

Summer 2016

Unmanned vehicles formation control in 3D space and cooperative search

Saba Ramazani

Follow this and additional works at: <https://digitalcommons.latech.edu/dissertations>



Part of the [Electrical and Computer Engineering Commons](#)

**UNMANNED VEHICLES FORMATION CONTROL IN 3D SPACE
AND COOPERATIVE SEARCH**

by

Saba Ramazani, B.Sc., M.S.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

**COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY**

August 2016

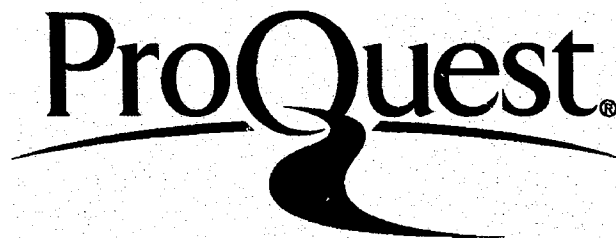
ProQuest Number: 10301324

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10301324

Published by ProQuest LLC(2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code.
Microform Edition © ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY
THE GRADUATE SCHOOL

May 20, 2016

Date

We hereby recommend that the dissertation prepared under our supervision
by Saba Ramazani

entitled Unmanned Vehicles Formation Control in 3D Space and Cooperative Search

be accepted in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy

R. Šelmić (S.Z.)

Supervisor of Dissertation Research

Head of Department

COLLEGE OF ENGINEERING & SCIENCE

Department

Recommendation concurred in:

[Signature]

MSaleh

Advisory Committee

Jinko Kanno

J.K.

Matthias R. Brust

Approved:

[Signature]

Director of Graduate Studies

Approved:

[Signature]

Dean of the Graduate School

Hisham Hegab/SD

Dean of the College

ABSTRACT

The first problem considered in this dissertation is the decentralized non-planar formation control of multiple unmanned vehicles using graph rigidity. The three-dimensional formation control problem consists of n vehicles operating in a plane Q and r vehicles that operate in an upper layer outside of the plane Q . This can be referred to as a layered formation control where the objective is for all vehicles to cooperatively acquire a predefined formation shape using a decentralized control law. The proposed control strategy is based on regulating the inter-vehicle distances and uses backstepping and Lyapunov approaches. Three different models, with increasing level of complexity are considered for the multi-vehicle system: the single integrator vehicle model, the double integrator vehicle model, and a model that represents the dynamics of a class of robotics vehicles including wheeled mobile robots, underwater vehicles with constant depth, aircraft with constant altitude, and marine vessels. A rigorous stability analysis is presented that guarantees convergence of the inter-vehicle distances to desired values. Additionally, a new Neural Network (NN)-based control algorithm that uses graph rigidity and relative positions of the vehicles is proposed to solve the formation control problem of unmanned vehicles in 3D space. The control law for each vehicle consists of a nonlinear component that is dependent on the closed-loop error dynamics plus a NN component that is linear in the output weights (a one-tunable layer NN is used). A Lyapunov analysis shows that the proposed

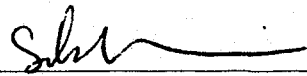
distance-based control strategy achieves the uniformly ultimately bounded stability of the desired infinitesimally and minimally rigid formation and that NN weights remain bounded. Simulation results are included to demonstrate the performance of the proposed method.

The second problem addressed in this dissertation is the cooperative unmanned vehicles search. In search and surveillance operations, deploying a team of unmanned vehicles provides a robust solution that has multiple advantages over using a single vehicle in efficiency and minimizing exploration time. The cooperative search problem addresses the challenge of identifying target(s) in a given environment when using a team of unmanned vehicles by proposing a novel method of mapping and movement of vehicle teams in a cooperative manner. The approach consists of two parts. First, the region is partitioned into a hexagonal beehive structure in order to provide equidistant movements in every direction and to allow for more natural and flexible environment mapping. Additionally, in search environments that are partitioned into hexagons, the vehicles have an efficient travel path while performing searches due to this partitioning approach. Second, a team of unmanned vehicles that move in a cooperative manner and utilize the Tabu Random algorithm is used to search for target(s). Due to the ever-increasing use of robotics and unmanned systems, the field of cooperative multi-vehicle search has developed many applications recently that would benefit from the use of the approach presented in this dissertation, including: search and rescue operations, surveillance, data collection, and border patrol. Simulation results are presented that show the performance of the Tabu Random search algorithm method in combination with hexagonal partitioning.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author 

Date 7, 25, 16

DEDICATION

I dedicate this dissertation to my loving parents. I would not have been able to embark on and complete this journey without their support, endless hard work, and sacrifices.

TABLE OF CONTENTS

ABSTRACT	iii
DEDICATION	vi
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
ACKNOWLEDGMENTS	xv
CHAPTER 1 INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Outline of Dissertation.....	4
CHAPTER 2 BACKGROUND AND RELATED WORK	6
2.1 Formation Control	6
2.2 Rigid Graph Theory.....	8
2.3 Neural Networks.....	14
2.4 Cooperative Search and Partitioning Methods	16
CHAPTER 3 NON-PLANAR MULTI-VEHICLE LAYERED FORMATION CONTROL.....	24
3.1 Problem Formulation	25
3.1.1 Non-Planar Multi-Vehicle Layered Formation Control with Sin- gle Integrator Model.....	26
3.1.2 Non-Planar Multi-Vehicle Layered Formation Control with Dou- ble Integrator Model.....	27

3.2	Control Algorithms	27
3.2.1	Single Integrator Vehicle Model.....	27
3.2.2	Double Integrator Vehicle Model.....	34
3.3	Simulation Results	39
3.3.1	Single Integrator Model.....	39
3.3.2	Double Integrator Model	45
3.3.3	Robotic Vehicle Dynamics.....	50
CHAPTER 4	NEURAL NETWORK-BASED FORMATION CONTROL OF UNMANNED VEHICLES IN 3D SPACE.....	55
4.1	Problem Formulation	55
4.2	Control Algorithm.....	56
4.3	Simulation Results	59
4.3.1	Single Integrator Model without External Disturbances	59
4.3.2	Single Integrator Model with External Disturbances	64
CHAPTER 5	UNMANNED VEHICLES SEARCH IN 2D SPACE.....	71
5.1	Methodology	71
5.1.1	Partitioning Method.....	71
5.1.2	Search Algorithm.....	71
5.2	Simulation.....	75
5.2.1	Simple Environment with Randomized Deployment	76
5.2.2	Simple Environment with Localized Deployment.....	77
5.2.3	Complex Environment with Randomized Deployment.....	79
5.2.4	Complex Environment with Localized Deployment.....	80

5.3	Results	84
5.3.1	Simple Environment	84
5.3.2	Complex Environment	86
CHAPTER 6	CONCLUSIONS AND FUTURE WORK	89
BIBLIOGRAPHY	92

LIST OF TABLES

Table 5.1:	Results of the search for the target by both vehicles in the simple environment using RD	84
Table 5.2:	Results of the search for the target by both vehicles in the simple environment using LD.....	85
Table 5.3:	Results of the search and percentage of target detection by each vehicle in the simple environment using RD	85
Table 5.4:	Results of the search and percentage of target detection by each vehicle in the simple environment using LD	85
Table 5.5:	Results of the search for the targets by all the vehicles in the complex environment using RD and beehive structure.....	86
Table 5.6:	Results of the search for the targets by all the vehicles in the complex environment using RD and square partitioning.....	87
Table 5.7:	Results of the search for the targets by all the vehicles in the complex environment using LD and beehive structure	87
Table 5.8:	Results of the search for the targets by all the vehicles in the complex environment using LD and square partitioning	88

LIST OF FIGURES

Figure 2.1: A noncongruent framework that depicts flip ambiguity (vertex 2 can be flipped over the edge (1,3) to the symmetric position 2').....	9
Figure 2.2: Flexible (left) and rigid (right) formations in two dimensions.....	10
Figure 2.3: A rigid but not infinitesimally and minimally rigid framework (left), an infinitesimally but not minimally rigid framework (middle), and an infinitesimally and minimally rigid framework (right) in two dimensions	11
Figure 2.4: An infinitesimally and minimally rigid graph in three dimensions	12
Figure 2.5: A two layer neural network.....	14
Figure 2.6: Triangular tiling, square tiling, and hexagonal tiling.....	20
Figure 2.7: Hexagonal (a) and square (b) travel path lengths	21
Figure 2.8: Comparison of hexagonal and square partition calculations.....	22
Figure 3.1: Formation control concept under the layered sensing framework	24
Figure 3.2: Formation control concept under the coning framework.....	25
Figure 3.3: Desired formation for six lower layer vehicles (circles) and one upper layer vehicle (star)	41
Figure 3.4: Vehicles' trajectories $p_{i_x}(t)$, $i = 1, \dots, 7$ with single integrator model (solid line) and desired formation (dotted line).....	41
Figure 3.5: Sample of distance errors for six lower layer vehicles and the upper layer vehicle	42
Figure 3.6: Control inputs $u_{i_x}(t)$, $i = 1, \dots, 7$ in the x -direction.....	42
Figure 3.7: Control inputs $u_{i_y}(t)$, $i = 1, \dots, 7$ in the y -direction.....	43

Figure 3.8: Control input in the z -direction for the upper layer vehicle	43
Figure 3.9: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 8$ with single integrator model (solid line) and desired formation (dotted line).....	44
Figure 3.10: Sample of distance errors for four lower layer vehicles and four upper layer vehicles.....	44
Figure 3.11: Desired formation for five lower layer vehicles and three upper layer vehicles with double integrator model.....	45
Figure 3.12: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 8$ with double integrator model (solid line) and desired formation (dotted line).....	46
Figure 3.13: Sample of distance errors for five lower layer vehicles and three upper layer vehicles.....	47
Figure 3.14: Control inputs $u_{i,x}(t)$, $i = 1, \dots, 8$ in the x -direction.....	47
Figure 3.15: Control inputs $u_{i,y}(t)$, $i = 1, \dots, 8$ in the y -direction.....	48
Figure 3.16: Control inputs in the z -direction for the upper layer vehicles	48
Figure 3.17: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 6$ with double integrator model (solid line) and desired formation (dotted line).....	49
Figure 3.18: Sample of distance errors for two lower layer vehicles and four upper layer vehicles	49
Figure 3.19: Vehicles' trajectories $q_i(t)$, $i = 1, \dots, 5$ (solid line) and desired formation (dotted line)	53
Figure 3.20: Sample of distance errors (m) vs. time (s) for four robotic vehicles and one upper layer agent.....	54
Figure 4.1: Desired formation for four UGVs and two UAVs	60
Figure 4.2: Vehicle trajectories $p_i(t)$, $i = 1, \dots, 6$ (solid line) and desired formation (dotted line)	61
Figure 4.3: Sample of distance errors $e_{ij}(t)$, $i, j \in V$	61
Figure 4.4: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the x -direction.....	62
Figure 4.5: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the y -direction.....	63

Figure 4.6: Control inputs $u_i(t)$, $i = 5, 6$ in the z -direction for the UAVs	63
Figure 4.7: Sample of NN weights W_{1ij}	64
Figure 4.8: Vehicle trajectories $p_i(t)$, $i = 1, \dots, 6$ (solid line) and desired formation (dotted line) in the presence of external disturbances.....	66
Figure 4.9: Sample of distance errors $e_{ij}(t)$, $i, j \in V$ with external disturbances..	66
Figure 4.10: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the x -direction.....	67
Figure 4.11: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the y -direction.....	67
Figure 4.12: Control inputs $u_i(t)$, $i = 5, 6$ in the z -direction for the UAVs	68
Figure 4.13: Sample of NN weights W_{1ij} with external disturbances.....	68
Figure 4.14: Comparison of $\sum e_{ij} $ for the control laws with and without NN component and without external disturbances	69
Figure 4.15: Comparison of $\sum e_{ij} $ for the control laws with and without NN component in the presence of external disturbances.....	70
Figure 5.1: Hexagonal partitioned environment with one obstacle, one target, and two unmanned vehicles with RD	76
Figure 5.2: Square partitioned environment with one obstacle, one target, and two unmanned vehicles with RD	77
Figure 5.3: Hexagonal partitioned environment with one obstacle, one target, and two unmanned vehicles with LD	78
Figure 5.4: Square partitioned environment with one obstacle, one target, and two unmanned vehicles with LD	78
Figure 5.5: Hexagonal partitioned environment with two obstacles, three targets, and four unmanned vehicles with RD	79
Figure 5.6: Square partitioned environment with two obstacles, three targets, and four unmanned vehicles with RD	80
Figure 5.7: Hexagonal partitioned environment with two obstacles, three targets, and four unmanned vehicles with LD	80

Figure 5.8:	Square partitioned environment with two obstacles, three targets, and four unmanned vehicles with LD	81
Figure 5.9:	Collision occurs if the same adjacent cell is chosen by the vehicles to move into.....	83

ACKNOWLEDGMENTS

The completion of this dissertation would not have been possible without the guidance, support, and encouragement of many people, only some of whom are mentioned here.

I am grateful to my advisor, Dr. Rastko Selmic, for his guidance, mentoring, and consistent help throughout my graduate studies. I would like to thank Dr. Marcio de Queiroz, Dr. Jinko Kanno, Dr. Galen Turner, and Dr. Matthias Brust for serving on my committee and for their valuable advise which has helped me tremendously in my work. I would also like to appreciate my research team members, Andrew Gardner and Delvin Jackson, for their support and help. Thanks also to NASA, Louisiana Space Consortium (LaSPACE), and Louisiana Tech College of Engineering and Science for their financial support.

Last but not least, I thank all of family and friends that motivated and supported me throughout this journey. Special thanks to Heath for always inspiring, encouraging, and challenging me to reach my full potential and achieve my goals. Thanks to my wonderful siblings Sina, Saleh, and Sadaf for their continuous love and support. Many thanks to Heather who loved me like a sister and always supported me. She will live forever in my heart. I would also like to thank June and Steve Berry for taking me in, caring for me, and encouraging me.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Unmanned systems are widely used in the military, in defense operations, and in space applications [1], [2]. The use of unmanned units has provided significant new capabilities for defense, space, and civil applications including surveillance, reconnaissance, battle damage, first response units, atmospheric radiation monitoring, meteorology, coastal patrol, earth science, land management, homeland security, and planetary exploration. Combining novel sensing approaches and unmanned platforms with advanced logistic control algorithms is of extreme importance to the shifting role of technology.

Formation control of multiple unmanned vehicles, also referred to as a multi-agent system, has attracted considerable attention in recent years due to its many applications in military and defense operations, environmental monitoring, and space missions. The concept of unmanned vehicle formation is inspired by the collective behavior of biological systems in nature, e.g. a flock of birds or a school of fish. These biological systems often display formation-type behavior. In this type of behavior, the group moves as a cohesive whole from one point to another while performing complex tasks and maintaining the original formation shape. In nature, the behavior of groups

of birds, fish, and bees is also distributed and decentralized as each individual member of the group has its own local sensing and control mechanism without global knowledge or planning [3].

With this inspiration, formations of Unmanned Ground Vehicles (UGVs), Unmanned Underwater Vehicles (UUVs), and Unmanned Aerial Vehicles (UAVs) are deployed to perform surveillance, reconnaissance, and search of an area [4], [5], [6]. Performing such tasks using formation of unmanned vehicles is more effective than using one vehicle for various reasons such as robustness when one or more vehicles fail, more complex task execution, and reducing the sensor uncertainty by taking advantage of the merging of overlapping information from the cooperating vehicles; therefore, increasing the efficiency of the mission [7], [8].

In a formation-type behavior, the group of vehicles move together from one point to another to perform a task while maintaining the original formation structure. This structure is a geometric shape, and maintaining this shape implies that the formation at one instant of time is congruent to the formation at another instant of time. This behavior is also displayed in nature by flocks of birds and schools of fish [9].

Graphs have been used as a tool to model multi-vehicle formations. In a graph that corresponds to a formation structure, each vehicle corresponds to a vertex, and for each pair of vehicles i and j , there must be an edge (i, j) if the distance between i and j is to be maintained at all times [10]. Rigid graph theory [11], [12] plays a crucial role in analyzing the multi-vehicle formation shape and describing the information architecture of the system. In this case, the rigidity matrix is important for the

stability analysis of the formation control. Some previous work that used rigid graph theory to address the multi-agent formation problem can be found in [9], [13], [14], [15].

Cooperative search of an environment using multiple unmanned vehicles is a process aiming to improve the performance of the vehicles involved in costly and time consuming individual searches for target(s). For example, consider a team of UAVs that are deployed in a region to search for a hidden emitter source. The deployed UAVs use their on-board sensors to detect the Electromagnetic (EM) source. Then by communicating with other UAVs in the region they will aggregate in the perimeter of the target. By forming a coalition and searching the environment cooperatively, the UAVs can share or re-use the information (that otherwise might have been discarded) regarding the search environment between themselves. The cooperative multi-vehicle search of an environment is useful for many applications such as search and rescue operations, surveillance, data collection, and border patrol [1], [16], [17].

Algorithms for searching an environment have been studied extensively in the past decade. However, the recent advances in autonomous systems technology, robotics, and wireless communication have created the need for studying the cooperative search methods. The development of a decentralized search algorithm for intelligent unmanned vehicles that are capable of wireless communication and are equipped with various sensor devices is considered in this dissertation.

Partitioning a search environment is a method that decomposes the area into cells for effective coverage. There are two method of uniformly partitioning an environment: square partitioning and hexagonal partitioning. Most of the cooperative search algorithms are utilized in a square partitioned area. This dissertation discusses

the advantages of using hexagonal partitioning and compares the effectiveness of the proposed search algorithm using both partitioning methods.

1.2 Outline of Dissertation

In this work, we first propose control strategies for stabilization of multi-vehicle formations in three-dimensional space followed by a new search algorithm for cooperative unmanned vehicles search of an environment.

Chapter 2 begins with discussing the topic of formation control and giving an overview of the recent work in literature on this topic. Then, the concept of rigid graph theory and its use in solving the multi-vehicle formation control problems are discussed. Some preliminary results which are used in the subsequent chapters are also presented. Next, some background information on NNs is given. Finally, the general cooperative search methods and the related work in the area of mobile agents search are introduced in this chapter.

The non-planar multi-vehicle formation control topic is discussed in Chapter 3 and Chapter 4 which begin by giving an overview of the addressed problem and then presenting the detailed problem formulation for multi-vehicle formation control. Next, the control algorithms and the proposed strategies for stabilizing the multi-vehicle formations are presented. Finally, the simulation scenarios and results are discussed.

The cooperative unmanned vehicles search of an environment is presented in Chapter 5. This chapter discusses the partitioning method used in this research and describes the developed search algorithm for the vehicles. The simulations performed

with different scenarios using the designed algorithm are described in Section 5.2 and the results of these simulations are presented in Section 5.3.

Chapter 6 concludes the dissertation by stating the conclusions that can be drawn from the research work carried out and discusses the future research work that can be done as a results of the work presented in this dissertation.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Formation Control

The formation control problems are generally categorized into consensus and distance-based. In the consensus-based formation control, the relative displacements between the vehicles are controlled to achieve the desired formation. In the distance-based control, the vehicles sense the relative position of their neighboring vehicles, using onboard sensors, with respect to their own local coordinate systems. Therefore, a main advantage of this control, compared to consensus-based algorithms [18], is that position measurements in a global coordinate frame are not required [19]. This is especially useful in global positioning system (GPS)-denied environments where unmanned vehicles are used for planetary explorations, indoor/outdoor navigation, and target tracking. Here, we consider the decentralized formation control problem where each vehicle uses its onboard sensors such as ultrasonic or infrared-based relative positioning sensors to obtain locally sensed information about the other vehicles.

Most of the available distance-based formation control results in the literature use the single integrator model for the vehicles' motion in the plane [20], [21], [22], [23], [24], [25]. In this work, in addition to the single integrator vehicle model which is beneficial for studying fundamental properties, the more realistic and practical double

integrator model for the vehicles' motion is also presented. Another contribution of this work is to provide a theoretical framework for real world applications which are often in three-dimensional space as opposed to the plane. Recently, some work has been done to extend the multi-vehicle formations to three-dimensional space [14], [26]. As opposed to the control approach presented in [26] for a three-dimensional tetrahedral formation with only four agents, there is no limit in this dissertation for the number of vehicles. The control of rigid formations in three dimensions using single and double integrator models is considered in [14]. The approach in this dissertation differs in that here the infinitesimally and minimally rigid three-dimensional formations are considered. In this case minimally rigid refers to multi-vehicle formations with minimum number of communication and control links between the vehicles. The formation acquisition of n agents in the plane was addressed in [7]. Here, the results in [7] are extended to three-dimensional space and stability analysis and sufficient conditions for the initial conditions that guarantees the convergence of the vehicles' formation to the desired framework are provided. The NN-based control of unmanned vehicle formations is also presented in this work. The formation control of vehicles with absolute positions in two-dimensional space using Hopfield NNs was presented in [27]. We note that there are a number of existing results in the literature that use NN for the control of multi-agent systems (e.g., [28], [30], [31], [33]). However, to the best of our knowledge, none use rigid graph theory to formulate and solve the formation control problem.

This work considers a non-hierarchical formation structure where there is no leader-follower in the formation. The formation structures with no hierarchy,

as opposed to the leader-follower or hierarchical structures, allow a balanced task distribution among the vehicles and are expected to be more robust to atmospheric disturbances and variations in the speed of the individual agents [32].

The concepts of layered sensing and formation control are combined in this work to address the multi-vehicle layered formation control problem. Layered sensing was first introduced by the U.S. Air Force Research Laboratory. It refers to the appropriate sensor or combination of sensors/platforms, infrastructure, and exploitation capabilities that generates situational awareness and directly supports tailored effects [29].

There are many applications for multi-vehicle layered formations. A layered formation of UAVs can be used for data collection, mapping, and inspection in industries such as forestry, agriculture, and oil extraction. A team of UGVs moving on the ground and multiple UAVs operating at a certain altitude in a formation can perform complex, cooperative surveillance tasks. A formation of unmanned systems can also be used for other tasks such as minesweeping and target tracking. An example of layered formation is a group of UUVs that move underwater and a coordinating ship that operates on the surface of water.

2.2 Rigid Graph Theory

The formation shape of a multi-vehicle system is represented by an undirected graph $G = (V, E)$ where $V = \{1, 2, \dots, n\}$ is the vertex set of this graph that represents the vehicles and E is the edge set that represents the communication links between the vehicles. The number of vertices and edges of G are denoted by $|V|$ and $|E|$, respectively.

A framework is a realization of a graph at given points in \mathbb{R}^d where $d \in \{2, 3\}$. A d -dimensional framework F is a pair (G, p) where $p = (p_1, \dots, p_n) \in \mathbb{R}^{dn}$ and $p_i \in \mathbb{R}^d$ is the coordinate of vertex i [7]. Given an arbitrary ordering of the edges of G , an edge function $\Phi_G : \mathbb{R}^{dn} \rightarrow \mathbb{R}^{|E|}$ associated with (G, p) is given by

$$\Phi_G(p) = (\dots, \|p_i - p_j\|^2, \dots), \quad (i, j) \in E, \quad (2.1)$$

where $\|\cdot\|$ denotes the Euclidean norm. The rigidity matrix $R(p) : \mathbb{R}^{dn} \rightarrow \mathbb{R}^{|E| \times dn}$ of (G, p) is defined as

$$R(p) = \frac{1}{2} \frac{\partial \Phi_G(p)}{\partial p}. \quad (2.2)$$

Two frameworks (G, p) and (G, q) are equivalent if $\Phi_G(p) = \Phi_G(q)$ and are congruent if $\|p_i - p_j\| = \|q_i - q_j\|$ for all $i, j \in V$ [7], [34]. In the case where two frameworks are equivalent but not congruent, then they are flip ambiguous [35]. The notion of flip ambiguity is illustrated in Figure 2.1.

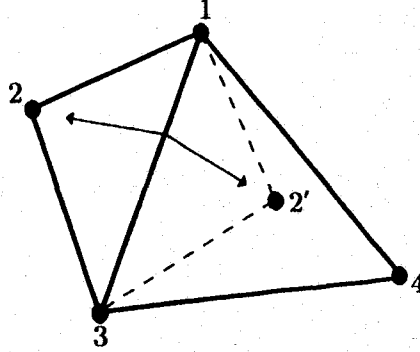


Figure 2.1: A noncongruent framework that depicts flip ambiguity (vertex 2 can be flipped over the edge (1,3) to the symmetric position 2').

Rigid graph theory plays a crucial role in solving the distance-based formation control problems since it naturally ensures that the inter-vehicle distance constraints of the desired formation are enforced through graph rigidity. Therefore, collisions

between vehicles are avoided while they acquire a formation [7]. In order to determine the rigidity of a formation, both the number of the edges and their distribution among the graph vertices matter. Laman's theorem is known to be a key result that is used in solving the rigidity-based formation control problems in two dimensions.

Theorem 2.1 ([36]). *A graph $G = (V, E)$ modeling a formation in two dimensions is rigid if and only if there exists a subgraph $G' = (V, E')$, $E' \subset E$ with $|E'| = 2|V| - 3$ such that for any $V' \subset V$, the associated induced subgraph $G'' = (V', E'')$ of G' with $E'' \subset E'$, satisfies $|E''| \leq 2|V'| - 3$.*

Figure 2.2 shows an example of flexible and rigid formations in two dimensions. In the flexible formation, part of the framework can be deformed by a smooth motion, while the distance between the vehicles that are connected with edges remain unchanged. In the rigid formation, the only smooth motions of the framework correspond to translation or rotation; therefore the formation cannot be deformed.

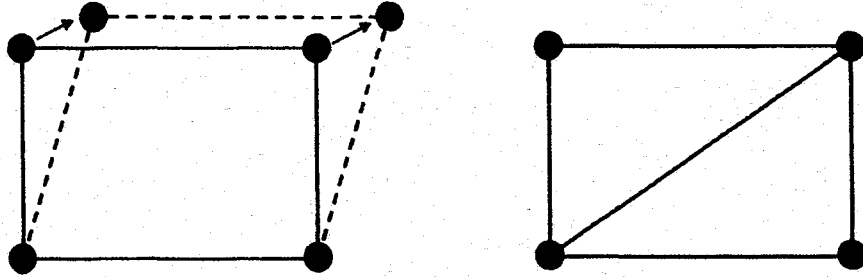


Figure 2.2: Flexible (left) and rigid (right) formations in two dimensions.

A graph is *minimally rigid* if it is rigid and if no single edge can be removed from the graph without causing the graph to lose its rigidity [37]. A graph $G = (V, E)$ is minimally rigid in two or three dimensions if and only if $|E| = 2|V| - 3$ or $|E| = 3|V| - 6$, respectively [35], [38].

