COOPERATIVE COVERAGE CONTROL OF MULTI-AGENT SYSTEMS

Farid Sharifi

A thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

Concordia University

Montréal, Québec, Canada

September 2014

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Farid Sharifi**

Entitled: **Cooperative Coverage Control of Multi-Agent Systems**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Doctor of Philosophy)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Asim Al-Khalili

_____ Chair

Dr. Simon X. Yang

_____ External Examiner

Dr. Shahin Hashtrudi Zad

_____ Examiner

Dr. Chun-Yi Su

_____ Examiner

Dr. Wen-Fang Xie

_____ Examiner

Dr. Youmin Zhang

_____ Supervisor

Dr. Amir G. Aghdam

_____ Co-supervisor

Approved _____

Chair of Department or Graduate Program Director

_____ 20 _____ _____

Dean of the Faculty of Engineering and Computer Science

Abstract

Cooperative Coverage Control of Multi-Agent Systems

Farid Sharifi, Ph.D.

Concordia University, 2014

In this dissertation, motion coordination strategies are proposed for multiple mobile agents over an environment. It is desired to perform surveillance and coverage of a given area using a Voronoi-based locational optimization framework. Efficient control laws are developed for the coordination of a group of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) with double-integrator and non-holonomic dynamics. The autonomous vehicles aim to spread out over the environment while more focus is directed towards areas of higher interest. It is assumed that the so-called "operation costs" of different agents are not the same. The center multiplicatively-weighted Voronoi configuration is introduced, which is shown to be the optimal configuration for agents. A distributed control strategy is also provided which guarantees the convergence of the agents to this optimal configuration. To improve the cooperation performance and ensure safety in the presence of inter-agent communication delays, a spatial partition is used which takes the information about the delay into consideration to divide the field. The problem is also extended to the case when the sensing effectiveness of every agent varies during the mission, and a novel partition is proposed to address this variation of the problem. To avoid obstacles as well as collision between agents in the underlying coverage control problem, a distributed navigation-function-based controller is developed. The field is partitioned to the Voronoi cells first, and the agents are relocated under the proposed controller such that a pre-specified cost function is minimized while collision and obstacle avoidance is guaranteed. The coverage problem in uncertain environments is also investigated, where a number of search vehicles are deployed to explore the environment. Finally, the effectiveness of all proposed algorithms in this study is demonstrated by simulations and experiments on a real testbed.

Acknowledgments

First of all, I would like to express my profound gratitude to my supervisors, Dr. Youmin Zhang and Dr. Amir G. Aghdam for their invaluable support, encouragement and supervision without which this thesis would never have materialized. I am very grateful for the research freedom I was given and more so for the necessary corrections along my road of inquiry. I feel fortunate to work closely with two brilliant advisors that have different expertise, style and interest.

I would like to acknowledge my colleagues at Concordia University, who provided me with some fruitful collaborations. In particular, I would like to thank Dr. Brandon W. Gordon, Dr. Jalal Habibi, Dr. Hamid Mahboubi, Dr. Abbas Chamseddine and Mostafa Mirzaei with whom I co-authored a number of papers. Also I would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial assistance for this research.

I would like to extend my deepest gratitude and appreciation to my parents who have always encouraged me to move forward in my academic pursuits. I am grateful for their prayers and support in my entire life. Words cannot express my appreciation to my great brothers for their supports and goodwill.

Last but not least, I am blessed to have an outstanding wife, Samira. I would like to thank her for constant support, care, patience and faith in me. Without whom my life would never have been so glamorous and exciting; to whom I dedicate this thesis.

Contents

List of Figures **ix**

List of Tables **xiv**

1 Introduction **1**

1.1 Literature Review 3

1.2 Dissertation Outline 8

1.3 Research Contributions 9

2 Coverage Control of Multi-Agent Systems **15**

2.1 Introduction 15

2.2 Problem Statement 17

2.3 Centroidal Configuration 19

2.4 Dynamic Model of Vehicles 20

2.4.1 Model of a Quadrotor UAV 20

2.4.2 Model of UGV 23

2.5 Coverage with Multiple UGVs and UAVs 24

2.5.1 Control of UAVs 25

2.5.2 Control of UGVs 27

2.6 Simulation Results 30

3 Deployment Strategy for a Network of Non-Identical Agents 34

3.1 Problem Statement 35

3.2 Locational Optimization 37

3.2.1 Background 37

3.2.2 Optimal Configuration 38

3.3 Distributed Coverage Control 41

3.4 Simulation Results 44

3.5 Experimental Results 47

3.5.1 Unmanned Aerial Vehicle: the Qball-X4 47

3.5.2 Unmanned Ground Vehicles: the Qbot UGV and the Quanser QGV 49

3.5.3 Experimental Tests 50

4 Coverage Control of Multi-Agent Systems subject to Communication Delay 55

4.1 Problem Formulation 56

4.2 Inter-Agent Communication Delays 57

4.3 Guaranteed Voronoi Partitioning 59

4.4 Distributed Coverage Control in Faulty Situations 60

4.4.1 Deployment Strategy 61

4.4.2 Motion Control 61

4.4.3 Motion Coordination Algorithm 63

4.5 Coverage Control of Multi-Agent Systems subject to Communication Delay and Health Degradation 65

4.5.1 Guaranteed Multiplicatively-Weighted Voronoi Partitioning . . 67

4.5.2 Motion Control 69

4.6 Simulation Results 71

5 Coverage Strategy with Guaranteed Collision Avoidance in Multi-Agent Systems **77**

5.1 Problem Formulation 78

5.2 Navigation Functions 80

5.2.1 The Goal Function 80

5.2.2 The Collision Avoidance Function 81

5.3 Convergence Analysis 82

5.4 Simulation Results 86

6 Coverage Control of Multi-Agent Systems in Uncertain Environments **91**

6.1 Forest Fire Monitoring and Detection: A Practical Application 93

6.2 Problem Statement 95

6.3 Search Problem 96

6.3.1 Updating the Probability Map 97

6.3.2 Dynamic Programming Formulation 99

6.3.3 Cooperative Decision Making 101

6.4 Coverage Problem 103

6.4.1 Priority Function 104

6.4.2 Distributed Coverage Controllers 105

6.5 Simulation Results 107

6.6 Experimental Results 112

6.6.1 Experimental Tests 113

7 Conclusions and Future Work **117**

7.1 Conclusions 117

7.2 Future Work 119

Bibliography **120**

List of Figures

2.1 A schematic illustration of capabilities of multi-agent systems. 16

2.2 Schematic Voronoi diagram of 7 agents. 18

2.3 The structure of a quadrotor UAV and its frames. 21

2.4 A schematic diagram of the i-th unicycle-type robot. 24

2.5 The initial configuration of planar position of a combined UAV-UGV team. (The helicopters represent the aerial vehicles and robots represent the ground vehicles.) 31

2.6 The trajectories of the position of eight vehicles in 3D plane (the final center of each Voronoi region that corresponds to final planar position of each vehicle in XY plane is marked by black $*$). 31

2.7 The final configuration of planar position of a combined UAV-UGV team. (The helicopters represent the aerial vehicles and robots represent the ground vehicles.) 31

2.8 The mean error between the vehicle's planar positions and the centroids of their Voronoi regions. 32

2.9 The height of each quadrotor UAV in the mission. 32

2.10 The yaw angle of each quadrotor in the mission. 33

3.1 The MW-Voronoi region for a node S_1 with four neighbors $S_2, \ldots, S_5$. 38

3.2 An example of the MW-Voronoi diagram for a group of 20 nodes with different weights in a network. 39

3.3 The initial and final positions of agents in the first scenario along with their trajectories under the proposed deployment algorithm. The corresponding MW-Voronoi regions' boundaries are also depicted by dashed curves. 45
3.4 The initial and final positions of agents in the second scenario along with their trajectories under the proposed deployment algorithm. The corresponding MW-Voronoi regions' boundaries are also depicted by dashed curves. 46
3.5 The cost function $\mathcal{H}$ obtained by using the proposed deployment strategy for: (a) first scenario, and (b) second scenario. 46
3.6 The average cost function $\mathcal{H}$ obtained by using 50 different random initial configurations for the first scenario. 47
3.7 The experimental environment. 48
3.8 The Qball-X4 UAV. 48
3.9 The unmanned ground vehicles used in the experiments. 50
3.10 Trajectories of vehicles along with their corresponding MW-Voronoi regions in the first scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB. 51
3.11 Trajectories of vehicles along with their corresponding MW-Voronoi regions in the second scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively, and the gray level of the field is proportional to the value of the Gaussian priority function. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB. 52

3.12 Trajectories of vehicles along with their corresponding MW-Voronoi regions in the third scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB. 53

3.13 Trajectories of vehicles along with their corresponding MW-Voronoi regions in the fourth scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively, and the gray level of the field is proportional to the value of the Gaussian priority function. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB. 53

3.14 Velocities of all vehicles in the X and Y directions in the four scenarios. (a) First scenario; (b) Second scenario; (c) Third scenario; (d) Fourth scenario. 54

4.1 The inter-agent communication between two agents with faulty communication channel. 58

4.2 An example of the guaranteed Voronoi diagram for a group of 7 agents. 60

4.3 An example of the GMW-Voronoi diagram for a group of 9 agents with different weights. 68

4.4 The initial positions of the nine agents in the first scenario. 72

4.5 The trajectories of agents and their final positions under the proposed deployment algorithm in the first scenario. 72

4.6 The distances between the blue agent and other agents in the conventional Voronoi partitioning. 73

4.7 The distance between the blue agent and other agents in the guaranteed Voronoi partitioning. 74

4.8 The coverage performance $\mathcal{H}$ obtained by using the proposed deployment strategy (solid curve), and the conventional Voronoi partitioning in the first scenario. 75
4.9 The initial and final positions of agents in the second scenario along with their trajectories under the proposed deployment Algorithm 2. . 75
4.10 The coverage performance $\mathcal{H}$ obtained by using the proposed deployment strategy (solid curve), and the conventional MW-Voronoi partitioning in the second scenario. 76
5.1 The collision region of the i-th agent in the presence of obstacles, and its Voronoi region. 79
5.2 An example of the collision function $\beta_{il}(\|p_{il}\|)$, with $R_{col} = 8$. 82
5.3 The initial positions of the five agents along with their corresponding Voronoi regions in the first scenario. 87
5.4 The trajectories and final positions of the agents along with their Voronoi regions in the first scenario. 87
5.5 The coverage function $\mathcal{H}$ for the first scenario. 88
5.6 Distances between every pair of agents in the first scenario. 88
5.7 The initial positions of the nine agents along with their corresponding Voronoi regions in the second scenario. 89
5.8 The trajectories and final positions of the agents along with their corresponding Voronoi regions in the second scenario. 90
5.9 The coverage function $\mathcal{H}$ in the second scenario using the proposed control strategy and the VMW-Voronoi-based method. 90
6.1 Future position of a vehicle for three steps look-ahead. 102

6.2 The problem environment; the grey rectangles are the uncertainty regions of different objects and * denotes the actual position of objects. Search UAVs and service UAVs are shown by ▷ and ∘ markers respectively. 108
6.3 (a) the initial probability map, (b) the final probability map, (c) the initial priority function, and (d) the final priority function. 109
6.4 Left: The configuration and the trajectories of all UAVs and the exact priority function. The color intensity is proportional to the value of priority function. Search UAVs and service UAVs are shown by ▷ and ⊙ respectively. Right: The configuration of service UAVs and the corresponding priority function based on the probability maps. (a) and (b) t=100 sec, (c) and (d) t=160 sec, (e) and (f) t=200 sec, (g) and (h) t= 240 sec 111
6.5 The average number of detected targets and the value of coverage function for 25 simulations. 112
6.5 Experimental results: The configuration and the trajectories of all service vehicles and the corresponding priority function based on the probability maps. The color intensity is proportional to the value of priority function. The UGVs are shown by ⊙ marker and their trajectories are shown by solid lines. (a) t=30 sec, (b) t=50 sec, (c) t=65 sec. (d) t= 80 sec 116
6.6 The value of coverage function. 116

List of Tables

1 Simulation parameters 30

2 The coverage function $\mathcal{H}$ at different times for the scenario with different communication ranges 110

Chapter 1

Introduction

Cooperative control of multi-agent systems is concerned with a group of dynamic agents that are working collectively to meet a common objective. Typical common objectives include consensus, persistent surveillance, monitoring and serving. These types of systems are used in a wide range of applications including air traffic control, automated highway systems, search and rescue missions, satellite networks, forest fire monitoring, to name only a few. In these applications, the objective can be achieved more efficiently and reliably using a team of cooperative agents rather than a single agent. Technological advances and development of relatively inexpensive communication, navigation, and computational systems have enabled greater autonomy in multi-agent systems. Therefore, there has been a shift towards cooperative systems over the past few decades in order to achieve the control objectives more efficiently.

The control of a group of interconnected agents (which are, in fact, subsystems of the whole network) is sometimes performed in a centralized manner, where a central controller communicates with each agent and coordinates their actions. However, a centralized control structure has important shortcomings in practice. First of all, it may not be a reliable configuration as it has a single point of failure. In other words, the failure of the control decision maker can lead to the failure of the entire network.

Secondly, a centralized control system can become computationally inefficient as the number of agents increases. The required computation and communication resources rapidly grow with the number of group members. Due to the aforementioned problems, there has been significant interest in the development of distributed controllers for these types of systems over the past several years [1–3].

In a distributed cooperative control systems, the control decisions are made locally by either individual agents or subgroups of agents, requiring less information flow between the control units and agents. Therefore, a distributed control structure yields increased autonomy while reducing the computational burden. Unlike a centralized control scheme which uses the state of the overall systems to determine the control inputs for all agents, in a distributed control regime the agents use the information of certain agents only (often called neighboring agents) to autonomously compute their own control inputs in order to achieve the overall objective of the network. The information flow between a local controller and the neighboring agents is usually carried out through inter-agent communication and onboard sensors. The performance of a distributed cooperative control law is highly dependent upon the communication structure of the multi-agent system. The communication structure determines the information available to each agent in the network, which may be constrained by sensing, communication, and computational limitations. In many applications, agents only communicate with those neighbors that are located within a certain distance from them.

As mentioned, cooperative control of multiple agent systems covers a broad range of applications. In this research, novel cooperative control techniques are proposed for coverage control of multi-agent systems. The problem of covering an environment (or serving different points of the environment) is thoroughly investigated in this work from different aspects.

1.1 Literature Review

A considerable amount of research in cooperative control of multi-agent systems is focused on the coverage problem. In particular, the deployment of a group of agents (mobile sensors, unmanned aerial/ground/surface/underwater vehicles, etc.) over an environment to carry out distributed surveillance and sensing task is called *coverage*. Some important applications in this area that are investigated in the literature includes forest fire surveillance and detection [4–7], gas pipelines monitoring [8], and environmental monitoring [9].

In the coverage problem, it is important to place the nodes in the network such that the covered area is maximized. For instance, in the traditional art gallery problem in computational geometry [10] it is desired to determine, for some polygonal environment, the minimum number of cameras that can be placed such that the entire environment is observed. In the problem of sensing coverage, the autonomous agents are deployed to collectively maximize a prescribed objective function related to the quality of coverage [11–15].

The coverage control problem considered in this research work is based on the optimization formulation introduced in [11] which uses the geometrical notion of a Voronoi partition to assign the agents to different parts of the environment. This work employs some concepts from locational optimization [16], which is concerned with the optimal placement of industrial facilities, and can be formulated in the context of the classical problem of finding the geometric median points. The coverage controller in [11] drives the robots towards a centroidal Voronoi configuration [17]. The same problem is considered in [12] with a more realistic model, where the sensing range is restricted to a bounded region. In [18], a directional-search online control schemes is proposed for this type of sensors in order to achieve the optimum coverage. A Voronoi-based algorithm is developed in [19] for network coverage in a mobile sensor

network (MSN) which does not require any global location assurance condition for the sensors. In [20], a deterministic annealing technique is used to relocate the final robot configurations to improve covered area by the network. The Voronoi diagram is used in [21] to discover the coverage holes, and different sensor deployment strategies are proposed to increase coverage. In [22], the authors provide an efficient deployment strategy for a heterogeneous group of robots with different sensor footprints.

The Voronoi partition is generalized in [23], where the power diagram is introduced to achieve equitable mass partitions. The problem of coverage control in an environment with the time-varying priority function is addressed in [24]. In [25], the coverage problem for target points that appear sporadically over time in a bounded environment is investigated (see also [26, 27] for a more cohesive presentation of this problem). An entropy-based metric is used in [28] to construct a map that determines the reachable regions of the environment. While the mobile robots explore the environment, they also use a centroid geodesic Voronoi tessellation to distribute themselves such that proper coverage is maintained.

In addition, some research works have addressed coverage control in a non-convex domain. The coverage problem in non-convex polygonal environments is presented in [29]. A path-planning algorithm is used to compute the trajectory of mobile robots around obstacles and corners. Another common method for coverage of environments with non-convex boundaries by applying the geodesic distance measure to Voronoi coverage is presented in [22]. Using a proper diffeomorphism, the non-convex environment is transformed in [30] to a convex region where conventional Voronoi partition can be applied. Probabilistic scenarios that use a similar control strategy for coverage problem in non-convex environment have also been considered in [14, 31]. In [32], a novel discrete partitioning and coverage control algorithm for a non-convex environment is presented. This method requires only short-range communication between

pairs of robots.

Some other Voronoi-based coverage strategies, on the other hand, consider a more realistic environment. For example, in [33], an energy-aware coverage strategy is proposed for mobile sensor networks where the agents (sensors) have limited power to move. A self-triggered coordination algorithm is presented in [34] for a group of agents performing an optimal deployment task when individual agents do not have up-to-date information about each other's locations. A dynamic awareness model is proposed in [35] to cover events dynamically taking place over a given task domain using a multi-vehicle sensor network with intermittent communications. The work [36] provides a lower bound on the communication range of an agent in a sensor network in order to optimize the coverage of the field. In [37], a coverage algorithm for unicycle vehicles like wheeled mobile robots is investigated via hybrid modeling.

A centralized mission planning is presented in [38] for persistent surveillance using a team of small unmanned vehicles with a centralized health manager. The system controls the group such that a certain number of vehicles always cover a region of interest while it considers the refueling needs of vehicles based on a stochastic fuel consumption model. A generalized Voronoi partition is used in [39] for the problem of area-constrained coverage. The overall area of the region assigned to each agent is assumed to be fixed, and a Jacobi iterative algorithm is used to assign weights to the generalized Voronoi partition that satisfies the area constraints. The problem of target tracking using a heterogeneous mobile sensor network is studied in [40], and a novel space-partitioning algorithm is provided in [41] to tackle the same problem.

