time-dependent communication links. *Automatic Control, IEEE Transactions on*, **50**, 169–182. 20

Ngo, V.T., Nguyen, A. & Ha, Q.P. (2005). Integration of planning and control in robotic forma- tion. *in Proc. the Australasian Conference on Robotics and Automation*, 1–8. 21, 22

Nilsson, N.J. (1980). *Principles of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 101

Oh, K.K. & Ahn, H.S. (2010). A survey of formation of mobile agents. In *Intelligent Control (ISIC), 2010 IEEE International Symposium on*, 1470 – 1475. 17

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *Automatic Control, IEEE Transactions on*, **51**, 401 – 420. 20

Olfati-Saber, R. & Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, **49**, 1520 – 1533. 17, 19, 20

Olfati-saber, R., Fax, J.A. & Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE*, 2007. 17

Paden, B.E. & Sastry, S.S. (1987). A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators. **34**, 73–81. 46, 63, 72

Peng, K. & Yang, Y. (2009). Leader-following consensus problem with a varying-velocity leader and time-varying delays. *Physica A: Statistical Mechanics and its Applications*, **388**, 193 – 208. 20

Ren, W. & Beard, R. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *Automatic Control, IEEE Transactions on*, **50**, 655 – 661. 17, 19, 20, 33

Ren, W., Beard, R. & Atkins, E. (2007). Information consensus in multivehicle cooperative control. *Control Systems, IEEE*, **27**, 71 –82. 17

Reynolds, C.W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, 25–34, ACM, New York, NY, USA. 19

Sandeep, S., Fidan, B. & Yu, C. (2006). Decentralized cohesive motion control of multi-agent formations. In *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on*, 1 –6. 22

Shames, I., Yu, C., Fidan, B. & Anderson, B. (2007). Externally excited coordination of autonomous formations. In *Control Automation, 2007. MED '07. Mediterranean Conference on*, 1 –6. 21, 22, 101

Shevitz, D. & Paden, B. (1994). Lyapunov stability theory of nonsmooth systems. *Automatic Control, IEEE Transactions on*, **39**, 1910 –1914. 46, 47, 72, 73

Shi, H., Wang, L. & Chu, T. (2006). Virtual leader approach to coordinated control of multiple mobile agents with asymmetric interactions. *Physica D: Nonlinear Phenomena*, **213**, 51 – 65. 20

Shi, H., Wang, L. & Chu, T. (2009). Flocking of multi-agent systems with a dynamic virtual leader. *International Journal of Control*, **82**, 43–58. 20

SU, H., CHEN, G., WANG, X. & LIN, Z. (2011). Adaptive second-order consensus of networked mobile agents with nonlinear dynamics. *Automatica (Oxford)*, **47**, 368–375, eng. 20

Summers, T., Yu, C. & Anderson, B. (2008). Decentralized closing ranks in vehicle formations and sensor networks. In *Control and Automation, 2008 16th Mediterranean Conference on*, 1137 –1143. 26

Summers, T.H., Yu, C. & Anderson, B.D.O. (2009). Addressing agent loss in vehicle formations and sensor networks. *International Journal of Robust and Nonlinear Control*, **19**, 1673–1696. 26

Tanner, H.G., Jadbabaie, A. & Pappas, G.J. (2007). Flocking in fixed and switching networks. *Automatic Control, IEEE Transactions on*, **52**, 863 –868. 46, 63, 72

TAY, T. & WHITELEY, W. (1985). Generating isostatic frameworks. *Structural Topology*, 21–69. 22, 25, 90, 120

VICSEK, T., CZIROK, A., BEN-JACOB, E., COHEN, I. & SOCHET, O. (1995). Novel type of phase transition in a system of self-driven particles. *PHYS REV LETT.*, **75**, 1226. 19

WANG, X., YADAV, V. & BALAKRISHNAN, S. (2007). Cooperative uav formation flying with obstacle/collision avoidance. *Control Systems Technology, IEEE Transactions on*, **15**, 672 –679. 23

WEN, G. (2009). *Research on motion control of multi-agent persistent formation*. Master thesis in Beijing Jiaotong University. 102

WEN, G., RAHMANI, A. & YU, Y. (2010). Decentralized closing ranks in 2-dimensional rigid multi-agent formations. In *Intelligent Control and Information Processing (ICICIP), 2010 International Conference on*, 310 –315. 89

WEN, G., PENG, Z., YU, Y. & RAHMANI, A. (2012 a). Planning and control of three-dimensional multi-agent formations. *IMA Journal of Mathematical Control and Information*, doi:10.1093/imamci/dns028. 99, 103

WEN, G., PENG, Z., YU, Y. & RAHMANI, A. (2012 b). Consensus tracking for second-order multi-agents systems with nonlinear inherent dynamics. *International journal of System Science*, (Accepted). 36, 39, 41, 42, 45