A graph is *infinitesimally rigid* if it cannot flex or deform by even a very small amount. A framework (G, p) where $n > d$ and p is generic (the affine span of p is all of \mathbb{R}^d) is infinitesimally rigid if and only if $\text{rank}[R(p)] = dn - \binom{d+1}{2}$. Therefore, (G, p) is flexible if $|E| < dn - \binom{d+1}{2}$ [11], [40], [41]. A rigid framework is not always infinitesimally rigid. Figure 2.3 shows an example of rigid, infinitesimally rigid, and infinitesimally and minimally rigid frameworks in two dimensions. Note that the framework on the left in Figure 2.3 is not infinitesimally rigid, since it will have an infinitesimal deformation if force is applied to vertex c .

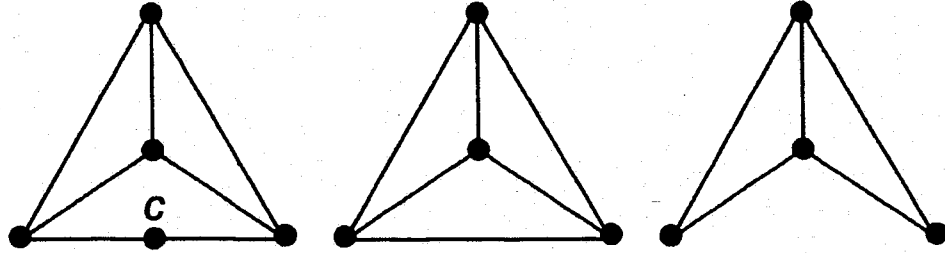


Figure 2.3: A rigid but not infinitesimally and minimally rigid framework (left), an infinitesimally but not minimally rigid framework (middle), and an infinitesimally and minimally rigid framework (right) in two dimensions.

Lemma 2.2. *If the framework $F = (G, p)$ is infinitesimally and minimally rigid in three dimensions, then $R(p)R^T(p)$ is invertible.*

Proof. We know that if F is infinitesimally and minimally rigid in three dimensions, then $\text{rank}[R(p)] = 3|V| - 6$ and $|E| = 3|V| - 6$. Therefore, $R(p)$ has full row rank. Since, $R^T(p)$ has full column rank and $\text{rank}[R(p)] = \text{rank}[R(p)R^T(p)]$, then $R(p)R^T(p) \in \mathbb{R}^{|E| \times |E|}$ is invertible. \square

The rigidity matrix $R(p)$ for the framework F is constructed with an arbitrary ordering of vertices and edges and has $3|V|$ columns and $|E|$ rows. The rows of $R(p)$

correspond to the edges of G . If there is an edge between vertices i and j then the entries $x_i - x_j$, $y_i - y_j$, $z_i - z_j$, $x_j - x_i$, $y_j - y_i$, and $z_j - z_i$ will fill columns $3i - 2$, $3i - 1$, $3i$, $3j - 2$, $3j - 1$, and $3j$ of $R(p)$, respectively. The entries of the other columns will be zero.

An undirected infinitesimally and minimally rigid graph $G = (V, E)$ in three dimensions is shown in Figure 2.4.

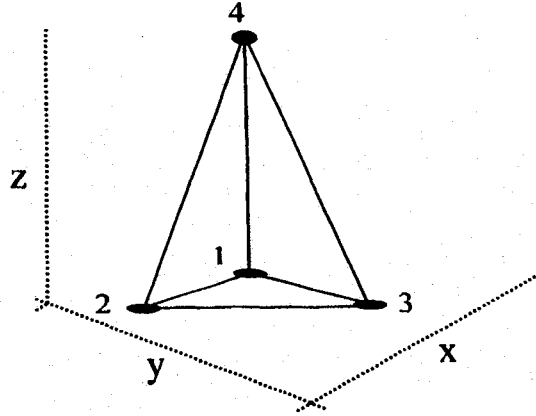


Figure 2.4: An infinitesimally and minimally rigid graph in three dimensions.

The corresponding rigidity matrix $R(p)$ is given by

$$R(p) = \begin{bmatrix} \tilde{p}_{12}^T & \tilde{p}_{21}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{p}_{23}^T & \tilde{p}_{32}^T & \mathbf{0} \\ \tilde{p}_{13}^T & \mathbf{0} & \tilde{p}_{31}^T & \mathbf{0} \\ \tilde{p}_{14}^T & \mathbf{0} & \mathbf{0} & \tilde{p}_{41}^T \\ \mathbf{0} & \tilde{p}_{24}^T & \mathbf{0} & \tilde{p}_{42}^T \\ \mathbf{0} & \mathbf{0} & \tilde{p}_{34}^T & \tilde{p}_{43}^T \end{bmatrix}, \quad (2.3)$$

where $\tilde{p}_{ij}^T = [x_i - x_j, y_i - y_j, z_i - z_j]$ and each $\mathbf{0}$ is a 1×3 vector of zeroes.

A surface in \mathbb{R}^3 is triangulated if it is covered with a collection of triangles and if any two triangles intersect, their intersection is a common edge or vertex [39]. A

graph $G = (V, E)$ is polyhedral if and only if there exists $p = (p_1, \dots, p_n) \in \mathbb{R}^{3|V|}$ such that $p_i \neq p_j$ for $i \neq j$ and $(i, j) \in E$ are the edges of a convex polyhedron in \mathbb{R}^3 [42]. A triangulated convex polyhedral surface with vertices only in the natural edges, is infinitesimally rigid. The natural edges are one-dimensional intersections of a support plane with the convex polyhedral surface [43].

Coning is a technique in rigidity which takes frameworks in \mathbb{R}^N to frameworks in \mathbb{R}^{N+1} , where $N \in \{1, 2, 3, \dots\}$, while preserving first-order rigidity (or infinitesimal rigidity) [44], [45]. In this work, the concept of coning is used for the case when there is only one upper layer vehicle. A graph $G = (V, E)$ is coned by adding a new vertex c , and adding edges from this vertex to all the original vertices in G . This will create the cone graph $G * \{c\}$ with $V(G * \{c\}) = V(G) \cup \{c\}$ where the cone vertex c is distinct from the vertices of G . If p_c is a configuration for $G * \{c\}$ and $H \cong \mathbb{R}^N$ is a hyperplane, then we denote p_H to be the projection of p_c from the cone vertex into H . We call p_c and p_H a projection pair of configurations.

A general version of the infinitesimal rigidity coning theorem is presented in [45] and its proof can be found in [44]. The following lemma is directed towards the cone framework (one upper layer vehicle in the framework) of interest in this work.

Lemma 2.3 ([44]). *Let G be a graph that represents a group of n agents in \mathbb{R}^2 , $G * \{c\}$ be the cone graph with cone vertex c representing the coordinating agent, and p^* and p_H be a projection pair of configurations. Then, $(G * \{c\}, p^*)$ is infinitesimally rigid in \mathbb{R}^3 if and only if (G, p_H) is infinitesimally rigid in \mathbb{R}^2 .*

2.3 Neural Networks

Artificial NNs, which are based on biological neuronal structures of interconnected nodes, have properties such as learning and adaptation, function approximation, classification, generalization, etc. A two-layer NN (Figure 2.5) is commonly used for closed-loop control purposes. This NN consists of a hidden layer with L nodes and an output layer with c nodes. The NN inputs are $x_1(t), x_2(t), \dots, x_k(t)$. The output $y(t)$ can be written as

$$y = W^T \sigma(V^T x), \quad (2.4)$$

where V and W are the first and second layer augmented weight matrices which contain biases in the first row and $\sigma(\cdot)$ is an activation function. In this work, the sigmoid (logistic) function was selected as the activation function.

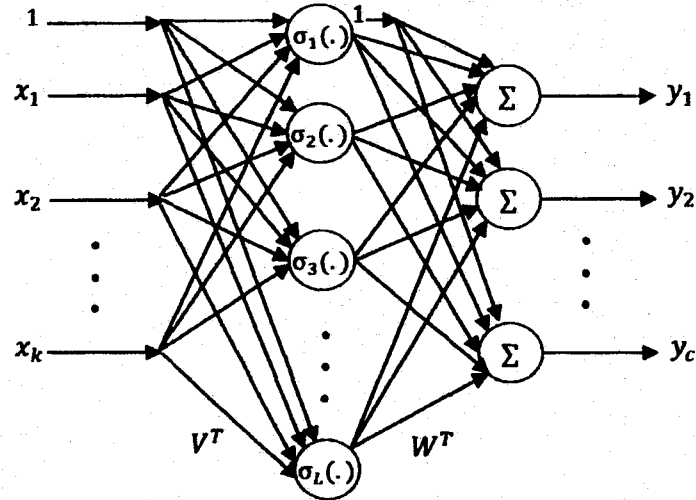


Figure 2.5: A two layer neural network.

The function approximation property of NNs plays an important role in control applications. The basic approximation result states that any smooth function $f(x)$ can be approximated arbitrarily closely over a compact set $\Omega \in \mathbb{R}^k$. That is, for a

positive constant ε_N , there exists a two layer NN with an ideal weight matrix W and a number of hidden layer nodes L such that

$$f(x) = W^T \sigma(V^T x) + \varepsilon, \quad (2.5)$$

where ε is the NN function approximation error and satisfies $\|\varepsilon\| < \varepsilon_N$. If the first layer weights and biases V are fixed, the NN is Linear-in-the-Parameter (LIP) and the approximation property can be satisfied by tuning the second layer weights and biases W . The first layer weights V are selected randomly and are not tuned. The second layer weights are tunable. The approximation holds [46] for such NN, with approximation error convergence to zero of order $O(C/\sqrt{L})$, where C is independent of L . It is assumed that the approximating weights W are bounded such that $\|W\|_F \leq W_m$, where $\|W\|_F$ is the Frobenius norm. Given a matrix $A = [a_{ij}]$, the Frobenius norm is defined by

$$\|A\|_F^2 = \sum_{i,j} a_{ij}^2 = \text{tr}(A^T A), \quad (2.6)$$

with $\text{tr}()$ being the trace.

Definition 2.4 ([67]). *Consider the nonlinear system*

$$\dot{x} = f(t, x). \quad (2.7)$$

The solutions of (2.7) are uniformly ultimately bounded with ultimate bound b if there exists positive constants a and c , independent of $t_0 \geq 0$, and for every $a \in (0, c)$, there is $T = T(a, b) \geq 0$, independent of t_0 , such that

$$\|x(t_0)\| \leq a \Rightarrow \|x(t)\| \leq b, \quad \forall t \geq t_0 + T. \quad (2.8)$$

2.4 Cooperative Search and Partitioning Methods

The use of autonomous vehicles such as UGVs and UAVs for performing tasks such as environmental monitoring, hazardous chemical detection, exploration of dangerous areas, and search of an environment has many applications to the commercial and defense sector. Additionally, through the use of sensors and wireless radios for communication, intelligent unmanned vehicles can coordinate complex tasks and missions to further enhance their capabilities.

A local search metaheuristic is an algorithmic process of selecting one potential solution by iterating through a set of solutions within a local neighborhood. The fundamental concept of this search algorithm is applied in the Simulated Annealing, Genetic Algorithm, and Tabu Search [47]. Local search heuristics are applicable in optimization problems such as determining optimal solutions for the traveling salesman problem [48] and performing navigational-based search with autonomous vehicles [49]. The well known success of Simulated Annealing, Tabu Search, and Genetic Algorithm has resulted in a flow of literature in recent years. Some of the factors that distinguish these metaheuristics from others include: their reference to optimization mechanisms in nature (in the case of Simulated Annealing and Genetic Algorithm), general applicability, and flexibility of the approach [47]. This section aims to describe the basic versions of these three general heuristic approaches in more details.

Simulated Annealing algorithm originates from thermodynamics and metallurgy and mimics the process of annealing (in which a liquid (e.g. molten iron) is cooled slowly until it becomes stable in a solid form) to obtain an optimum solution. In this

algorithm, at the start, a solution is drawn randomly. The next solution is chosen by perturbing the current solution by an amount $pert(T)$ which is found using [49], [51]

$$pert(T) = k_d T \zeta, \quad (2.9)$$

where k_d is a constant used to scale the results, T is the so-called current ‘temperature’, and ζ is a random number between 0 and 1. If the current solution does not have a better evaluation value than the current best solution, then the following Equation is used to calculate a probability based on the evaluation value of the current solution and that of the current best solution:

$$P = \frac{F_{prev} - F_{new}}{T}, \quad (2.10)$$

where P is the probability, F_{prev} is the current best evaluation value, and F_{new} is the current evaluation value. This stage of the algorithm is called the Metropolis Criterion. The calculated probability is then compared to a random number γ which has a range from 0 to 1. If the value of $P > \gamma$, then the current solution is selected as the new best solution and if $P < \gamma$, then the solution is rejected. This feature allows the algorithm to avoid local minimum. A local minimum is a solution with a minimum value compared to its neighborhood environment, but it does not have the global minimum value which belongs to the target solution. Therefore, when the algorithm reaches a local minimum point, it will not move from this solution since it is surrounded with solutions that have a larger evaluation function values. Another stage of the Simulated Annealing algorithm is the Annealing Schedule which is used to systematically reduce the distance from the current solution and the next solution

as the search proceeds. The Annealing Scheduling is given by

$$AS(T) = \eta^k T_0, \quad (2.11)$$

where η is the rate of decay of the Annealing Schedule ranged from 0 to 1, k is the number of iterations, and T_0 is the initial temperature.

Genetic Algorithm is inspired by the genetic evolution of a species and deals with searching the neighborhood of a population rather than the neighborhood of a single solution [47]. Note that many terms from Genetics are used in the Genetic Algorithm literature. At first, the algorithm randomly generates a population called Chromosomes as a representation of the parameters that are to be optimized. This population is then evolved toward better solutions. In the next stage of the algorithm which is called Selection, the performance of each chromosome is evaluated and the parents of the next generation are chosen. There are different methods such as rank-based and probability-based for selecting candidates for reproduction [49]. The process of reproduction is represented by the next stage of the algorithm called Crossover. In this process, a number of the genes from one of the parents are swapped with the same number of the genes from the other parent. The chromosomes that are produced from this process are called children and they replace the adults in the next generation. The next stage of the algorithm is called mutation. In nature, mutation occurs when there is a sudden random change in the chromosomal properties. In the Genetic Algorithm, mutation is the random selection of a percentage of the children and slightly altering the value of their genes. The process continues by evaluating new populations until the stop condition is met [47], [50], [51].

Tabu search, which in its present form was first introduced by Glover [52], has many applications such as graph partitioning [53] and vehicle routing [54]. Different implementations of Tabu search have been used to solve vehicle routing problems and have been amongst the most effective methods used [54]. Two main advantages of Tabu search are local optima avoidance and cycling avoidance [54]. Cycling in search happens when the algorithm continuously moves between certain groups of solutions [49]. Two main aspects of Tabu search are Tabu list and Aspiration Criteria. Tabu list [52], [54] is a beneficial function used in Tabu search that records the recent moves by the algorithm during the search process. The items are stored in the Tabu list until the list reaches its maximum length and new items replace them on the list or a certain length of time is passed [49]. Tabu list prevents moving back to previously visited solutions by functioning like a short memory. Sometimes a point in the Tabu list may have a better evaluation value than any other currently available solution in the surrounding environment. Aspiration Criteria is a method used in Tabu search that decides if a solution that currently exists on the Tabu list should be removed from this list and become available as a solution that can be moved to [49], [54].

The proposed search algorithm in this dissertation is a variant of Tabu Search algorithm which combines the standard Tabu and Random search algorithms for finding the target(s) while providing obstacle avoidance capability. In a search performed using Random search algorithm, the solutions are chosen randomly and tested. This type of search will continue until a stop condition is met [49]. In this work, in order to address the challenge of searching a given environment for a target using a team of unmanned vehicles, the proposed algorithm for coordination of these vehicles will be

used in a partitioned search area for effective coverage. There are only three ways of tiling an environment in a regular manner: triangular tilings, square tilings, and hexagonal tilings [55]. These three types of regular tilings are shown in Figure 2.6.

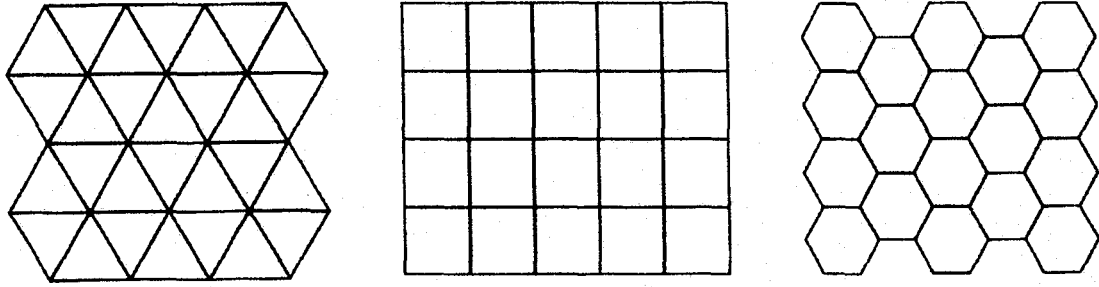


Figure 2.6: Triangular tiling, square tiling, and hexagonal tiling.

To guarantee symmetry, the tiling must be uniform. A tiling is uniform if a point, p , in one tile forms a lattice with the set of duplicates of p in all tiles [56]. A grid consisting of triangle shaped tiles will not be uniform because alternate triangles have to be rotated 180 degrees to create a continuous tiling pattern and a given point, p , in one tile does not form a lattice with all of its duplicates in the other tiles. Therefore, the only two possible uniform tilings are: square tiling and hexagonal tiling. We call the tiling of a given environment into squares and hexagons, square partitioning and hexagonal partitioning, respectively.

In this chapter, square and hexagonal partitioning which are the two possible types of uniform partitioning are discussed. Hexagonal partitioning has been presented in [57] to navigate mobile agents and Unmanned Aerial Vehicles (UAVs) in the search area. In [58], the authors used a degree-3 hexagonal lattice to cover the search region. When searching for a target using unmanned vehicles, one of the advantages of the hexagonal partitioning is that each vehicle can move in six different directions to the

adjacent hexagonal cells at each time step. In a hexagonal grid, the distance between the center of each hexagonal cell and the center of all the neighboring cells is constant, so the hexagonal partitioning provides for equidistant movement in every direction. This characteristic is shown in Figure 2.7a.

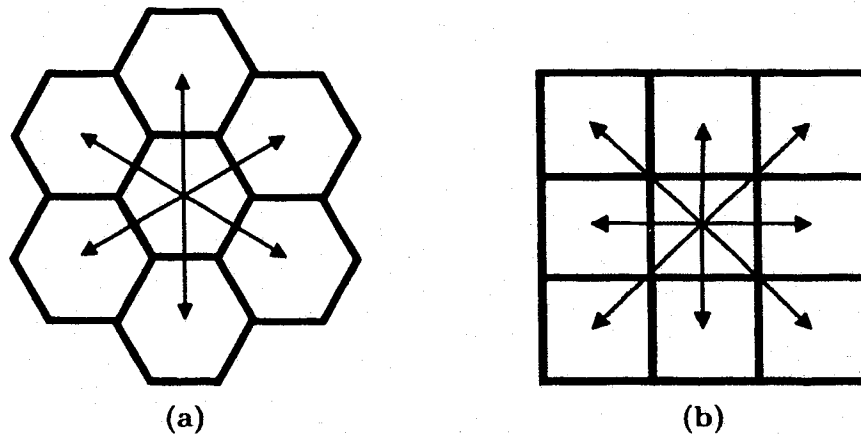


Figure 2.7: Hexagonal (a) and square (b) travel path lengths.

A number of search techniques have been developed using square partitioning of the search space [16], [17], [59]. In square partitioning, each square is surrounded by eight adjacent squares. However, even though unmanned vehicles can move in eight directions, not all of the neighboring cells share an edge. Additionally in this type of partitioning, some cells contact each other at only one point. This property of square grids causes difficulties because of its inconsistent nearest neighborhood problem. Therefore, if the vehicles move from the center of one square to the center of another cell, not all the movements in all directions will be equidistant. The vehicles must travel a greater distance to the center of each of the four diagonal adjacent square cells than the distance to the center of each adjacent cell that shares an edge with the

cell of the vehicle. The corresponding travel path length for the square partitioning is shown in Figure 2.7b.

One of the major advantages of using a hexagonal grid comes from the understanding that in an outdoor environment there are no walls, rooms, and corridors that arrange neatly into a square grid. Hexagonal grids, however, allow for a more natural and flexible mapping of an environment. The reason for this is that unlike squares which have all 90 degree angles, regular hexagons have 120 degree interior angles, making the structure more conformal than a square. This feature of hexagons makes hexagonal grids more suitable for mapping an outdoor environment and is ideal for integrating the non-ideal characteristics of realistic terrain features into the map.

Another advantage of using hexagonal partitioning is that the unmanned vehicles travel to less number of cells to perform the search and therefore save time and energy compared to if the area is partitioned into square cells. Consider the hexagonal and square cells with the same perimeter, p . The sides of the regular hexagonal cell is $\frac{p}{6}$ and in case of square cell is $\frac{p}{4}$, see Figure 2.8.

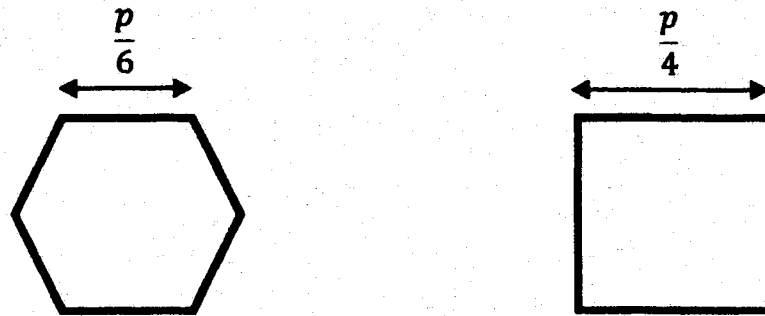


Figure 2.8: Comparison of hexagonal and square partition calculations.

Areas of hexagonal and square cells are given by:

$$A_H = \frac{\sqrt{3}p^2}{24} \quad (2.12)$$

and

$$A_S = \frac{p^2}{16}, \quad (2.13)$$

where A_H and A_S are areas of hexagonal and square cells, respectively.

Each square cell is composed of four vertices and any single non-boundary vertex is shared by four square cells. On the other hand, each hexagonal cell is composed of six vertices and any single non-boundary vertex is shared by three hexagonal cells. Therefore, when partitioning an area with hexagonal cells, fewer cells are needed compared to an equally sized area partitioned with square cells. Hexagonal grids have been used in path planning for UAVs and have been shown to result in a significantly better performance since these types of grids support simple dynamics [60].

Finally, many studies have been done on the beehive structure of honeycomb built by honeybees in the nature. Amongst the many reasons why honeybees construct their beehive with hexagonal cells are: optimized cell density, increased structural strength to store honey, and the maximized space for living and storage with the given material provided by these cells [61].

As a result of these advantages, the hexagonal structure was the type of grid chosen for partitioning the search environment in this work. Since hexagonal grids resemble a beehive structure in nature, this grid structure is referred to as a “beehive structure” and this method of partitioning is referred to as “beehive partitioning”.

CHAPTER 3

NON-PLANAR MULTI-VEHICLE LAYERED FORMATION CONTROL

The ability to have multiple vehicles, whether on the ground or airborne, autonomously perform searching, detecting, and tracking tasks while one or more vehicles move in an upper layer above the multi-vehicle group is the main objective of this research. In addition to autonomously performing a task, all the vehicles must acquire a formation structure. An example of this scenario is shown in Figure 3.1 and includes UGVs and UAVs. The upper layer vehicles can be with or without sensors for the ground supporting mission.

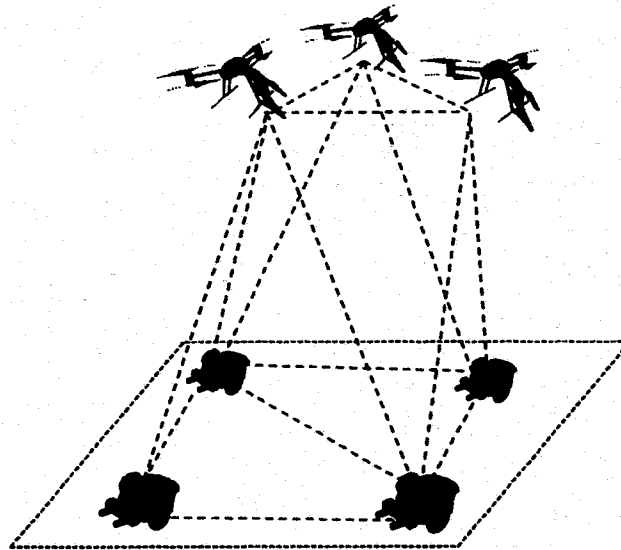


Figure 3.1: Formation control concept under the layered sensing framework.

In this dissertation, the *layered formation control* of n lower layer vehicles that belong to a plane Q , modeled by an undirected graph and r upper layer vehicles that are moving outside of the plane Q , as shown in Figure 3.1, is addressed. Triangulation of the layered formation is proposed to obtain an infinitesimally and minimally rigid framework in three-dimensional space.

As a special case, the layered formation control with one vehicle in the upper layer (i.e., $r = 1$) is also addressed (see Figure 3.2). Here, the concept of coning [44], [45] is used to create a three-dimensional formation that retains the properties of infinitesimal and minimal rigidity of the framework for the lower layer vehicles in plane Q .

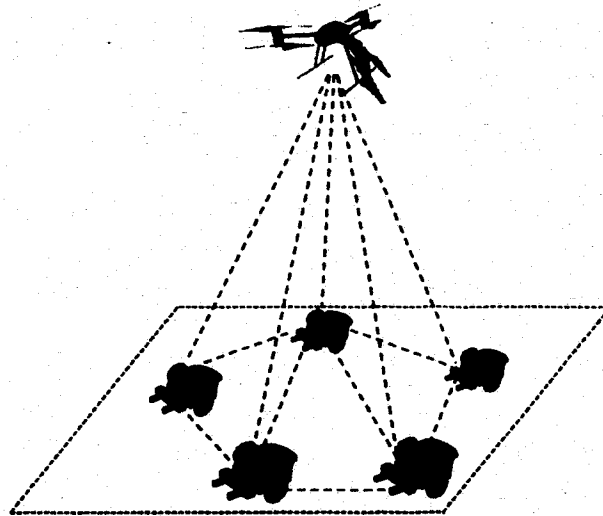


Figure 3.2: Formation control concept under the coning framework.

3.1 Problem Formulation

In this work, it is assumed that vehicles are equipped with sensors such as ultrasonic or infrared-based relative positioning sensors [62] that allow them to measure the distance and direction between selected vehicles.