A huge body of research work in the area of formation control has considered the collision and obstacle avoidance issues [42–44]. However, few works have addressed these issues on cooperative coverage control in multi-agent systems. An algorithm is presented to dynamically cover a field with flocking and collision avoidance properties

in [45]. In [46] a Voronoi-based coverage control is proposed for a group of wheeled mobile robots with a dynamic constraint, where a collision avoidance term is also incorporated in the kinematic controller using potential functions to avoid inter-agent collision. In [47], a deterministic full coverage method is presented based on computational geometry for a region with arbitrary boundary and obstacles with the regular and irregular shapes. In [48], an obstacle-resistant robot deployment algorithm is presented, where a robot deploys a near-minimal number of sensor nodes to achieve full sensing coverage. This problem is investigated in the environment containing unpredicted obstacles with regular or irregular shapes.

In [14] the authors consider a probabilistic network model and a density function to represent the frequency of random events taking place over the mission space. Using this information, an optimization problem is formulated which aims at maximizing coverage using sensors with limited ranges, while minimizing communication cost. A potential-field-based approach is presented in [49] for sensor deployment in an unknown environment. In [50], a receding-horizon-based path-planning algorithm is presented for time-sensitive cooperative surveillance using UAVs equipped with cameras. An algorithm is presented in [51] to add a relatively small number of mobile sensors to a set of static sensors in order to improve network coverage. The algorithm employs a strategy which aims to optimize the contribution of the mobile sensors to the overall coverage.

In dynamic coverage control, on the other hand, it is desired to develop a motion control strategy for coordinated multi-vehicle systems in order to dynamically cover a given arbitrarily-shaped domain. The objective is to survey the entire search domain such that the information collected at each point reaches a prescribed level. In [45], a feedback control law is presented that guarantees each point in the search domain

is surveyed by search vehicles until a certain preset threshold is achieved, while collision avoidance is guaranteed. The method is centralized, however, is not necessarily optimal. As an alternative approach for searching in an uncertain environment, simultaneous localization and mapping (SLAM) strategy is presented in [52]. In [53], a sweep coverage model is proposed to periodically monitor a set of points-of-interest in surveillance applications.

The problem of multi-vehicle search in an uncertain environment has been studied in recent years, and a number of approaches have been formulated. The problem is converted to a multi-vehicle path planning problem, and an optimization problem is subsequently developed where an optimal path is obtained by maximizing a global objective function subject to a set of constraints [54–59]. Dynamic programming (DP) and approximate dynamic programming (ADP) algorithms are the most widely used techniques for solving this type of problem. A combined deploy and search strategy using a centralized Voronoi partitioning is proposed in [60], where the mobile agents are autonomously deployed to maximize the reduction of uncertainty about the environment at each step. In [61], a set of mobile sensors collaborate with a group of stationary sensors in order to detect an event. A path planning algorithm based on receding horizon optimization is presented to move the mobile sensors towards the areas that are least covered by the stationary sensors.

A common assumption in all of the coverage control results cited in the previous paragraphs is that the distribution of sensory information in the environment is known *a priori* by all agents. This assumption was first relaxed in [62], where the agents approximate the sensory function (a function indicating the relative importance of different areas in the environment) from sensor measurements. Then, the problem of online learning of the priority function was addressed in [15]. In [63], the priority function estimation was proposed using a neural network approach. Although in the

above papers the priority function is unknown, it is assumed to be measurable by each agent locally. In the present dissertation also the priority function is unknown but is not locally measurable. Motivated by a number of real-world applications, the priority function is assumed to be a function of position of some unknown targets which can be detected by appropriate sensors. In order to accomplish the coverage task more efficiently, some search vehicles are assigned to find the targets. This leads to more effective coverage of uncertain environments. Furthermore, not much experimental work is available in the literature on this type of multi-vehicle systems. In one of this work [64], a mechanism is proposed for surveillance of an area using a group of three wheeled mobile robots subject to navigation failures. In [65], the coverage problem is investigated for an MSN with anisotropic sensor model which depends on the distance and orientation of a target point. The algorithm is then implemented on a mobile robot testbed.

1.2 Dissertation Outline

The dissertation is organized as follows. In Chapter 2, first the coverage control problem is introduced and then the optimal solution for it is provided. The dynamic model of a quadrotor UAV as well as a unicycle type UGV is presented, and a planar position controller is subsequently proposed for this group of vehicles.

In Chapter 3, the above problem is investigated for the case when serving (sensing) capabilities of different agents are not the same. The necessary background material on the multiplicatively-weighted Voronoi (MW-Voronoi) partition for weighted nodes is presented and some results concerning locational optimization are also provided. A distributed control strategy based on the MW-Voronoi diagram is also proposed in this chapter, and the theoretical developments are validated through experiments.

Two novel partitioning techniques are presented in Chapter 4 for coverage control of multi-agent systems in the presence of inter-agent communication delays and variable sensor effectiveness (health), respectively.

In Chapter 5, an effective coverage technique is developed for multi-agent systems using navigation function with guaranteed collision avoidance and obstacle avoidance. A distributed navigation function is introduced and its convergence properties are studied.

In Chapter 6, a new formulation is proposed which is suitable for the search and coverage problems in uncertain environments. The probability map updating rules, dynamic programming formulation and derivation of priority function based on the probability map are also presented in this chapter.

Finally, Chapter 7 summarizes the contributions of the thesis and provides suggestions for future research directions in this area.

1.3 Research Contributions

Contributions of this thesis lie in investigating different aspects of the coverage control problem for a network of cooperative autonomous agents. It is aimed to address some of the practical challenges and realistic assumptions concerning this type of network.

Most of the prior work in the Voronoi-based coverage control consider single-integrator dynamics for the agents. However, in practice a wide range of mobile agents such as unmanned vehicles have more complex dynamics, which can invalidate the performance of the algorithms developed for trivial dynamics. The present work is an attempt to contribute to this aspect by proposing a control law for the deployment of agents with different dynamics (quadrotor, double-integrator and unicycle-type dynamics). To this end, a planar position controller is proposed for a group of heterogeneous mobile agents.

The problem of providing a prescribed service (such as coverage) using a group of mobile agents is also investigated, where it is desired to minimize the overall service cost. The serving capabilities of different agents are assumed to be nonidentical, while in the literature they are often supposed to be the same. Note that in the case of agents with nonidentical operating costs, the conventional Voronoi diagram is no longer as effective in developing agent-deployment algorithms. Instead, the field is partitioned using the MW-Voronoi diagram introduced in the literature. A distributed coverage control law is then provided which guarantees the convergence of agents to the optimal configuration w.r.t. the above-mentioned cost function. To further generalize the results, a prescribed priority function is used to prioritize the importance of providing service to different points in the field.

Typically, each agent in a multi-agent system transmits its position information to a subset of agents (often referred to as neighbors) and receives similar information from them. Most of the existing results on the coverage control of sensor networks assume that this information exchange has no delay. However, it is known that neglecting the effect of delay in the analysis and design of multi-agent systems can lead to poor performance as well as unsafe behaviors such as inter-agent collision. While the effect of communication delay in the flocking and formation control of multi-agent systems has been investigated in the past few years, not much work has been done in cooperative coverage control of a network of agents subject to communication delay. The problem is also extended to the case where the sensors have varying effectiveness (health). The sensor effectiveness factor of agents is incorporated in the locational optimization problem. Hence, two different spatial techniques are introduced to partition the field into smaller regions (one for each agent) and study the coverage control problem for multi-agent systems, where the agents are subject to communication delays and health degradation. A motion coordination algorithm

is subsequently proposed for the agents based on the latest information received from their neighbors such that the coverage performance function is minimized over the regions assigned to agents.

There is a substantial body of research on collision and obstacle avoidance in multi-agent systems. However, not much work has been reported on cooperative coverage control of multi-agent systems. To address this problem, a distributed navigation function is used for both collision and obstacle avoidance in multi-agent coverage problem. The field is partitioned to the Voronoi cells first, and a control strategy is subsequently proposed to relocate the agents. This is performed in such a way that a prescribed coverage cost function is minimized while collision and obstacle avoidance are guaranteed. The convergence analysis is provided to show the stability of the network under the proposed control strategy.

Another common assumption in most of the existing results on the Voronoi-based coverage control is that the distribution of sensory information about the environment is required to be known *a priori* by all agents. However, this assumption is not realistic in many applications. Therefore, the coverage problem in uncertain environments is investigated using some search agents to explore the environment. The uncertainty in the environment is captured by an unknown priority function. Motivated by real applications, a new priority function is introduced which is a function of the position of some unknown targets in the environment. To cover this uncertain environment more efficiently and to improve the coverage performance, the task of finding the targets can be carried out by some search agents. The search agents possess the probability maps of targets in the environment, and update these probability maps based on the sensor measurements during the search mission. A Voronoi-based coverage control strategy is then proposed to modify the configuration of coverage agents such that a prescribed coverage cost function is minimized using the updated probability maps

which are provided by the search agents.

Finally, not many experimental results are available in the literature on this type of multi-vehicle systems. In order to show the effectiveness of the Voronoi-based coverage control of multi-agent systems as a practical framework for real-time applications (like forest fire detection and monitoring), some of the proposed algorithms are verified experimentally on a real testbed. To this end, a group of unmanned systems consisting of quadrotor helicopter and different ground vehicles are employed to perform some experiments in the Networked Autonomous Vehicles Lab (NAVL) of the Department of Mechanical and Industrial Engineering, Concordia University.

A list of publications resulted from this research is provided below.

Journal Papers

1. F. Sharifi, Y. M. Zhang, and A. G. Aghdam, "A distributed deployment strategy for multi-agent systems subject to health degradation and communication delays," *Journal of Intelligent and Robotic Systems*, Volume 73, Issue 1-4, pp. 623–633, 2014.

2. F. Sharifi, A. Chamseddine, H. Mahboubi, Y. M. Zhang, and A. G. Aghdam, "A distributed deployment strategy for a network of cooperative autonomous vehicles," *IEEE Transactions on Control Systems Technology* (Accepted for publication on 2014-06-19).

3. F. Sharifi, M. Mirzaei, Y. M. Zhang, and B. W. Gordon, "Cooperative multi-vehicle search and coverage problem in uncertain environments," *Unmanned Systems*, 2014 (Minor revision, revised version under 2nd review).

Conference Papers

1. F. Sharifi, Y. M. Zhang, and A. G. Aghdam, "Forest fire detection and monitoring using a network of autonomous vehicles," Accepted by the *International*

Conference on Intelligent Unmanned Systems (ICIUS), Montreal, Canada, 2014.

2. F. Sharifi, H. Mahboubi, Y. M. Zhang, and A. G. Aghdam, "A coverage strategy with guaranteed collision avoidance in multi-agent systems using navigation functions," in *Proceeding of the AIAA Guidance, Navigation, and Control (AIAA-GNC) Conference*, Boston, USA, 2013.

3. F. Sharifi, Y. M. Zhang, and A. G. Aghdam, "Coverage control in multi-vehicle systems subject to health degradation," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1119–1124, Atlanta, USA, 2013.

4. F. Sharifi, Y. M. Zhang, H. Mahboubi, and A. G. Aghdam, "Coverage control in multi-agent systems subject to communication delays," in *Proceedings of 8th IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA2012)*, pp. 269–274, Suzhou, China, 2012.

5. F. Sharifi, Y. M. Zhang, and B. W. Gordon, "Voronoi-based coverage control for multi-quadrotor UAVs", in *Proceedings of the ASME/IEEE International Conference on Mechatronics and Embedded Systems and Applications (MESA2011)*, pp. 991–996, Washington D.C., USA, 2011.

6. F. Sharifi, M. Mirzaei, B. W. Gordon, and Y. M. Zhang, "Fault tolerant control of a quadrotor UAV using sliding mode control," in *Proceedings of the Conference on Control and Fault-Tolerant Systems (SysTol'10)*, pp. 239–244, Nice, France, 2010.

7. F. Sharifi, B. W. Gordon, and Y. M. Zhang, "Decentralized sliding control of

cooperative multi agent systems subject to communication delay," in *Proceeding of the AIAA Guidance, Navigation, and Control (AIAA-GNC) Conference*, Toronto, Canada, 2010.

Some of the collaborative research results related to the present topic are also listed below.

1. H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. M. Zhang, "Distributed coordination of a network of agents with limited communication range in the presence of obstacles," *IEEE Transactions on Automatic Control* (Conditionally accepted on 2014-08-01).

2. H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. M. Zhang, "Distributed coordination of a network of nonidentical agents with limited communication capabilities in the presence of fixed obstacles," in *Proceedings of the American Control Conference (ACC) 2013*, pp. 5398–5403, Washington DC, USA, 2013.

3. H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. M. Zhang, "Distributed coordination of multi-agent systems for coverage problem in presence of obstacles," in *Proceedings of the American Control Conference (ACC) 2012*, pp. 5252–5257, Montreal, Canada, 2012.

4. M. Mirzaei, F. Sharifi, B. W. Gordon, C. A. Rabbath, and Y. M. Zhang, "Cooperative multi-vehicle search and coverage problem in uncertain environments," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 4140–4145, Orlando, USA, 2011.

5. M. Mirzaei, B. W. Gordon, C. A. Rabbath, and F. Sharifi, "Cooperative multi-UAV search problem with communication delay," in *Proceeding of the AIAA Guidance, Navigation, and Control (AIAA-GNC) Conference*, Toronto, Canada, 2010.

Chapter 2

Coverage Control of Multi-Agent Systems

2.1 Introduction

In general, an agent refers to a dynamical system. However, in the context of this work the term "agent" is interchangeable with "vehicle" or "robot". The multi-agent system is defined as a cooperating group of autonomous mobile agents capable of motion, communication, sensing and computation. In general, these capabilities can have some constraints: The motion is subject to dynamics of the platform, nonholonomic constraints and limits on velocities. Important constraints for the communication are maximum range, available bandwidth and delay. Sensing is typically limited by range or field of view of the sensors, and sensor accuracy. This is schematically illustrated in Fig. 2.1.

In the cooperative control one deals with a team of agents instead of one agent; hence, in addition to the teaming objective the *interaction* between agents must be taken into account. As some examples of such interactions one can point to physical interactions such as collision avoidance, formation keeping and communication

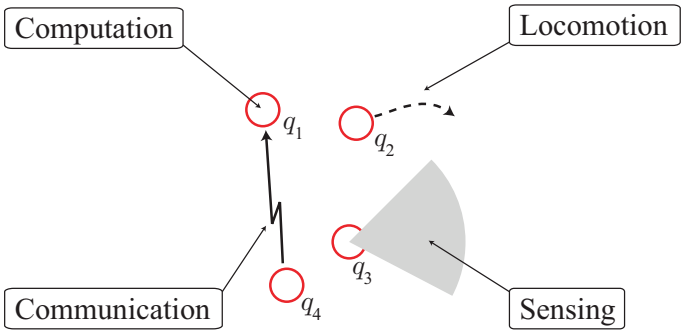


Figure 2.1: A schematic illustration of capabilities of multi-agent systems.

interaction. It is also assumed that there are no sensor errors, actuator errors, model uncertainty, obstacles in the environment or communication noise. These assumptions allow one to focus on the core issue of the problem "coverage control". However, the proposed approaches in this chapter will be extended to the cases above by suitably modifying the proposed algorithms to account for these non-ideal effects.

Cooperative control of multi-agent systems has received considerable attention over the last decade because of technological advances and development of relatively inexpensive communication, computation, and sensing devices. Then, a plenty of research work has been done on the coverage problem as one of the main applications of cooperative control. In [11], One of the remarkable references in this area, a decentralized control law is designed for mobile sensor networks to cover an area partitioned into Voronoi region, in the sense that continually driving the agents towards the centroids of their Voronoi cells. In this chapter, a motion coordination strategy for the deployment of multiple quadrotor UAVs and wheeled robot UGVs via Voronoi-based locational optimization framework introduced in [11] is proposed. Most of the prior works in the area of Voronoi-based coverage control assume single-integrator dynamics for the agents. However, in reality many current vehicles such as unmanned systems have non-trivial dynamics, which can invalidate the performance of the proposed algorithms. Moreover, aerial vehicles offer the advantage of wide

area coverage and relative insensitivity to terrain considerations. Hence, to overcome this problem with existing works, a planar position controller is proposed for heterogeneous vehicles with nonholonomic dynamics. In this chapter, first the locational optimization problem is reviewed and the dynamic model of vehicles are then presented. Furthermore, using LaSalle's invariance principle, it is proved that under proposed controller the network of vehicles converges to the optimal configuration. Finally numerical simulation is provided to show the effectiveness of the proposed method.

2.2 Problem Statement

Let Q be a convex field in R^2, and denote an arbitrary point in Q by q. Consider a group of n mobile agents randomly distributed in Q with the position of the i-th agent represented by p_i. Accordingly, the positions of all agents are denoted by the set $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$. Let also $\varphi : Q \to R_+$ be a priority function which assigns a weight to each point in the field, and represents an *a priori* measure of information on Q. The function φ can reflect a measure of relative importance of different points in the field or reflect a knowledge of the probability of occurrence of events in different regions, which means the agents should monitor the points with higher value of φ more closely.

Let $V = \{V_1, V_2, \ldots, V_n\}$ be the Voronoi partition of Q, where the agent positions are the generating points. By definition, the Voronoi region V_i of agent i is the locus of all points that are closer to that agent than to any other agents in the field. This can be expressed mathematically as:

$$V_i = \{q \in Q | \, \|q - p_i\| \leq \|q - p_j\| \, , \forall j \neq i\}, \; i \in \mathbf{n} := \{1, 2, \ldots, n\} \tag{2.1}$$

Agents i and j are called neighbors if $V_i \cap V_j \neq 0$ (or more precisely if they share an edge). The set of all neighbors of agent i is denoted by $\mathcal{N}_i$. One can see easily that at a fixed-agents location, the optimal partition of environment is the Voronoi partition in which the agent positions are the generator points. The schematic Voronoi diagram of 7 agents which are spread randomly over a region is depicted in Fig. 2.2.