WHITELEY, W. (1984). *Transactions of the American Mathematical Society*, **285**, 431–465. 123

WHITELEY, W. (1997). Handbook of discrete and computational geometry. chap. Rigidity and scene analysis, 893–916, CRC Press, Inc., Boca Raton, FL, USA. 90, 91

YU, C., HENDRICKX, J., FIDAN, B., ANDERSON, B.D. & BLONDEL, V.D. (2007). Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence. *Automatica*, **43**, 387 – 402. 89, 90, 95, 96, 97, 98

YU, W., CHEN, G., CAO, M. & KURTHS, J. (2010). Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **40**, 881 –891. 20

YU, W., CHEN, G. & CAO, M. (2011). Consensus in directed networks of agents with nonlinear dynamics. *Automatic Control, IEEE Transactions on*, **56**, 1436 –1441. 20

ZHAO, R., YU, Y. & WEN, G. (2010). Decentralized cohesive motion control of multi-agent formation in 3-dimensional space. In *Proceedings of the 2010 international conference on Artificial intelligence and computational intelligence: Part I*, AICI'10, 339–348, Springer-Verlag, Berlin, Heidelberg. 98, 102

ZHAO, W. & TIAUW, H. (2011). obust decentralized formation flight control. *International Journal of Aerospace Engineering*, **2011,Article ID 157590**, 1–13. 23

ZORAN GAJIC, M.T.J.Q. (1995). *Lyapunov Matrix Equation in System Stability and Control*. CMathematics in Science & Engineering, Academic Press, San Diego. 42

Contrôle Coopératif Distribué pour Systémes Multi-Agents

Résumé:Cette thèse considère principalement trois problèmes dans le domaine du contrôle distribué coopératif des systèmes multi-agents (SMA): le consensus, la navigation en formation et le maintien en formation d'un groupe d'agents lorsqu'un agent disparait. Nous proposons 3 algorithmes pour résoudre le problème du calcul distribué d'un consensus à partir de l'approche leadeur-suiveur dans le contexte SMA à dynamique non-linéaire. La référence est définie comme un leader virtuel dont on n'obtient, localement, que les données de position et de vitesse. Pour résoudre le problème du suivi par consensus pour les SMA à dynamique non-linéaire, nous considérons le suivi par consensus pour SMA de premier ordre. On propose des résultats permettant aux suiveurs de suivre le leadeur virtuel en temps fini en ne considérant que les positions des agents. Ensuite, nous considérons le suivi par consensus de SMA de second. Dans le cas de la planification de trajectoire et la commande du mouvement de la formation multi-agents. L'idée est d'amener la formation, dont la dynamique est supposée être en 3D, d'une configuration initiale vers une configuration finale (trouver un chemin faisable en position et orientation) en maintenant sa forme tout le long du chemin en évitant les obstacles. La stratégie proposée se décompose en 3 étapes. Le problème du Closing-Rank se traduit par la réparation d'une formation rigide multi-agents "endommagée" par la perte de l'un de ses agents. Nous proposons 2 algorithmes d'autoréparation systématique pour récupérer la rigidité en cas de perte d'un agent. Ces réparations s'effectuent de manière décentralisée et distribuée n'utilisant que des informations de voisinage.

Mots-clefs: Systèmes multi-agents, Contrôle coopératif, Contrôle distribué, Contrôle de la formation, Consensus, Maintien de la formation , Théorie des graphes.

Distributed Cooperative Control for Multi-Agent Systems

Abstract: This dissertation focuses on distributed cooperative control of multi-agent systems. First, the leader-following consensus for multi-agent systems with nonlinear dynamics is investigated. Three consensus algorithms are proposed and some sufficient conditions are obtained for the states of followers converging to the state of virtual leader globally exponentially. Second, the consensus tracking for multi-agent systems with nonlinear dynamics is investigated. Some consensus tracking algorithms are developed, and some sufficient conditions are obtained. Based on these consensus tracking algorithms and sufficient conditions, it is shown that in first-order multi-agent systems all followers can track the virtual leader in finite time, and in second-order multi-agent systems the consensus tracking can be achieved at least globally exponentially. Third, the path planning and motion control of multi-agent formation is studied, where a practical framework is provided. In order to find a collision-free and deadlock-free feasible path for the whole formation, an optimizing algorithm is given to optimize the path generated by A* search algorithm. In order to realize the cohesive motion of a persistent formation in 3-dimensional space, a set of decentralized control laws is designed. Finally, the formation keeping problem is studied. We mainly focus on the closing ranks problem, which deals with the addition of links to a rigid multi-agent formation that is “damaged" by losing one of its agents, in order to recover rigidity. Some graph theoretical results are obtained, and some systematic ‘self-repair’ operations are proposed to recover the rigidity in case of agent removals.

Keywords: Multi-agent systems, Cooperative Control, Distributed control, Formation control, Consensus, Formation keeping, Graph theory.