3.1.1 Non-Planar Multi-Vehicle Layered Formation Control with Single Integrator Model

Consider a system of n vehicles in the plane Q and r vehicles in an upper layer outside of the plane Q , modeled by the single integrator (model \mathcal{W}_1) [4], [21], [22], [63]

$$\mathcal{W}_1 : \quad \dot{p}_{l_i} = u_{l_i}, \quad i = 1, \dots, n, n+1, \dots, n+r, \quad (3.1)$$

where $p_{l_i} = (x_{l_i}, y_{l_i}, 0) \in \mathbb{R}^3$ for $i = 1, \dots, n$ is the location of the i -th vehicle in local coordinates in plane Q that is defined by $z = 0$.

The location of the upper layer vehicles is defined by $p_{l_i} = (x_{l_i}, y_{l_i}, z_{l_i}) \in \mathbb{R}^3$ for $i = n+1, \dots, n+r$, where z_{l_i} is the distance of the vehicles from plane Q . The control input for the i -th vehicle in plane Q and the upper layer vehicles is $u_{l_i} \in \mathbb{R}^3$ for $i = 1, \dots, n, n+1, \dots, n+r$.

Consider a formation in \mathbb{R}^3 that can be triangulated for the layered system mentioned above and described by the framework $F_l^* = (G_l^*, p_l^*)$ where $G_l^* = (V_l, E_l)$ and $p_l^* = (p_{l_1}^*, \dots, p_{l_n}^*, p_{l_{n+1}}^*, \dots, p_{l_{n+r}}^*)$. Let F_l^* represent the desired layered formation of the system of $n+r$ vehicles.

Given the actual formation $F_l(t) = (G_l^*, p_l(t))$ where $p_l = (p_{l_1}, \dots, p_{l_n}, p_{l_{n+1}}, \dots, p_{l_{n+r}})$ and assuming that at $t = 0$, $\|p_{l_i}(0) - p_{l_j}(0)\| \neq d_{l_{ij}}$ for $(i, j) \in E_l$, where $d_{l_{ij}} = \|p_{l_i}^* - p_{l_j}^*\| > 0$ is the constant desired distance between vehicles i and j in the layered formation, the control objective is to design the control input u_{l_i} such that

$$\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}} \text{ as } t \rightarrow \infty, \quad (i, j) \in E_l. \quad (3.2)$$

3.1.2 Non-Planar Multi-Vehicle Layered Formation Control with Double Integrator Model

In this section, the problem of non-planar multi-vehicle layered formation control is formulated using the double integrator model [7], [64]. The use of a double integrator model is more practical for a multi-vehicle formation. In this case, the formation reaches a state of balance when the agents are not scattered and their velocities match [65].

Consider a system of n vehicles in the plane Q and r vehicles in an upper layer outside of the plane Q , modeled by the double integrator (model \mathcal{W}_2)

$$\mathcal{W}_2 : \begin{cases} \dot{p}_{l_i} = v_{l_i} \\ \dot{v}_{l_i} = u_{l_i}, \quad i = 1, \dots, n, n+1, \dots, n+r, \end{cases} \quad (3.3)$$

where $v_{l_i} \in \mathbb{R}^3$ for $i = 1, \dots, n, n+1, \dots, n+r$ is the velocity and u_{l_i} is the acceleration-level control input for the i -th vehicle. The control objective is to design the control input u_{l_i} such that (3.2) is satisfied.

3.2 Control Algorithms

Here, the results from [7] are extended to solve the formation acquisition and stabilization problem in three dimensions where the upper layer vehicles are operating outside of the plane defined by the lower layer vehicles. The approach is presented for both the single and double integrator vehicle models.

3.2.1 Single Integrator Vehicle Model

Define the relative position of two vehicles in the layered formation as

$$\tilde{p}_{l_i, j} = p_{l_i} - p_{l_j}, \quad (i, j) \in E_l. \quad (3.4)$$

The distance error for the group of $n + r$ vehicles and the corresponding distance error dynamics are given by

$$e_{i,j} = \|\tilde{p}_{l_{ij}}\| - d_{l_{ij}}, \quad (3.5)$$

$$\begin{aligned} \dot{e}_{l_{ij}} &= \frac{d}{dt} \sqrt{\tilde{p}_{l_{ij}}^T \tilde{p}_{l_{ij}}} = (\tilde{p}_{l_{ij}}^T \tilde{p}_{l_{ij}})^{-\frac{1}{2}} \tilde{p}_{l_{ij}}^T (u_i - u_j) \\ &= \frac{\tilde{p}_{l_{ij}}^T (u_i - u_j)}{e_{l_{ij}} + d_{l_{ij}}}. \end{aligned} \quad (3.6)$$

For control algorithm development and stability analysis, consider the potential function [7], [21], [22]

$$M_{ij} = \frac{1}{4} e_{l_{ij}}^2 (e_{l_{ij}} + 2d_{l_{ij}})^2, \quad (3.7)$$

where M_{ij} is positive definite and radially unbounded in $e_{l_{ij}}$. Define

$$M(e_l) = \sum_{(i,j) \in E_l} M_{ij}(e_{l_{ij}}), \quad (3.8)$$

where $M_{ij}(e_{l_{ij}})$ was given in (3.7). The time derivative of (3.8) is then given by

$$\dot{M} = \sum_{(i,j) \in E_l} \frac{\partial M_{ij}}{\partial e_{l_{ij}}} \frac{\tilde{p}_{l_{ij}}^T (u_i - u_j)}{e_{l_{ij}} + d_{l_{ij}}}. \quad (3.9)$$

As shown in [7], it follows from (2.1), (2.2), (3.5), and (3.6) that (3.9) can be expressed as

$$\dot{M} = \beta^T(e_l) R(p_l) u_l, \quad (3.10)$$

where $u_l = (u_{l_1}, \dots, u_{l_n}, u_{l_{n+1}}, \dots, u_{l_{n+r}}) \in \mathbb{R}^{3(n+r)}$, $R(p_l) \in \mathbb{R}^{|E_l| \times 3(n+r)}$, $e_l = (\dots, e_{l_{ij}}, \dots) \in \mathbb{R}^{|E_l|}$, and

$$\begin{aligned} \beta(e_l) &= (\dots, \frac{\frac{\partial M_{ij}}{\partial e_{l_{ij}}}}{e_{l_{ij}} + d_{l_{ij}}}, \dots) \\ &= (\dots, e_{l_{ij}}(e_{l_{ij}} + 2d_{l_{ij}}), \dots), \quad (i, j) \in E_l. \end{aligned} \quad (3.11)$$

The terms in e_l and $\beta(e_l) \in \mathbb{R}^{|E_l|}$ are ordered the same way as in (2.1).

The following theorem relates the triangulation of the layered framework in three-dimensional space and the infinitesimal and minimal rigidity of the formation.

Theorem 3.1 ([43]). *Let $F_l^* = (G_l^*, p_l^*)$, which consists of n vehicles in the plane Q and r upper layer vehicles, be the desired framework for a layered formation in \mathbb{R}^3 . If the surface of F_l^* is arbitrarily triangulated, then the layered formation F_l^* is infinitesimally and minimally rigid in \mathbb{R}^3 .*

The proof for infinitesimal rigidity of the framework F_l^* in Theorem 3.1 is presented in [43], [66]. Such a framework has $|E_l| = 3|V_l| - 6$ edges and therefore is minimally rigid in three dimensions [35], [38].

Corollary 3.2 ([7]). *Define the function*

$$\Omega(F_l, F_l^*) = \sum_{(i,j) \in E_l} (\|p_{l_i} - p_{l_j}\| - \|p_{l_i}^* - p_{l_j}^*\|)^2. \quad (3.12)$$

If F_l^ is infinitesimally rigid and $\Omega(F_l, F_l^*) \leq \varepsilon$ where ε is a sufficiently small positive constant, then F_l is also infinitesimally rigid.*

The following theorem shows that the control law [7]

$$u_l = -k_c R^T(p_l) \beta(e_l), \quad (3.13)$$

where k_c is a positive control gain, achieves the local asymptotic stability of the desired layered formation $F_l^* = (G_l^*, p_l^*)$. Note that from (2.1), the rigidity matrix function R is dependent on \tilde{p}_l but in this work the argument of this function will be given as p_l .

Theorem 3.3. *Given a group of n lower layer vehicles and $r \geq 1$ upper layer vehicles with a layered formation $F_l(t) = (G_l^*, p_l(t))$ in \mathbb{R}^3 and modeled by \mathcal{W}_1 , let δ be a positive constant and $\tilde{p}_l(0)$, where $\tilde{p}_l = (\dots, \tilde{p}_{ij}, \dots) \in \mathbb{R}^{3|E_l|}$, is the initial condition. If*

$\tilde{p}_l(0) \in \mathcal{S}_1$ where

$$\mathcal{S}_1 = \left\{ \tilde{p}_l \in \mathbb{R}^{3|E_l|} \left| \left| \|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2 \right| \leq \frac{2\sqrt{\delta}}{\sqrt{|E_l|}}, (i, j) \in E_l \right. \right\}, \quad (3.14)$$

then $F_l(t)$ is infinitesimally and minimally rigid and the control law (3.13) achieves the local asymptotic stability of the desired formation F_l^* ensuring $\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$.

Proof. Theorem 3.1 ensures the infinitesimal and minimal rigidity of the layered formation F_l^* in three dimensions. Since F_l^* is minimally rigid and has the same number of edges as $F_l(t)$, we know that $F_l(t)$ is also minimally rigid for all time. Note that (3.8) can be written as

$$\begin{aligned} M(e_l) &= \frac{1}{4} \sum_{(i,j) \in E_l} [(\|\tilde{p}_{l_{ij}}\| - d_{l_{ij}})^2 (\|\tilde{p}_{l_{ij}}\| + d_{l_{ij}})^2] \\ &= \frac{1}{4} \sum_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2. \end{aligned} \quad (3.15)$$

Condition $M(e_l) \leq \delta$ is equivalent to

$$\sum_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 \leq 4\delta. \quad (3.16)$$

A sufficient condition for (3.16) is given by

$$\left| \|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2 \right| \leq \frac{2\sqrt{\delta}}{\sqrt{|E_l|}}, \quad (i, j) \in E_l. \quad (3.17)$$

Therefore $\tilde{p}_l \in \mathcal{S}_1$ implies $M(e_l) \leq \delta$ (valid for all $t > 0$). Lemma 2 in [7] establishes the equivalence between $M(e_l) \leq \delta$ and $\Omega(F_l, F_l^*) \leq \varepsilon$ and since F_l^* is infinitesimally rigid, Corollary 3.2 shows that the formation F_l is also infinitesimally rigid for $\tilde{p}_l \in \mathcal{S}_1$. Therefore, according to Lemma 2.2, $R(p_l)R^T(p_l)$ is invertible for $\tilde{p}_l \in \mathcal{S}_1$.

Substituting (3.13) into (3.10) yields

$$\dot{M} = -k_c \beta^T(e_l) R(p_l) R^T(p_l) \beta(e_l). \quad (3.18)$$

Therefore,

$$\dot{M} \leq -k_c \lambda_{\min}(R(p_l) R^T(p_l)) L(e_l) \quad \text{for } \tilde{p}_l(t) \in \mathcal{S}_l, \quad (3.19)$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue and $L(e_l) = \beta^T(e_l) \beta(e_l)$. It follows from the negative definiteness of (3.18) that the level sets of M are invariant [67] for $\tilde{p}_l(0) \in \mathcal{S}_l$ and $e_l = 0$ is asymptotically stable for $\tilde{p}_l(0) \in \mathcal{S}_l$. Therefore, $\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$ and the control law (3.13) achieves the local asymptotic stability of the desired formation $F_l^* = (G_l^*, p_l^*)$. \square

Remark 1. *Note that Theorem 3.3 provides a control law for local asymptotic stability of the formation and sufficient conditions for the set of allowed initial conditions \mathcal{S}_l that guarantees the convergence to the desired formation. This result establishes a region for local asymptotic stability, but it does not show how to determine such region. That is, it is an existence result, rather than a constructive result, with respect to the stability region. The value of δ can be found by trial-and-error only.*

Remark 2. *The developed control is decentralized in the sense that the control input for each vehicle is only dependent on the relative position and velocity of the neighboring agents in the formation and vehicles' own absolute velocity.*

The following lemma uses the absolute initial and desired positions of the vehicles and shows that (3.13) achieves the local asymptotic stability of the desired formation F_l^* . Therefore, Lemma 3.4 establishes the conditions that prevent convergence to ambiguous formations.

Lemma 3.4. *Given model \mathcal{W}_1 , let δ be a positive constant and $p_l(0) \in \mathbb{R}^{3(n+r)}$ is the initial condition. If $p_l(0) \in \mathcal{S}_2$ where*

$$\mathcal{S}_2 = \left\{ p_l \in \mathbb{R}^{3(n+r)} \mid \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\| + D_l \leq \sqrt[4]{\frac{\delta}{4|E_l|}} \right\}, \quad (3.20)$$

and $D_l = \max(d_{l_{ij}})$, then the control law (3.13) achieves the local asymptotic stability of the desired formation F_l^* and ensures $\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$.

Proof. A sufficient condition for (3.17) is

$$\max_{(i,j) \in E_l} \left| \|p_{l_i} - p_{l_j}\| - d_{l_{ij}} \right| \max_{(i,j) \in E_l} (\|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\| + d_{l_{ij}}) \leq \frac{2\sqrt{\delta}}{\sqrt{|E_l|}}. \quad (3.21)$$

Using inequality

$$\|p_{l_i} - p_{l_j}\| \leq \|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\| + d_{l_{ij}}, \quad (3.22)$$

a sufficient condition for (3.21) is

$$\max_{(i,j) \in E_l} (\|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\|) \max_{(i,j) \in E_l} (\|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\| + 2d_{l_{ij}}) \leq \frac{2\sqrt{\delta}}{\sqrt{|E_l|}}. \quad (3.23)$$

A sufficient condition for (3.23) is given by

$$\max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\| + D_l \leq \sqrt[4]{\frac{\delta}{4|E_l|}}. \quad (3.24)$$

Using Theorem 3.3, it follows that $\|p_{l_i}(t) - p_{l_i}^*\| \rightarrow d_{l_i}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$ and the control law (3.13) achieves the local asymptotic stability of the desired formation $F_l^* = (G_l^*, p_l^*)$. \square

Remark 3. *Note that Lemma 3.4 provides sufficient conditions for the formation convergence to the desired formation in terms of initial positions of each vehicle. An upper bound of the norm of initial position error for each vehicle is given by (3.24). Conservatism of the bound (3.24) depends on uniformity of distances d_{l_i} , i.e., more uniform distances will cause a less conservative bound.*

Remark 4. *The system of n vehicles in the plane Q and a single upper layer vehicle outside of the plane Q is considered a special case for non-planar multi-vehicle layered formation control with single [68] and double integrator models. The triangulation method still applies and is referred to as coning which allows for retaining the infinitesimal and minimal rigidity of the n planar vehicles in three-dimensional space.*

Note that the control algorithms in this section are applicable to a cone framework. In such a framework, the coning method (Lemma 2.3) ensures that the layered formation $F_l^*(t) = (G_l^*, p_l^*(t))$ is infinitesimally rigid in \mathbb{R}^3 [68]. Since the cone framework is triangulated and has $|E_l| = 3|V_l| - 6$ edges, F_l^* is also minimally rigid in three dimensions.

3.2.2 Double Integrator Vehicle Model

Define the relative position of two vehicles as in (3.4). The distance error for the the group of n lower layer vehicles and r upper layer vehicles is defined in (3.5) and distance error dynamics are given by

$$\dot{e}_{l_{ij}} = \frac{\tilde{p}_{l_{ij}}^T (v_{l_i} - v_{l_j})}{e_{l_{ij}} + d_{l_{ij}}}, \quad (3.25)$$

where (3.3) was used. Consider the potential function in (3.7). Define

$$M_1(e_l) = \sum_{(i,j) \in E_l} M_{ij}(e_{l_{ij}}). \quad (3.26)$$

The time derivative of (3.26) along (3.25) is given by

$$\dot{M}_1 = \sum_{(i,j) \in E_l} \frac{\partial M_{ij}}{\partial e_{l_{ij}}} \frac{\tilde{p}_{l_{ij}}^T (v_{l_i} - v_{l_j})}{e_{l_{ij}} + d_{l_{ij}}}. \quad (3.27)$$

It follows from (2.1), (2.2), and (3.5) that (3.27) can be written as

$$\dot{M}_1 = \beta^T(e_l) R(p_l) v_l, \quad (3.28)$$

where $v_l = (v_{l_1}, \dots, v_{l_n}, v_{l_{n+1}}, \dots, v_{l_{n+r}}) \in \mathbb{R}^{3(n+r)}$. Using the backstepping method [7], [69],

consider the variable $s_l = (s_{l_1}, \dots, s_{l_n}, s_{l_{n+1}}, \dots, s_{l_{n+r}}) \in \mathbb{R}^{3(n+r)}$ defined as

$$s_l = v_l - f_l, \quad (3.29)$$

where $f_l = (f_{l_1}, \dots, f_{l_n}, f_{l_{n+1}}, \dots, f_{l_{n+r}}) \in \mathbb{R}^{3(n+r)}$ is a virtual velocity input for the layered formation that is given by

$$f_l = -k_v R^T(p_l) \beta(e_l), \quad (3.30)$$

and k_v is a positive constant. Define the total potential function

$$M_2(e_l, s_l) = M_1(e_l) + \frac{1}{2} s_l^T s_l. \quad (3.31)$$

Taking the time derivative of (3.31) and using (3.3), (3.28), and (3.29) yields

$$\begin{aligned}\dot{M}_2 &= \beta^T(e_l)R(p_l)v_l + s_l^T \dot{s}_l \\ &= \beta^T(e_l)R(p_l)f_l + s_l^T[u_l + R^T(p_l)\beta(e_l) - \dot{f}_l],\end{aligned}\tag{3.32}$$

where $\dot{f}_l = -k_v(\dot{R}^T(p_l)\beta(e_l) + R^T(p_l)\dot{\beta}(e_l))$ and $u_l = (u_{l_1}, \dots, u_{l_n}, u_{l_{n+1}}, \dots, u_{l_{n+r}}) \in \mathbb{R}^{3(n+r)}$ is the control input for the layered formation.

The following theorem shows that the control law [7]

$$u_l = -k_a s_l + \dot{f}_l - R^T(p_l)\beta(e_l),\tag{3.33}$$

where k_a is a positive control gain, achieves the asymptotic stability of the desired layered formation $F_l^* = (G_l^*, p_l^*)$.

Theorem 3.5. *Given a group of n lower layer vehicles and $r \geq 1$ upper layer vehicles with a layered formation $F_l(t) = (G_l^*, p_l(t))$ in \mathbb{R}^3 and modeled by \mathcal{W}_2 , let δ be a positive constant and $\tilde{p}_l(0) \in \mathbb{R}^{3|E_l|}$ and $v_l(0) \in \mathbb{R}^{3(n+r)}$ are the initial conditions. If $(\tilde{p}_l(0), v_l(0)) \in \mathcal{T}_1$ where*

$$\begin{aligned}\mathcal{T}_1 = \left\{ (\tilde{p}_l, v_l) \in \mathbb{R}^{3|E_l|} \times \mathbb{R}^{3(n+r)} \left| \frac{1}{2}|E_l| \max_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 + |V_l| \max_{i \in V_l} (\|v_{l_i}\| \right. \right. \\ \left. \left. + k_v |\Psi_i| \max_{j \in \Psi_i} \|\tilde{p}_{l_{ij}}\|^3)^2 \leq 2\delta \right\},\end{aligned}\tag{3.34}$$

then $F_l(t)$ is infinitesimally and minimally rigid and the control law (3.33) achieves the asymptotic stability of the desired formation F_l^* and ensures $\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$.

Proof. Note that (3.31) can be written as

$$M_2(e_l, s_l) = \frac{1}{4} \sum_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 + \frac{1}{2} \sum_{i \in V_l} \|s_{l_i}\|^2.\tag{3.35}$$

Condition $M_2(e_l, s_l) \leq \delta$ is equivalent to

$$\frac{1}{2} \sum_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 + \sum_{i \in V_l} \|s_{l_i}\|^2 \leq 2\delta. \quad (3.36)$$

Substituting (3.29) into (3.36) and knowing that

$$f_{l_i} = -k_v \sum_{j \in \Psi_i} \tilde{p}_{l_{ij}} c_{l_{ij}} (c_{l_{ij}} + 2d_{l_{ij}}), \quad (3.37)$$

where $\Psi_i(E_l) = \{j \in V_l | (i, j) \in E_l\}$ is a set of all neighboring vertices to vertex (vehicle) i in graph G_l^* , a sufficient condition for (3.36) is

$$\frac{1}{2} |E_l| \max_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 + |V_l| \max_{i \in V_l} \|v_{l_i}\| + k_v \sum_{j \in \Psi_i} \tilde{p}_{l_{ij}} c_{l_{ij}} (c_{l_{ij}} + 2d_{l_{ij}}) \|^2 \leq 2\delta. \quad (3.38)$$

Using the triangle inequality, the sufficient condition is given by

$$\frac{1}{2} |E_l| \max_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 + |V_l| \max_{i \in V_l} (\|v_{l_i}\| + k_v \sum_{j \in \Psi_i} \|\tilde{p}_{l_{ij}}\| (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2))^2 \leq 2\delta. \quad (3.39)$$

$$\frac{1}{2} |E_l| \max_{(i,j) \in E_l} (\|\tilde{p}_{l_{ij}}\|^2 - d_{l_{ij}}^2)^2 + |V_l| \max_{i \in V_l} (\|v_{l_i}\| + k_v |\Psi_i| \max_{j \in \Psi_i} \|\tilde{p}_{l_{ij}}\|^3)^2 \leq 2\delta. \quad (3.40)$$

Therefore $(\tilde{p}_l(t), v_l(t)) \in \mathcal{T}_1$ implies $M_2(e_l, s_l) \leq \delta$ (valid for all $t > 0$). Lemma 2 in [7] establishes the equivalence between $M_1(e_l) \leq \delta$ and $\Omega(F_l, F_l^*) \leq \varepsilon$ and since F_l^* is infinitesimally rigid, Corollary 3.2 shows that the formation F_l is also infinitesimally rigid for $(\tilde{p}_l(t), v_l(t)) \in \mathcal{T}_1$. According to Lemma 2.2, $R(p_l)R^T(p_l)$ is invertible for $(\tilde{p}_l(t), v_l(t)) \in \mathcal{T}_1$.

Substituting (3.30) and (3.33) into (3.32) yields

$$\dot{M}_2 = -k_v \beta^T(e_l) R(p_l) R^T(p_l) \beta(e_l) - k_a s_l^T s_l, \quad (3.41)$$

therefore

$$\dot{M}_2 \leq -k_v \lambda_{\min}(R(p_l) R^T(p_l)) L(e_l) - k_a s_l^T s_l \quad \text{for } (\bar{p}_l(t), v(t)) \in \mathcal{T}_1. \quad (3.42)$$

It follows from the negative definiteness of (3.41) that the level sets of M_2 are invariant [67] for $(\bar{p}_l(0), v_l(0)) \in \mathcal{T}_1$ and therefore $(e_l, s_l) = 0$ is asymptotically stable for $(\bar{p}_l(0), v_l(0)) \in \mathcal{T}_1$. Therefore, $\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$ and the control law (3.33) achieves the local asymptotic stability of the desired formation $F_l^* = (G_l^*, p_l^*)$. \square

Note that the condition (3.40) gives a set of allowed initial conditions $(\bar{p}_l(0), v_l(0)) \in \mathcal{T}_1$ that will ensure the system stability.

Remark 5. *The infinitesimal rigidity characteristic of the layered formation framework F_l ensures collision avoidance between all the connected agents i and j at any time $t > 0$.*

The following lemma uses the absolute initial and desired positions of the vehicles and shows that (3.33) achieves the local asymptotic stability of the desired formation F_l^* . Therefore, Lemma 3.6 establishes the conditions that prevent convergence to ambiguous formations.

Lemma 3.6. Given model \mathcal{W}_2 , let δ be a positive constant and $p_l(0) \in \mathbb{R}^{3(n+r)}$ and $v_l(0) \in \mathbb{R}^{3(n+r)}$ are the initial conditions. Given $\Delta(G_l^*)$ being the maximum degree of the graph G_l^* , if $(p_l(0), v_l(0)) \in \mathcal{T}_2$ where

$$\mathcal{T}_2 = \left\{ (p_l, v_l) \in \mathbb{R}^{3(n+r)} \times \mathbb{R}^{3(n+r)} \left| 8|E_l| \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\|^2 (\|p_{l_i} - p_{l_i}^*\| + D_l)^2 \right. \right. \\ \left. \left. + |V_l| (\max_{i \in V_l} \|v_{l_i}\| + \Delta(G_l^*) k_v (2 \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\| + D_l)^3)^2 \leq 2\delta \right\}, \quad (3.43)$$

then $F_l(t)$ is infinitesimally and minimally rigid and the control law (3.33) achieves the asymptotic stability of the desired formation F_l^* and ensures $\|p_{l_i}(t) - p_{l_j}(t)\| \rightarrow d_{l_{ij}}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$.

Proof. Using (3.22), the sufficient condition for (3.38) is given by

$$\frac{1}{2} |E_l| \left(\max_{(i,j) \in E_l} (\|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\|)^2 \max_{(i,j) \in E_l} (\|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\| + 2d_{l_{ij}})^2 \right) \\ + |V_l| \max_{i \in V_l} \|v_{l_i}\| + k_v \sum_{j \in \Psi_i} \tilde{p}_{l_{ij}} e_{l_{ij}} (e_{l_{ij}} + 2d_{l_{ij}}) \|^2 \leq 2\delta. \quad (3.44)$$

$$8|E_l| \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\|^2 (\|p_{l_i} - p_{l_i}^*\| + D_l)^2 \\ + |V_l| \max_{i \in V_l} \|v_{l_i}\| + k_v \sum_{j \in \Psi_i} \tilde{p}_{l_{ij}} e_{l_{ij}} (e_{l_{ij}} + 2d_{l_{ij}}) \|^2 \leq 2\delta. \quad (3.45)$$

Using the triangle inequality, the sufficient condition is given by

$$8|E_l| \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\|^2 (\|p_{l_i} - p_{l_i}^*\| + D_l)^2 + |V_l| \max_{i \in V_l} (\|v_{l_i}\| + k_v \sum_{j \in \Psi_i} \|\tilde{p}_{l_{ij}}\|^3)^2 \leq 2\delta. \quad (3.46)$$

$$8|E_l| \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\|^2 (\|p_{l_i} - p_{l_i}^*\| + D_l)^2 \\ + |V_l| \max_{i \in V_l} (\|v_{l_i}\| + k_v \sum_{j \in \Psi_i} (\|p_{l_i} - p_{l_i}^*\| + \|p_{l_j} - p_{l_j}^*\| + d_{l_{ij}})^3)^2 \leq 2\delta. \quad (3.47)$$

A sufficient condition for (3.47) is given by

$$8|E_l| \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\|^2 (\|p_{l_i} - p_{l_i}^*\| + D_l)^2 + |V_l| (\max_{i \in V_l} \|v_{l_i}\| + \Delta(G_l^*) k_r (2 \max_{i \in V_l} \|p_{l_i} - p_{l_i}^*\| + D_l)^3)^2 \leq 2\delta. \quad (3.48)$$

Using Theorem 3.5, it follows that $\|p_{l_i}(t) - p_{l_i}(t)\| \rightarrow d_{l_i,j}$ as $t \rightarrow \infty$ for all $(i, j) \in E_l$ and the control law (3.33) achieves the local asymptotic stability of the desired formation $F_l^* = (G_l^*, p_l^*)$. \square

Remark 6. *The set (3.43) specifies allowed initial conditions in terms of positions and velocities of mobile agents. It is a balanced trade-off between large initial velocity and large initial position errors. For example, for zero initial velocities, the condition sets an upper bound on initial position errors that ensures the system stability.*

3.3 Simulation Results

A set of simulations was performed to demonstrate the performance of the proposed method for solving the layered formation control problem.