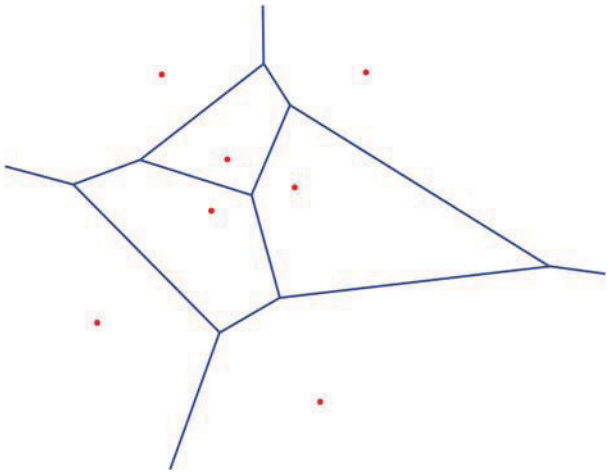


Figure 2.2: Schematic Voronoi diagram of 7 agents.

The cost of covering (or sensing) a point q by the i-th agent is denoted by $f_i(q)$, where $f_i : R \to R_+$ is a known strictly increasing convex function. The cost of covering incurred by an agent is closely related to the distance of the point to be covered from the agents; the longer the distance the higher the cost. It is assumed that each agent is in charge of covering all points in its own Voronoi region. Since the whole field is partitioned by all Voronoi regions, the coverage performance function as a measure of the system performance is defined as:

$$\mathcal{H}(\mathbf{P}) = \sum_{i=1}^{n} \int_{V_i} f_i(q)\varphi(q)dq \tag{2.2}$$

Note that (2.2) represents the coverage function which measures the ability of the coverage provided by the mobile sensing network in Q. Roughly speaking, the function $\mathcal{H}$ quantifies how well the agents are located inside the field Q. More precisely, a smaller

$\mathcal{H}$ means the agents are deployed more effectively and a higher value corresponds to a poor coverage performance. Then, it is desired to minimize it. Throughout this chapter, it is assumed that the coverage cost function of each agent has the following form:

$$f_i(q) = \frac{1}{2} \|p_i - q\|^2, \ \forall i \in \mathbf{n} \tag{2.3}$$

2.3 Centroidal Configuration

In general for every region V_i two following terms can be defined.

Definition 2.1. *The mass and center of mass of a region V_i with respect to the priority function φ are respectively defined as follows:*

$$M_{V_i} = \int_{V_i} \varphi(q) dq \tag{2.4}$$

$$C_{V_i} = \frac{1}{M_{V_i}} \int q\varphi(q) dq \tag{2.5}$$

If there is only one agent in the field, then minimizing $\mathcal{H}_1 = \int_Q f_i(q)\varphi(q)dq$ is often referred to as the *1-center* problem [66]. In the special case, if the cost of coverage of the *1-center* problem is chosen as (2.3), then the optimal position of the agent is the *centroid* (also called the center of mass) of the field [66], i.e.:

$$C_Q = \frac{\int_Q q\varphi(q)dq}{\int_Q \varphi(q)dq} \tag{2.6}$$

We will show that this result is valid in the case of n agent in the field. Assume function $\mathcal{H}_i$ defined as follows:

$$\mathcal{H}_i = \int_{V_i} \frac{1}{2} \|p_i - q\|^2 \varphi(q)dq \tag{2.7}$$

Let the above function $\mathcal{H}_i$ be differentiable everywhere on its domain so that its partial derivative with respect to the agent's position is well-defined. Easily one can show that

$$\frac{\partial \mathcal{H}_i}{\partial p_i} = \int_{V_i} (p_i - q)\varphi(q)dq = M_{V_i}(p_i - C_{V_i}) \tag{2.8}$$

So the partial derivative of $\mathcal{H}_i$ with respect to the position of the i-th agent only depends on its own position and the position of its Voronoi neighbors. Therefore the computation of the derivative of $\mathcal{H}_i$ with respect to the agents' location is distributed in the sense of Voronoi. It is clear that each partial derivative must be zero for a local minimum.

Clearly, the extreme points of $\mathcal{H}_i$ are those in which every agent is at the centroid of its Voronoi region, $p_i = C_{V_i},\ \forall i \in \mathbf{n}$. The resulting partition of the environment is commonly called a Centroidal Voronoi configuration. Readers are referred to [26] for more discussions on this configuration.

2.4 Dynamic Model of Vehicles

UAVs have desirable features such as high speed of coverage and a wide-area view of the region, while certain tasks such as precise localization may be better suited to UGVs. One way to combine the best features of both ground- and aerial-based vehicles is to examine a heterogeneous team. In this chapter, a group of quadrotor UAVs and wheeled robot UGVs is considered.

2.4.1 Model of a Quadrotor UAV

One type of UAV with a strong potential for both indoor and outdoor flight is the rotorcraft and the so-called quadrotor helicopter UAV has been chosen by many researchers as the most promising vehicle [67–69]. A quadrotor UAV is an aircraft

that is lifted and propelled by four rotors. Control of quadrotor UAV can be achieved by varying the relative speed of each rotor to change the thrust and torque produced by each propeller. The ability of quadrotor helicopters to take off and land vertically, to perform hover flight, as well as their agility, makes them ideal vehicles, specifically in the search and coverage problems.

In this section, the dynamic model of a quadrotor UAV has been studied briefly. A body-fixed frame B and the earth-fixed frame E are assumed to be at the center of gravity of the quadrotor UAV, where the z-axis is pointing upwards, as seen in Fig. 2.3. The position of the quadrotor UAV in earth frame is given by a vector (x, y, z). The orientation of quadrotor UAV that referred to as roll, pitch, and yaw is given by a vector (ϕ,θ,ψ) which is measured with respect to the earth coordinate frame E.

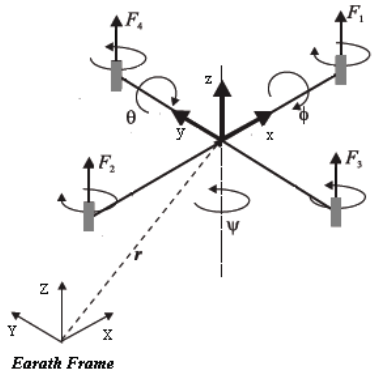


Figure 2.3: The structure of a quadrotor UAV and its frames.

The transformation of vectors from the body-fixed frame to the earth-fixed frame can be obtained based on Euler angles and the rotation matrix R_{EB}.

$$R_{EB} = \begin{bmatrix} C_\psi C_\theta & -S_\psi C_\phi + C_\psi S_\theta S_\phi & S_\psi S_\phi + C_\psi S_\theta C_\phi \\ S_\psi C_\theta & C_\psi C_\phi + S_\psi S_\theta S_\phi & -C_\psi S_\phi + S_\psi S_\theta C_\phi \\ -S_\theta & C_\theta S_\phi & C_\phi C_\theta \end{bmatrix} \tag{2.9}$$

where the abbreviations $S_{(.)}$ and $C_{(.)}$ have been used for representing sin(.) and cos(.), respectively. It is important to note that $R_{EB} = R_{BE}^T$. The trust force generated by rotor i ($\forall i \in \{1, 2, 3, 4\}$) is $F_i = b\omega_i^2$ where b is the thrust factor and ω_i is the speed of rotor i. In the body-fixed frame, the forces are defined as follows:

$$F_B = \begin{bmatrix} F_{xB} \\ F_{yB} \\ F_{zB} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} F_i \end{bmatrix}$$

In the earth-fixed frame, the forces can be described as:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = R_{EB}.F_B = (\sum_{i=1}^{4} F_i) \begin{bmatrix} S_\psi S_\phi + C_\psi S_\theta C_\phi \\ -C_\psi S_\phi + S_\psi S_\theta C_\phi \\ C_\phi C_\theta \end{bmatrix} \tag{2.10}$$

Therefore, the equations of motion (x, y, z) in the earth- fixed frame are represented as:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} F_x - K_1\dot{x} \\ F_y - K_2\dot{y} \\ F_z - mg - K_3\dot{z} \end{bmatrix} = \begin{bmatrix} (\sum_{i=1}^{4} F_i(S_\psi S_\phi + C_\psi S_\theta C_\phi)) - K_1\dot{x} \\ (\sum_{i=1}^{4} F_i(-C_\psi S_\phi + S_\psi S_\theta C_\phi)) - K_2\dot{y} \\ (\sum_{i=1}^{4} F_i(C_\phi C_\theta)) - mg - K_3\dot{z} \end{bmatrix} \tag{2.11}$$

where K_i is the drag coefficient. Note that these coefficients are negligible at low speed. Also, assuming total thrust approximately counteracts gravity, $\sum_{i=1}^{4} F_i \approx \tilde{F} = mg$, except in the Z axis. Finally, by applying the small angle approximation to

the rotation matrices, one can obtain:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & \tilde{F} & 0 \\ -\tilde{F} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \sum_{i=1}^{4} F_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{2.12}$$

By neglecting the gyroscopic effect of propeller rotation, body rotation, drag effects, and assuming the equal amount for all inertia I_x, I_y and I_z, the following linear model for angular equations can be obtained:

$$\begin{aligned} \ddot{\phi} &= \frac{l}{I_x} U_1 \\ \ddot{\theta} &= \frac{l}{I_y} U_2 \\ \ddot{\psi} &= \frac{1}{I_z} U_3 \end{aligned} \tag{2.13}$$

where l and U_i are the lever factor and system's inputs respectively.

It becomes obvious that the quadrotor UAV model can be decomposed into one subset of differential equations that describes the dynamics of the attitude (i.e. the angles) and one subset that describes the translation of the UAV.

2.4.2 Model of UGV

Each UGV is modeled as a nonholonomic two-wheeled mobile robot. Each UGV moves on a horizontal plane and the kinematic equation of the vehicle is given by:

$$\begin{aligned} \dot{x_i} &= v_i \cos\theta_i \\ \dot{y_i} &= v_i \sin\theta_i \\ \dot{\theta_i} &= w_i \end{aligned} \tag{2.14}$$

where (x_i, y_i) is the coordinate of a point p_i located at the mid-axis of the rear wheels of the i-th robot, θ_i is the orientation of the i-th robot with respect to the X-axis of the coordinate frame, v_i and w_i are the linear and angular velocities of the center of mass of the i-th robot, respectively (see Fig. 2.4). The dynamic equation of the mobile robot can be described as follows:

$$\begin{aligned} m\dot{v}_i &= F \\ j\dot{w}_i &= T \end{aligned} \tag{2.15}$$

where F and T represent the force and torque exerted on the robot respectively. And m and j are the mass and moment of inertia of the robot respectively.

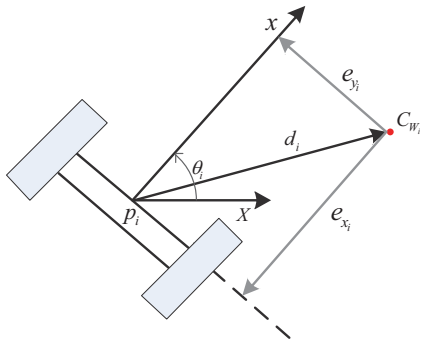


Figure 2.4: A schematic diagram of the i-th unicycle-type robot.

2.5 Coverage with Multiple UGVs and UAVs

In this section, the problem of deploying a group of heterogeneous vehicles to cover an environment is investigated. Then it is desired to design a set of planar position controllers for vehicles with different dynamics (the decoupled model of Quadrotor UAV and unicycle-type dynamics for UGVs). For the purpose of coordinating multiple vehicles to cover the points in the environment, it is desired to design a position-control strategy based on the centroidal Voronoi configuration. To achieve this goal,

the following assumptions are needed.

Assumption 2.1. *Every vehicle has complete knowledge of its own dynamics.*

Assumption 2.2. *The vehicles have the ability to compute their own Voronoi regions in a distributed manner.*

Assumption 2.3. *Each vehicle can communicate with other vehicles in its neighboring Voronoi regions.*

We consider n vehicles each of which belonging to a group of n_a quadrotor UAVs or a group of n_g wheeled robot UGVs ($n = n_a + n_g$).

2.5.1 Control of UAVs

As shown in Subsection 2.4.1, the linear model of quadrotor UAV is decoupled. Then the height and yaw angle independently can be easily controlled by any conventional control methods. However for the purpose of coordinating of multiple vehicles to cover a planar environment, the position controller based on the Cenroidal Voronoi configuration is designed. Consider that the planar position of the i-th vehicle in the earth frame is denoted by $p_i = (x_i, y_i)$ and $C_{V_i} = (C_{V_{i,x}}, C_{V_{i,y}})$ is the center of Voronoi that corresponds to the i-th vehicle. The following position control law is proposed for the i-th UAV:

$$\begin{aligned} u_x^i &= \frac{1}{g}\left(k_1^i M_{V_i}(C_{V_{i,x}} - x_i) - k_2^i \dot{x}_i\right) \\ u_y^i &= -\frac{1}{g}\left(k_1^i M_{V_i}(C_{V_{i,y}} - y_i) - k_2^i \dot{y}_i\right) \end{aligned} \tag{2.16}$$

where k_1^i and k_2^i are the positive gains and M_{V_i} mass of the i-th Voronoi region which is defined in (2.4).

Theorem 2.1. *Consider a group of n_a quadrotor UAVs whose dynamic models are described by (2.12) and (2.13). Let the assumptions (2.1) through (2.3) hold. Under control law (2.16), it is guaranteed that the whole system is asymptotically stable and the planar positions of quadrotor UAVs converge to a centroidal Voronoi configuration.*

Proof. Consider a Lyapunov function candidate as:

$$\vartheta = \sum_{i=1}^{n_a} k_1^i \mathcal{H}_i + \sum_{i=1}^{n_a} \frac{1}{2} \dot{p}_i \dot{p}_i^T \tag{2.17}$$

where $\mathcal{H}_i$ is defined as (2.7), $\dot{p}_i = (\dot{x}_i, \dot{y}_i)$ is the velocity of the i-th vehicle and k_1^i is the controller gain. Since k_1^i is positive number and also $\|.\|^2$ and $\varphi(q)$ are positive functions, thus the candidate Lyapunov function ϑ is lower-bounded by zero. By taking the time derivative of ϑ along the trajectories of systems one arrives at:

$$\dot{\vartheta} = \sum_{i=1}^{n_a} k_1^i \dot{\mathcal{H}}_i + \sum_{i=1}^{n_a} \ddot{p}_i \, \dot{p}_i^T \tag{2.18}$$

Using (2.8) and the fact $\frac{\partial \mathcal{H}_i}{\partial z} = \frac{\partial \mathcal{H}_i}{\partial \phi} = \frac{\partial \mathcal{H}_i}{\partial \theta} = \frac{\partial \mathcal{H}_i}{\partial \psi} = 0$, the (2.18) can be rewritten as follows:

$$\dot{\vartheta} = \sum_{i=1}^{n_a} \left[\left(k_1^i M_{V_i} \left(p_i - C_{V_i} \right) + \ddot{p}_i \right) \dot{p}_i^T \right] \tag{2.19}$$

Because of the decoupling characteristics of linear model of the quadrotor UAV, the pitch and roll angles of the quadrotor are the control input for planar position equation. The dynamic equation of planar position (2.12) can be easily rewritten as follows:

$$\ddot{p}_i = g \begin{bmatrix} u_x^i \\ -u_y^i \end{bmatrix}^T \tag{2.20}$$

By substituting (2.20) into (2.19), and using control input (2.16), the time derivative

of Lyapunov function can be obtained as follows:

$$\dot{\vartheta} = \sum_{i=1}^{n_a} \left(-k_2^i \dot{p}_i \dot{p}_i^T \right)$$

which is clearly non-positive. Let $\mathcal{S}$ be the set of all points in Q where $\dot{\vartheta} = 0$. Due to the convexity of the region Q, one can conclude that each of the Voronoi centroids C_{V_i} lies in the interior of the i-th Voronoi region and so in the interior of the region Q. Then the vehicles move towards the interior of the region Q and never leave it. This in turn means that $\mathcal{S}$ is a positive invariant set for the trajectories of the closed-loop system. Since this set is closed and bounded, one can use LaSalle's invariance principle to infer that quadrotors planar positions converge to the largest invariant subset of the set $\mathcal{S}$. For any trajectory belonging to the set $\mathcal{S}$, it results from the planar model (2.20) and the control law (2.16) that:

$$\dot{p}_i \equiv 0 \Rightarrow \ddot{p}_i \equiv 0 \Rightarrow u_x^i = u_y^i \equiv 0 \Rightarrow p_i = C_{V_i}, \ \forall i \in \{1, \ldots, n_a\}$$

One can then conclude that for any $i \in \{1, \ldots, n_a\}$, $p_i = C_{V_i}$ is the largest invariant set corresponding to the centroidal Voronoi configuration. Therefore, under the control law (2.16), the closed-loop system is asymptotically stable and the quadrotors planar positions converge to the centroidal Voronoi configuration. □

2.5.2 Control of UGVs

In this subsection, a position control law for UGVs with unicycle-type dynamics is designed based on the centroidal Voronoi configuration. The kinematic equation of UGVs mentioned earlier in (2.14) in which the position of the i-th vehicle in the earth frame is denoted by $p_i = (x_i, y_i)$ and θ_i is the orientation of the i-th vehicle with respect to the X-axis of the earth frame. Similar to the UAVs case, the following

control law is proposed for the i-th UGV with unicycle dynamics:

$$\begin{aligned} v_i &= -k_v^i \cdot e_{x_i} \\ w_i &= k_w^i \cdot \arctan(\frac{e_{y_i}}{e_{x_i}}) \end{aligned} \tag{2.21}$$

where k_v^i, k_w^i are positive gains, and e_{x_i}, e_{y_i} are error variables in the vehicle coordinate system defined as follows:

$$\begin{bmatrix} e_{x_i} \\ e_{y_i} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} (p_i - C_{V_i})^T \tag{2.22}$$

Theorem 2.2. *Consider a group of n_g wheeled robot UGVs whose kinematic models described by (2.14). Let the assumptions (2.1) through (2.3) hold. Under the control law (2.21), it is guaranteed that the whole system is asymptotically stable and the positions of UGVs converge to a centroidal Voronoi configuration.*

Proof. Consider a Lyapunov function candidate as:

$$\vartheta = \sum_{i=1}^{n_g} k_v^i \, \mathcal{H}_i \tag{2.23}$$

Since k_v^i is positive and $\mathcal{H}_i$ is also an integral of strictly positive terms, thus the candidate Lyapunov function ϑ is lower-bounded by zero. Substituting (2.14), (2.8) and (2.22) into the time derivative of ϑ along the system trajectory and using the control input (2.21) results in:

$$
\begin{aligned}
\dot{\vartheta} &= \sum_{i=1}^{n_g} k_v^i \; M_{V_i} \left(p_i - C_{V_i}\right) \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} v_i \\
&= \sum_{i=1}^{n_g} k_v^i \; M_{V_i} \left(p_i - C_{V_i}\right) \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \cdot (-k_v^i) \begin{bmatrix} \cos\theta_i & \sin\theta_i \end{bmatrix} \left(p_i - C_{V_i}\right)^T \qquad (2.24) \\
&= -M_{V_i} \; {k_v^i}^2 \; \sum_{i=1}^{n_g} \|p_i - C_{V_i}\|^2
\end{aligned}
$$

which is clearly non-positive. Analogous to the proof of Theorem 2.1, it can be concluded that for any $i \in \{1, \ldots, n_g\}$, $p_i = C_{V_i}$ is the largest invariant set corresponding to the centroidal Voronoi configurations. Therefore, under the control law (2.21), the closed-loop system is asymptotically stable and the configuration of UGVs converge to the centroidal Voronoi configuration. □

It is worth to mention that although each controller depends on its Voronoi centroid, calculation of the center of Voronoi depends on the neighboring Voronoi region. So each vehicle needs to communicate with other vehicles in its neighboring Voronoi regions to compute its Voronoi region. Another advantage of the Voronoi approach is the inherent collision avoidance feature of this partitioning technique. The vehicles applying their control law will move towards the centroid of their Voronoi region. Due to the convexity of region, the centroid is always inside the Voronoi region. Also by designing a suitable controller for heights of multiple vehicles, they can fly at different levels. Therefore, the collision avoidance can be guaranteed in the entire mission even for the large dimension vehicles.

2.6 Simulation Results

The proposed distributed coverage algorithm has been demonstrated via numerical simulations in MATLAB/Simulink environment.

The region Q to be surveyed is a convex set in R^2. A combined UAV-UGV team, consisted of $n_a = 4$ quadrotor aerial vehicles and $n_g = 4$ wheeled robot ground vehicles is started from random initial positions. The heights and heading angles of UAVs and orientation angles of UGV are zero. Nominal parameters of the quadrotors, and controllers in the simulation are shown in Table 1. In this simulation, the priority function is assumed to be a bimodal Gaussian function as follows:

$$\varphi(q) = \frac{1}{\sigma\sqrt{2\pi}} \left(e^{-\frac{(q-\mu_1)^2}{2\sigma^2}} + e^{-\frac{(q-\mu_2)^2}{2\sigma^2}} \right)$$

where $\mu_1 = (-0.5, 0.4)$, $\mu_2 = (1, 0)$, and $\sigma = 0.2$. The initial configuration of planar position, the trajectories of positions of the group in 3D plane and the final configuration of each vehicle are shown in Figs. 2.5-2.7 respectively. The contributing Gaussian functions are shown in blue in those figures with a color intensity proportional to the value of the function.

Table 1: Simulation parameters

Quadrotor parameters		Controller parameters	
m	$1.4\ Kg$	k_1	10
l	$0.2\ m$	k_2	1
I_x, I_y, I_z	$0.03\ Kg.m^2$	k_v	3
j	$10^{-4}\ Kg.m^2$	k_w	6
		Q	$\begin{pmatrix} 1 & 0 \\ 0 & 0.001 \end{pmatrix}$
		R	0.1

The error between each vehicle's planar position and the centroid of its Voronoi region is used as a performance criterion. The time history of the mean error measured in this way is depicted in Fig. 2.8. Note that the error diminishes over time as would

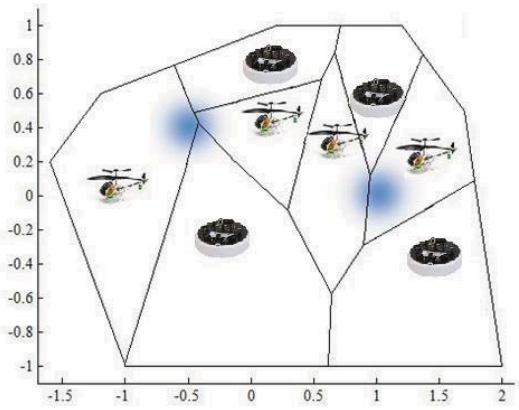


Figure 2.5: The initial configuration of planar position of a combined UAV-UGV team. (The helicopters represent the aerial vehicles and robots represent the ground vehicles.)

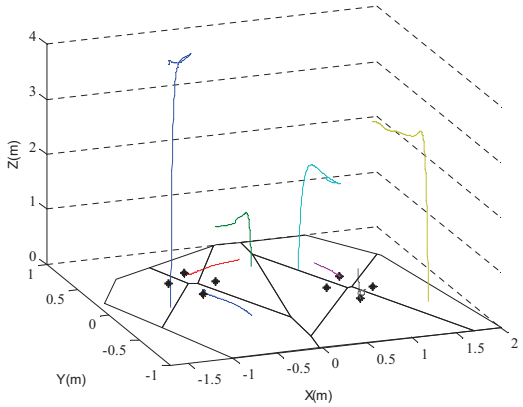


Figure 2.6: The trajectories of the position of eight vehicles in 3D plane (the final center of each Voronoi region that corresponds to final planar position of each vehicle in XY plane is marked by black $*$).

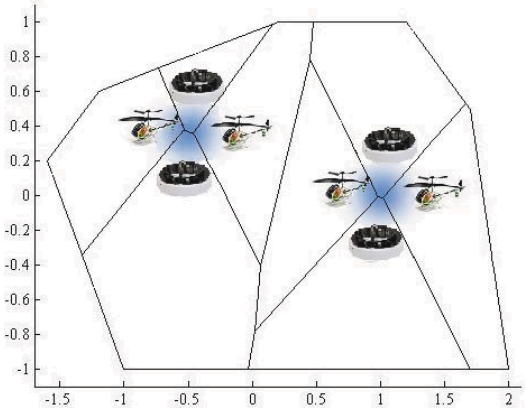


Figure 2.7: The final configuration of planar position of a combined UAV-UGV team. (The helicopters represent the aerial vehicles and robots represent the ground vehicles.)

be expected. A LQR method is used for controlling the height and heading orientation

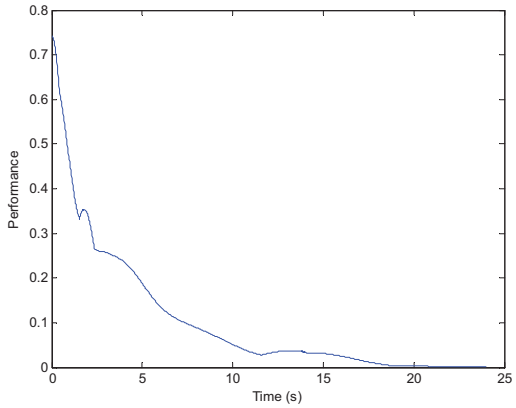


Figure 2.8: The mean error between the vehicle's planar positions and the centroids of their Voronoi regions.

of each quadrotor UAV. The parameters of LQR controller are presented in Table 1. The height and yaw angle of each quadrotor UAV is shown in Figs. 2.9 and 2.10 respectively. For the purpose of collision avoidance, each quadrotor UAV flies in a different level. Moreover, the desired yaw angles for all quadrotor UAVs in a group are 20 deg. The performance of the proposed controller is clearly demonstrated in the simulation results.

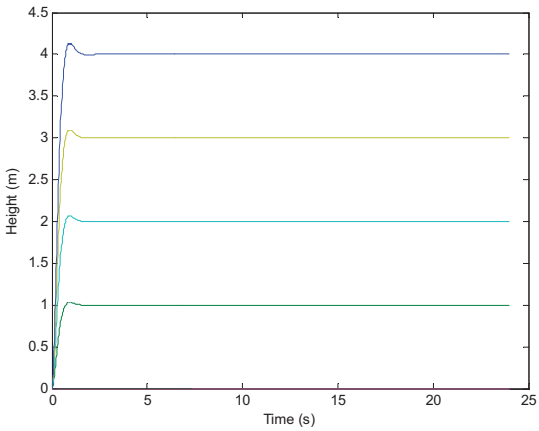


Figure 2.9: The height of each quadrotor UAV in the mission.

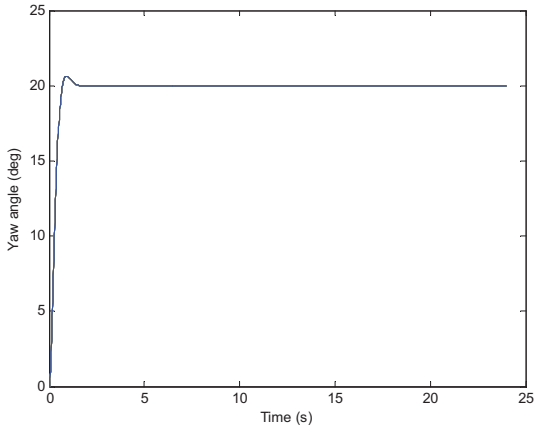


Figure 2.10: The yaw angle of each quadrotor in the mission.

Chapter 3

Deployment Strategy for a Network of Non-Identical Agents

The problem of providing a prescribed service (such as coverage) using a group of heterogeneous mobile agents is investigated in this chapter, where it is desired to minimize the overall service cost. First, the problem of incorporating heterogeneity of the agents is addressed by allowing them to have different dynamics and different serving capabilities. In fact, when the operating costs of agents are different, the conventional Voronoi diagram is not effective for agents deployment in the network. To address this issue, the field is partitioned using multiplicatively-weighted Voronoi (MW-Voronoi) diagram. To further generalize the results, a priority function is assumed to be given to prioritize the importance of giving service to different points in the field. The proposed approach is then evaluated experimentally using four unmanned systems consisting of one quadrotor helicopter and three ground vehicles available in the Networked Autonomous Vehicles Lab (NAVL).

3.1 Problem Statement

Similar to the problem defined in Section 2.2, let Q be a convex field in R^2 and consider a group of n agents randomly distributed in Q with the position of the i-th agent denoted by p_i for any $i \in \mathbf{n} := \{1, 2, \ldots, n\}$. Accordingly the positions of all agents are denoted by the set $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$. Let also $\varphi : Q \to R_+$ be a priority function which represents the likelihood of an event taking place at any arbitrary point $q \in Q$.

Denote by $f_i(q)$ the cost of serving an event taken place at point q by the i-th agent, where $f_i : R \to R_+$ is a known strictly increasing convex function. For points located far from an agent, the cost of serving incurred by the agent is higher. In this type of problem, the cost function can encode different quantities of interest such as the travel time or the energy consumption required to serve any point in the field. It is to be noted that while the function f_i is not necessarily the same for different agents, it is assumed to be identical for any set of points located at the same distance from the corresponding agent. In other words, $f_i(q_j) = f_i(q_k)$, for any $q_j, q_k \in Q$ such that $\|p_i - q_j\| = \|p_i - q_k\|$, where $\|.\|$ denotes the Euclidean norm.

The objective of this chapter is to develop a proper agent-deployment algorithm such that the following cost function (performance index) is minimized:

$$\bar{\mathcal{H}}(\mathbf{P}, \mathbf{W}) = \sum_{i=1}^{n} \int_{W_i} f_i(q)\varphi(q)dq \tag{3.1}$$

while each point in the field is serviced by exactly one agent. The set $\mathbf{W} = \{W_1, W_2, \ldots, W_n\}$ represents a partition of the field Q into n regions, where the i-th agent is in charge of serving all points in region W_i. Roughly speaking, the function $\bar{\mathcal{H}}$ shows how well the agents are located inside the field Q. More precisely, a lower value of $\bar{\mathcal{H}}$ represents a more effective deployment of agents. This problem will hereafter be

referred to as the *service optimization problem*. Minimizing the above cost function implies maximizing the overall quality of service.

If the functions f_i are all the same for different agents (i.e., $f_i(q) = f_j(q),\ \forall i, j \in \mathbf{n}$), then for a fixed set of agent locations $\mathbf{P}$ the conventional Voronoi diagram provides the optimal partitioning of the field [11], [39]. It is also shown in the previous chapter, the centroidal Voronoi configuration is the optimal configuration for agents in this case. However, when f_i is agent-dependent, conventional Voronoi partitioning is no longer optimal. The problem of finding the optimal partitioning in this case can be very cumbersome, in general, but can be simplified significantly under a realistic assumption given below.

Assumption 3.1. *Throughout this chapter it is assumed that the service function of each agent has the following form:*

$$f_i(q) = \alpha_i \left\| p_i - q \right\|^2,\ \forall i \in \mathbf{n} \tag{3.2}$$

where α_i's are pre-specified strictly positive coefficients. The performance index (3.1) with the above service functions will be denoted by $\mathcal{H}(\mathbf{P}, \mathbf{W})$ [70].

The service optimization problem introduced above is investigated in previous chapter and in [26, 39] for the case where the coefficients $\alpha_1, \ldots, \alpha_n$ are equal. We showed that the center of mass of each Voronoi region corresponding to each agent is the optimal configuration of agents to move towards it. In the next section, the service optimization problem introduced here will be investigated in the case of nonidentical coefficients.

3.2 Locational Optimization

3.2.1 Background

Let $\mathbf{S}$ denote a set of n distinct weighted nodes $\{(S_1, w_1), (S_2, w_2), \ldots, (S_n, w_n)\}$ in a 2D field, where $w_i > 0$ is the weighting factor associated with the node S_i, for any $i \in \mathbf{n}$. The weighted distance is defined below.

Definition 3.1. *The weighted distance of a point q from the node (S_i, w_i), $i \in \mathbf{n}$, is defined as:*

$$d_w(q, S_i) = \frac{d(q, S_i)}{w_i}$$

where $d(q, S_i)$ denotes the Euclidean distance between the point q and the node S_i.

Assume that the field is partitioned into n regions such that each region contains only one node, which is the nearest node, in the sense of weighted distance, to any point inside the region. The diagram obtained by this type of partitioning is called the multiplicatively-weighted Voronoi (MW-Voronoi) diagram [17]. The resultant regions $\Pi_1, \ldots, \Pi_n$ obtained by such partitioning are described mathematically as follows:

$$\Pi_i = \left\{ q \in \mathbb{R}^2 \mid d_w(q, S_i) \leq d_w(q, S_j), \forall j \in \mathbf{n} - \{i\} \right\} \tag{3.3}$$

for any $i \in \mathbf{n}$.

Definition 3.2. *The* Apollonian circle *of the segment AB, denoted by $\Omega_{AB,k}$, is the locus of all points E such that $\frac{AE}{BE} = k$ [71].*

To construct the i-th MW-Voronoi region, the Apollonian circles $\Omega_{S_iS_j,\frac{w_i}{w_j}}$ are found for all $S_j \in \mathbf{S}\backslash\{S_i\}$, as described in [72]. This generates a number of closed regions in the field, and the smallest region containing the i-th node is referred to the i-th MW-Voronoi region (e.g., see Fig. 3.1).

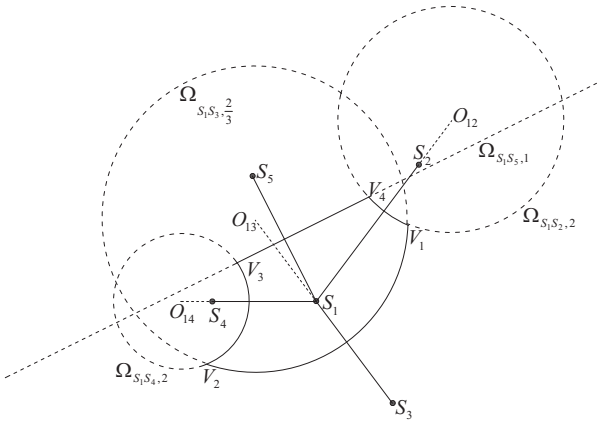


Figure 3.1: The MW-Voronoi region for a node S_1 with four neighbors $S_2, \ldots, S_5$.

Definition 3.3. *The mass and center of mass of a region W_i with respect to the priority function φ are respectively defined as follows:*

$$M_{W_i} = \int_{W_i} \varphi(q) dq \tag{3.4}$$

$$C_{W_i} = \frac{1}{M_{W_i}} \int q\varphi(q) dq \tag{3.5}$$

Definition 3.4. *A center multiplicatively-weighted Voronoi (CMWV) configuration of n distinct weighted nodes $\{(S_1, w_1), (S_2, w_2), \ldots, (S_n, w_n)\}$ in the prioritized field Q, is a configuration where each node is located in the center of mass of its MW-Voronoi region. This configuration will hereafter be denoted by $CMWV_n(w_1, w_2, \ldots, w_n)$.*

An example of the MW-Voronoi diagram for a group of 20 weighted nodes is depicted in Fig. 3.2. The MW-Voronoi diagram is used in this chapter to develop a distributed deployment algorithm for multi-agent systems.

3.2.2 Optimal Configuration

In what follows, it is desired to find agents configuration such that the underlying cost function is minimized for the case when agents have heterogeneous serving capabilities

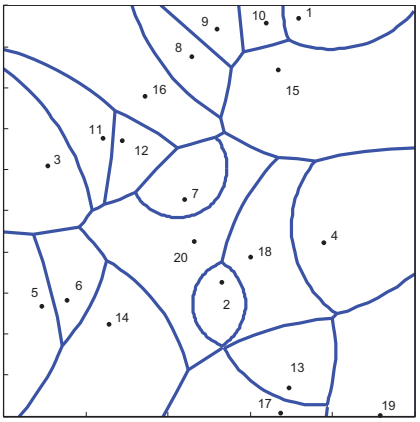


Figure 3.2: An example of the MW-Voronoi diagram for a group of 20 nodes with different weights in a network.

(i.e., non-identical coefficients as described in Section 3.1). The optimal configuration problem is addressed in the sequel.

Proposition 3.1. *Assume that the agent positions* $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$ *are fixed. If the coefficients* $\alpha_1, \ldots, \alpha_n$ *in* $\mathcal{H}$ *are not identical, then the MW-Voronoi diagram of the set of weighted nodes* $\left\{(p_1, \frac{1}{\sqrt{\alpha_1}}), (p_2, \frac{1}{\sqrt{\alpha_2}}), \ldots, (p_n, \frac{1}{\sqrt{\alpha_n}})\right\}$ *is the optimal partitioning of* Q *for minimizing* $\mathcal{H}$.