3.3.1 Single Integrator Model

In this section, the simulation results for two scenarios with different numbers of vehicles with layered formations are presented to demonstrate the performance of the control law in (3.13) for a *three-dimensional case* (the upper layer vehicles are outside of the plane where the lower layer vehicles operate).

The following initial conditions were chosen for the vehicles

$$p_{l_i}(0) = p_{l_i}^* + \alpha I_1, \quad i = 1, \dots, n, n+1, \dots, n+r, \quad (3.49)$$

where α is a uniform random number on the interval $(0,1)$ and I_1 generates a 3×1 unit vector of uniformly distributed random values on the interval $(0, 2\pi)$. In the simulations, the value of k_c was set to 1 in (3.13). Note that for this simulation, $\delta = 23.44$ in (3.14) and $\delta = 3532.4$ in (3.20).

The first simulation was conducted using six lower layer vehicles and one upper layer vehicle. The desired formation of the vehicles that was chosen for this simulation is shown in Figure 3.3 where vertices 1 through 6 represent the six lower layer vehicles and vertex 7 represents the upper layer vehicle. This simulation scenario can emulate, for instance, a formation of UUVs that move underwater and a supervising ship that moves on the surface of water.

The desired layered framework was chosen in the shape of a cone that is infinitesimally and minimally rigid. The edges of the desired framework in Figure 3.3 are indexed by their vertices, e.g., edge 13 connects vertices 1 and 3. Using this edge notation, the desired distance between each pair of vertices was set using the Euclidean distance. For instance, in Figure 3.3, the distance between vertices 1 and 2 is $d_{l_{12}} = \|p_{l_1}^* - p_{l_2}^*\|$. Figure 3.4 shows the trajectories for the six lower layer vehicles and the upper layer vehicle as they move from their initial position to the final position to form the desired layered formation. Figure 3.5 demonstrates the distance errors $e_{l_{ij}}$ approaching zero. The control input $u_i(t)$ for $i = 1, \dots, 7$ in the x and y directions are shown in Figure 3.6 and Figure 3.7, respectively. Figure 3.8 shows the control input $u_i(t)$ for only $i = 7$ in the z -direction since the location of the other vehicles is defined by $p_{l_i} = (x_{l_i}, y_{l_i}, 0) \in \mathbb{R}^3$. An additional simulation was performed with four lower layer vehicles and four upper layer vehicles. In Figure 3.9, the trajectories of

the vehicles as they form the desired formation are shown. A sample of the distance errors for this formation is presented in Figure 3.10.

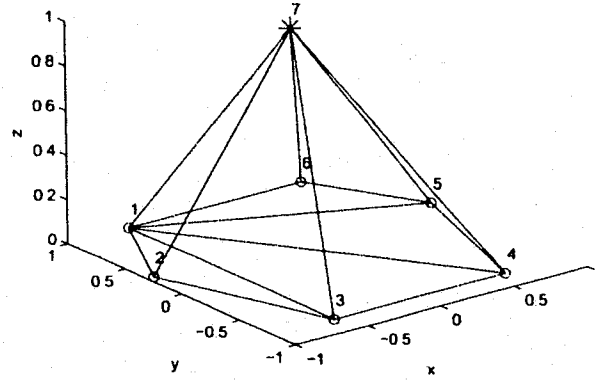


Figure 3.3: Desired formation for six lower layer vehicles (circles) and one upper layer vehicle (star).

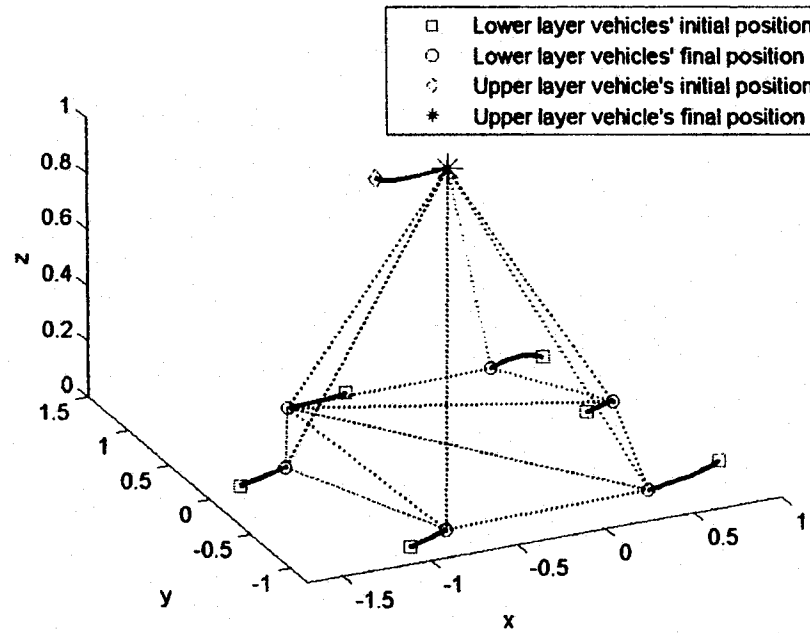


Figure 3.4: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 7$ with single integrator model (solid line) and desired formation (dotted line).

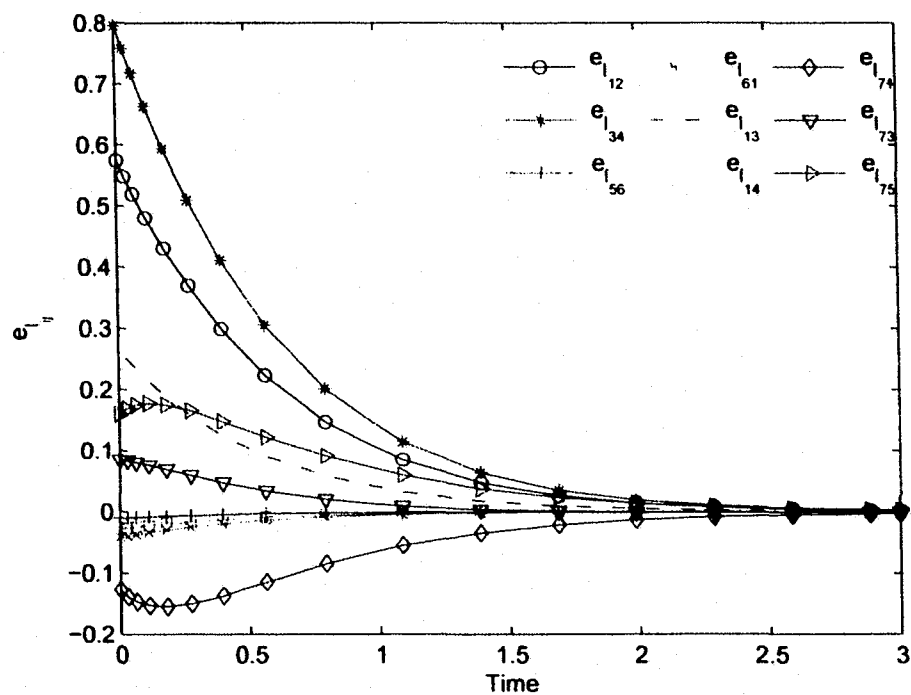


Figure 3.5: Sample of distance errors for six lower layer vehicles and the upper layer vehicle.

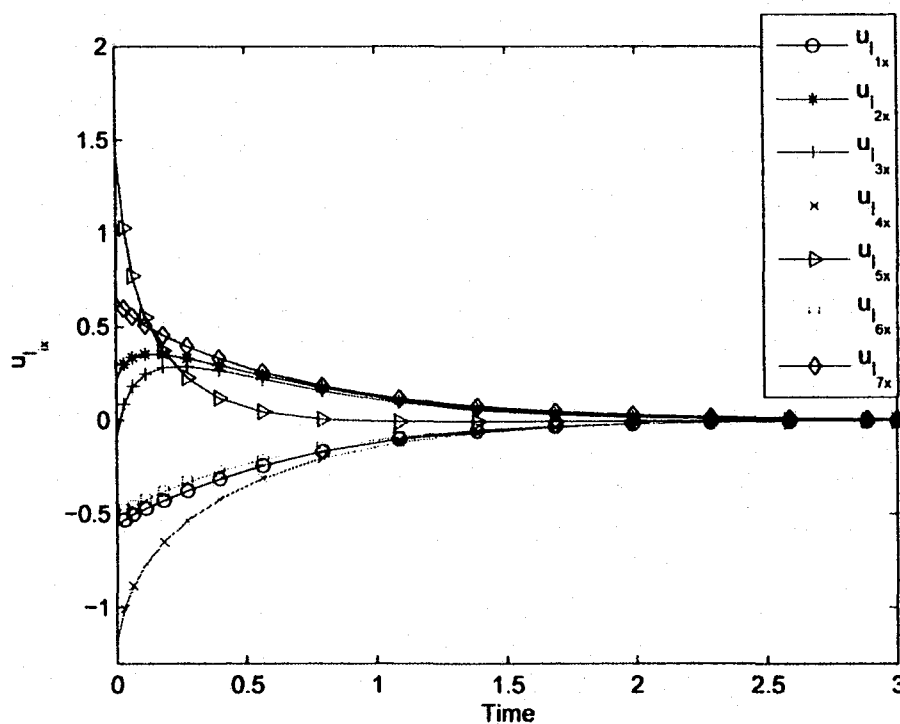


Figure 3.6: Control inputs $u_{l_{ix}}(t)$, $i = 1, \dots, 7$ in the x -direction.

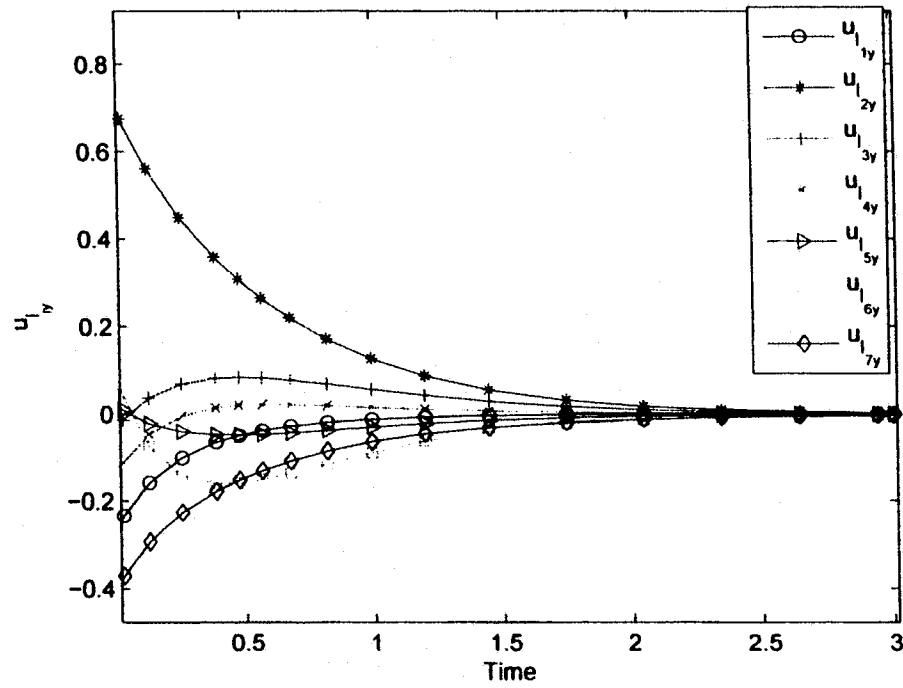


Figure 3.7: Control inputs $u_{i,y}(t)$, $i = 1, \dots, 7$ in the y -direction.

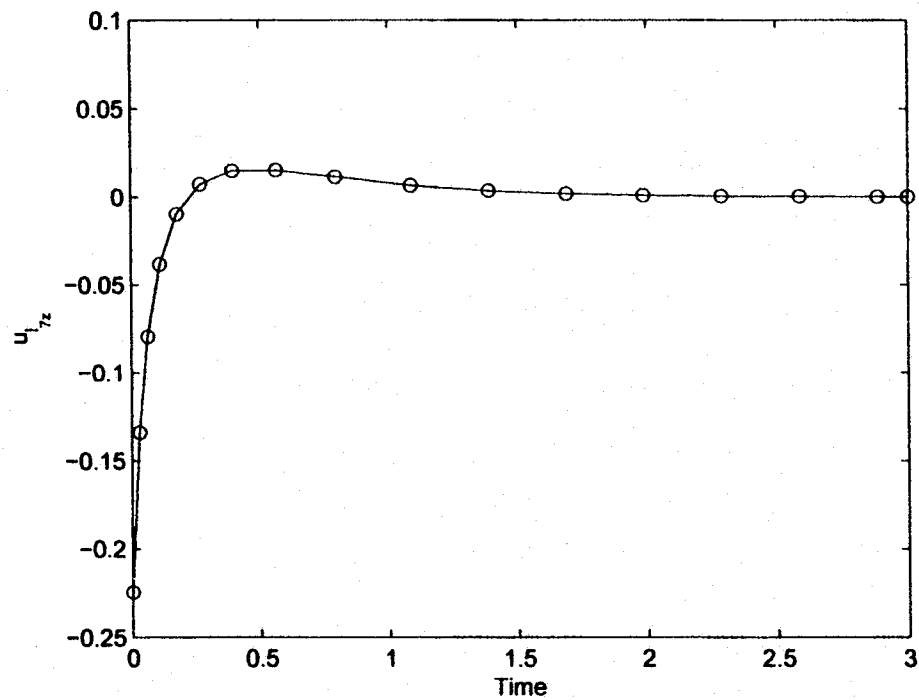


Figure 3.8: Control input in the z -direction for the upper layer vehicle.

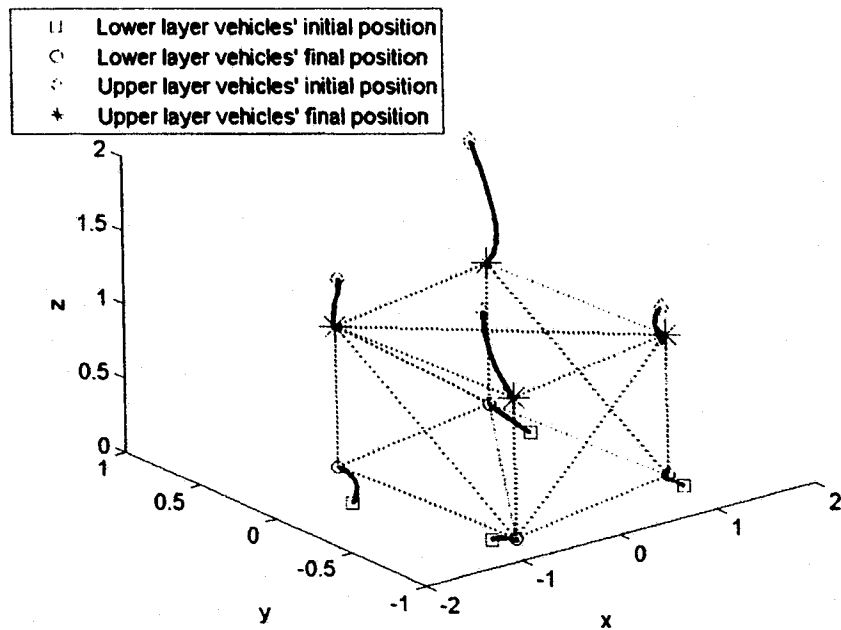


Figure 3.9: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 8$ with single integrator model (solid line) and desired formation (dotted line).

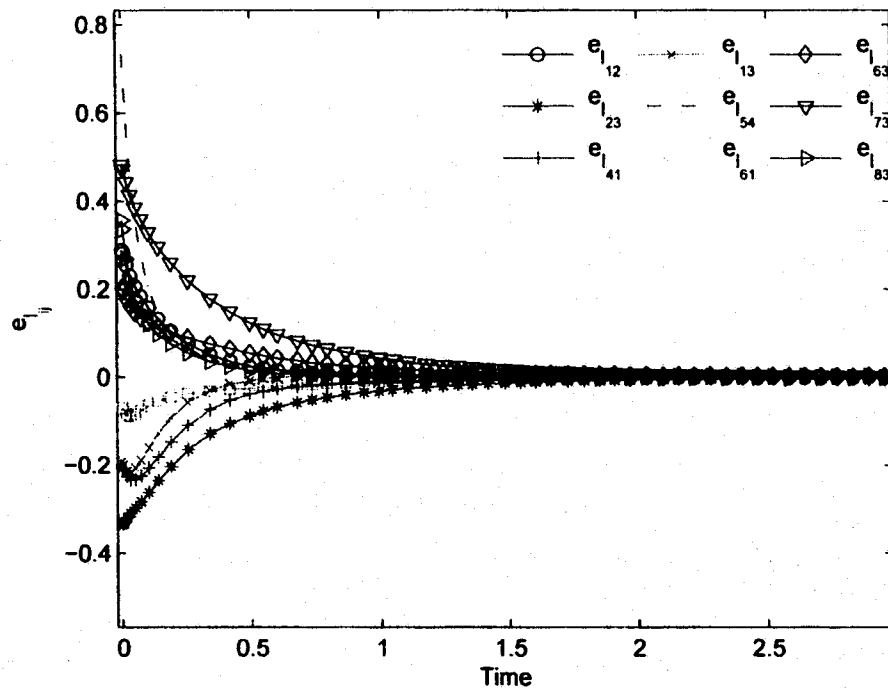


Figure 3.10: Sample of distance errors for four lower layer vehicles and four upper layer vehicles.

3.3.2 Double Integrator Model

The simulations in this section were performed for two scenarios with different number of vehicles using the proposed control law (3.33).

The initial conditions that were chosen for the vehicles were (3.49) and

$$v_i(0) = \alpha - 0.5I_2, \quad i = 1, \dots, n, n+1, \dots, n+r, \quad (3.50)$$

where I_2 generates a 3×1 unit vector of uniformly distributed random values on the interval $(0, 1)$. The control gains k_a and k_v were set to 1 in (3.33).

In the first simulation, the desired formation and trajectories of the vehicles as they move from the initial position to the final position are shown in Figure 3.11 and Figure 3.12, respectively.

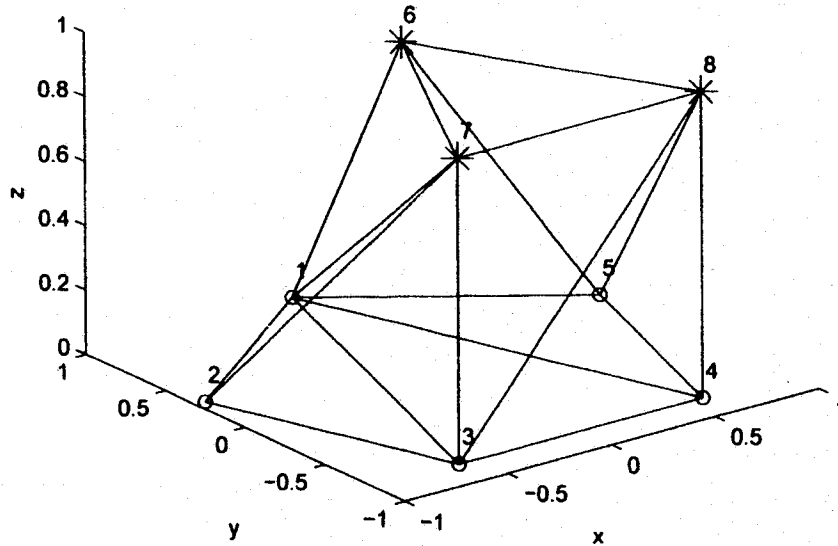


Figure 3.11: Desired formation for five lower layer vehicles and three upper layer vehicles with double integrator model.

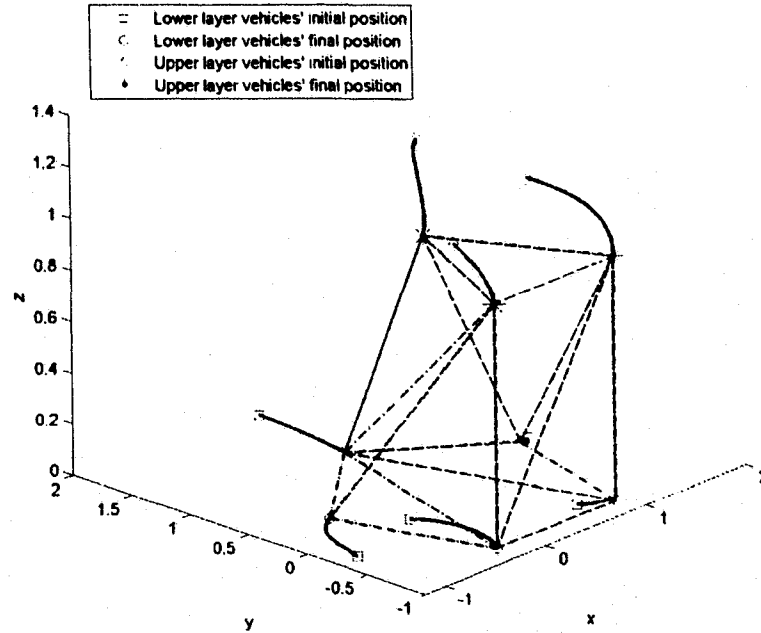


Figure 3.12: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 8$ with double integrator model (solid line) and desired formation (dotted line).

This simulation case can emulate a battlefield scenario where multiple UGVs move in a formation on the ground and multiple UAVs supervise them at a certain altitude, while providing intelligence and situational awareness to the UGVs for performing a mission. Figure 3.13 shows the distance errors e_{l_i} , approaching zero. The control input $u_{l_i}(t)$ for $i = 1, \dots, 8$ in the x and y directions are presented in Figure 3.14 and Figure 3.15 and $u_{l_i}(t)$ for $i = 6, 7, 8$ in the z -direction is shown in Figure 3.16. The second simulation was performed with two lower layer vehicles and four upper layer vehicles. Figure 3.17 shows the trajectories of all the vehicles and Figure 3.18 presents a sample of the distance errors.

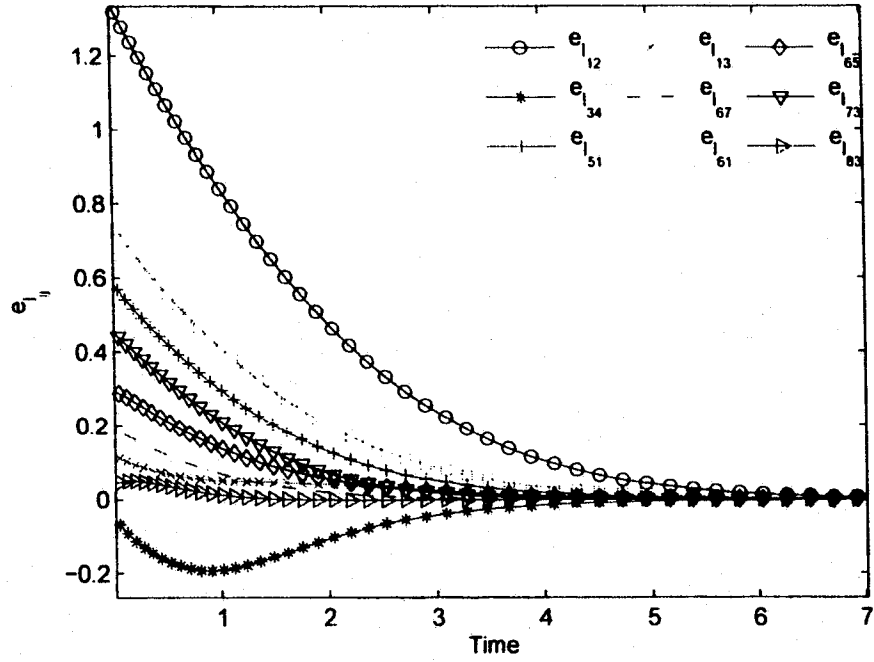


Figure 3.13: Sample of distance errors for five lower layer vehicles and three upper layer vehicles.

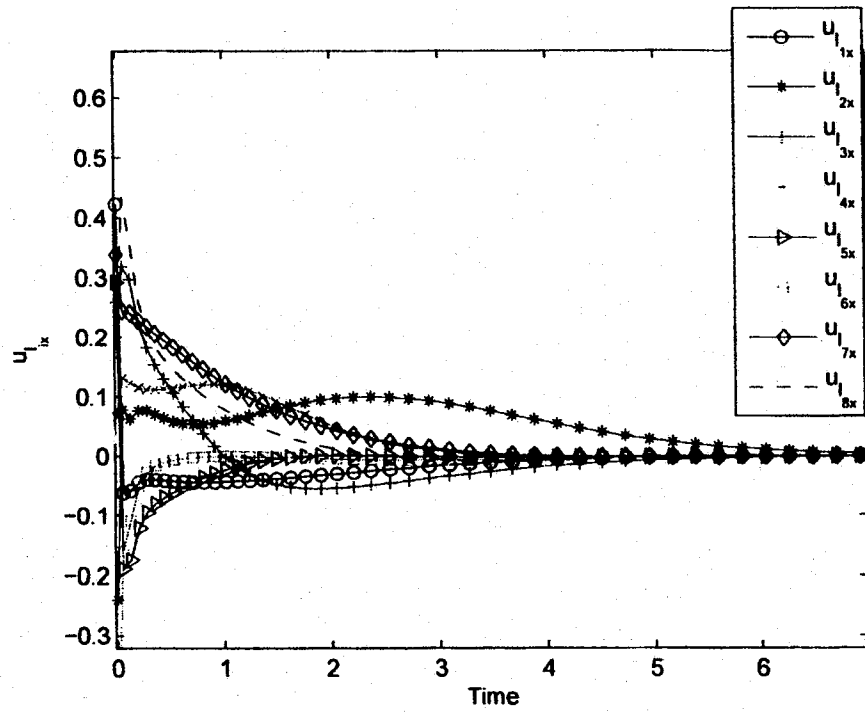


Figure 3.14: Control inputs $u_{l_{ix}}(t)$, $i = 1, \dots, 8$ in the x -direction.