Proof. Let the above-mentioned MW-Voronoi diagram be denoted by $\mathbf{MW}(\mathbf{P}, \alpha)$, where $\alpha := [\alpha_1, \alpha_2, \ldots, \alpha_n]$. Let also the i-th region of $\mathbf{MW}(\mathbf{P}, \alpha)$ be represented by $MW_i(\mathbf{P}, \alpha)$. Suppose that $\mathbf{W} = \{W_1, W_2, \ldots, W_n\}$ is the optimal partitioning of Q w.r.t. $\mathcal{H}$, and that $\mathbf{W} \neq \mathbf{MW}(\mathbf{P}, \alpha)$. Therefore:

$$\exists\ Q_1 \in Q,\ \exists\ i, j \in \mathbf{n}\ :\ \begin{cases} Q_1 \subseteq W_i \\ Q_1 \cap MW_i(\mathbf{P}, \alpha) = \emptyset, \\ Q_1 \subseteq MW_j(\mathbf{P}, \alpha),\ j \neq i \end{cases}$$

Now, consider the new partitioning $\acute{\mathbf{W}} = \left\{\acute{W}_1, \acute{W}_2, \ldots, \acute{W}_n\right\}$ of Q be defined as:

$$\acute{W}_k = W_k,\ \forall k \neq i, j, \quad \acute{W}_i = W_i - Q_1, \quad \acute{W}_j = W_j \cup Q_1$$

Denote the values of the cost function $\mathcal{H}$ corresponding to the partitionings $\mathbf{W}$ and $\acute{\mathbf{W}}$ by $\mathcal{H}_{\mathbf{W}}$ and $\mathcal{H}_{\acute{\mathbf{W}}}$, respectively. It is straightforward to verify that the following equation holds:

$$\mathcal{H}_{\acute{\mathbf{W}}} = \mathcal{H}_{\mathbf{W}} + \int_{Q_1} (\alpha_j \left\|p_j - q\right\|^2 - \alpha_i \left\|p_i - q\right\|^2)\varphi(q)dq \tag{3.6}$$

On the other hand, since $\forall q \in Q_1$, $q \in MW_j(\mathbf{P}, \alpha)$ and $q \notin MW_i(\mathbf{P}, \alpha)$ one arrives at:

$$\frac{\left\|p_j - q\right\|}{1/\sqrt{\alpha_j}} < \frac{\left\|p_i - q\right\|}{1/\sqrt{\alpha_i}},\ \forall q\ \in Q_1 \tag{3.7}$$

or equivalently:

$$\alpha_j \left\|p_j - q\right\|^2 < \alpha_i \left\|p_i - q\right\|^2,\ \forall q\ \in Q_1 \tag{3.8}$$

From (3.6) and (3.8), it can be concluded that:

$$\mathcal{H}_{\acute{\mathbf{W}}} < \mathcal{H}_{\mathbf{W}} \tag{3.9}$$

which contradicts the initial assumption that $\mathbf{W}$ is the optimal partitioning of Q. This completes the proof. □

Remark 3.1. *The above result shows that for any arbitrary set of points* $\mathbf{P}$*, the optimal partition for the underlying service problem is the MW-Voronoi partition, i.e., the following relation holds:*

$$\mathcal{H}\left(\mathbf{P}, \mathbf{MW}\left(\mathbf{P}, \alpha\right)\right) \leq \mathcal{H}(\mathbf{P}, \mathbf{W})$$

where $\mathbf{W}$ *is any arbitrary partition of the field.*

In the case the partitions in the field are fixed, the problem can be considered as n independent 1-center optimization problem. As mentioned in Section 2.3, the center

of mass of each partition is the optimal position for each agent. Then, the following result is a straightforward extension of the 1-center problem.

Remark 3.2. *Assume* $\mathbf{W} = \{W_1, W_2, \ldots, W_n\}$ *is a fixed partition of the field* Q*. Then the center of mass of each region is the optimal position for the corresponding agent in order to minimize the cost function* $\mathcal{H}$ *with non-identical coefficients* $\alpha_1, \ldots, \alpha_n$ *in (3.2).*

According to Remark 3.2, for any arbitrary partition $\mathbf{W}$ of the field the agent positions for the optimal service are the centers of mass of each region. More precisely:

$$\mathcal{H}(\{C_{W_1}, C_{W_2}, \ldots, C_{W_n}\}, \mathbf{W}) \leq \mathcal{H}(\mathbf{P}, \mathbf{W})$$

where $\mathbf{P}$ is any arbitrary set of n points (one in each region). This motivates the definition of the centroidal configuration for the MW-Voronoi diagram.

Based on Definition 3.4, one can easily establish the following optimality result.

Remark 3.3. *Given* n *distinct weighted nodes* $\mathbf{S}$ *and service functions (3.2), the configuration* $CMWV_n(1/\sqrt{\alpha_1}, 1/\sqrt{\alpha_2}, \ldots, 1/\sqrt{\alpha_n})$ *is the solution to the service optimization problem min*$\mathcal{H}$*.* □

3.3 Distributed Coverage Control

In this section, it is desired to design a set of planar position controllers for agents with double-integrator dynamics. Most of the existing Voronoi-based coverage control schemes assume single-integrator dynamics for the agents [11]. However, many unmanned systems have non-trivial dynamics, which can invalidate the performance of the corresponding algorithms. The motion of a broad class of mobile agents can be expressed by a double-integrator dynamic model. In addition, the dynamics of many

agents can be feedback linearized to a double-integrator form. As will be shown in Section 3.5, double-integrator dynamics is a proper model for the quadrotor helicopter system.

For the purpose of coordinating multiple mobile agents to service the points in the environment, it is desired to design a position-control strategy based on the CMWV configuration. To achieve this goal, the following assumption is needed.

Assumption 3.2. *Each mobile agent can communicate with other vehicles in its neighboring MW-Voronoi regions.*

In the remainder of the chapter, C_{W_i} represents the center of mass of the MW-Voronoi region that corresponds to the i-th mobile agent, currently positioned at p_i.

Consider a set of agents modeled as *double-integrator* point masses moving in Q as follows:

$$\ddot{p}_i = u_i \tag{3.10}$$

where u_i is the control input of the i-th agent. A position-control law of the following form is proposed for the i-th agent:

$$u_i = 2\alpha_i k_1^i M_{W_i}(C_{W_i} - p_i) - k_2^i \dot{p}_i \tag{3.11}$$

where k_1^i and k_2^i are some positive gains to be chosen by the designer, and M_{W_i} is the mass of the i-th MW-Voronoi region as defined in (3.4). The following theorem presents the convergence result.

Theorem 3.1. *Consider a group of n mobile agents with dynamic models described by (3.10). The overall system is asymptotically stable under the local control laws of the form (3.11), and the positions of agents converge to the CMWV configuration.*

Proof. Consider a Lyapunov function candidate as:

$$\vartheta = \sum_{i=1}^{n} k_1^i \mathcal{H}_i + \sum_{i=1}^{n} \frac{1}{2} \dot{p}_i \dot{p}_i^T \tag{3.12}$$

where $\mathcal{H}_i = \int_{W_i} \alpha_i \left\| p_i - q \right\|^2 \varphi(q) dq$. Since α_i, k_1^i are positive numbers and also $\|.\|^2$ and $\varphi(q)$ are positive functions, thus the candidate Lyapunov function ϑ is lower-bounded by zero. Moreover, it can be easily shown that:

$$\frac{\partial \mathcal{H}_i}{\partial p_i} = 2\alpha_i \int_{W_i} (p_i - q)\varphi(q) dq = 2\alpha_i M_{W_i} (p_i - C_{W_i}) \tag{3.13}$$

By substituting (3.10) and (3.13) into the time derivative of ϑ along the system trajectory and using the control input (3.11), one arrives at:

$$\begin{aligned} \dot{\vartheta} &= \sum_{i=1}^{n} k_1^i \dot{\mathcal{H}}_i + \sum_{i=1}^{n} \ddot{p}_i \ \dot{p}_i^T \\ &= \sum_{i=1}^{n} \left[\left(2\alpha_i k_1^i M_{W_i} \left(p_i - C_{W_i} \right) + \ddot{p}_i \right) \dot{p}_i^T \right] \\ &= \sum_{i=1}^{n} \left(-k_2^i \dot{p}_i \dot{p}_i^T \right) \end{aligned}$$

which is clearly non-positive. Let $\mathcal{M}$ be the set of all points in Q where $\dot{\vartheta} = 0$. Since every center of mass C_{W_i}, $i \in \mathbf{n}$ lies in the interior of the region Q, hence the mobile agents never leave their regions. This in turn means that $\mathcal{M}$ is a positive invariant set for the trajectories of the closed-loop system. Since this set is closed and bounded, one can use LaSalle's invariance principle to infer that the positions of mobile agents converge to the largest invariant subset of the set $\mathcal{M}$. For any trajectory belonging to the set $\mathcal{M}$, it results from the model (3.10) and the control law (3.11) that:

$$\dot{p}_i \equiv 0 \Rightarrow \ddot{p}_i \equiv 0 \Rightarrow u_i \equiv 0 \Rightarrow p_i = C_{W_i}, \ \forall i \in \mathbf{n}$$

One can then conclude that for any $i \in \mathbf{n}$, $p_i = C_{W_i}$ is the largest invariant set corresponding to the CMWV configuration. Therefore, under the control law (3.11) the closed-loop system is asymptotically stable and the positions of agents converge to the CMWV configuration. □

The proposed distributed deployment strategy to coordinate the mobile agents to service the points in the environment utilizes the position control law (3.11). At each time step, agent i detects its neighboring agents and exchanges position information with them. Then, agent i constructs its MW-Voronoi cell (as explained in Section 3.2) and calculates its center of mass in order to construct the control input, under which the agent moves towards C_{W_i}. At every time step all agents in the group follow the above steps simultaneously to finally converge to the optimal configuration.

3.4 Simulation Results

In this section, the efficacy of the proposed distributed agent-deployment algorithm is demonstrated by simulations.

The environment used in the simulations is a 10m $\times$ 10m square. A group of 10 agents with double-integrator dynamics are deployed to service the points in the environment. The coefficients $\alpha_1, \alpha_2, \ldots, \alpha_{10}$ in $\mathcal{H}$ (which reflect the operating cost of different agents) are given as $\frac{25}{9}, \frac{25}{9}, \frac{25}{9}, \frac{25}{16}, \frac{25}{16}, 1, 1, 1, 1, \frac{25}{49}$, respectively.

Two different scenarios are considered in the sequel. In the first scenario, the initial positions of agents are chosen randomly, and the serving priority is given by the following Gaussian function:

$$\varphi(q) = \frac{1}{\sigma\sqrt{2\pi}} \left(e^{-\frac{(q-\mu)^2}{2\sigma^2}} \right) \tag{3.14}$$

where $\mu = (5, 5)$ and $\sigma = 0.6$. The initial and final configurations of agents along with

their trajectories under the presented deployment algorithm are shown in Fig. 3.3. In this figure, the initial position of agents are shown by $\times$, and the final position of each agent is depicted by a circle whose radius is equal to the inverse of the square root of the corresponding coefficient in $\mathcal{H}$ (e.g., the radius of the 4th circle is $4/5$ because $\alpha_4 = 25/16$). Moreover, the Gaussian priority function $\varphi(.)$ is shown in gray with a color intensity proportional to the value of the function.

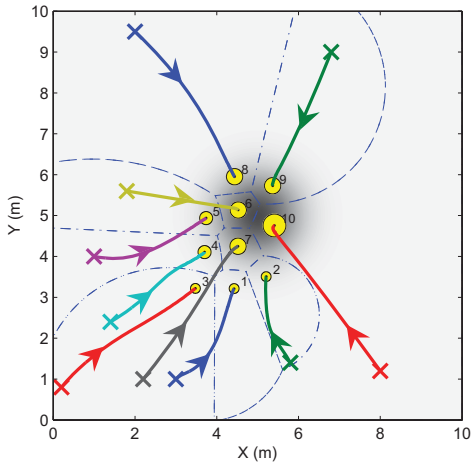


Figure 3.3: The initial and final positions of agents in the first scenario along with their trajectories under the proposed deployment algorithm. The corresponding MW-Voronoi regions' boundaries are also depicted by dashed curves.

In the second scenario, the same group of agents are considered, but with bimodal Gaussian priority function as follows:

$$\varphi(q) = \frac{1}{\sigma\sqrt{2\pi}} \left(e^{-\frac{(q-\mu_1)^2}{2\sigma^2}} + e^{-\frac{(q-\mu_2)^2}{2\sigma^2}} \right) \tag{3.15}$$

where $\mu_1 = (8, 2)$, $\mu_2 = (2, 8)$, and $\sigma = 0.6$. Unlike the previous case, it is assumed that agents start their move from the border of the field. Similar to the first scenario, each agent computes its MW-Voronoi region and moves towards its mass center based on the control input (3.11). The results are provided in Fig. 3.4, analogously to those in Fig. 3.3. As shown in this figure, all agents aggregate around the more important

areas in the field. The cost functions resulted from using the proposed controller are depicted in Figs. 3.5(a), (b) for both scenarios. As it can be observed from Fig. 3.5, in both cases the cost function decreases by time significantly.

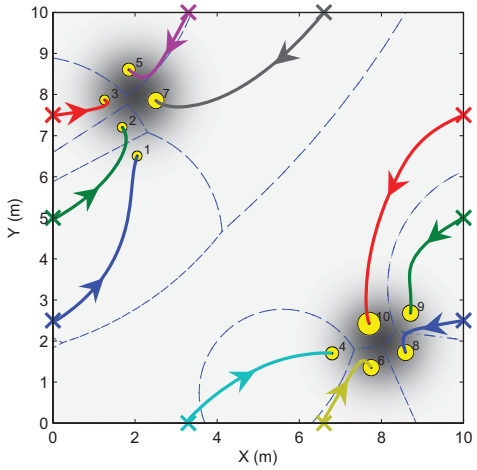
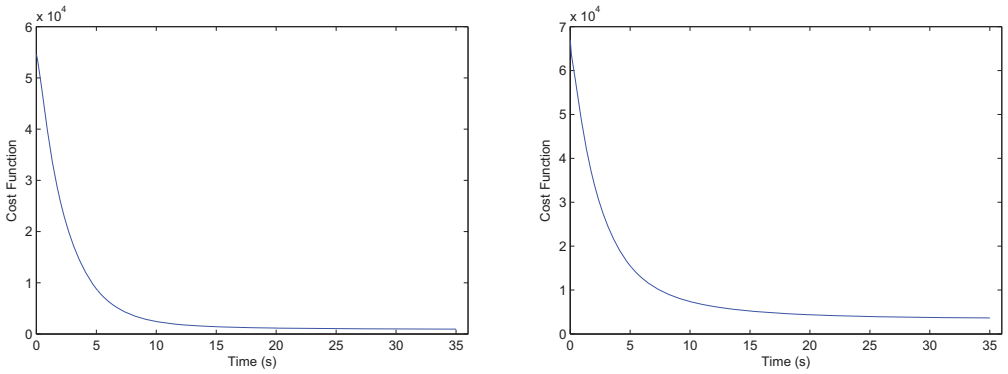


Figure 3.4: The initial and final positions of agents in the second scenario along with their trajectories under the proposed deployment algorithm. The corresponding MW-Voronoi regions' boundaries are also depicted by dashed curves.



(a) (b)

Figure 3.5: The cost function $\mathcal{H}$ obtained by using the proposed deployment strategy for: (a) first scenario, and (b) second scenario.

In order to evaluate the average performance of the proposed deployment strategy, 50 different simulations have been carried out with random initial positions for the first scenario. The resultant average cost function is depicted in Fig. 3.6, which demonstrates the effectiveness of the approach in general.

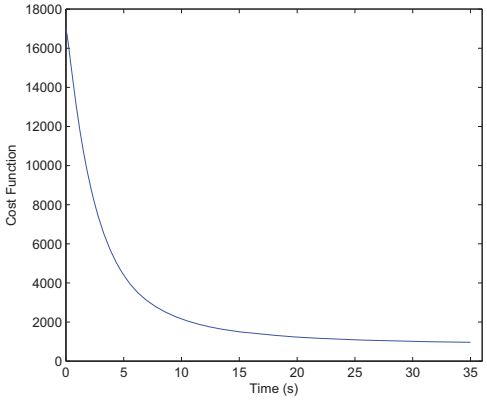


Figure 3.6: The average cost function $\mathcal{H}$ obtained by using 50 different random initial configurations for the first scenario.

3.5 Experimental Results

To demonstrate the efficacy of the presented theoretical developments, some experiments are conducted on four unmanned systems consisting of one quadrotor helicopter and three ground vehicles. These unmanned systems are equipped with a QuaRC-powered single-board Gumstix embedded computer (QuaRC is the real-time control software by Quanser Inc). QuaRC allows to rapidly implement controllers designed in MATLAB/Simulink environment for real-time control of vehicles. Sensors measurement, data logging and parameter tuning are supported between the ground host computer and the target vehicles. The experiments are performed indoor with no GPS signals, and a network of OptiTrack camera systems from NaturalPoint is employed to provide the systems positions in the 3D space. In Fig. 3.7, the experimental environment consisting of the four unmanned vehicles, a host computer, and the network of OptiTrack cameras is illustrated.

3.5.1 Unmanned Aerial Vehicle: the Qball-X4

The quadrotor UAV available at the Networked Autonomous Vehicles (NAV) Lab in the Department of Mechanical and Industrial Engineering of Concordia University is

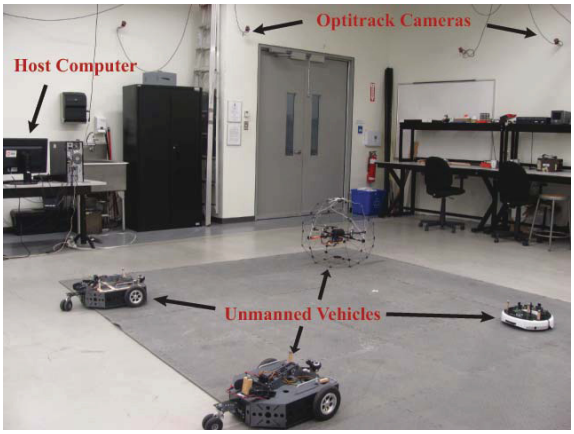


Figure 3.7: The experimental environment.

the Qball-X4 testbed shown in Fig. 3.8. Detailed description of the system can be found in [73].

Figure 3.8: The Qball-X4 UAV.