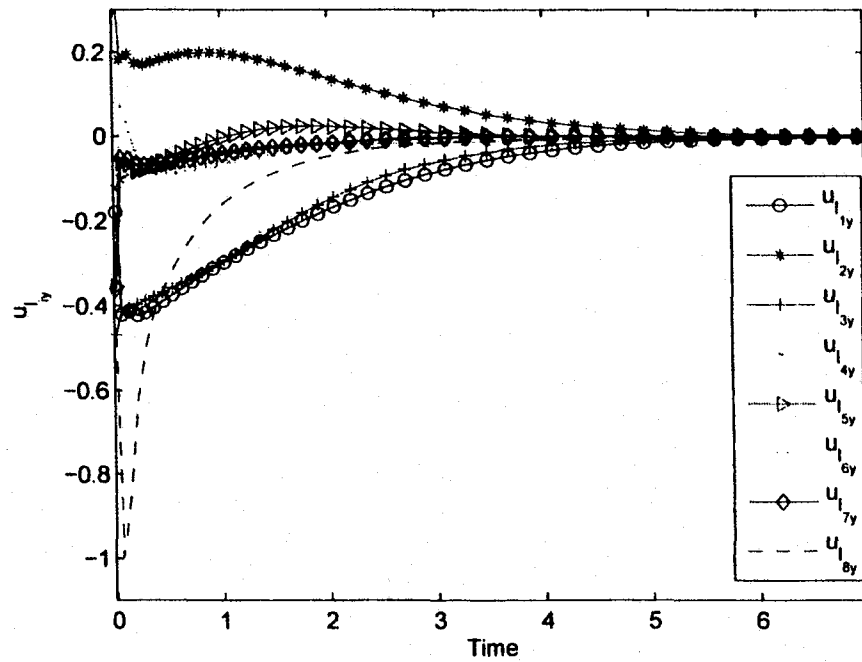


Figure 3.15: Control inputs $u_{i,y}(t)$, $i = 1, \dots, 8$ in the y -direction.

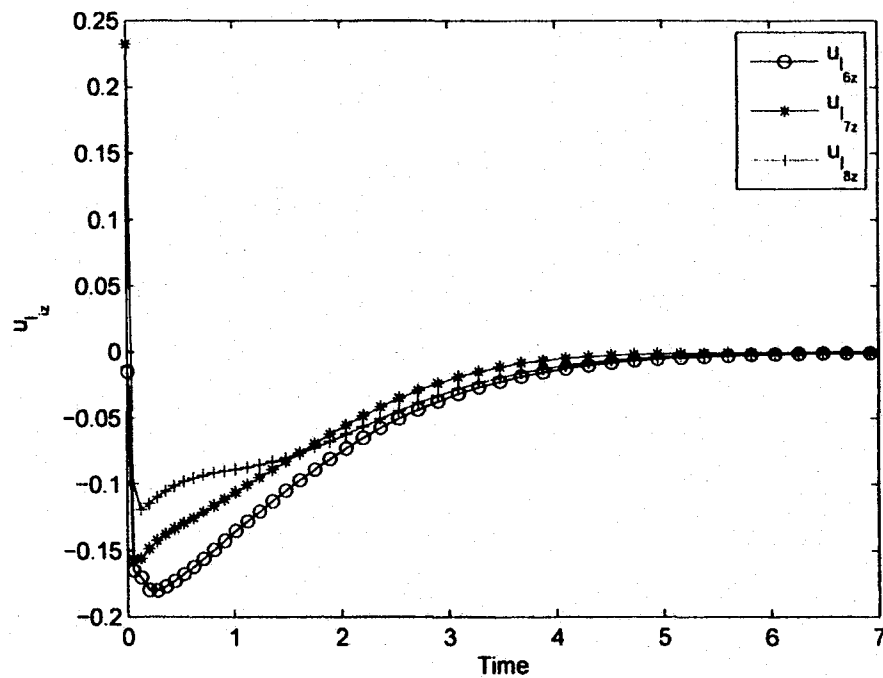


Figure 3.16: Control inputs in the z -direction for the upper layer vehicles.

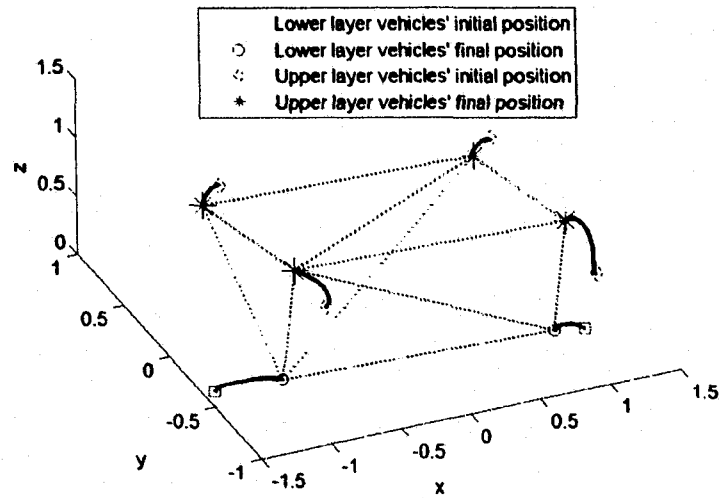


Figure 3.17: Vehicles' trajectories $p_i(t)$, $i = 1, \dots, 6$ with double integrator model (solid line) and desired formation (dotted line).

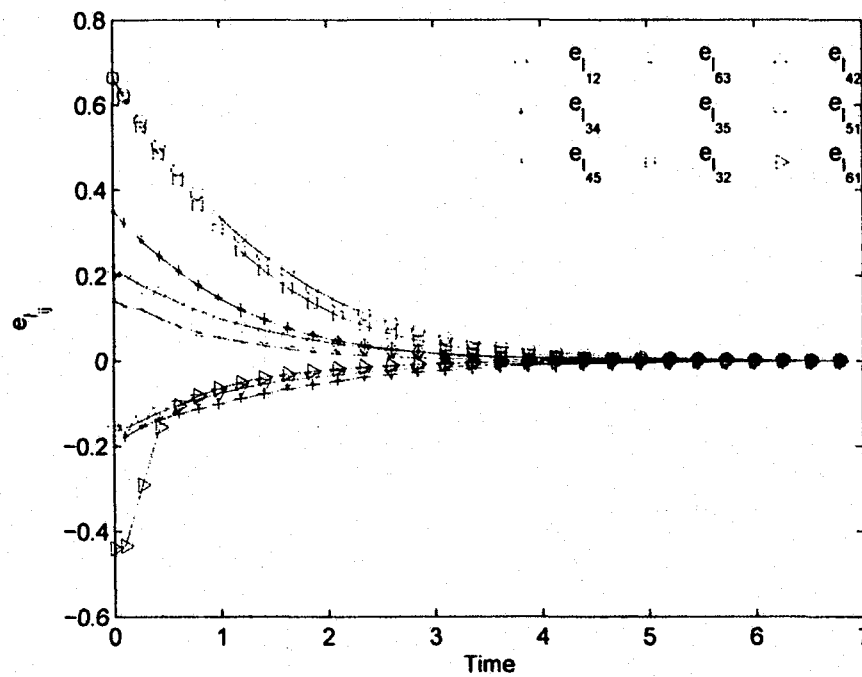


Figure 3.18: Sample of distance errors for two lower layer vehicles and four upper layer vehicles.

3.3.3 Robotic Vehicle Dynamics

Here, we present the simulation results for the layered formation of multiple robotic vehicles and one upper layer vehicle. We use the dynamics of a class of robotic vehicles (includes wheeled mobile robots, underwater vehicles with constant depth, aircraft with constant altitude, and marine vessels) that is given by the following model [13]

$$\begin{cases} \dot{q}_l = \omega_l \\ J_l(q_l) \dot{\omega}_l = \bar{u}_l - C_l(q_l, \dot{q}_l) \omega_l - H_l(q) \omega_l, \end{cases} \quad (3.51)$$

where $q_l = (x_l, y_l) \in \mathbb{R}^2$ is the *hand position* of the robotic vehicle with respect to an earth-fixed coordinate frame, $\omega_l \in \mathbb{R}^2$ is the velocity of the vehicles relative to an Earth-fixed frame, $\bar{u}_l \in \mathbb{R}^2$ represents the force/torque level control input and is given in (3.33), $J_l(q_l) \in \mathbb{R}^{2n \times 2n}$ is the mass matrix, $C_l(q_l, \dot{q}_l) \in \mathbb{R}^{2n \times 2n}$ is the Coriolis matrix, and $H_l(q_l) \in \mathbb{R}^{2n \times 2n}$ is the damping matrix. The dynamic model of a mobile robot can be feedback linearized if the orientation of the robot is ignored and the control is focused on an off-wheel axis point on the mobile robot, called robot hand point [70]. More precisely, the robot hand point q_l is defined as a point that is located at a distance L from the center of mass of robot and on the robot's orientation axis.

The mass matrix in (3.51) is given by $J_l(q_l) = \text{diag}(J_{l_i}(q_{l_i}))$ for $i = 1, \dots, n$. The term $J_{l_i}(q_{l_i})$ is defined as $J_{l_i}(q_{l_i}) = \eta^T \bar{J}_l \eta$ where $\bar{J}_l = \text{diag}(m, I)$ and

$$\eta(\theta_{l_i}) = \begin{bmatrix} \cos \theta_{l_i} & \sin \theta_{l_i} \\ -\frac{\sin \theta_{l_i}}{L} & \frac{\cos \theta_{l_i}}{L} \end{bmatrix}, \quad (3.52)$$

$$J_{l_i}(q_{l_i}) = \begin{bmatrix} J_{l_{i1}} & J_{l_{i2}} \\ J_{l_{i21}} & J_{l_{i22}} \end{bmatrix}, \quad (3.53)$$

$$J_{l_{i1}} = m \cos^2(\theta_{l_i}) + \frac{I}{L^2} \sin^2(\theta_{l_i})$$

$$J_{l_{i2}} = (m - \frac{I}{L^2}) \cos\theta_{l_i} \sin\theta_{l_i}$$

$$J_{l_{i21}} = (m - \frac{I}{L^2}) \cos\theta_{l_i} \sin\theta_{l_i}$$

$$J_{l_{i22}} = m \sin^2(\theta_{l_i}) + \frac{I}{L^2} \cos^2(\theta_{l_i}).$$

The Coriolis matrix is given by $C_l(q_l, \dot{q}_l) = \text{diag}(C_{l_i}(q_{l_i}, \dot{q}_{l_i}))$ for $i = 1, \dots, n$. The term $C_{l_i}(q_{l_i}, \dot{q}_{l_i})$ is defined as $C_{l_i}(q_{l_i}, \dot{q}_{l_i}) = \eta^T \bar{J}_l \dot{\eta}$ and is given by

$$C_{l_i}(q_{l_i}, \dot{q}_{l_i}) = \begin{bmatrix} C_{l_{i1}} & C_{l_{i2}} \\ C_{l_{i21}} & C_{l_{i22}} \end{bmatrix}, \quad (3.54)$$

$$C_{l_{i1}} = -(m - \frac{I}{L^2}) \dot{\theta}_{l_i} \cos(\theta_{l_i}) \sin(\theta_{l_i})$$

$$C_{l_{i2}} = m \dot{\theta}_{l_i} \cos^2(\theta_{l_i}) + \frac{I}{L^2} \dot{\theta}_{l_i} \sin^2(\theta_{l_i})$$

$$C_{l_{i21}} = -m \dot{\theta}_{l_i} \sin^2(\theta_{l_i}) - \frac{I}{L^2} \dot{\theta}_{l_i} \cos^2(\theta_{l_i})$$

$$C_{l_{i22}} = (m - \frac{I}{L^2}) \dot{\theta}_{l_i} \cos(\theta_{l_i}) \sin(\theta_{l_i}).$$

The damping matrix is given by $H_l(q_l) = \text{diag}(H_{l_i}(q_{l_i}))$ for $i = 1, \dots, n$. The term $H_{l_i}(q_{l_i})$ is defined as $H_{l_i}(q_{l_i}) = \eta^T \bar{H}_{l_i} \eta$. The derivation of the matrices $J_l(q_l)$, $C_l(q_l, \dot{q}_l)$, and $H_l(q_l)$ can be found in [13].

In this simulation, there are four robotic vehicles and one upper layer agent. The model used for the upper layer vehicle was the same as in Section 3.3.2. The initial positions of the robotic vehicles were set to

$$q_{l_i}(0) = q_{l_i}^* + \alpha I_3, \quad i = 1, \dots, 4, \quad (3.55)$$

where $q_{l_i}^*$ is the desired position of the agents and I_3 generates a 2×1 unit vector of uniformly distributed random values on the interval $(0, 2\pi)$. The initial orientations and velocities of the vehicles were chosen as

$$\theta_{l_i}(0) = \sigma_1 I_4, \quad i = 1, \dots, 4, \quad (3.56)$$

$$\omega_{l_i}(0) = \sigma_2 [I_5 - 0.5], \quad i = 1, \dots, 4, \quad (3.57)$$

where $\sigma_1 = 1$, I_4 is a uniformly distributed random number on the interval $(0,1)$, $\sigma_2 = 2\pi$, and I_5 generates a 2×1 unit vector of uniformly distributed values on the interval $(0,1)$. The parameters such as mass and moment of inertia of the vehicles that are used in matrices $J_l(q_l)$ and $C_l(q_l, \dot{q}_l)$ were set to $m = 4 \text{ kg}$ and $I = 0.0405 \text{ kg-m}^2$, respectively. The distance between the center of mass of the vehicle and the hand position was chosen as $L = 0.15 \text{ m}$. The constant damping matrix $\bar{H}_{l_i} = \text{diag}(0.4 \text{ kg/s}, 0.005 \text{ kg-m}^2/\text{s})$ for $i = 1, \dots, n$ was chosen for the simulation.

The initial position of the upper layer vehicle was chosen as (4.24) and its velocity was set to

$$\omega_{l_5}(0) = \sigma_2 [I_2 - 0.5]. \quad (3.58)$$

The control gains k_a and k_v in (3.33) were set to 20 and 1, respectively.

An infinitesimally and minimally rigid formation, shown with dotted lines in Figure 3.19, with nine communication/control links was chosen as the desired formation. Figure 3.19 shows the trajectories of the vehicles, as they move from their initial positions to the final positions and get into the desired formation. Figure 3.20 shows the distance errors e_{li} approaching zero.

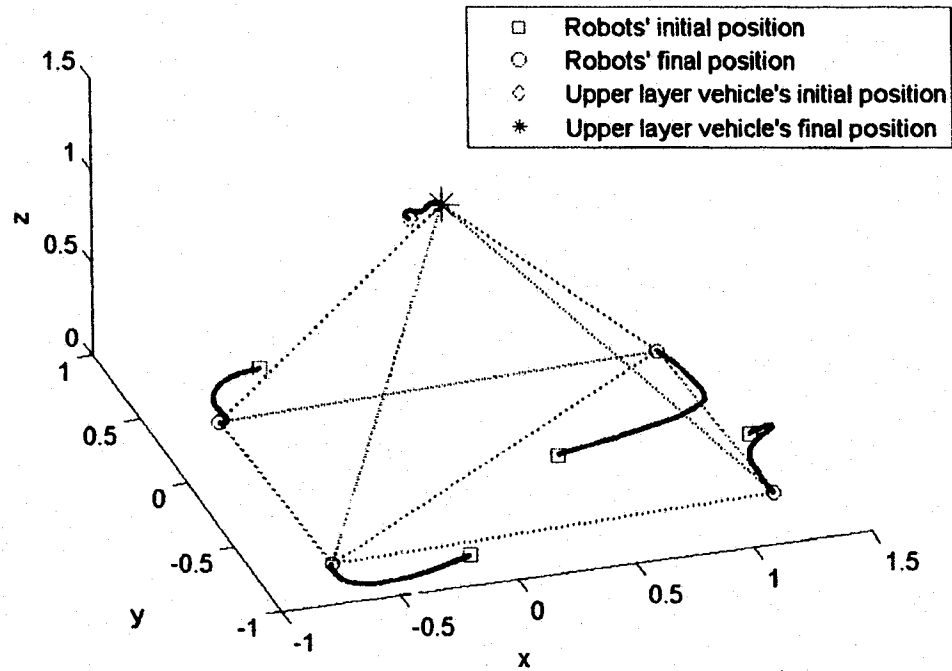


Figure 3.19: Vehicles' trajectories $q_i(t)$, $i = 1, \dots, 5$ (solid line) and desired formation (dotted line).

From these simulation examples, it can be seen that distance errors converge to zero with time. Notice that in the case of vehicle dynamics, the error converges to the small bounded neighborhood of zero. That is to be expected since the robotic vehicle dynamics is not an ideal integrator model, rather a dynamic model with extra terms.

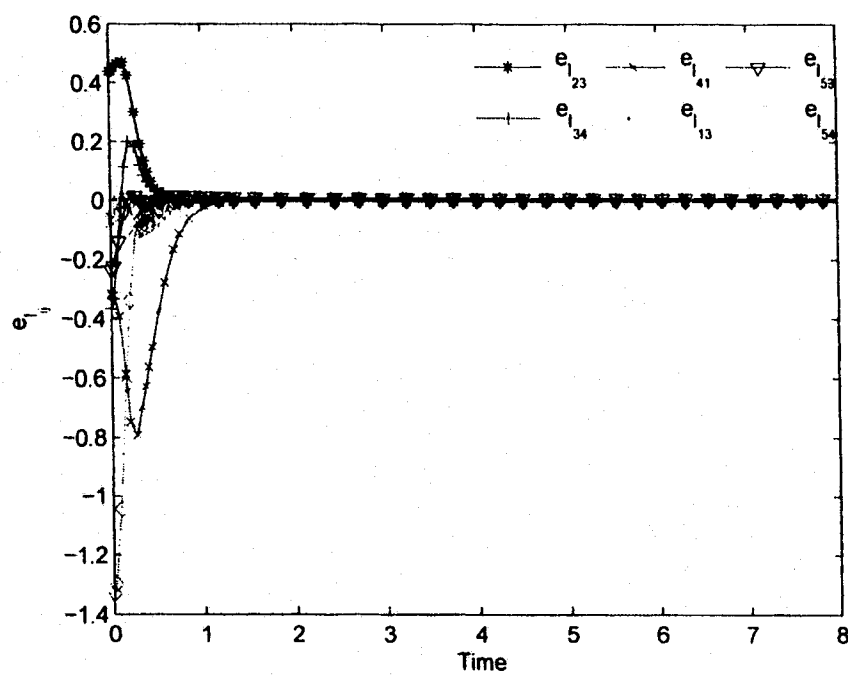


Figure 3.20: Sample of distance errors (m) vs. time (s) for four robotic vehicles and one upper layer agent.

CHAPTER 4

NEURAL NETWORK-BASED FORMATION CONTROL OF UNMANNED VEHICLES IN 3D SPACE

This chapter addresses the unmanned vehicles formation control problem in three-dimensional space using graph rigidity and NNs control technique. In this work, it is assumed that vehicles are equipped with sensors such as ultrasonic or infrared-based relative positioning sensors that allow them to measure the distance and direction between selected vehicles.

4.1 Problem Formulation

Consider a system of n agents (vehicles) in space modeled by the single integrator

$$\dot{p}_i = u_i, \quad i = 1, \dots, n, \quad (4.1)$$

where $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ for $i = 1, \dots, n$ is the location of the i -th vehicle and $u_i \in \mathbb{R}^3$ is the velocity-level control input.

In addition, consider an infinitesimally and minimally rigid framework as the desired formation described by $F^* = (G^*, p^*)$ where $G^* = (V, E)$ and $p^* = (p_1^*, \dots, p_n^*)$.

Given the actual formation $F(t) = (G^*, p(t))$ where $p = (p_1, \dots, p_n)$ and assuming that at $t = 0$, $\|p_i(0) - p_j(0)\| \neq d_{ij}$ for $(i, j) \in E$, where $d_{ij} = \|p_i^* - p_j^*\| > 0$ is

the constant desired distance between vehicles i and j in the formation, the control objective is to design the control input u_i such that the distance error

$$e_{ij} = \|p_i(t) - p_j(t)\| - d_{ij}, \quad (4.2)$$

is uniformly ultimately bounded.

4.2 Control Algorithm

Define the relative position of two vehicles in the formation as

$$\tilde{p}_{ij} = p_i - p_j, \quad (i, j) \in E. \quad (4.3)$$

The distance error dynamics for the group of n vehicles is given by

$$\begin{aligned} \dot{e}_{ij} &= \frac{d}{dt} \sqrt{\tilde{p}_{ij}^T \tilde{p}_{ij}} = (\tilde{p}_{ij}^T \tilde{p}_{ij})^{-\frac{1}{2}} \tilde{p}_{ij}^T (u_i - u_j) \\ &= \frac{\tilde{p}_{ij}^T (u_i - u_j)}{e_{ij} + d_{ij}}. \end{aligned} \quad (4.4)$$

Let the potential function related to vehicles i and j be given by

$$M_{ij}(e_{ij}) = \frac{1}{4} e_{ij}^2 (e_{ij} + 2d_{ij})^2, \quad (4.5)$$

and note that it is positive definite and radially unbounded. For control algorithm development and stability analysis, consider the total potential function

$$M(e, \tilde{W}) = \sum_{(i,j) \in E} M_{ij}(e_{ij}) + \frac{1}{2} \text{tr}(\tilde{W}^T S \tilde{W}), \quad (4.6)$$

where S is symmetric positive definite and \tilde{W} is the weight estimation error matrix and is defined as $\tilde{W} = W - \hat{W}$ with \hat{W} being the actual weight matrix. The time derivative of (4.6) is given by

$$\dot{M} = \sum_{(i,j) \in E} \frac{\partial M_{ij}}{\partial e_{ij}} \frac{\tilde{p}_{ij}^T (u_i - u_j)}{e_{ij} + d_{ij}} + \text{tr}(\tilde{W}^T S \dot{\tilde{W}}). \quad (4.7)$$

It follows from (2.1), (2.2), and (4.4) that (4.7) can be expressed as

$$\dot{M} = \beta^T(e)R(p)u + \text{tr}(\bar{W}^T S \dot{\bar{W}}). \quad (4.8)$$

where $u = (u_1, \dots, u_n) \in \mathbb{R}^{3n}$, $e = (\dots, e_{ij}, \dots) \in \mathbb{R}^{|E|}$, and

$$\begin{aligned} \beta(e) &= (\dots, \frac{\frac{\partial M_{ij}}{\partial e_{ij}}}{e_{ij} + d_{ij}}, \dots) \\ &= (\dots, e_{ij}(e_{ij} + 2d_{ij}), \dots), \quad (i, j) \in E. \end{aligned} \quad (4.9)$$

The terms in e and $\beta(e) \in \mathbb{R}^{|E|}$ are ordered the same way as in (2.1). The following theorem gives the control and NN tuning laws for the formation control problem.

Theorem 4.1. *Let $F(t) = (G^*, p(t))$ in \mathbb{R}^3 be the formation of a group of n vehicles. Select the control input as*

$$u = R^T(p)[R(p)R^T(p)]^{-1}(-k_p\beta(e) + \hat{W}^T\sigma(V^T\bar{x})), \quad (4.10)$$

where k_p is a positive control gain and x is the relative position \tilde{p}_{ij} of the vehicles. Let the estimated NN weights be given by the NN tuning algorithm.

$$\dot{\hat{W}} = -S^{-1}\sigma(V^T\bar{x})\beta^T(e) - k_c\|\beta(e)\|S^{-1}\hat{W}, \quad (4.11)$$

where k_c is a user selected constant. Then, by properly selecting the control gain and the design parameters, the distance error e and the NN weights \hat{W} are uniformly ultimately bounded.

Proof. Substituting (4.10) into (4.8) yields

$$\begin{aligned} \dot{M} &= -k_p\beta^T(e)R(p)R^T(p)[R(p)R^T(p)]^{-1}\beta(e) \\ &\quad + \beta^T(e)R(p)R^T(p)[R(p)R^T(p)]^{-1}\hat{W}^T\sigma(V^T\bar{x}) + \text{tr}(\bar{W}^T S \dot{\bar{W}}). \end{aligned} \quad (4.12)$$

Choose $\dot{\bar{W}}$ (or equivalently $\dot{\hat{W}}$) such that \dot{M} is negative definite outside of a compact set around the origin. From Lemma (2.2), we can state that $R(p)R^T(p)$ is invertible

in the compact set. Then, the previous equation is equivalent to

$$\dot{M} = -k_p \beta^T(e) \beta(e) + \beta^T(e) (W^T - \bar{W}^T) \sigma(V^T \bar{x}) + \text{tr}(\bar{W}^T S \dot{\bar{W}}), \quad (4.13)$$

$$\dot{M} = -k_p \beta^T(e) \beta(e) + \beta^T(e) W^T \sigma(V^T \bar{x}) - \beta^T(e) \bar{W}^T \sigma(V^T \bar{x}) + \text{tr}(\bar{W}^T S \dot{\bar{W}}), \quad (4.14)$$

$$\dot{M} = -k_p \beta^T(e) \beta(e) + \beta^T(e) W^T \sigma(V^T \bar{x}) + \text{tr}[\bar{W}^T (S \dot{\bar{W}} - \sigma(V^T \bar{x}) \beta^T(e))]. \quad (4.15)$$

Using the tuning law (4.11), we have

$$\dot{M} = -k_p \beta^T(e) \beta(e) + \beta^T(e) W^T \sigma(V^T \bar{x}) + k_c \|\beta(e)\| \text{tr}(\bar{W}^T \dot{\bar{W}}). \quad (4.16)$$

From the Cauchy-Schwartz inequality we know that

$$\beta^T(e) W^T \sigma(V^T \bar{x}) \leq \|\beta(e)\| \|W^T \sigma(V^T \bar{x})\| \leq \|\beta(e)\| W_m L. \quad (4.17)$$

It can be shown, from the definitions of trace and the Frobenius norm, that for any two matrices X and Y the following inequality holds:

$$\text{tr}[X(Y - X)] \leq \|X\|_F \|Y\|_F - \|X\|_F^2. \quad (4.18)$$

Therefore one has,

$$\begin{aligned} \dot{M} &= -k_p \beta^T(e) \beta(e) + \beta^T(e) W^T \sigma(V^T \bar{x}) + k_c \|e\| \text{tr}(\bar{W}^T \dot{\bar{W}}) \\ &\leq -k_p \|\beta(e)\|^2 + \|\beta(e)\| W_m L + k_c \|\beta(e)\| \left(\|\bar{W}\|_F \|W\|_F - \|\bar{W}\|_F^2 \right) \\ &\leq -\|\beta(e)\| \left(k_p \|\beta(e)\| + k_c \|\bar{W}\|_F^2 - W_m L - k_c \|\bar{W}\|_F W_m \right) \\ &= -\|\beta(e)\| \left(k_p \|\beta(e)\| + k_c \left(\|\bar{W}\|_F - \frac{1}{2} W_m \right)^2 - W_m L - \frac{k_c}{4} W_m^2 \right). \end{aligned} \quad (4.19)$$

It follows that $\dot{M} < 0$ if either of the following is true:

$$\|\beta(e)\| > \frac{N}{k_p} \quad (4.20)$$

$$\|\bar{W}\|_F > \sqrt{\frac{N}{k_c}} + \frac{1}{2} W_m, \quad (4.21)$$

where

$$N = \frac{k_c}{4} W_m^2 + W_m L. \quad (4.22)$$

This shows that \dot{M} is negative definite outside of a compact set, which can be reduced arbitrarily by increasing the gain k_p . Therefore, the distance error converges to the bounded neighborhood of zero. \square

Note that bounds in terms of the error e can be derived from (4.20):

$$e_{ij} < \frac{N}{k_p}. \quad (4.23)$$

Furthermore, note that the bounds are functions of the number of hidden layer nodes L , which is to be expected.

4.3 Simulation Results

A set of simulations was performed to test the performance of the control law (4.10) first without any external disturbances in the single integrator model and then with external disturbances included in the model.

4.3.1 Single Integrator Model without External Disturbances

Here, a simulation with six vehicles using the vehicle model in (4.1) was conducted. This simulation case emulates a battlefield scenario where multiple UGVs move in a formation on the ground and multiple UAVs supervise them at a certain altitude while providing intelligence and situational awareness to the UGVs relevant to mission performance. The desired formation was chosen as an infinitesimally and minimally rigid shape shown in Figure 4.1.

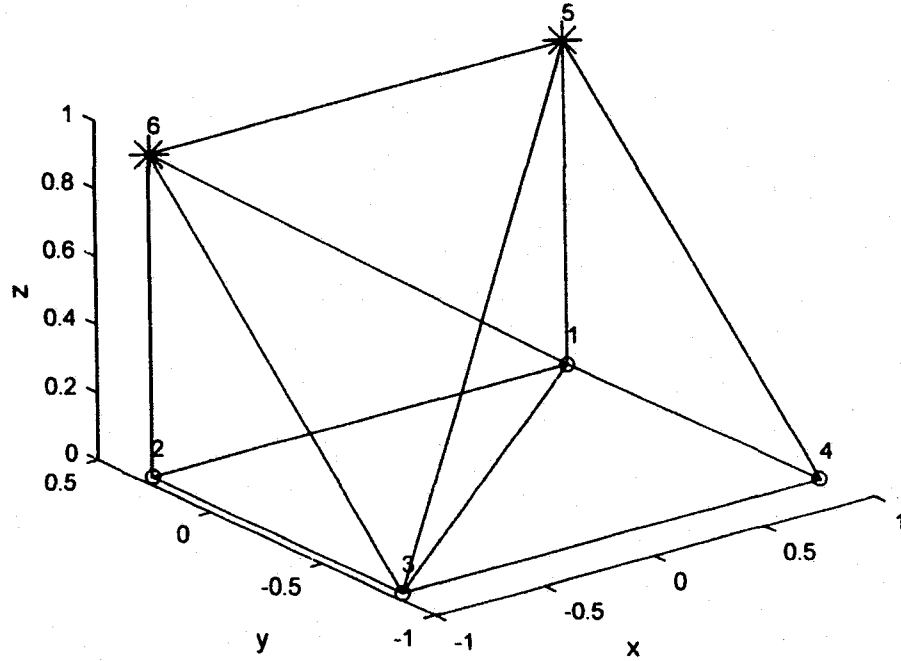


Figure 4.1: Desired formation for four UGVs and two UAVs.

The following initial conditions were chosen for the vehicles

$$p_i(0) = p_i^* + \alpha[I - 0.5], \quad i = 1, \dots, n, \quad (4.24)$$

where α is the maximum offset which was set to 1 and I generates a random 3×1 vector whose elements are uniformly distributed on the interval $(0, 1)$. In the simulation, the value of k_p was set to 1 in (4.10), $L = 240$ was chosen as the total number of hidden layer nodes, the entries of matrix V were random values, the weight matrix W was initiated at zero, S was a diagonal matrix with 0.1 on the main diagonal, and k_c was set to 1 in (4.11).

Figure 4.2 shows the trajectories for the four UGVs and two UAVs as they move from their initial position to the final position to form the desired formation.

Figure 4.3 demonstrates a sample of distance errors e_{ij} approaching zero.

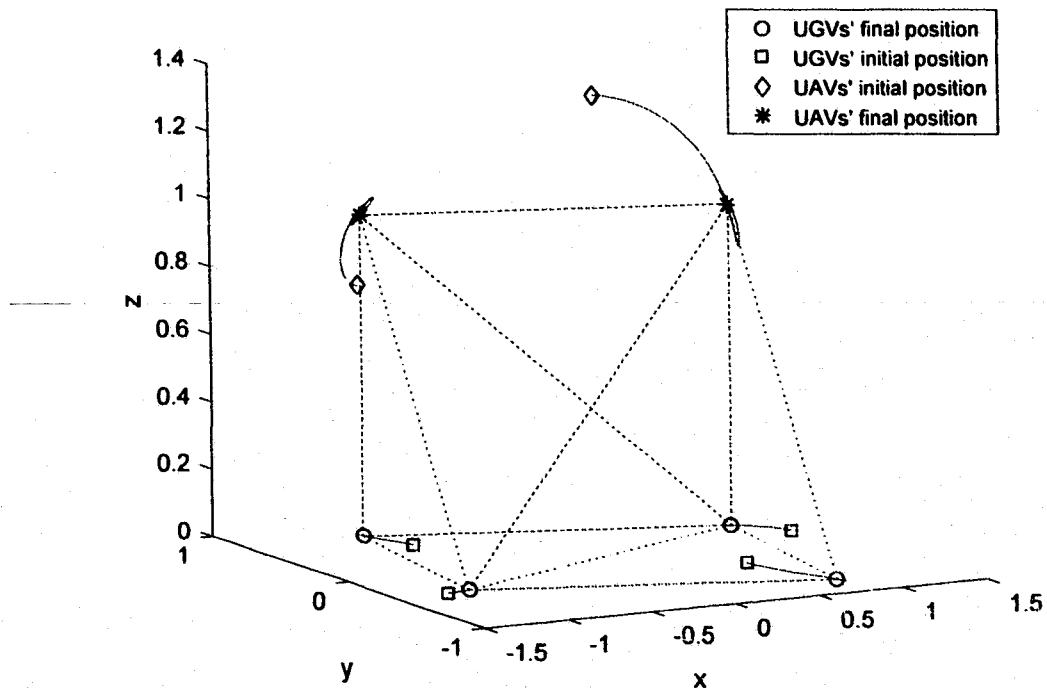


Figure 4.2: Vehicle trajectories $p_i(t)$, $i = 1, \dots, 6$ (solid line) and desired formation (dotted line).

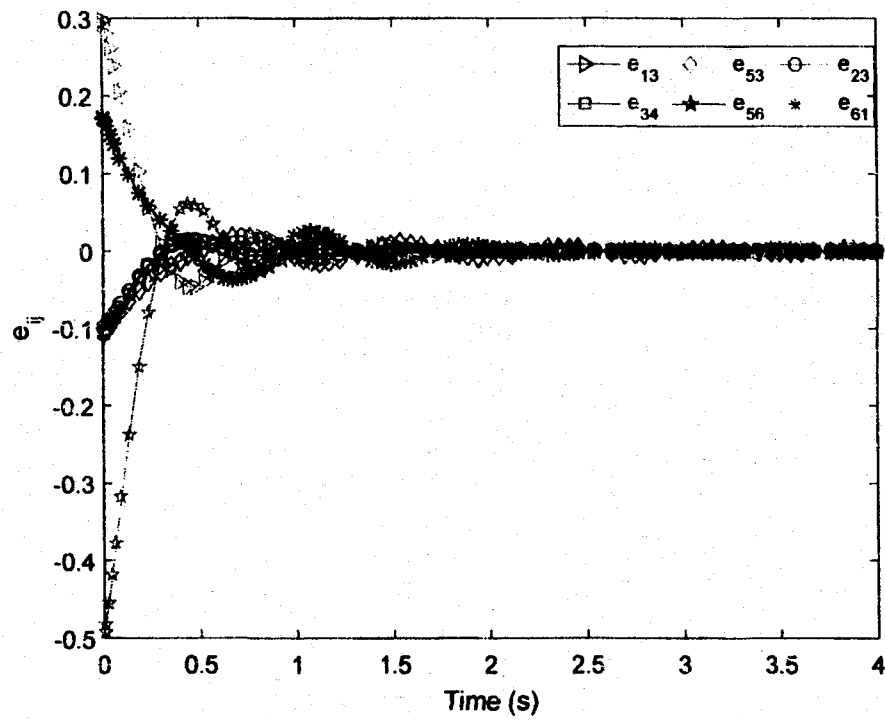


Figure 4.3: Sample of distance errors $e_{ij}(t)$, $i, j \in V$.

The control input $u_i(t)$ for $i = 1, \dots, 6$ in the x and y directions are shown in Figure 4.4 and Figure 4.5, respectively. Since this simulation emulated a scenario with four UGVs and two UAVs, the position of each UGV was defined by $p_i(t) = (x_i, y_i, 0)$, $i = 1, \dots, 4$ and the position of each UAV was defined by $p_i(t) = (x_i, y_i, z_i)$, $i = 5, 6$. Figure 4.6 shows the control input $u_i(t)$ for $i = 5, 6$ in the z -direction. Each of the NN's outputs and thus each column of W is associated with an edge. Elements of W plotted in Figure 4.7 are chosen to correspond to the sampled errors in Figure 4.3 and are indexed in a similar manner.

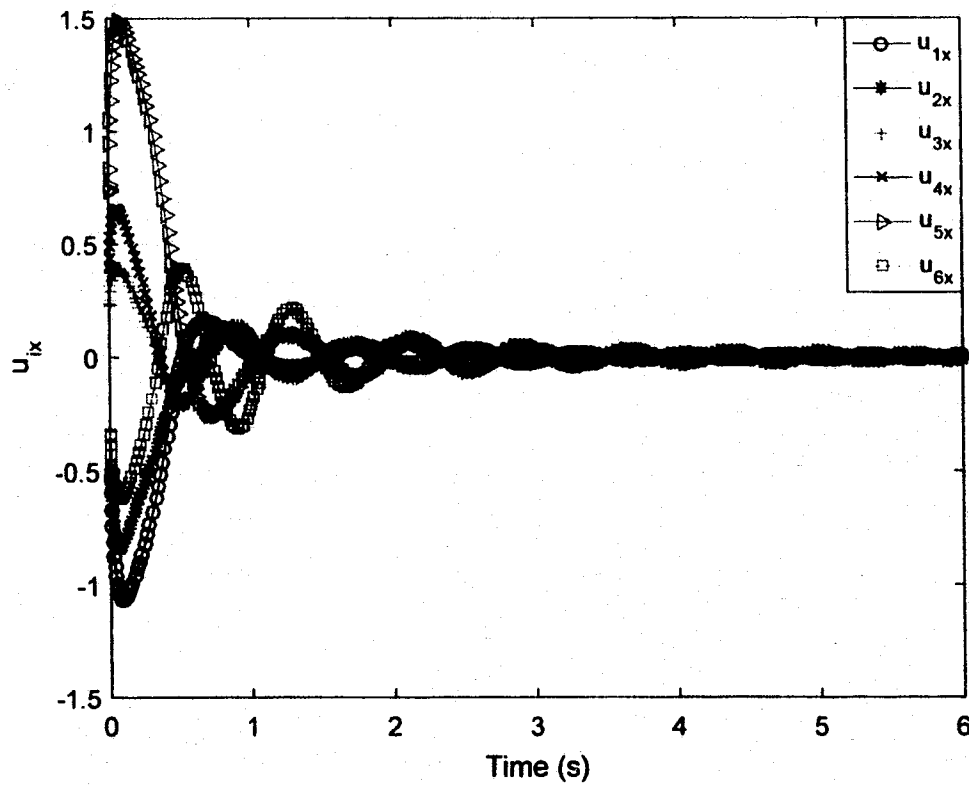


Figure 4.4: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the x -direction.

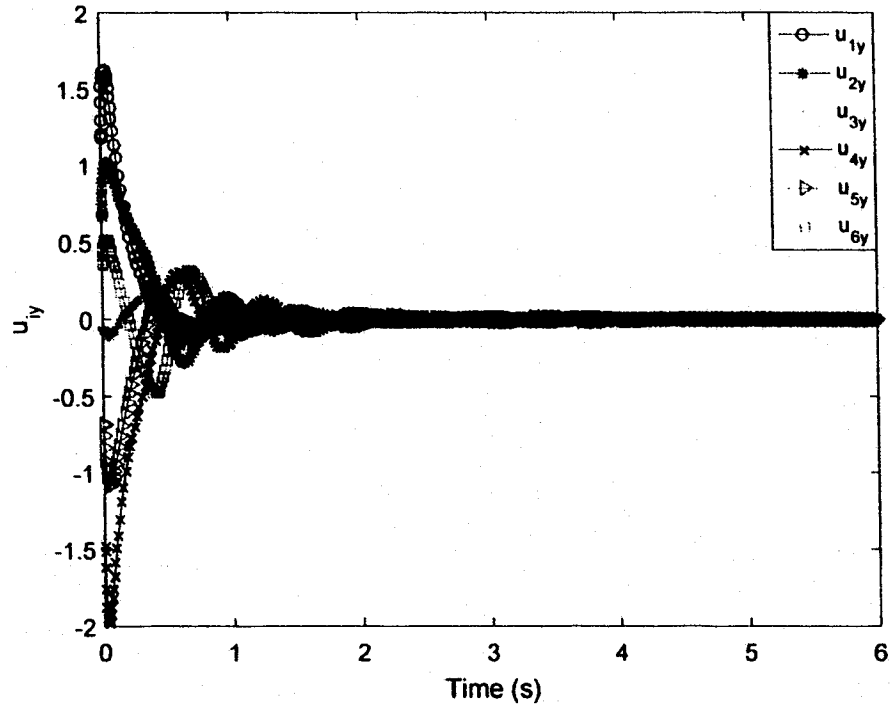


Figure 4.5: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the y -direction.

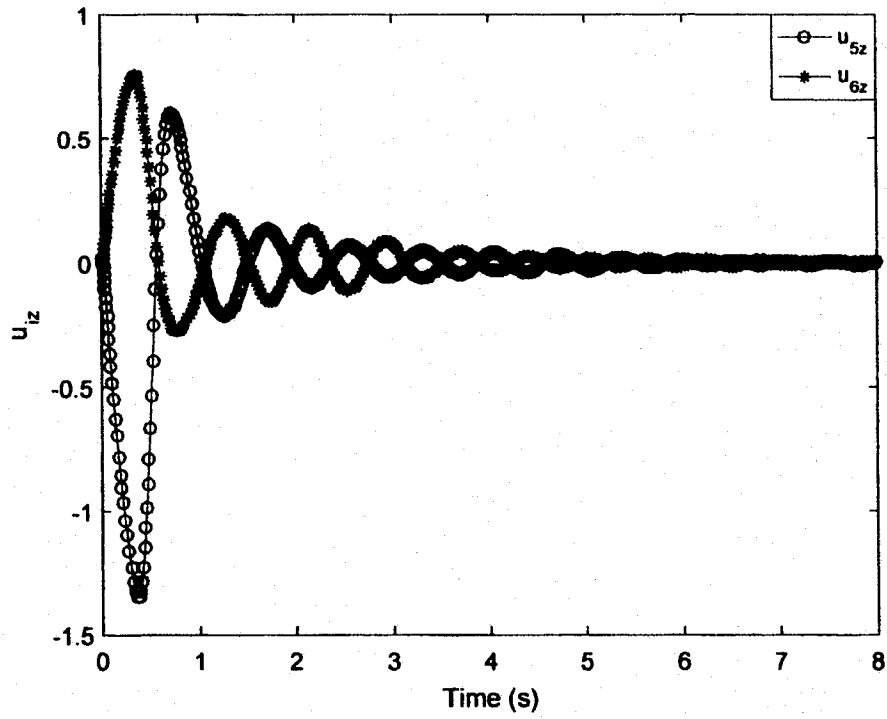


Figure 4.6: Control inputs $u_i(t)$, $i = 5, 6$ in the z -direction for the UAVs.

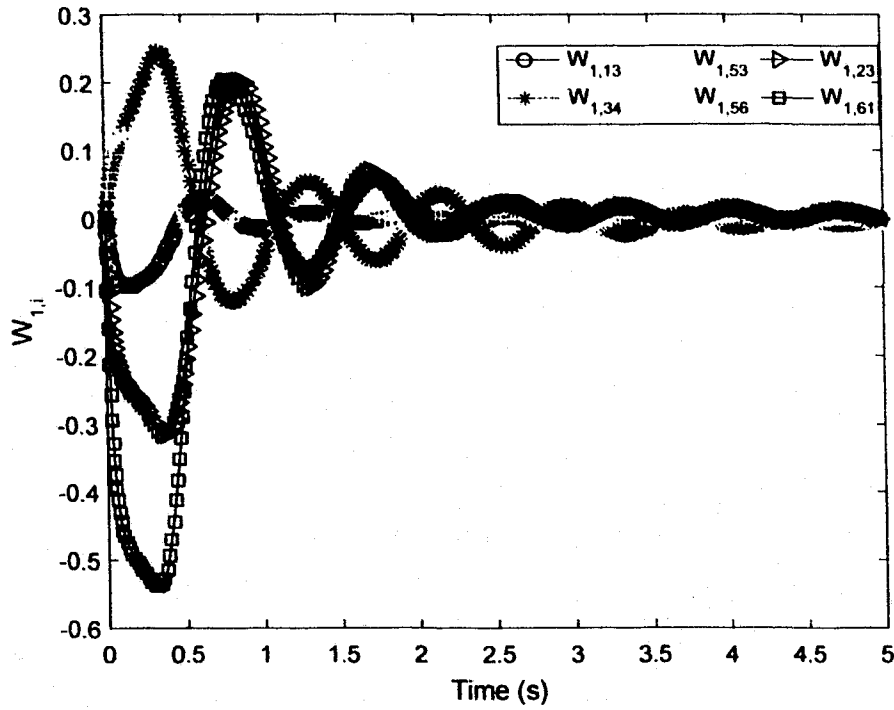


Figure 4.7: Sample of NN weights W_{1ij} .

4.3.2 Single Integrator Model with External Disturbances

In practical applications of unmanned vehicles formations, there exist various uncertainties that act on the vehicles due to factors such as imprecise sensor measurements and external disturbances. NNs have the capability of approximating any smooth functions over a compact set to arbitrary accuracy. Therefore, NN is a powerful technique for control of systems when there are large uncertainties and nonlinearities.

An additional set of simulation was performed to show the performance of the control law (4.10) in the presence of external disturbances. The following dynamics were used for the vehicles

$$\dot{p}_i = u_i + \Delta_i, \quad i = 1, \dots, n, \quad (4.25)$$

where $\Delta_i \in \mathbb{R}^3$ represents the external disturbances. The initial conditions used for these simulations were the same as in (4.24). The value of k_p was set to 1 in (4.10), $L = 240$ was chosen as the total number of hidden layer nodes, the entries of matrix V were random values, the weight matrix W was initiated at zero, S was a diagonal matrix with 0.1 on the main diagonal, and k_c was set to 1 in (4.11). The external disturbance Δ_i was modeled as

$$\Delta_1^T = [\cos(t^2), \sin(t^2), 0]$$

$$\Delta_2^T = [\sin(t^2), \cos(t^2), 0]$$

$$\Delta_3^T = [\cos(t^2), \sin(t^2), 0]$$

$$\Delta_4^T = [-\sin(t^2), \cos(t)\sin(-3t), 0]$$

$$\Delta_5^T = [\sin(t)\cos(-3t), \sin(t)\cos(t^2), \cos(t)\sin(-3t)]$$

$$\Delta_6^T = [\sin(-3t), \sin(t^2), \cos(t)\sin(-3t)].$$

Figure 4.8 shows the trajectories for the four UGVs and two UAVs as they move from their initial position to the final position to form the desired formation in spite of the external disturbances. Figure 4.9 demonstrates a sample of distance errors e_{ij} approaching bounded neighborhood of zero. The control input $u_i(t)$ for $i = 1, \dots, 6$ in the x and y directions are shown in Figure 4.10 and Figure 4.11. Figure 4.12 shows the control input $u_i(t)$ for $i = 5, 6$ in the z -direction. Figure 4.13 shows the NN weights in the presence of external disturbances. As it can be seen in these simulations, the system is stable in the presence of external disturbances and the vehicles achieve the desired formation.

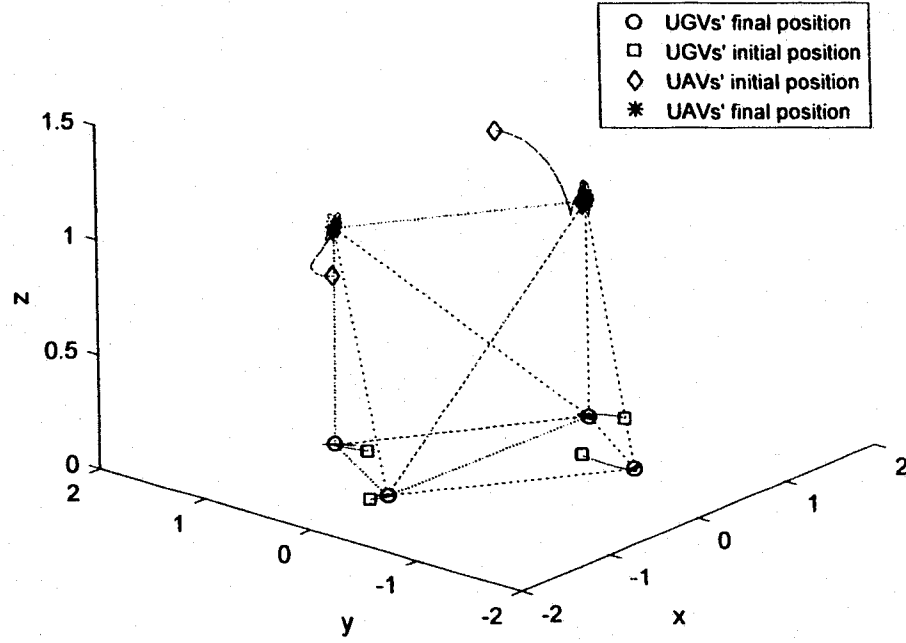


Figure 4.8: Vehicle trajectories $p_i(t)$, $i = 1, \dots, 6$ (solid line) and desired formation (dotted line) in the presence of external disturbances.

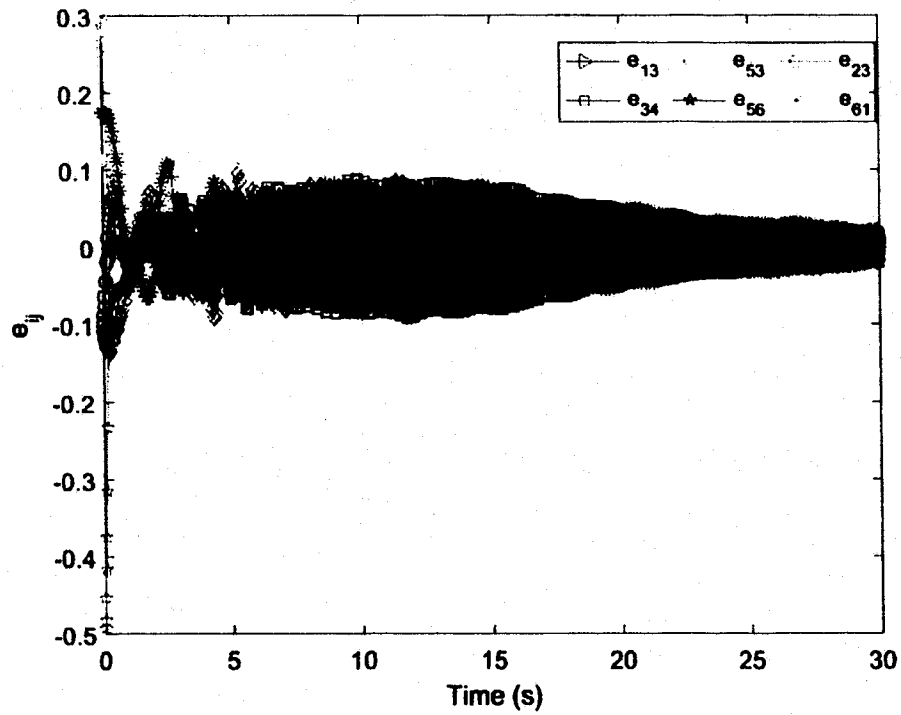


Figure 4.9: Sample of distance errors $e_{ij}(t)$, $i, j \in V$ with external disturbances.

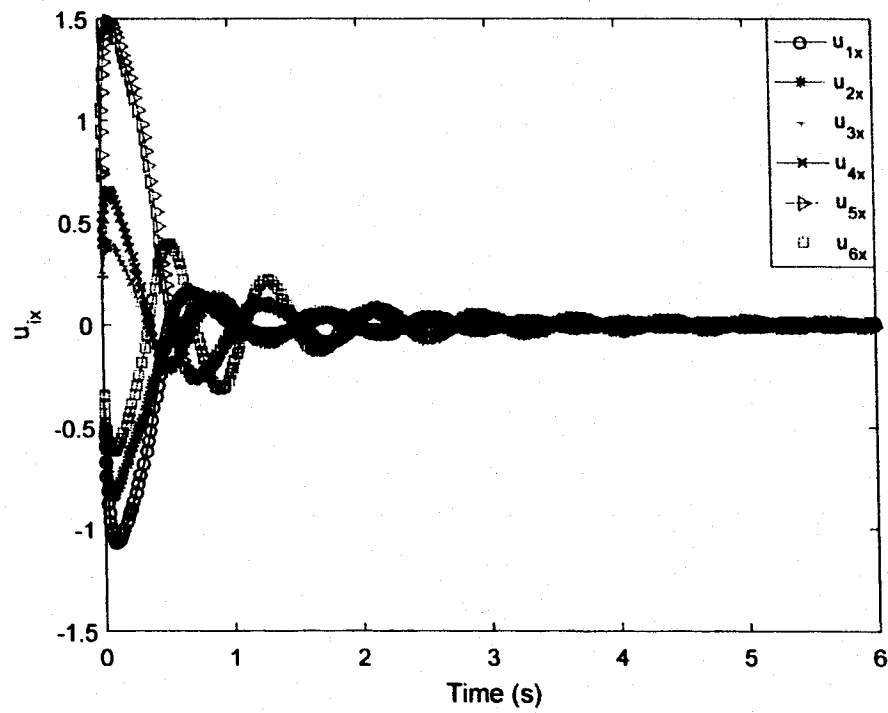


Figure 4.10: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the x -direction.