A simplified linear model can be derived for the Qball-X4 by assuming hovering conditions ($u_z \approx mg$ in the x and y directions) without yawing ($\psi = 0$) and with small roll and pitch angles:

$$\begin{aligned} \ddot{x} &= \theta g; & J_1\ddot{\theta} &= u_\theta \\ \ddot{y} &= -\phi g; & J_2\ddot{\phi} &= u_\phi \\ \ddot{z} &= u_z/m - g; & J_3\ddot{\psi} &= u_\psi \end{aligned} \tag{3.16}$$

where x, y and z are the coordinates of the quadrotor UAV center of mass in the

earth-frame, and θ, ϕ and ψ are the Euler angles in pitch, roll and yaw, respectively. The lift u_z generated by the four propellers is applied to the quadrotor UAV in the z-direction (body-fixed frame), and the torques u_θ, u_ϕ and u_ψ tend to rotate the quadrotor in the directions of θ, ϕ and ψ, respectively. Moreover, m is the mass and J_1, J_2 and J_3 are the moments of inertia along the y, x and z axes, respectively. The four motors of the system are driven by pulse width modulated (PWM) inputs, with the following relations between the lift/torques and the PWM inputs:

$$\begin{aligned} u_z &= K(u_1 + u_2 + u_3 + u_4) \\ u_\theta &= KL(u_1 - u_2) \\ u_\phi &= KL(u_3 - u_4) \\ u_\psi &= KK_\psi(u_1 + u_2 - u_3 - u_4) \end{aligned} \tag{3.17}$$

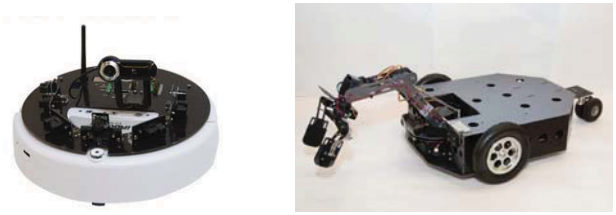
where $u_1, \ldots, u_4$ are the PWM inputs, K and K_ψ are positive constants and L is the distance between the motor and the quadrotor's center of mass.

It is important to note that since the linear model of the quadrotor is decoupled along the three axes, one can easily control the height and yaw angle independently by using conventional control methods. Moreover, for the coordination of multiple vehicles in order to cover a planar environment, the position controller is designed based on the CMWV configuration.

3.5.2 Unmanned Ground Vehicles: the Qbot UGV and the Quanser QGV

Qbot (shown in Fig. 9(a)) is an autonomous ground vehicle and an ideal platform for research in an indoor lab environment. It has a low power Gumstix Verdex XL6P 600 MHz on-board computer operated by Linux OS. The Quanser QGV (depicted in

Fig. 9(b)) is a vehicle platform suitable for a wide variety of UGV related research. Both the Qbot and the Quanser QGV are differential drive ground robots. The Quanser QGV is also equipped with a 4 degree-of-freedom robotic manipulator with gripper but will not be used in the experiments since control of these grippers is not the objective of this thesis. The dynamic model of the QGV (without gripper) and the Qbot is considered as exactly the same as (2.14) for this coverage control problem, therefore the structure of control inputs for these vehicles is analogous to (2.21).



(a) The Qbot (b) The QGV

Figure 3.9: The unmanned ground vehicles used in the experiments.

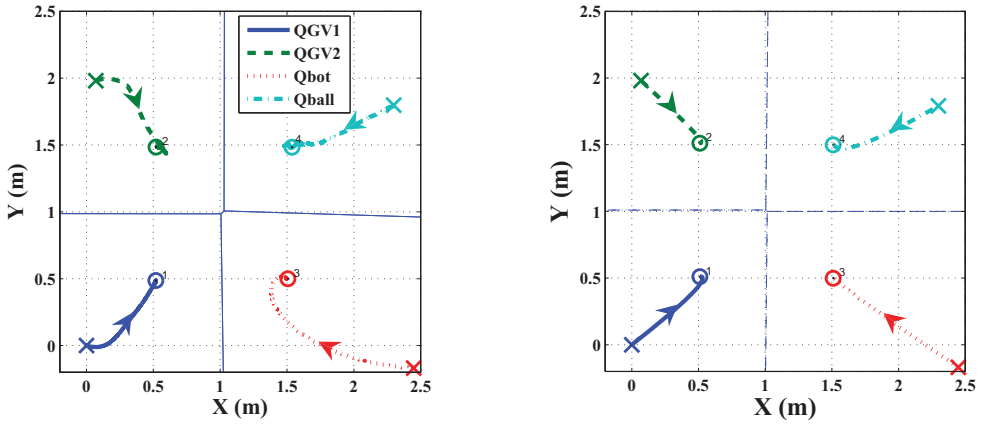
3.5.3 Experimental Tests

The environment used in the experiments is a 2m $\times$ 2m square surface with high ceiling to allow the UAV to fly without any physical restriction. A group of two QGV UGVs, one Qbot UGV and one Qball-X4 UAV is deployed to cover the environment. The vehicles start their mission from the corner of the field. To validate the proposed theoretical results, several experiments are conducted through the following four scenarios.

Scenario 1

The coefficients of $\mathcal{H}$ are identical and equal to 1 (i.e., $\alpha_1 = \ldots = \alpha_4 = 1$), and the priority function is uniform (i.e., $\varphi(q) = 1,\ \forall\ q \in Q$). Each vehicle constructs its MW-Voronoi cell based on the positions of the neighboring vehicles, and then

moves to the center of mass of its own cell by the proposed control input. The experimental results of this scenario are presented in Fig. 3.10(a) along with the corresponding computer simulation results in Fig. 3.10(b) for completeness. Note that since the priority function is uniform and all coefficients are identical, the field is divided symmetrically between the four vehicles in this case.



(a) (b)

Figure 3.10: Trajectories of vehicles along with their corresponding MW-Voronoi regions in the first scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB.

Scenario 2

The coefficients of $\mathcal{H}$ are again identical and equal to 1 but the priority function in this case is the Gaussian function defined in (3.14) with $\mu = (1.2, 0.6)$ and $\sigma = 0.2$. It is noteworthy to mention that the priority function is known *a priori* by all vehicles, as required by the proposed deployment strategy. The results are depicted in Figs. 3.11(a), (b), analogously to those in Fig. 3.10. It can be observed from these figures that the vehicles eventually aggregate around the higher priority points in the field, shown by higher gray levels.

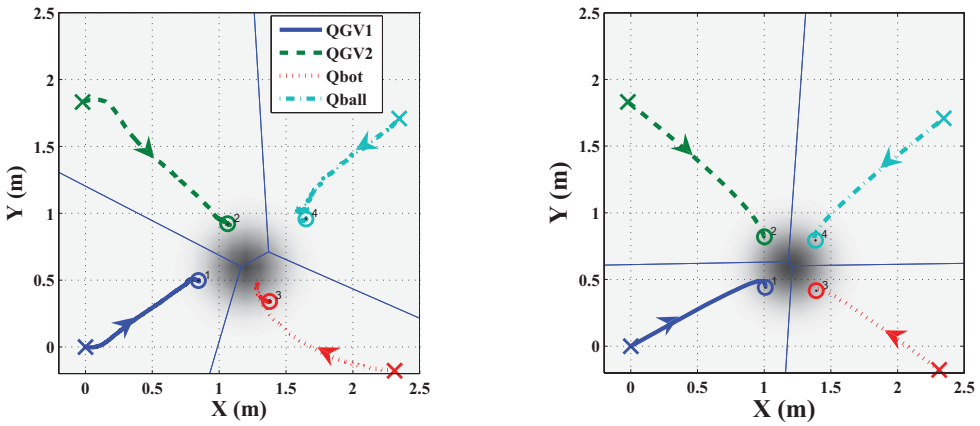


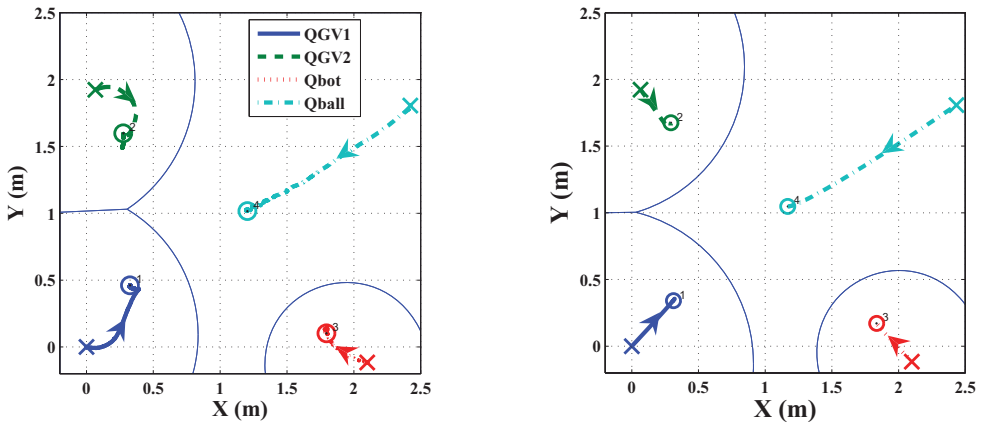
Figure 3.11: Trajectories of vehicles along with their corresponding MW-Voronoi regions in the second scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively, and the gray level of the field is proportional to the value of the Gaussian priority function. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB.

Scenario 3

In this scenario, it is assumed that the priority function of the field is uniform but the operating cost of each vehicle are different. To this end, let the coefficients $\alpha_1, \ldots, \alpha_4$ in $\mathcal{H}$ (which reflect different serving capabilities of vehicles) are chosen as $\frac{1}{4}, \frac{1}{4}, 1, \frac{1}{25}$, respectively. The results are depicted in Figs. 3.12(a), (b) analogously to those in Fig. 3.10. As shown in Fig. 3.12, vehicles with higher operating costs (i.e., those with smaller coefficients) are assigned to regions with larger area.

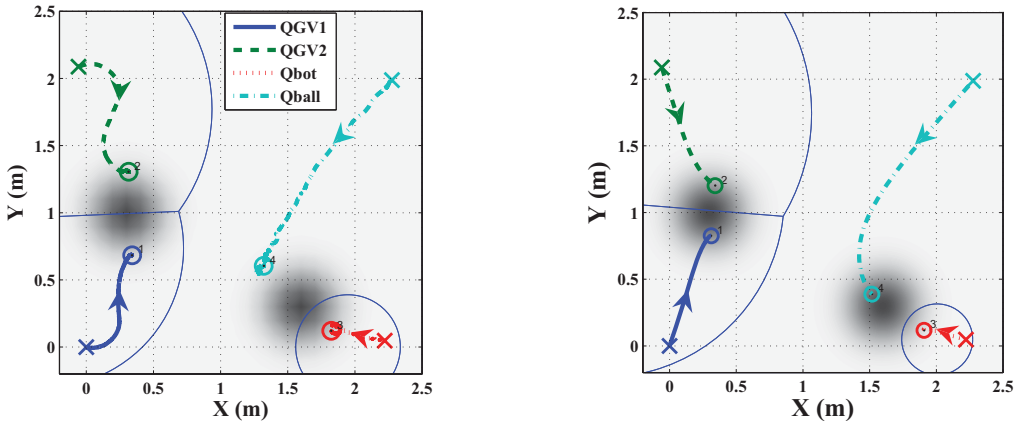
Scenario 4

In the fourth scenario (which is the most complete one), the priority function is assumed to be a bimodal Gaussian function as defined in (3.15), with $\mu_1 = (0.3, 1)$, $\mu_2 = (1.6, 0.3)$ and $\sigma = 0.2$. The operating costs of vehicles are the same as the previous scenario. The experimental results of this scenario along with the corresponding computer simulations are depicted in Figs. 3.13(a), (b). The resultant trajectories in this case show that the vehicles eventually cover the most important areas.



(a) (b)

Figure 3.12: Trajectories of vehicles along with their corresponding MW-Voronoi regions in the third scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB.



(a) (b)

Figure 3.13: Trajectories of vehicles along with their corresponding MW-Voronoi regions in the fourth scenario. The initial and final planar positions of each vehicle are indicated by $\times$ and $\odot$, respectively, and the gray level of the field is proportional to the value of the Gaussian priority function. (a) Experimental results obtained in the lab; (b) simulation results obtained by MATLAB.

The velocities of all vehicles at X and Y directions for all these four scenarios are also depicted in Fig. 3.14. It can be observed from this figure that as the vehicles converge to the optimal configuration at the end of mission, the velocities of the vehicles approach zero.

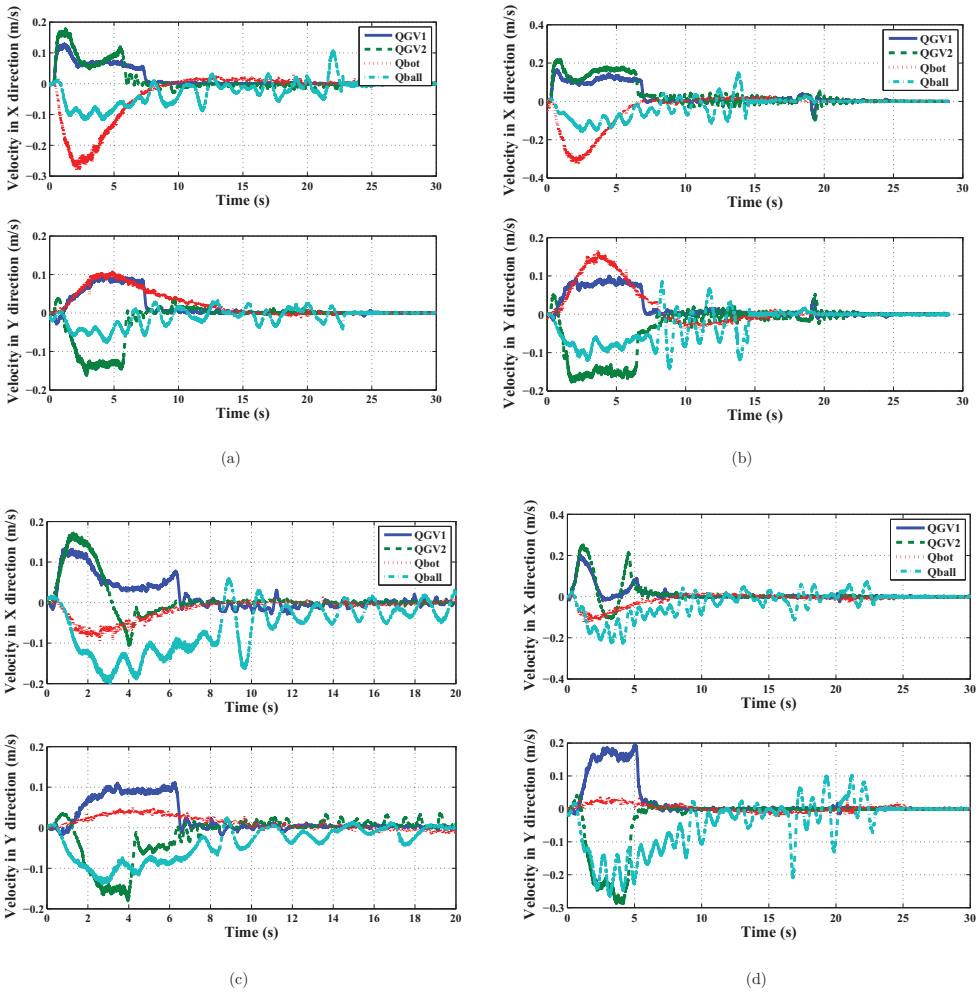


Figure 3.14: Velocities of all vehicles in the X and Y directions in the four scenarios. (a) First scenario; (b) Second scenario; (c) Third scenario; (d) Fourth scenario.

Finally, these four scenarios are carried out consecutively, where the final positions of vehicles in each scenario are their initial positions in the subsequent scenario. The video that shows the experimental results in this case is available in [74].

Chapter 4

Coverage Control of Multi-Agent Systems subject to Communication Delay

Usually in the multi-agent systems, the agents in the team transmit/receive their position information to/from a subset of agents. Most of the existing works on sensor network coverage control assume that this information exchange has no delay. However, it is known that neglecting the effect of delay in the analysis and design of multi-agent systems can lead to poor performance as well as unsafe behaviors such as inter-agent collision. While the effect of communication delay in the flocking and formation control of multi-agent systems has been investigated in the past few years (e.g., see [75], [76]), not much work has been done in the area of cooperative coverage control of a network of agents subject to communication delay.

Motivated by [34], a spatial partitioning technique is considered in this chapter to address the problem of coverage control subject to inter-agent communication delays induced by certain communication faults. A strategy is introduced for updating

the information vector of each agent and constructing its guaranteed Voronoi region based on the latest information received from its neighbors. A motion coordination algorithm is subsequently proposed for the agents such that a prescribed coverage performance function over the guaranteed Voronoi cell of every agent is minimized. One of the important properties of the proposed algorithm is inter-agent collision avoidance in the faulty conditions, which is also confirmed by simulation.

The above problem is also extended for the case sensors of agents may have variable effectiveness (health). The sensor effectiveness factor of agents is incorporated in the locational optimization problem, and a novel partitioning technique is also introduced to address this problem. Analogous to the motion coordination algorithm in the previous case, an algorithm is proposed to drive agents in such a way that the coverage performance function is minimized over the regions assigned to agents. The effectiveness of the proposed algorithm is evaluated by simulations.

4.1 Problem Formulation

Similar to the problem of Chapter 2, Q is a convex field in R^2, and an arbitrary point in Q is denoted by q. Consider a group of n mobile agents randomly distributed in Q with the position of the i-th agent represented by p_i, and the positions of all agents are denoted by the set $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$. Let also $\varphi : Q \to R_+$ be a priority function which assigns a weight to each point in the field, representing the likelihood of an event taking place at any arbitrary point in Q. Let $V = \{V_1, V_2, \ldots, V_n\}$ be the Voronoi partition of Q, where the agent positions are the generating points. It is assumed that each agent is in charge of covering all points in its own Voronoi region.

Then, the coverage performance function is defined as:

$$\mathcal{H}(\mathbf{P}) = \sum_{i=1}^{n} \int_{V_i} f_i(q)\varphi(q)dq \tag{4.1}$$

where $f_i(q)$ the cost of covering (or sensing) a point q by the i-th agent is chosen as (2.3).

The objective here is to develop an algorithm to properly place the agents in the field where individual agents receive the information about each others' locations with some delay caused by communication faults. In the next section, the inter-agent communication delays under certain fault conditions are introduced first, and a specific partitioning technique is then presented to divide the field among all agents.

4.2 Inter-Agent Communication Delays

Each agent in the team receives/sends the information from/to neighboring agents through a communication channel. Normally, all agents receive/send information from/to their neighbors with negligible processing or transmission delays. However, when the communication channel of an agent fails, the received and transmitted information from/to faulty agents are both subject to delays. Moreover, it is assumed that faulty agents and the magnitudes of their communication delays can be easily diagnosed with a proper fault detection and diagnosis (FDD) algorithm such as the one proposed in [77] (development of FDD schemes is beyond the scope of this thesis). To proceed further, some important assumptions are made in the sequel.