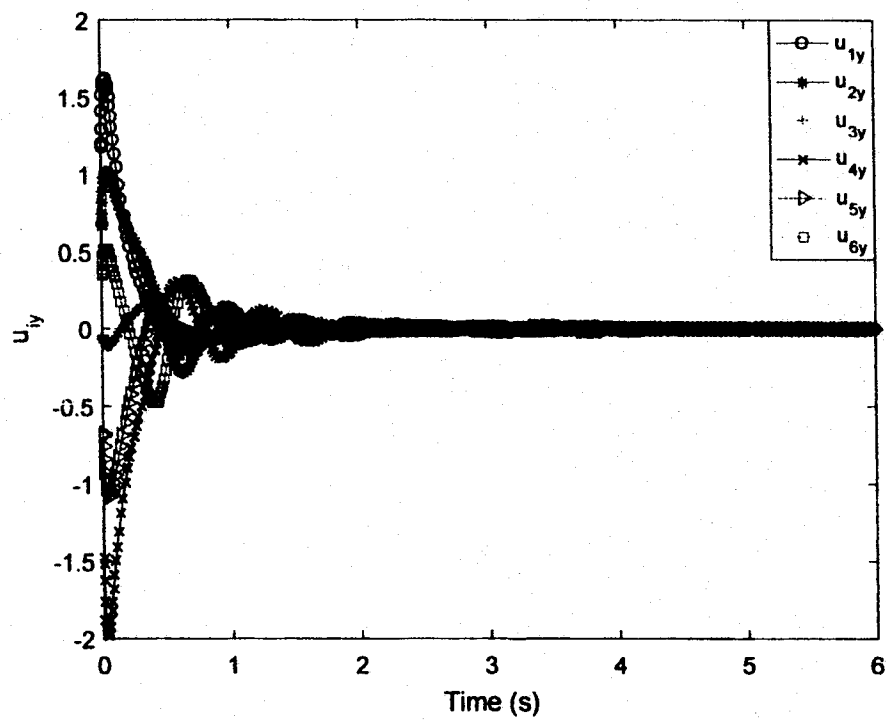


Figure 4.11: Control inputs $u_i(t)$, $i = 1, \dots, 6$ in the y -direction.

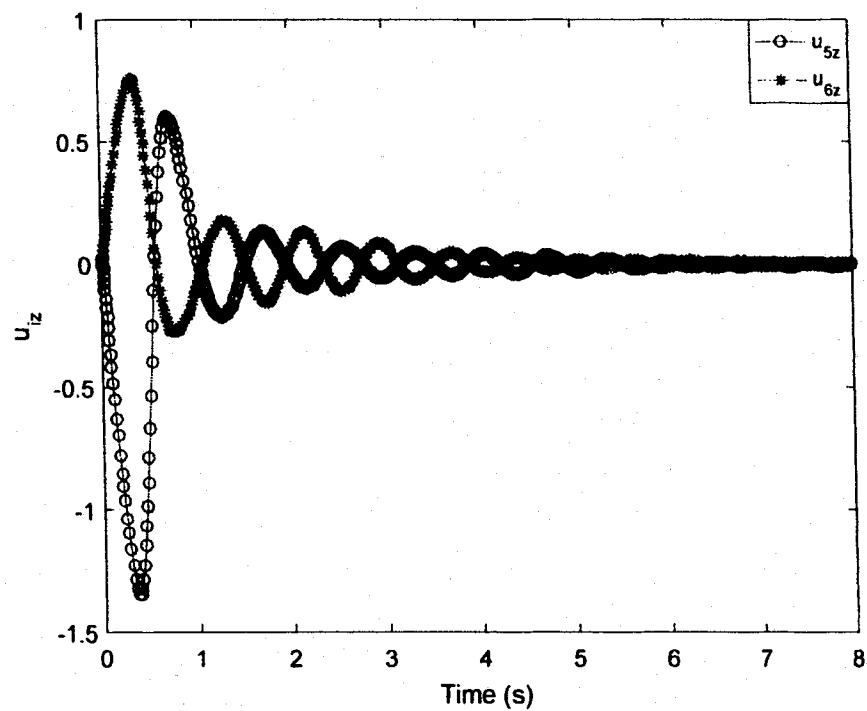


Figure 4.12: Control inputs $u_i(t)$, $i = 5, 6$ in the z -direction for the UAVs.

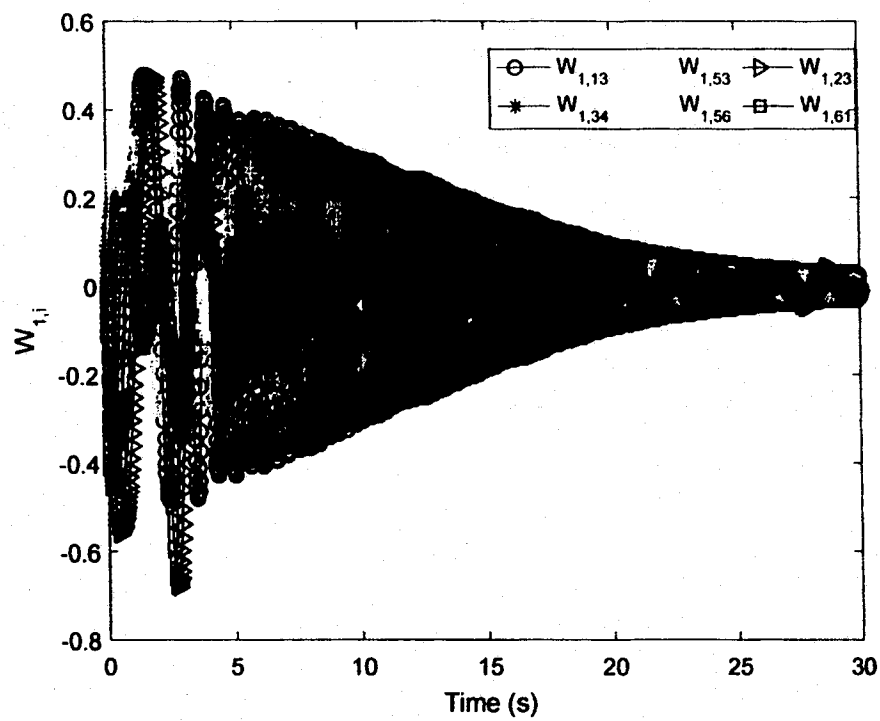


Figure 4.13: Sample of NN weights W_{1ij} with external disturbances.

The result of the simulation of the NN control law with and without external disturbances was compared with the result of the simulation of the single integrator control law in (3.13). The comparison of $\sum |e_{ij}|$ for all $(i, j) \in E$ without the external disturbances is shown in Figure 4.14. It can be seen that using both control laws, $\sum |e_{ij}| \rightarrow 0$. Figure 4.15 shows the comparison of $\sum |e_{ij}|$ for all $(i, j) \in E$ in the presence of external disturbances. This comparison shows that NN control law performs better than the control law in (3.13) as $\sum |e_{ij}| \rightarrow 0$ faster in the beginning and approaches to a smaller neighborhood of zero towards the end of the simulation.

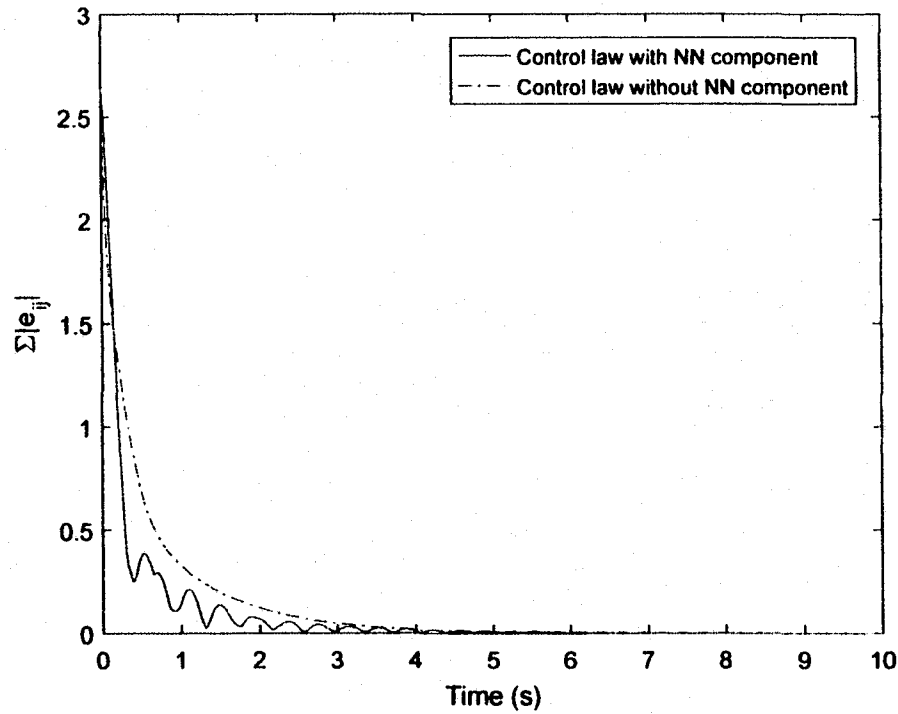


Figure 4.14: Comparison of $\sum |e_{ij}|$ for the control laws with and without NN component and without external disturbances.

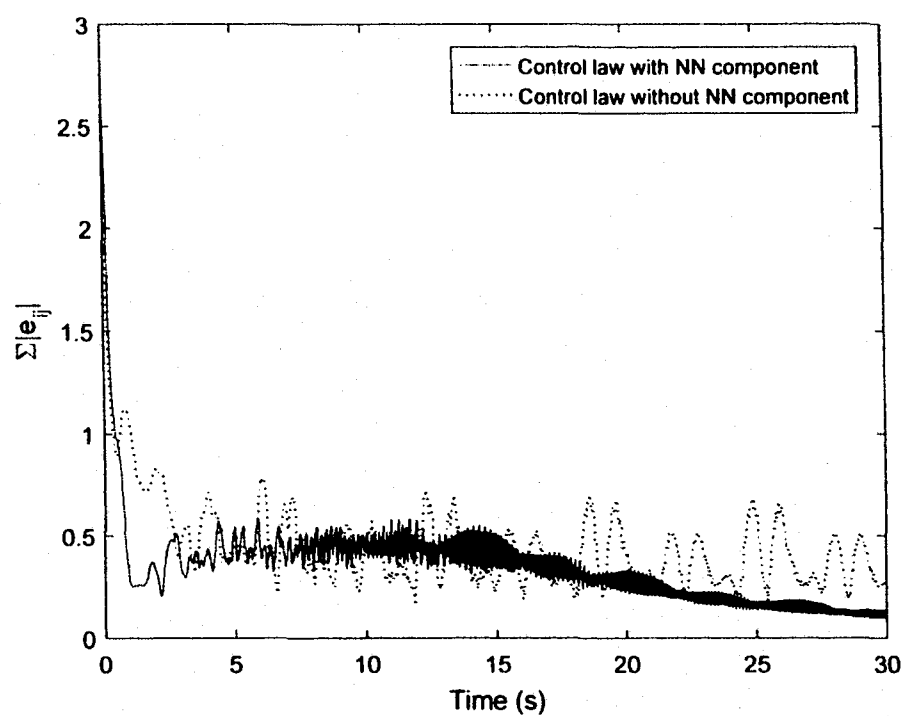


Figure 4.15: Comparison of $\sum |e_{ij}|$ for the control laws with and without NN component in the presence of external disturbances.

CHAPTER 5

UNMANNED VEHICLES SEARCH IN 2D SPACE

5.1 Methodology

In this research, the exploration of an unknown environment that is partitioned into hexagonal shapes by unmanned vehicles that use the designed search metaheuristic is studied.

5.1.1 Partitioning Method

After the environment is partitioned into a set of hexagonal cells and a map of search space is created, the vehicles will update their maps. This update includes omnidirectional sensor measurements and information obtained through communication with other vehicles in the field. Hexagonal cells in a beehive structure have been proved to be more efficient for mobile agents that are equipped with a circular tool because hexagonal grids provide a better approximation for that tool, which in our case is an omnidirectional antenna [71].

5.1.2 Search Algorithm

In this work, a Tabu-based search algorithm is used to support and improve a random search for target(s) in the search environment. This hybrid algorithm is called Tabu Random search algorithm and is distributed among multiple vehicles within the

given environment. The vehicles act independently of each other in their efforts to avoid hazards within the partitioned space and to search for the target.

In the simulation, it is assumed that the unmanned vehicles are equipped with a sonar sensor that is used for detecting physical obstacles and nearby vehicles in neighboring cells. The unmanned vehicles are also equipped with a sensor that is capable of detecting the target(s) based on the Received Signal Strength (RSS). A target is found by the vehicles when the value of RSS is larger than zero. Each vehicle is capable of wirelessly communicating information about the environment and is also capable of moving in a synchronized motion. This ad-hoc communication occurs as long as the vehicles are within each others communication range causing them to search cooperatively and move in a formation. Within such communication framework, vehicles are able to notify other nearby vehicles of their individual movement plan. The individual movement plan for each vehicle involves the vehicle notifying all vehicles of the cell it is currently occupying and the adjacent cell that it plans to move into. If a vehicle picks a cell that is occupied or has been chosen by another vehicle, the vehicle picks another adjacent cell and evaluates its availability. If all of the adjacent cells are not available, the vehicle remains in its current cell until all other connected vehicles have moved. This procedure prevents collision by preventing vehicles from travelling into cells that could cause vehicles to collide with each other. Once each vehicle has picked its plan of action and is ready to move, the synchronized movement between cells occurs by the vehicles cooperatively electing a leader to commence and lead the movement. This leader-based flocking occurs when vehicles are in close proximity of each other and causes the vehicles to simultaneously move in formation. This

formation prevents vehicles from colliding with each other because potential collisions are resolved before the leader commences the synchronized movement. The leader based flocking method provides cohesion to vehicles that are in close proximity of each other, which helps to prevent collisions [72]. However, since the proposed algorithm is distributed, it does not require persistent connection with other vehicles and will continue to search for target(s) if connectivity is lost with the leader or other vehicles. By using sensors and communicating with nearby vehicles, an individual vehicle is able to determine if the path from its currently occupied cell to a neighboring cell is safe for travel. This movement from one cell to an adjacent cell is called a logical step. A vehicle goes through a sequence of machine instructions to determine which cell it should travel to based on certain conditions. A vehicle uses a Tabu Random search heuristic for stochastically selecting an adjacent cell that is safe to move into.

The Tabu Random search is a meta-heuristic based on conventional random search where items are randomly selected, but the items are then added to a fixed sized sequence called a Tabu list [53]. Each vehicle has a local database that stores information regarding each cell in the partitioned space with a timestamp based on the time the information on the cell was last updated. After each logical step, the vehicle updates its local database with the latest information by synchronizing its database with other connected vehicles. The Boolean attributes for each cell in the database are tabu, hazard, and target near. Each cell also has a timestamp, t_c , which is updated every time an attribute is changed. In certain situations, a vehicle might be in a cell that is fully surrounded by tabu cells. In these cases, the vehicle might need to determine the aspiration criteria of a cell to override its tabu status to reach other

non-tabu cells [73]. For the Tabu Random search Algorithm, the aspiration criteria are determined by finding the neighboring cell with the smallest t_c timestamp. The following provides the pseudo code for the execution loop of this distributed search algorithm.

Algorithm 1

```

OtherAgents = (get other agents in ad-hoc network);
if RSS > 0 then
    Set attribute target near for current Cell to True;
end if
Add current cell to Tabu list and set tabu attribute to True;
options = (make a list of adjacent cells that are not tabu and not hazardous);
notAllow = (make a list of adjacent cells that are considered hazardous or contain
agents);
adTabuCells = (make a list of adjacent tabu cells);
NewCell = null;
while options > 0 do
    NewCell = (randomly select and remove a cell in the options list);
    Face the direction of this cell and scan cell for obstacles using sonar sensor;
    if obstacle is detected: then
        Add cell to notAllow list;
        if "cell is not occupied by an agent set hazard attribute to True" then
            NewCell = null;
        end if
    else
         $t_p$  = (current time);
        Break out of loop;
    end if
end while
if NewCell == null then
    Newcell = (tabu cell in adTabuCells list with highest aspiration rating);
     $t_p$  = (current time);
end if
Leader = pick a leader agent that has the smallest  $t_p$  between itself and the
OtherAgents list;
Leader moves agent to NewCell;
Repeat Loop;

```

The assumptions made in this work are:

- Assumption 1: The boundaries of the search area and its size are known and an initial map of the search area is provided for the unmanned vehicles.
- Assumption 2: The unmanned vehicles move from the center of one cell to the center of another cell in the search area.
- Assumption 3: If there is an obstacle in a cell, the entire cell is considered occupied.

5.2 Simulation

Here, the efficiency of using the Tabu Random search algorithm along with a beehive partitioning structure to search an environment that contains obstacles for target(s) by a cooperative team of unmanned vehicles is demonstrated. The search scenario adds obstacles into the search area that would possibly be encountered in a real life scenario such as buildings, cars, mountains, and trees. etc.

In order to establish the efficiency of the presented method, a series of simulations have been conducted and the results are presented in this section. The simulations are performed in two different search environments: simple and complex. The search environment is the area that is used for the simulation of the proposed method. For the simulations, the search environments needed to be varied enough to provide a suitable data set to be collected and analyzed based off of the performance of the unmanned vehicles system. Two different types of deployments are used in each environment: Randomized and Localized. In the randomized deployment, the vehicles are inserted randomly into the search area. In the localized deployment, the

vehicles are inserted into the search area from its upper right corner. The simulations were performed using a simulator created with the Jython programming language. Jython is an implementation of Python programming language written in Java. The two different search environments that were designed for simulations are as follows.

5.2.1 Simple Environment with Randomized Deployment

The first environment is a search area that contains one stationary obstacle and one stationary target. Two unmanned vehicles were deployed randomly in this area to search for the target. For the simulations in the simple environment, a rectangular area was partitioned with hexagonal cells and an area of equal size was partitioned using square cells. The unmanned vehicles were placed in the same coordinates in both environments. The vehicles did not have knowledge of the position or existence of the target and the obstacle. The simple environment with Randomized Deployment (RD) and hexagonal and square partitioning is shown in Figure 5.1 and Figure 5.2.

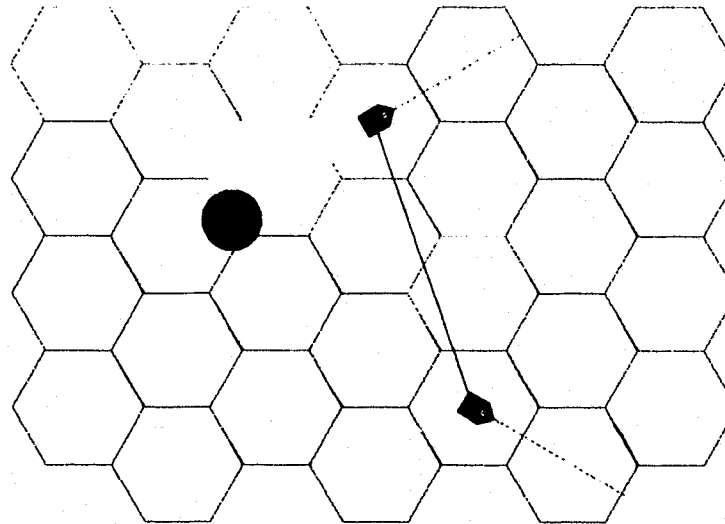


Figure 5.1: Hexagonal partitioned environment with one obstacle, one target, and two unmanned vehicles with RD.

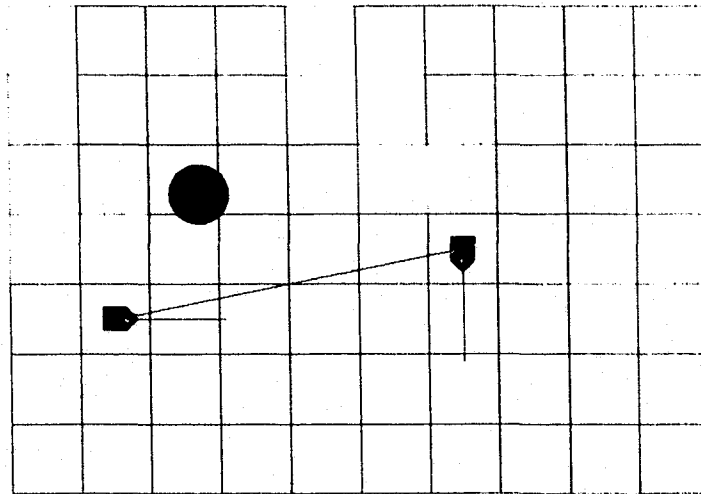


Figure 5.2: Square partitioned environment with one obstacle, one target, and two unmanned vehicles with RD.

In Figure 5.1 and Figure 5.2, the pentagon shapes represent the unmanned vehicles. The small circle on the vehicles represents their on-board sonar sensor and the line attached to the vehicles represents the sensing range of the sensor. The black filled-in circle is the obstacle and the other circle is the target in the search area. Moreover, in the simulator, the cells that have not been explored by the vehicles are designated with the color blue, the cells that have been explored by the vehicles are designated with the color red, and the cells that contain an obstacle are designated as hazard cells. When the vehicles are in each other's communication range, a black line will appear between them.

5.2.2 Simple Environment with Localized Deployment

For the second simple environment simulations, the vehicles were inserted into the search area from the upper right corner of the environment. This simulates the scenario where vehicles enter the search area at the same location. For the

simulations of the simple environment using this method of deployment, the search area was partitioned with hexagonal cells and square cells. The simple environment with Localized Deployment (LD) and hexagonal and square partitioning is shown in Figure 5.3 and Figure 5.4.

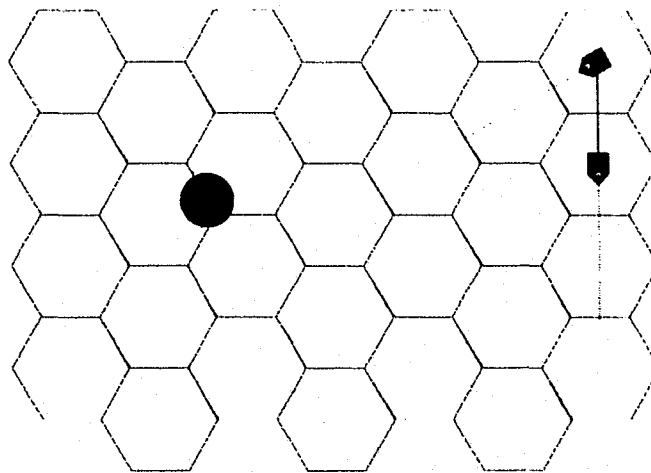


Figure 5.3: Hexagonal partitioned environment with one obstacle, one target, and two unmanned vehicles with LD.

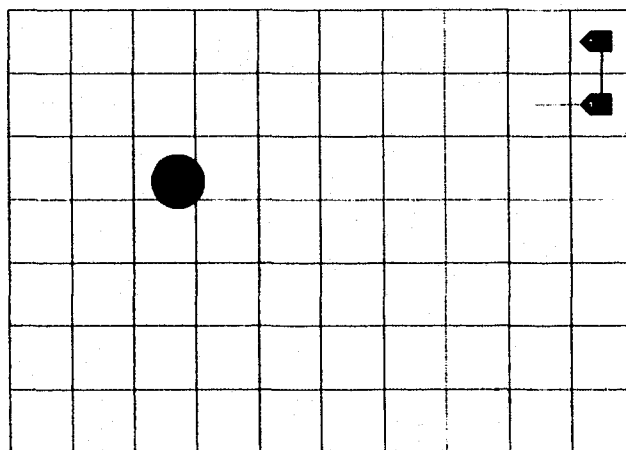


Figure 5.4: Square partitioned environment with one obstacle, one target, and two unmanned vehicles with LD.

5.2.3 Complex Environment with Randomized Deployment

The second environment is a search area that has larger number of cells and is more complex than the first environment. This search area consists of three stationary targets and two stationary obstacles. Four unmanned vehicles were deployed in this area to search for the target. For the simulations in the complex environment, a rectangular area was partitioned with hexagonal cells and an area of equal size was partitioned using square cells. The unmanned vehicles were placed randomly in the same coordinates in both environments. The vehicles did not have knowledge of the position or existence of the targets and the obstacles. The complex environment with RD and hexagonal and square partitioning is shown in Figure 5.5 and Figure 5.6. In these figures, the black filled-in circles are the obstacle and the other circles are the targets in the search area.

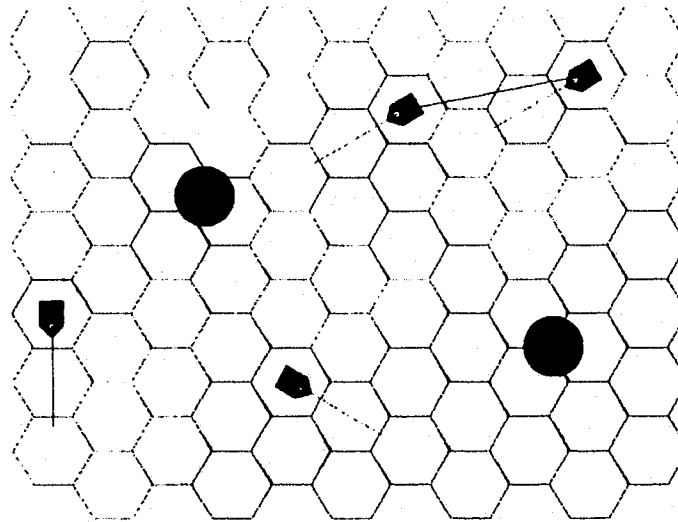


Figure 5.5: Hexagonal partitioned environment with two obstacles, three targets, and four unmanned vehicles with RD.

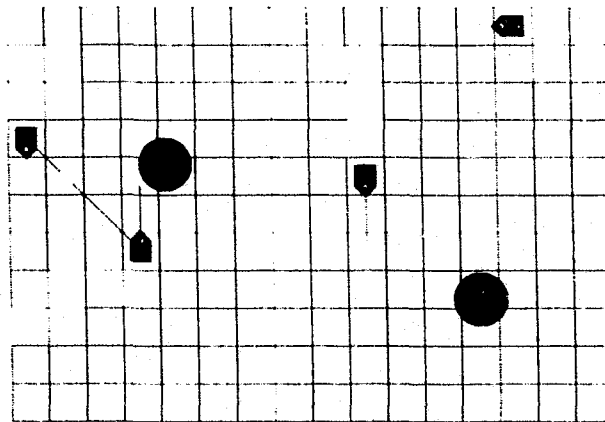


Figure 5.6: Square partitioned environment with two obstacles, three targets, and four unmanned vehicles with RD.