Assumption 4.1. *The clocks of all agents are synchronous, i.e., the time steps for all agents start at $t_k = t_0 + k\Delta t$ for $k \in \mathbb{N}$, where Δt is the smallest time step (which is known a priori), and t_0 is the initial time.*

Assumption 4.2. *The communication delays in the faulty conditions are larger than*

the smallest time step, i.e. $\tau > \Delta t$*. Further, to ensure synchronization between the communication delay and discrete time step it is assumed that* $\tau = d\Delta t$*, where* $d \in \mathbb{N}$*.*

Assumption 4.3. *The maximum velocity of every agent is bounded. For simplicity and without loss of generality it is assumed that this maximum velocity is the same for all agents. Denote this maximum velocity by* v_{max}*; the movement of agents at each time step will be bounded according to the following relation:*

$$\frac{p_i(t_{k+1}) - p_i(t_k)}{\Delta t} \leq v_{max} \tag{4.2}$$

Fig. 4.1 shows the inter-agent communication between two agents in the presence of delay. As noted earlier, agents receive delayed information from faulty neighbors and non-delayed information from healthy neighbors. Thus, the current information of agent i from a faulty neighbor, say agent j, is based on the last known position p_j^i, and the elapsed time $\tau_j^i \in R_+$ after the information was received from agent j. One can use this formulation to describe the information flow between agent i and all of its neighbors, and set τ_k^i to zero if agent $k \in \mathcal{N}_i$ is healthy. It is also assumed that the position of agent i is available for itself with no delay, i.e., $p_i^i = p_i$ and $\tau_i^i = 0$.

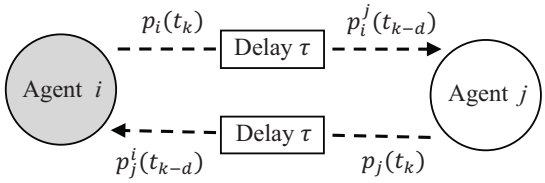


Figure 4.1: The inter-agent communication between two agents with faulty communication channel.

Since the last information available to agent i from the neighboring agent j, $\forall j \in \mathcal{N}_i$, at time t is $p_j^i(t - \tau_j^i)$, according to Assumption 4.3 agent i knows that the maximum displacement of agent j from the time instant $t - \tau_j^i$ to t is upper bounded by $r_j^i = v_{max}\tau_j^i$. In other words, based on the information available to agent i at

time $t - \tau_j^i$, it is guaranteed that at time t neighboring agent j is inside the ball $\bar{B} = (p_j^i, r_j^i)$. Consequently, the information vector of agent i at time t is updated to $D^i = [p_i, (p_j^i, r_j^i)]; j \in \mathcal{N}_i$. It is noteworthy that if the information of a fault-free neighboring agent j at time t is available with no delay, then r_j^i is zero.

Since the exact locations of some agents are not available in the faulty case, the conventional Voronoi partitioning cannot be used to optimize the coverage function $\mathcal{H}(\mathbf{P})$. Hence, the guaranteed Voronoi diagram introduced in [78] is described briefly in the next section to address the coverage problem subject to communication delays.

4.3 Guaranteed Voronoi Partitioning

Let D be a set of regions in Q, denoted by $D = \{D_1, D_2, \ldots, D_n\}$, each region containing exactly one agent. The guaranteed Voronoi region G_i, associated with agent i is the locus of all points that are closer to this agent than to any other agents, i.e.:

$$G_i = \{q \in Q | \max_{x \in D_i} \|q - x\| \leq \min_{y \in D_j} \|q - y\|, \forall j \neq i\}, \; i \in \mathbf{n} := \{1, \ldots, n\} \tag{4.3}$$

An example of a guaranteed Voronoi diagram with disc-shaped regions is given in Fig. 4.2. As it can be observed from this figure, some of the points in the field do not belong to any guaranteed Voronoi region. In other words, unlike the conventional Voronoi diagram, the guaranteed Voronoi diagram does not partition the field. In this section, it is assumed that region D_i is a ball of radius of r_i centered at p_i. Then, the boundary shared between the guaranteed Voronoi region G_i and its neighboring region G_j is a set of points q satisfying the following condition:

$$\|q - p_i\| + r_i = \|q - p_j\| - r_j$$

where p_i and p_j are the centers and r_i and r_j are the radii of the circular regions of guaranteed Voronoi regions G_i and G_j, respectively. Since r_i and r_j are constant, the points on the boundary lie on the hyperbola arm with foci p_i and p_j, closer to p_i. In the special case when the circles' radii are both zero, the points on the boundary lie on the perpendicular bisector of $p_i p_j$.

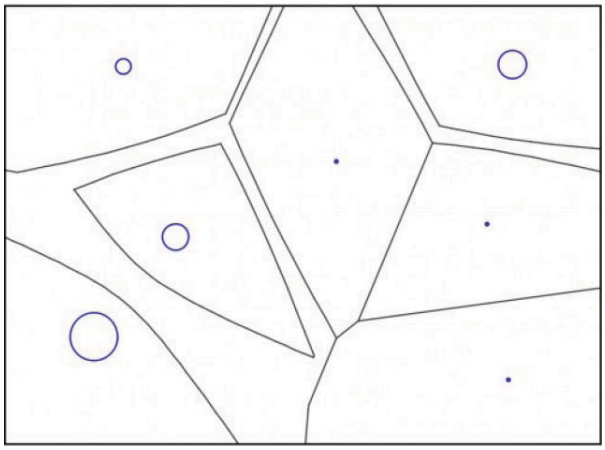


Figure 4.2: An example of the guaranteed Voronoi diagram for a group of 7 agents.

In general, for any region D_i containing the agent p_i, the guaranteed Voronoi region G_i is a subset of the corresponding conventional Voronoi region V_i (i.e., $G_i \subseteq V_i$). In the special case, when the region D_i has only one point, then G_i will be the same as the conventional Voronoi partition V_i.

4.4 Distributed Coverage Control in Faulty Situations

In this section, the coverage control problem subject to communication delays induced by faulty communication of a group of mobile agents is investigated. A motion coordination strategy for each agent is proposed based on the available information in faulty conditions.

4.4.1 Deployment Strategy

In this work, it is assumed that each agent is responsible for sensing or covering the points in its own Voronoi region. In the fault-free case, the exact location of agents is available, and coverage function $\mathcal{H}$ can be optimized accordingly. In the presence of communication delays between agents, on the other hand, the field is divided based on the guaranteed Voronoi diagram. As noted before, in such partitioning, there is some neutral region in the field. These points are not assigned to any agent, and hence cannot be covered. Therefore, there is no guarantee to find the global optimum for the coverage function $\mathcal{H}$ in the general case.

Although the guaranteed Voronoi diagram does not partition the field, one can still use the results of the *1-center* problem optimization to drive each agent towards its centroid. The optimization in each guaranteed Voronoi region G_i can be considered as the *1-center* problem optimization. Therefore, the centroid of the guaranteed Voronoi region G_i is the optimal destination for the agent i.

4.4.2 Motion Control

Each mobile agent is modeled as a *double-integrator* point mass moving in Q as follows:

$$\ddot{p}_i = u_i \tag{4.4}$$

where u_i is the control input of the i-th agent. It is assumed that each agent is able to compute its own guaranteed Voronoi region at every time step based on its last updated information vector. Subsequently, each agent calculates the centroid of its guaranteed Voronoi region and applies its control input to move towards its centroid. A control law of the following form is proposed for the i-th agent:

$$u_i = k_1^i M_{G_i}(C_{G_i} - p_i) - k_2^i \dot{p}_i \tag{4.5}$$

where k_1^i and k_2^i are some positive gains to be chosen by the designer. Also, M_{G_i} and C_{G_i} are the mass and centroid of the i-th guaranteed Voronoi region, respectively. Let $\mathcal{H}_{G_i} = \int_{G_i} \frac{1}{2} \|p_i - q\|^2 \varphi(q) dq$. Thus:

$$\begin{aligned} \frac{\partial \mathcal{H}_{G_i}}{\partial p_i} &= -\int_{G_i} (q - p_i)\varphi(q) dq \\ &= -M_{G_i}(C_{G_i} - p_i) \end{aligned} \tag{4.6}$$

Proposition 4.1. *The control input (4.5) can drive the i-th agent to the centroid of the guaranteed Voronoi region G_i by using $\mathcal{H}_{G_i}$.*

Proof. Consider the Lyapunov function candidate as:

$$\vartheta = k_1^i \mathcal{H}_{G_i} + \dot{p}_i \dot{p}_i^T$$

This function ϑ is lower-bounded by zero since φ and $\|.\|^2$ are both positive. By substituting (4.4) and (4.6) into the time derivative of ϑ along the system trajectory and using the control input (4.5), the time derivative of the above Lyapunov function can be obtained as follows:

$$\begin{aligned} \dot{\vartheta} &= k_1^i \frac{\partial \mathcal{H}_{G_i}}{\partial p_i} \dot{p}_i^T + \ddot{p}_i \ \dot{p}_i^T \\ &= \left(k_1^i M_{G_i} \left(p_i - C_{G_i} \right) + \ddot{p}_i \right) \dot{p}_i^T \\ &= -k_2^i \dot{p}_i \dot{p}_i^T \end{aligned}$$

which is clearly non-positive. Let $\mathcal{M}$ be the set of all points in G_i where $\dot{\vartheta} = 0$. Since the center of mass C_{G_i} lies in the interior of the region G_i, hence the i-th mobile agent always stays in the interior of G_i. This in turn means that $\mathcal{M}$ is a positive invariant set. Since this set is closed and bounded, one can use LaSalle's invariance principle to infer that the position of the i-th mobile agent converges to the largest invariant

subset of the set $\mathcal{M}$ (i.e. C_{G_i}). □

Remark 4.1. *The control input (4.5) of the i-th agent depends on its own position and centroid of its dominance region. For the computation of C_{G_i}, however, the positions of its neighboring guaranteed Voronoi regions are needed, and this shows the distributed nature of the proposed controller.*

Remark 4.2. *Due to the convexity of each guaranteed Voronoi region, the centroid of every region lies in its interior. Since each mobile agent moves towards the centroid of its associated guaranteed Voronoi region, inter-agent collision avoidance is guaranteed (even in the presence of inter-agent communication delays) if agents initially start their movement from a safe configuration.*

4.4.3 Motion Coordination Algorithm

The proposed motion coordination algorithm utilizes the motion control law with the updated information vector of agents. Every agent uses its information vector at each time step to construct its guaranteed Voronoi cell. The shape of each guaranteed Voronoi region and the size of neutral regions depend heavily on the communication delays between that agent and its neighboring agents. The higher communication delays, the larger neutral regions.

As mentioned in Section 4.2, the information vector of each agent contains the position of itself and those of its neighboring agents, as well as communication delays between the agent and its neighbors. The constructed guaranteed Voronoi cell and the motion coordination mechanism that each agent follows at every time step are presented in Algorithm 1.

According to Algorithm 1, agent i at each time step detects its neighboring agents and exchanges position information with them. The magnitudes of the communication delays of its faulty neighbors are subsequently obtained, and its information

Algorithm 1

1. At time step t_k, every agent $i \in \{1, \ldots, n\}$:
   i. finds the set of neighboring agents $\mathcal{N}_i$, and receives their positions p_j, $\forall j \in \mathcal{N}_i$;
   ii. updates its information vector $D^i = [p_i, (p_j^i, r_j^i)]$, $j \in \mathcal{N}_i$, and sets $D = D^i$;
   iii. constructs the guaranteed Voronoi region G_i and calculates its centroid C_{G_i};
   iv. computes the control input u_i, and applies it to the agent.
2. terminate the algorithm if $\mathcal{H}|_{t_k} - \mathcal{H}|_{t_{k-1}}$ is less than a predefined threshold; otherwise set $k = k + 1$ and go to Step 1.

vector is updated as described in Section 4.2. Then, agent i constructs its guaranteed Voronoi cell as explained in details in [34], and calculates its centroid in order to construct the control input (4.5), under which the agent moves towards C_{G_i}. At every time step all agents in the group follow the above steps simultaneously until the difference between the coverage performance $\mathcal{H}$ at time step t_k and t_{k-1} is less than a prescribed threshold.

In the next section, a problem similar to the one introduced in Section 4.1 is investigated where the sensor effectiveness of each agent varies during the mission. The sensor effectiveness factor of agents is incorporated in the locational optimization problem, and a novel partitioning technique is also introduced to address this problem. The inter-agent interaction in the presence of delay in this case is considered as the same as introduced in Section 4.2.

4.5 Coverage Control of Multi-Agent Systems subject to Communication Delay and Health Degradation

Analogous to (3.2), the cost of covering (or sensing) a point q by the i-th agent is denoted by $f_i(q, h_i)$, where $h_i \in [0, 1]$ is a sensor effectiveness factor for the i-th agent which represents varying health conditions. The sensor effectiveness factor $h_i = 1$ denotes the healthy i-th agent, and $h_i = 0$ corresponds to total failure in the sensor of i-th agent. As the distance of points from an agent increases, the cost of covering by the corresponding agent grows as well. Moreover, the sensing quality of the agents varies for various reasons (e.g., in the outdoor environment, weather conditions can degrade the quality of sensing). Variations of sensor effectiveness can lead to the variations of sensing quality. It is to be noted that while the function f_i is not necessarily the same for different agents, it is assumed to be identical for any set of points located at the same distance from the corresponding agent. In other words, $f_i(q_j, h_i) = f_i(q_k, h_i)$, for any $q_j, q_k \in Q$ such that $\|p_i - q_j\| = \|p_i - q_k\|$, where $\|.\|$ denotes the Euclidean norm.

It is desired that regardless of the inter-agent communication delay, the field is properly partitioned into n regions and the agents are relocated in the field such that each point in the field is covered by exactly one agent and the following coverage performance function is minimized:

$$\mathcal{H}(\mathbf{P}, \mathbf{W}) = \sum_{i=1}^{n} \int_{W_i} f_i(q, h_i)\varphi(q)dq \tag{4.7}$$

where the set $\mathbf{W} = \{W_1, W_2, \ldots, W_n\}$ represents a partition of the field Q, and the i-th agent is in charge of covering all points in the region W_i. Minimizing the above

coverage performance function implies maximizing the overall quality of covering. Furthermore, it is assumed that each mobile agent can communicate with other agents in its neighboring partition, and the set of all neighbors of agent i is denoted by $\mathcal{N}_i$.

If the functions f_i, and sensor effectiveness factor h_i are all the same for different agents (i.e., $h_i = h_j$ and $f_i(q, h_i) = f_j(q, h_j),\ \forall i, j \in \mathbf{n}$), then for a fixed set of agent locations $\mathbf{P}$ the conventional Voronoi diagram provides the optimal partitioning of the field [11], [39]. However, when f_i is agent-dependent, conventional Voronoi partitioning is no longer optimal. The problem of finding the optimal partitioning in this case can be very cumbersome, in general. However, it can be simplified significantly under a realistic assumption as follows:

$$f_i(q, h_i) = \frac{1}{h_i} \|p_i - q\|^2,\ \forall i \in \mathbf{n} \tag{4.8}$$

where h_i is pre-specified strictly positive coefficients. It is assumed that the magnitudes of the sensor effectiveness factor can be measured with a proper method. In fact, the h_i affects the performance of agent i's sensor. In the case all other variables are constant, a higher h_i results in a lower cost for the i-th agent and vice versa (e.g., in the case of fully healthy agent $h_i = 1$, the cost of covering is minimum and for the degraded health sensor this cost will increase). It is to be noted that when the i-th agent is fully faulty and its sensor effectiveness factor is zero, the cost of covering of any point in the field by the i-th agent would be infinity. Therefore, the i-th faulty agent can not incorporate in the group to cover the environment.

Regardless of communication delay, the coverage optimization problem with the cost functions (4.8) is investigated in Chapter 3 when the coefficients $\frac{1}{h_1}, \ldots, \frac{1}{h_n}$ are non-identical. In this case, it is shown that the center of mass of multiplicatively-weighted Voronoi (MW-Voronoi) region is the optimal configuration. The objective

here is to develop an algorithm to properly place the agents with varying health conditions in the field where individual agents receive the information about each others' locations with some delay caused by communication faults. Since the exact locations of some agents are not available in the faulty case, the MW-Voronoi partitioning cannot be used to optimize the coverage function $\mathcal{H}$. On the other hand, in the faulty situation agents can not precisely construct their regions, and communication delays can also result in overlap between some regions. Consequently, as the agents of the overlapped regions move to the center of mass of their regions, there is a chance that they collide. Hence, a specific partitioning technique is presented to divide the field among all agents in the next subsection.

4.5.1 Guaranteed Multiplicatively-Weighted Voronoi Partitioning

Let D be a set of regions in Q, denoted by $D = \{D_1, D_2, \ldots, D_n\}$, and each region contains exactly one weighted node. It is worth to mention that the weighted distance is defined in Subsection 3.2.1. The GMW-Voronoi region GMW_i, associated with node S_i is the locus of all points that are closer to this node than to any other nodes, i.e.:

$$GMW_i = \{q \in Q \mid \max_{S_i \in D_i} d_w(q, S_i) \leq \min_{S_j \in D_j} d_w(q, S_j), \forall j \neq i\}, \; i \in \mathbf{n} \quad (4.9)$$

An example of a GMW-Voronoi diagram for 9 nodes with arbitrary weights and with disc-shaped regions is given in Fig. 4.3. As it can be observed from this figure, some of the points in the field that are shown by dark blue do not belong to any GMW-Voronoi region. In other words, unlike the conventional Voronoi diagram (or MW-Voronoi diagram), the GMW-Voronoi diagram does not partition the field.

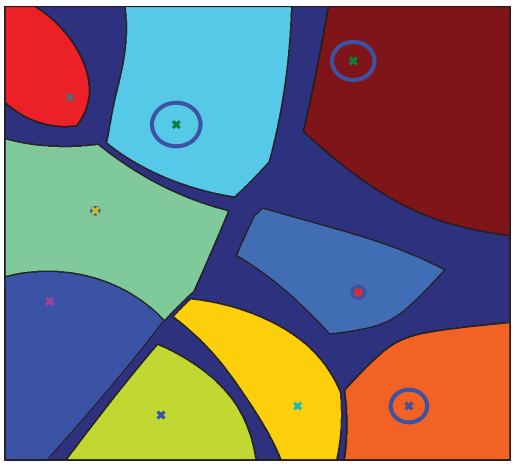


Figure 4.3: An example of the GMW-Voronoi diagram for a group of 9 agents with different weights.