5.2.4 Complex Environment with Localized Deployment

For the second complex environment simulations, the vehicles were inserted into the search area from the upper right corner of the environment. For these simulations, the search area was partitioned with hexagonal cells and square cells. The complex environment with LD and hexagonal and square partitioning is shown in Figure 5.7 and Figure 5.8.

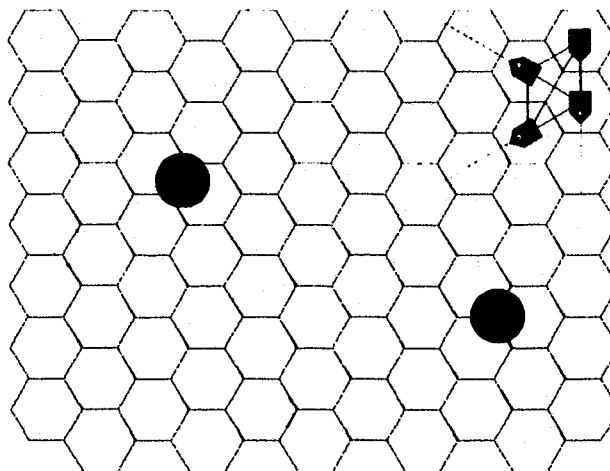


Figure 5.7: Hexagonal partitioned environment with two obstacles, three targets, and four unmanned vehicles with LD.

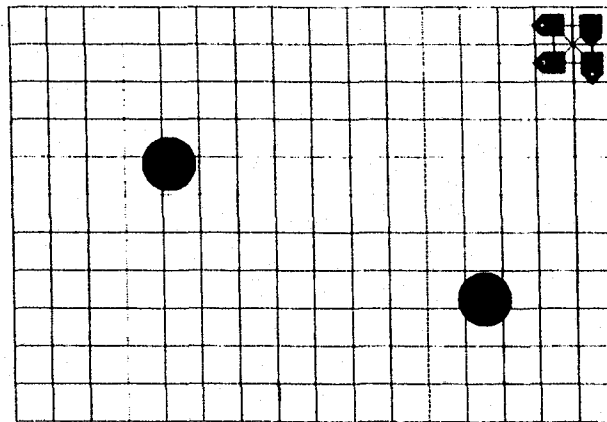


Figure 5.8: Square partitioned environment with two obstacles, three targets, and four unmanned vehicles with LD.

In both simple and complex environments, the unmanned vehicles have an on-board sonar sensor. If the perimeter of a hexagonal cell and square cell is designated with p , then the sonar's sensing range for the two partitioned areas can be calculated as a function of perimeter:

$$S_H = \frac{\sqrt{3}p}{4} \quad (5.1)$$

and

$$S_S = \frac{3p}{8}, \quad (5.2)$$

where S_H is the sonar's sensing range in a hexagonal partitioned area and S_S is the sonar's sensing range in a square partitioned area. The sensing range needs to be this size so that the vehicles can sense the presence of a physical obstacle in the adjacent cell. The vehicles detect obstacles in an adjacent cell by rotating their facing direction in such a way that the sonar sensor's orientation angle covers the adjacent cell. The sonar sensor's orientation angle is 60 degrees for the hexagonal partitioned area and it

is 90 degrees for the square partitioned area. The unmanned vehicles are also equipped with a sensor for detecting the EM source. The sensing range of this sensor is twice the size of one side of a hexagonal cell in beehive partitioned area and twice the size of one side of a square cell in a square partitioned area. The sensing orientation angle of the sensor used for detecting the target is 360 degrees in both partitioned search areas. During the search, the vehicles are capable of wirelessly communicating information about the environment. This ad-hoc communication is possible when the vehicles are within each other's communication range. If the perimeter of a hexagonal cell and square cell is designated with p , then each vehicle's communication range can be calculated as a function of perimeter using

$$C_H \geq \frac{5\sqrt{3}p}{12} \quad (5.3)$$

and

$$C_S \geq \frac{5p}{8}, \quad (5.4)$$

where C_H is each vehicle's communication range in a hexagonal partitioned area and C_S is each vehicle's communication range in a square partitioned area. The reason why the communication range needs to be this size is because a vehicle needs to be able to communicate with vehicles that are at least two cells away to prevent collision. Otherwise, if a vehicle moves in an adjacent cell that has been chosen by another vehicle that is not within the communication range, a collision will occur (as shown in Figure 5.9). With a communication range of this size, vehicles can infer that an adjacent cell is safe to travel into.

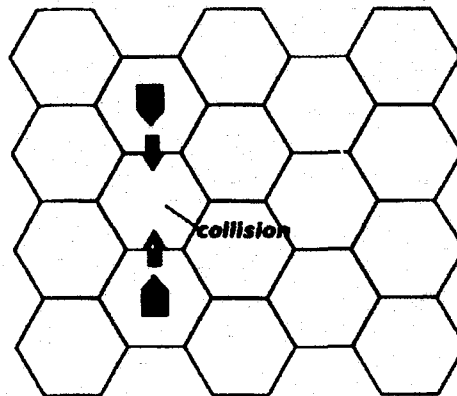


Figure 5.9: Collision occurs if the same adjacent cell is chosen by the vehicles to move into.

The simulations were performed using the Tabu Random search algorithm for the two search environments mentioned above. We are interested in comparing the distance that the unmanned vehicles travel to find the target(s) and the number of logical steps they take to find the target(s) in both partitioned areas. After a number of runs using the simulation of the Tabu Random search algorithm with the same environment and conditions, different simulation results were observed every time. Therefore, the simulations were run in the two environments mentioned above 50 times for each partitioned area for a total of 400 simulations. Then, the average travelled distance and the number of logical steps taken by the vehicles using this algorithm to find the target during the simulations were calculated.

5.3 Results

In this section the results of the simulations in both environments with hexagonal and square partitioning are presented. In the following tables, the travelled distance by the vehicles is presented based on unit length.

5.3.1 Simple Environment

This section presents the simulation results for the simple environment during the 50 runs of simulation for the beehive structured area and 50 simulations for square partitioned area using the RD method. The simulation results for the simple environment during the 50 runs of simulation for the beehive structured area and 50 simulations for square partitioned area using the LD method are demonstrated in Table 5.1 and Table 5.2, respectively. Note that the travelled distance is presented with units of meters.

Table 5.1: Results of the search for the target by both vehicles in the simple environment using RD.

Randomized Deployment	Beehive Structure		Square Partitioning	
	Vehicle 1	Vehicle 2	Vehicle 1	Vehicle 2
Avg travelled distance	10.47	10.51	14.05	14.05
Avg number of logical steps	6.54	6.56	11.58	11.58
Avg total travelled distance	20.98		28.09	
Avg total number of logical steps	13.1		23.16	

As it can be seen in Table 5.1 and Table 5.2, the average travelled distance and the number of logical steps taken by each of the vehicles using both methods of deployment are smaller in the beehive partitioned area than in the square partitioned area. Also, the average total travelled distance and number of logical steps taken by

Table 5.2: Results of the search for the target by both vehicles in the simple environment using LD.

Localized Deployment	Beehive Structure		Square Partitioning	
	Vehicle 1	Vehicle 2	Vehicle 1	vehicle 2
Avg travelled distance	5.38	5.60	8.59	8.59
Avg number of logical steps	3.36	3.50	7.08	7.08
Avg total travelled distance	10.98		6.86	
Avg total number of logical steps	17.18		14.16	

the vehicles using both LD and RD methods are smaller in the beehive partitioned area than in square partitioned area.

Table 5.3 and Table 5.4 present the percentage of target detection by each vehicle during the 50 runs of the simulation in beehive structured area and 50 simulations for square partitioned area using RD and LD methods. These tables also present the distance between each vehicle and the target in four search scenarios.

Table 5.3: Results of the search and percentage of target detection by each vehicle in the simple environment using RD.

Randomized Deployment	Beehive Structure		Square Partitioning	
	Vehicle 1	Vehicle 2	Vehicle 1	Vehicle 2
% of target detection	22.00	78.00	16.00	84.00
Avg travelled distance to the target	12.67	9.89	17.89	13.32
Avg number of logical steps to the target	7.91	6.18	14.75	10.97
Distance to the target	7.91	4.89	7.91	5.51

Table 5.4: Results of the search and percentage of target detection by each vehicle in the simple environment using LD.

Localized Deployment	Beehive Structure		Square Partitioning	
	Vehicle 1	Vehicle 2	Vehicle 1	Vehicle 2
% of target detection	10.00	90.00	28.00	72.00
Avg travelled distance to the target	10.89	4.77	8.96	8.79
Avg number of logical steps to the target	6.80	2.98	7.67	7.25
Distance to the target	4.67	3.31	4.76	3.94

As it can be seen in Table 5.3 and Table 5.4, the percentage of target detection by vehicle 1 is larger than vehicle 2 in the beehive structured area and the square partitioned area using both RD and LD methods. Also, the average travelled distance and the number of logical steps taken by vehicle 1 is larger than the ones for vehicle 2. It can be concluded from the results presented in Table 5.3 and Table 5.4 that in the simple environment, a vehicle that is closer to the target will find the target with smaller average travelled distance and by taking less number of logical steps. This is true for both beehive structured area and square partitioned area.

5.3.2 Complex Environment

Table 5.5 and Table 5.6 present the simulation results for the complex environment during the 50 runs of simulation for the beehive structured area and 50 simulations for square partitioned area using both RD and LD methods. The travelled distance and number of logical steps presented in this table are from the beginning of the search until all three targets are found and the search is stopped.

Table 5.5: Results of the search for the targets by all the vehicles in the complex environment using RD and beehive structure.

	Randomized Deployment		Beehive Structure	
	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4
Avg travelled distance	15.34	15.35	15.68	15.49
Avg number of logical steps	15.82	15.82	16.16	15.96
Avg total travelled distance	61.86			
Avg total number of logical steps	63.76			
% of target detection	16.67	12	24.67	46.67
Avg travelled distance to the target	9.98	14.29	12.09	5.56
Avg number of logical steps to the target	10.32	14.72	12.46	5.74

As it can be seen in Table 5.5 and Table 5.6, the average travelled distance and the number of logical steps taken by each of the vehicles using both methods of

Table 5.6: Results of the search for the targets by all the vehicles in the complex environment using RD and square partitioning.

Randomized Deployment	Square Partitioning			
	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4
Avg travelled distance	15.42	15.63	16.34	15.58
Avg number of logical steps	20.56	21.16	22.1	21.08
Avg total travelled distance	62.97			
Avg total number of logical steps	84.9			
% of target detection	26.00	13.33	25.33	35.33
Avg travelled distance to the target	2.85	14.21	12.27	8.32
Avg number of logical steps to the target	3.85	19.20	16.63	11.26

deployment are smaller in the beehive partitioned area than in the square partitioned area. Also, the average total travelled distance and number of logical steps taken by the vehicles using both LD and RD methods are smaller in the beehive partitioned area than in square partitioned area.

Table 5.7 and Table 5.8 present the percentage of target detection by each vehicle during the 50 runs of the simulation in beehive structured area and 50 simulations for square partitioned area using RD and LD methods.

Table 5.7: Results of the search for the targets by all the vehicles in the complex environment using LD and beehive structure.

Localized Deployment	Beehive Structure			
	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4
Avg travelled distance	14.66	16.87	16.89	16.42
Avg number of logical steps	15.20	17.42	17.42	16.90
Avg total travelled distance	64.84			
Avg total number of logical steps	66.94			
% of target detection	6.66	20.00	38.66	34.66
Avg travelled distance to the target	12.81	13.62	9.71	11.61
Avg number of logical steps to the target	13.20	14.07	10.02	11.96

The results in Table 5.7 and Table 5.8 show that in the beehive structured area using both methods of deployment, the vehicles with the highest percentage of target

Table 5.8: Results of the search for the targets by all the vehicles in the complex environment using LD and square partitioning.

Localized Deployment	Square Partitioning			
	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4
Avg travelled distance	25.16	27.97	26.69	25.99
Avg number of logical steps	34.12	37.88	36.08	35.30
Avg total travelled distance	105.82			
Avg total number of logical steps	143.38			
% of target detection	18.00	25.33	26.66	30.00
Avg travelled distance to the target	20.61	25.34	10.73	14.83
Avg number of logical steps to the target	27.89	34.31	14.50	20.07

detection had to travel a shorter distance and take less number of logical steps to find the target(s). However, the same conclusion cannot be drawn from the results of the simulation in the square partitioned area using both methods of deployment.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this work, first the problem of formation acquisition for a group of n lower layer vehicles that belong to a plane Q and r upper layer vehicles that move outside of the plane Q was considered. Graph coning and triangulation concepts were used to create a three-dimensional framework that retains the infinitesimal and minimal rigidity characteristics of the framework of n lower layer vehicles. By using the proposed approach, the inter-vehicle distances were stabilized to acquire a pre-defined shape in three dimensions. Sufficient conditions were provided for the initial conditions that guarantee convergence of the layered formation to the desired framework. The simulation results showed that the proposed control method, using single and double integrator models, yields asymptotic stability of the desired formation.

Additionally, to solve the formation control problem for an n vehicle system in a 3D space, a decentralized NN-based controller was introduced to stabilize inter-vehicle distances to desired values. This method which makes use of graph rigidity, and a Lyapunov analysis showed that the designed control law ensures the uniformly ultimately bounded stability of the infinitesimally and minimally rigid desired formation. The nonlinear control law consists of a nonlinear component that depends on inter-vehicle distances and a neural net component.

Another problem considered in this dissertation was the cooperative unmanned vehicles search. The use of Tabu Random Search algorithm as a solution to the problem of searching for target(s) in an environment that contains obstacle(s) by a team of unmanned vehicles that move in a cooperative manner was presented. In this method, beehive partitioning of the search area was used along with the Tabu Random search algorithm in order to develop a novel approach for searching for the target(s) while avoiding the obstacles. Tabu list and Aspiration Criteria are two main components of Tabu Random search algorithm used in this work. The aspiration criteria prevents the vehicles from cycling and being trapped in the search area. A simulator was developed to demonstrate the efficiency of using the Tabu Random search algorithm along with a beehive partitioning structure as a solution to the cooperative unmanned vehicles search problem of an environment. The results of the simulations in two different search environments using Randomized and Localized Deployment of the vehicles show that the proposed search algorithm works effectively along with beehive partitioning structure to search an environment for target(s). The simulations were also performed in square partitioned environments and the results were compared to the results obtained from the simulations in the beehive structured environments. The comparisons show that the use of Tabu Random search algorithm along with a beehive partitioning structure is more effective for searching an environment than using this search algorithm along with a square partitioning of the environment.

As a future work, the control algorithms for layered formations would be implemented on UAVs available at Micro-Aerial Vehicles and Sensor Networks (MAVSeN) laboratory at Louisiana Tech University to test their performance on actual hardware.

For the NN-based formation control, the current results would be extended to double integrator vehicle model and the performance of the NN approach would be examined further using different types of noise and disturbances. A possible future direction for the cooperative unmanned vehicles search study would be to compare the performance of Tabu Random search with the performance of another distributed search algorithm such as Simulated Annealing and also to implement the proposed search algorithm on the UAVs at MAVSeN laboratory.

BIBLIOGRAPHY

- [1] M. Mirzaei, F. Sharifi, B. Gordon, C. Rabbath, and Y. Zhang, "Cooperative multi-vehicle search and coverage problem in uncertain environments," in *Proc. of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Dec 2011, pp. 4140–4145.
- [2] P. Sujit and D. Ghose, "Search using multiple uavs with flight time constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 491–509, April 2004.
- [3] V. Gazi and K. M. Passino, *Swarm Stability and Optimization*. Springer-Verlag Berlin Heidelberg, 2011.
- [4] X. Cai and M. de Queiroz, "Multi-agent formation maintenance and target tracking," in *Proc. of American Control Conference (ACC)*, June 17–19 2013, pp. 2521–2526.
- [5] D. Pack, P. DeLima, G. Toussaint, and G. York, "Cooperative control of uavs for localization of intermittently emitting mobile targets," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 4, pp. 959–970, 2009.
- [6] S. Ramazani, D. L. Jackson, and R. R. Selmic, "Cooperative mobile agents search using beehive partitioned structure and tabu random search algorithm," in *Proc. of SPIE 8741, Unmanned Systems Technology XV*, 2013, pp. 874 108–874 108.
- [7] X. Cai and M. de Queiroz, "Rigidity-based stabilization of multi-agent formations," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 1, January 2014.
- [8] W. Burgard and F. Schneider, "Collaborative exploration of unknown environments with teams of mobile robots," in *Proc. of Dagstuhl Seminar on Plan-based Control of Robotic Agents*. Springer Verlag, 2002, pp. 52–70.

- [9] B. D. Anderson, C. Yu, S. Dasgupta, and A. S. Morse, "Control of a three-coleader formation in the plane," *Systems and Control Letters*, vol. 56, no. 9 10, pp. 573–578, 2007.
- [10] C. Yu, J. M. Hendrickx, B. Fidan, B. D. Anderson, and V. D. Blondel, "Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence," *Automatica*, vol. 43, no. 3, pp. 387–402, 2007.
- [11] L. Asimow and B. Roth, "The rigidity of graphs, II," *Journal of Mathematical Analysis and Applications*, vol. 68, no. 1, pp. 171–190, 1979.
- [12] T.-S. Tay and W. Whiteley, "Generating isostatic frameworks," *Structural Topology*, vol. 11, pp. 21–69, 1985.
- [13] X. Cai and M. de Queiroz, "Adaptive rigidity-based formation control for multirobotic vehicles with dynamics," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 389–396, January 2015.
- [14] K.-K. Oh and H.-S. Ahn, "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, no. 12, pp. 1809–1820, 2014.
- [15] ———, "Distance-based control of cycle-free persistent formations," in *Proc. of IEEE Multi-Conference on Systems and Control*, Sept 2011, pp. 816–821.
- [16] M. Gates, R. Selmic, and R. Ordonez, "Cooperative control of MAVs for a hidden emitter localization," in *Proc. of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, vol. 8361, 2012.
- [17] P. Sujit and R. Beard, "Cooperative path planning for multiple uavs exploring an unknown region," in *Proc. of American Control Conference (ACC)*, July 2007, pp. 347–352.
- [18] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer-Verlag London, 2008.
- [19] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

- [20] T. Summers, C. Yu, S. Dasgupta, and B. D. O. Anderson, "Control of minimally persistent leader-remote-follower and coleader formations in the plane," *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2778–2792, 2011.
- [21] K.-K. Oh and H.-S. Ahn, "Formation control of mobile agents based on inter-agent distance dynamics," *Automatica*, vol. 47, no. 10, pp. 2306–2312, 2011.
- [22] L. Krick, M. Broucke, and B. Francis, "Stabilization of infinitesimally rigid formations of multi-robot networks," in *Proc. of 47th IEEE Conference on Decision and Control (CDC)*, 2008, pp. 477–482.
- [23] S.-M. Kang, M.-C. Park, B.-H. Lee, and H.-S. Ahn, "Distance-based formation control with a single moving leader," in *American Control Conference (ACC)*, June 2014, pp. 305–310.
- [24] M.-C. Park, K. Jeong, and H.-S. Ahn, "Formation stabilization and resizing based on the control of inter-agent distances," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 14, pp. 2532–2546, 2015.
- [25] M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos, "Coordinate-free formation stabilization based on relative position measurements," *Automatica*, vol. 57, pp. 11–20, 2015.
- [26] M.-C. Park, Z. Sun, B. Anderson, and H.-S. Ahn, "Stability analysis on four agent tetrahedral formations," in *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014, pp. 631–636.
- [27] R. Kelly and C. Monroy, "Interactive animation of agent formation based on hopfield neural networks," in *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 530–536.
- [28] Z.-G. Hou, L. Cheng, and M. Tan, "Decentralized robust adaptive control for the multiagent system consensus problem using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 3, pp. 636–647, 2009.
- [29] M. Bryant, P. Johnson, B. M. Kent, M. Nowak, and S. Rogers, "Layered sensing," Air Force Research Laboratory, White Paper Version 6, 1 May 2008.

- [30] A. Das and F. L. Lewis, "Cooperative adaptive control for synchronization of second-order systems with unknown nonlinearities," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 13, pp. 1509–1524, 2011.
- [31] P. Shi and Q. Shen, "Cooperative control of multi-agent systems with unknown state-dependent controlling effects," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 827–834, 2015.
- [32] B. Anderson, B. Fidan, C. Yu, and D. Walle, "UAV formation control: Theory and application," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences. Springer London, 2008, vol. 371, pp. 15–33.
- [33] A.-M. Zou and K. D. Kumar, "Neural network-based adaptive output feedback formation control for multi-agent systems," *Nonlinear Dynamics*, vol. 70, no. 2, pp. 1283–1296, 2012.
- [34] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *Journal of Combinatorial Theory, Series B*, vol. 94, no. 1, pp. 1–29, 2005.
- [35] B. D. O. Anderson, C. Yu, B. Fidan, and J. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Systems*, vol. 28, no. 6, pp. 48–63, 2008.
- [36] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, no. 4, pp. 331–340, 1970.
- [37] C. Yu, B. D. O. Anderson, S. Dasgupta, and B. Fidan, "Control of minimally persistent formations in the plane," *SIAM Journal of Control and Optimization*, vol. 48, no. 1, pp. 206–233, 2009.
- [38] W. Whiteley, "Some matroids from discrete applied geometry," 1996.
- [39] ———, "The rigidity of polyhedral surfaces," *Mathematics Magazine*, vol. 52, no. 5, pp. 275–283, Nov 1979.
- [40] R. Connelly, "Generic global rigidity," *Discrete Comput. Geom.*, vol. 33, no. 4, pp. 549–563, Apr. 2005.

- [41] H. Maehara, "Geometry of frameworks," *Yokohama Mathematical Journal*, 1999.
- [42] L. Asimow and B. Roth, "The rigidity of graphs," *Transactions of the American Mathematical Society*, vol. 245, pp. 279–289, 1978.
- [43] R. Connelly, "The rigidity of certain cabled frameworks and the second-order rigidity of arbitrarily triangulated convex surfaces," *Advances in Mathematics*, vol. 37, no. 3, pp. 272–299, 1980.
- [44] W. Whiteley, "Cones, infinity and 1-story buildings," *Structural Topology*, no. 8, pp. 53–70, 1983.
- [45] R. Connelly and W. Whiteley, "Global rigidity: The effect of coning," *Discrete & Computational Geometry*, vol. 43, no. 4, pp. 717–735, 2010.
- [46] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.
- [47] M. Pirlot, "General local search methods," *European Journal of Operational Research*, vol. 92, no. 3, pp. 493 – 511, 1996.
- [48] P. Merz and B. Freisleben, "Genetic local search for the tsp: new results," in *IEEE International Conference on Evolutionary Computation, 1997*, 1997, pp. 159–164.
- [49] K. Worrall, "Guidance and search algorithms for mobile robots: application and analysis within the context of urban search and rescue," Ph.D. dissertation, University of Glasgow, 2008.
- [50] L. M. Schmitt, "Theory of genetic algorithms ii: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling," *Theoretical Computer Science*, vol. 310, no. 13, pp. 181–231, 2004.
- [51] E. McGookin, "Optimisation of sliding mode controllers for marine applications: a study of methods and implementation issues," Ph.D. dissertation, University of Glasgow, 1997.

- [52] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986, applications of Integer Programming.
- [53] ———, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.
- [54] M. Gendreau, "An introduction to tabu search," in *Handbook of Metaheuristics*, ser. International Series in Operations Research & Management Science, F. Glover and G. Kochenberger, Eds. Springer US, 2003, vol. 57, pp. 37–54.
- [55] D. Herrmann, T. Kamphans, and E. Langetepe, "Exploring Simple Triangular and Hexagonal Grid Polygons Online," *Computational Geometry*, 2010.
- [56] V. E. Brimkov and R. P. Barneva, "honeycomb vs square and cubic models," *Electronic Notes in Theoretical Computer Science*, vol. 46, pp. 321 – 338, 2001.
- [57] P. Sujit and D. Ghose, "Multiple agent search of an unknown environment using game theoretical models," in *Proc. of American Control Conference (ACC)*, vol. 6, June 2004, pp. 5564–5569.
- [58] J. Riehl, G. Collins, and J. Hespanha, "Cooperative graph-based model predictive search," in *Proc. of 46th IEEE Conference on Decision and Control (CDC)*, Dec 2007, pp. 2998–3004.
- [59] Y. Yang, A. Hilinai, and M. Polycarpou, "Decentralized cooperative search in uav's using opportunistic learning," in *Proc. of the AIAA Guidance, Navigation, and Control Conference*, August 2002.
- [60] L. Chrupa and A. Komenda, "Smoothed hex-grid trajectory planning using helicopter dynamics," in *Proc. of the 3rd International Conference on Agents and Artificial Intelligence (ICAART)*, vol. 1, January 28–30 2011, pp. 629–632.
- [61] T. C. Hales, "The Honeycomb Conjecture," *Discrete and Computational Geometry*, pp. 1–22, 2001.
- [62] J. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "3-d relative positioning sensor for indoor flying robots," *Autonomous Robots*, vol. 33, no. 1–2, pp. 5–20, 2012.

- [63] F. Dörfler and B. Francis, "Geometric analysis of the formation problem for autonomous robots," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2379–2384, 2010.
- [64] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2609–2614, Oct 2012.
- [65] D. Jin and L. Gao, "Stability analysis of a double integrator swarm model related to position and velocity," *Transactions of the Institute of Measurement and Control*, vol. 30, no. 3–4, pp. 275–293, 2008.
- [66] H. Gluck, "Almost all simply connected closed surfaces are rigid," *Geometric Topology*, vol. 438, pp. 225–239, 1975.
- [67] H. Khalil, *Nonlinear systems*. Prentice Hall, 2002.
- [68] S. Ramazani, R. Selmic, and M. de Queiroz, "Non-planar multi-agent formation control using coning graphs," in *Proc. of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct 2014, pp. 3091–3096.
- [69] Y.-Y. Chen and Y.-P. Tian, "A backstepping design for directed formation control of three-coleader agents in the plane," *International Journal of Robust and Nonlinear Control*, vol. 19, no. 7, pp. 729–745, 2009.
- [70] Y. Zou and O. S. U. M. Engineering, *Distributed Control of Multiple Vehicle Systems Using Constraint Forces*. Oklahoma State University, 2008.
- [71] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geometry*, vol. 17, no. 12, pp. 25–50, 2000.
- [72] Z. Xin, Y. Maode, and J. Yongfeng, "A tabu search based flocking algorithm of motion control for multiple mobile robots," in *Fifth International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Jan 2012, pp. 48–52.
- [73] F. Glover, E. Taillard, and E. Taillard, "A user's guide to tabu search," *Annals of Operations Research*, vol. 41, no. 1, pp. 1–28, 1993.