In this section, it is assumed that region D_i is a ball of radius of r_i centered at p_i, and the weight of w_i corresponds to the i-th agent. Then, the boundary shared between the GMW-Voronoi region GMW_i and its neighboring region GMW_j is a set of points q satisfying the following condition:

$$d_w(q, p_i) + r_i = d_w(q, p_j) - r_j$$

where p_i and p_j are the centers and r_i and r_j are the radii of the circular regions of GMW-Voronoi regions GMW_i and GMW_j, respectively. In the special case when the circles' radii are both zero, the points on the boundary lie on the Apollonian circles $\Omega_{p_i p_j, \frac{w_i}{w_j}}$ (the locus of all points q such that $\frac{p_i q}{p_j q} = \frac{w_i}{w_j}$ [71]). In general, for any region D_i containing the agent p_i with weight of w_i, the GMW-Voronoi region GMW_i is a subset of the corresponding conventional MW-Voronoi region. In the special case, when the region D_i has only one point, then GMW_i will be the same as the conventional MW-Voronoi partition.

As mentioned in Section 3.2 in the previous chapter, in the case there is no inter-agent communication delays and when the coefficients $\frac{1}{h_1}, \ldots, \frac{1}{h_n}$ in

$\mathcal{H}$ are not identical, the MW-Voronoi diagram of the set of weighted nodes $\{(p_1, \sqrt{h_1}), (p_2, \sqrt{h_2}), \ldots, (p_n, \sqrt{h_n})\}$ is the optimal partitioning of Q for the fixed agent positions $\{p_1, p_2, \ldots, p_n\}$. According to this result when the network of agents is subject to communication delay, the same set of weighted nodes here is used for the GMW-Voronoi diagram (weight of $\sqrt{h_i}$ for the i-th agent).

4.5.2 Motion Control

It is assumed that each agent is responsible for sensing or covering the points in its own dominance region. In the presence of communication delays between agents and varying health conditions, the field is divided based on the GMW-Voronoi diagram. As noted before, in such partitioning, there is some neutral region in the field. These points are not assigned to any agent, and hence cannot be covered. Therefore, there is no guarantee to find the global optimum for the coverage function $\mathcal{H}$ in the general case. Although the GMW-Voronoi diagram does not partition the field, one can still use the results of the *1-center* problem optimization to drive each agent towards its centroid. The optimization in each GMW-Voronoi region GMW_i can be considered as the *1-center* problem optimization. Therefore, for the optimization problem $min\mathcal{H}$ with the cost functions (4.8), the configuration $CGMWV_n(\sqrt{h_1}, \sqrt{h_2}, \ldots, \sqrt{h_n})$ can be considered as the candidate destination for the agents to move towards it.

Suppose each agent in the team is modeled as a *single-integrator* point mass moving in Q as follows:

$$\dot{p}_i = u_i \tag{4.10}$$

It is assumed that each agent is able to compute its own GMW-Voronoi region at every time step based on its last updated information vector. Subsequently, each agent calculates the centroid of its GMW-Voronoi region and applies its control input to move towards its centroid. A control law of the following form is proposed for the

i-th agent:

$$u_i = k_1^i(C_{GMW_i} - p_i) \tag{4.11}$$

where C_{GMW_i} is the center of mass of the i-th GMW-Voronoi region, and k_1^i is positive gain to be chosen by the designer. Similar to Proposition 4.1, one can prove that the control input (4.11) will drive the i-th agent to the centroid of the GMW-Voronoi region GMW_i.

Remark 4.3. *The GMW-Voronoi regions are mutually disjoint, and the centroid of every region lies in its interior. Since each mobile agent moves towards the centroid of its associated GMW-Voronoi region, inter-agent collision avoidance is guaranteed (even in the presence of inter-agent communication delays) if agents initially start their movement from a safe configuration.*

Using the motion control law and the updated information vector of agents, a motion coordination algorithm can be proposed analogously to Algorithm 1. Every agent uses its information vector at each time step to construct its GMW-Voronoi cell. The shape of each GMW-Voronoi region and the size of neutral regions depend heavily on the inter-agent communication delays, and the weights of agents correspond to their health conditions. As mentioned in Section 4.2, the information vector of each agent contains the position of itself and those of its neighboring agents, as well as communication delays between the agent and its neighbors. The constructed GMW-Voronoi cell and the motion coordination mechanism that each agent follows at every time step are presented in Algorithm 2.

According to Algorithm 2, agent i at each time step detects its neighboring agents and exchanges position information and sensor effectiveness factor with them. The magnitudes of the communication delays of its faulty neighbors are subsequently obtained, and its information vector is updated as described in Section 4.2. Then, agent

Algorithm 2

1. At time step t_k, every agent $i \in \{1, \ldots, n\}$:
   i. finds the set of neighboring agents $\mathcal{N}_i$, and receives their positions p_j, and sensor effectiveness factors h_j $\forall j \in \mathcal{N}_i$;
   ii. updates its information vector $D^i = [p_i, (p_j^i, r_j^i)]$, $j \in \mathcal{N}_i$, and sets $D = D^i$;
   iii. constructs the GMW-Voronoi region GMW_i and calculates its centroid C_{GMW_i};
   iv. computes the control input u_i, and applies it to the agent.
2. Terminate the algorithm if $\mathcal{H}|_{t_k} - \mathcal{H}|_{t_{k-1}}$ is less than a predefined threshold; otherwise set $k = k + 1$ and go to Step 1.

i constructs its GMW-Voronoi cell, and calculates its centroid in order to construct the control input (4.11), under which the agent moves towards C_{GMW_i}.

4.6 Simulation Results

In this section, the effectiveness of the proposed distributed agent-deployment algorithms are demonstrated by simulations in two different scenarios. The environment used in this simulation is a 10m $\times$ 10m square. In the first scenario, a group of nine agents deployed to cover the environment. The maximum velocity of agents is $v_{max} = 2\ m/s$, with faulty communication channels for agents 3 and 5, generating delays of $\tau_3 = 0.5\ s$ and $\tau_5 = 0.8\ s$ between these agents and their neighbors. Moreover, the field coverage is prioritized by the following bimodal Gaussian function:

$$\varphi(q) = \frac{1}{\sigma\sqrt{2\pi}} \left(e^{-\frac{(q-\mu_1)^2}{2\sigma^2}} + e^{-\frac{(q-\mu_2)^2}{2\sigma^2}} \right) \tag{4.12}$$

where $\mu_1 = (2, 2)$, $\mu_2 = (8, 8)$, and $\sigma = 1\ m$. The mobile agents start their move from the area indicated in the top-left corner of the field, as shown in Fig. 4.4. Agents 3 and 5 are assumed to be faulty agents, which are respectively shown in red and blue.

Furthermore, the priority function $\varphi(q)$ (which is concentrated in two disjoint areas) is shown in gray in the figure, with a color density proportional to the value of the function. The final configuration of agents along with their trajectories under the proposed deployment Algorithm 1 are shown in Fig. 4.5.

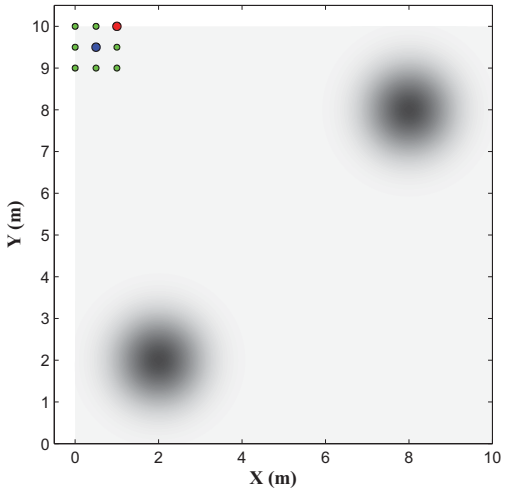


Figure 4.4: The initial positions of the nine agents in the first scenario.

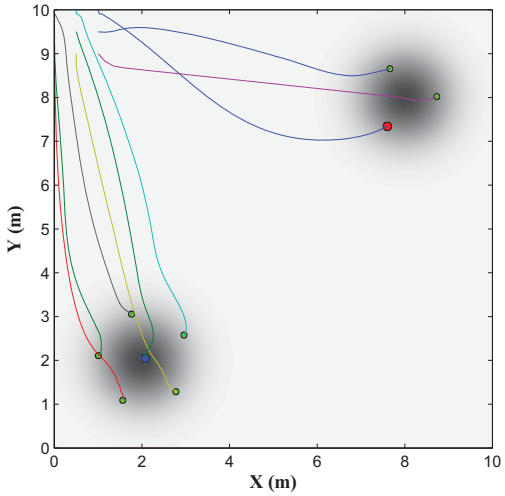


Figure 4.5: The trajectories of agents and their final positions under the proposed deployment algorithm in the first scenario.

One of the important properties of a deployment algorithm based on the guaranteed Voronoi diagram is inter-agent collision avoidance. However, this is not necessarily the case with other partitioning methods. To demonstrate this, the simulation is also performed with conventional Voronoi partitioning. In this case, each agent constructs its Voronoi polygon based on the latest position information received from the neighboring agents, and moves towards its centroid. It is to be noted that in the presence of the communication delays (for faulty agents), the current position of a faulty agent is different from the position information received by its neighbors. The distance between the blue agent (faulty agent) and all other agents using both partitioning is depicted in Figs. 4.6 and 4.7. As it can be observed from the curves in these figures, the faulty agent collides with one of the agents when the conventional Voronoi diagram is utilized in the agent deployment procedure but no collision happens when the guaranteed Voronoi partitioning is used instead.

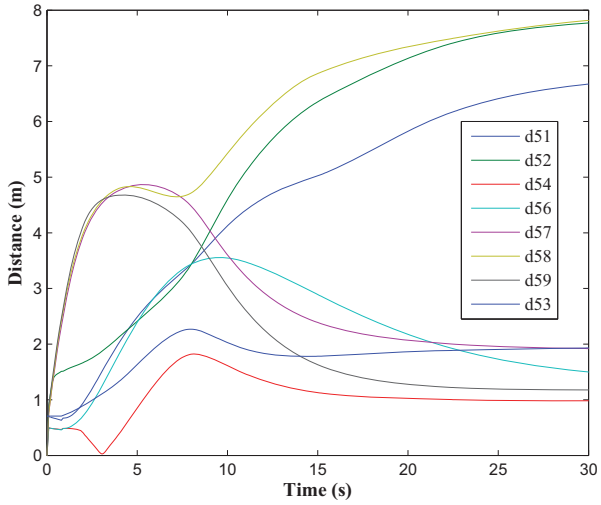


Figure 4.6: The distances between the blue agent and other agents in the conventional Voronoi partitioning.

As mentioned in Section 4.3, there are some neutral regions in the field when the guaranteed Voronoi partitioning is used. Thus, one cannot compare the coverage performance $\mathcal{H}$ obtained by conventional Voronoi partitioning with that computed

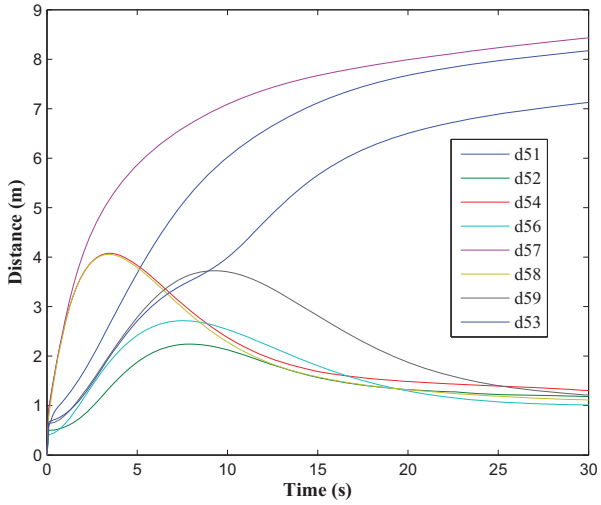


Figure 4.7: The distance between the blue agent and other agents in the guaranteed Voronoi partitioning.

based on the guaranteed Voronoi diagram. Hence, in order to compare the coverage performance, at each time step after updating the position of every agent using Algorithm 1 with the guaranteed Voronoi diagram, the field is partitioned by the conventional Voronoi polygons, and the coverage performance $\mathcal{H}$ is measured w.r.t this diagram. The result obtained is compared with the coverage performance measured based on the conventional Voronoi partitioning, as depicted in Fig. 4.8. The figure demonstrates that the field is covered more effectively when the guaranteed Voronoi partitioning is used in the presence of faulty agents.

In the second scenario, a group of eight single-integrator agents is considered in the same field as before. All agents start their move from the border of the environment, and the sensor effectiveness factors $h_1, h_2, \ldots, h_8$ in $\mathcal{H}$ are chosen as $1, 0.25, 0.2, 0.66, 0.33, 0.1, 1, 0.66$, respectively. It is also assumed that the communication channels of agents 3 and 7 are faulty, generating delays of $\tau_3 = 0.5\ s$ and $\tau_5 = 0.25\ s$ between these agents and their neighbors. The field is prioritized again by the bimodal Gaussian density function (4.12) in where $\mu_1 = (2, 8)$, $\mu_2 = (8, 2)$,

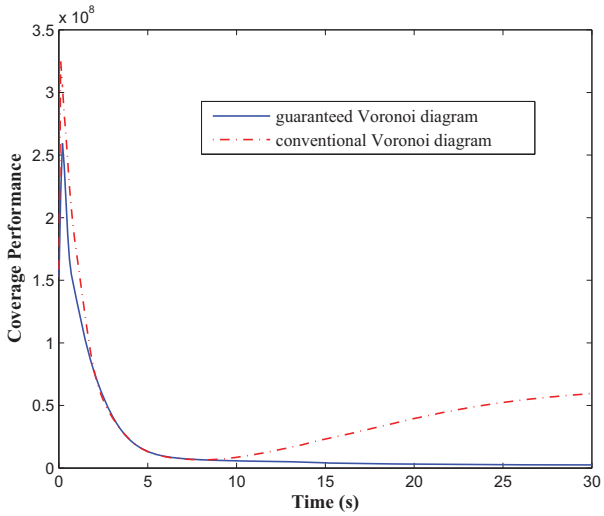


Figure 4.8: The coverage performance $\mathcal{H}$ obtained by using the proposed deployment strategy (solid curve), and the conventional Voronoi partitioning in the first scenario.

and $\sigma = 0.6\ m$. The initial configuration of agents, their trajectories and final configuration are depicted in Fig. 4.9. In this figure, the initial and final position of agents are shown by $\times$ and $\circ$ markers, respectively. As expected, the agents cover the most important area.

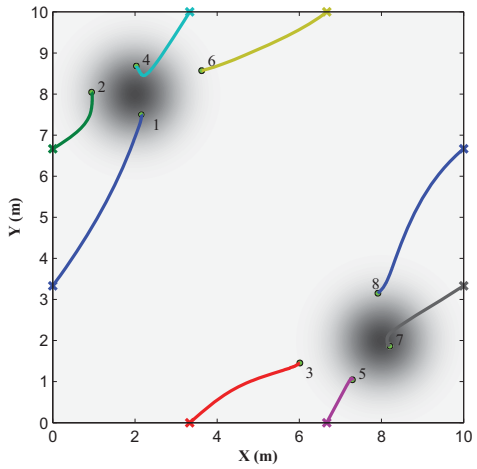


Figure 4.9: The initial and final positions of agents in the second scenario along with their trajectories under the proposed deployment Algorithm 2.

As mentioned previously, there are some neutral regions in the field when the

GMW-Voronoi partitioning is used. Thus, one cannot compare the coverage performance $\mathcal{H}$ obtained by conventional MW-Voronoi partitioning with that computed based on the GMW-Voronoi diagram. Hence, in order to compare the coverage performance, at each time step after updating the position of every agent using Algorithm 2 with the GMW-Voronoi diagram, the field is partitioned by the conventional MW-Voronoi polygons, and the coverage performance $\mathcal{H}$ is measured w.r.t. this diagram. The weights of all agents which correspond to h_i's coefficients in $\mathcal{H}$ are the same in the both partitioning techniques. The result obtained is compared with the coverage performance measured based on the conventional MW-Voronoi partitioning, as depicted in Fig. 4.10. The figure demonstrates that the field is covered more effectively when the GMW-Voronoi partitioning is used in the presence of faulty agents.

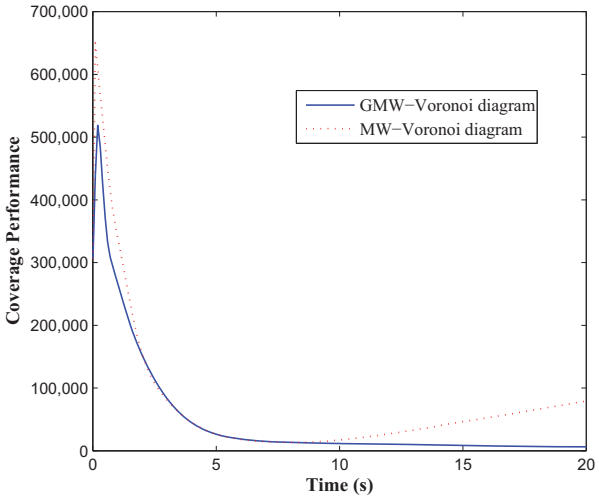


Figure 4.10: The coverage performance $\mathcal{H}$ obtained by using the proposed deployment strategy (solid curve), and the conventional MW-Voronoi partitioning in the second scenario.

Chapter 5

Coverage Strategy with Guaranteed Collision Avoidance in Multi-Agent Systems

There is a substantial body of research works addressing collision and obstacle avoidance in multi-agent systems [42–44]. However, few works have been reported on cooperative coverage control in multi-agent systems. An algorithm is presented in [45] to dynamically cover a field with guaranteed collision avoidance. In [46] a Voronoi-based coverage control is proposed for a group of wheeled mobile robots with a dynamic constraint, where a collision avoidance term is also incorporated in the kinematic controller using potential functions. However, none of these works addresses the problem of obstacle avoidance in multi-agent systems.

This chapter aims to develop an effective coverage technique in a multi-agent system with both collision avoidance and obstacle avoidance properties. The proposed technique employs a navigation function which consists of two main components. One of these components aims to drive the agent to the desired location, while the other

component repels the agents from obstacles and from each other. A priority function is also defined for different points in the field to prioritize different points in the field in terms of coverage.

5.1 Problem Formulation

Similar to the problem in previous chapters, let q be an arbitrary point in a convex field in $Q \subset R^2$. Consider a group of n mobile agents whose dynamic is single-integrator (4.10). The agents are randomly distributed in Q, and the position of the i-th agent is denoted by p_i. Let $\varphi : Q \to R_+$ be a prescribed priority function that assigns a weight to every point in the field, reflecting the relative importance of different points in Q. Moreover, there exist M obstacles in the field which are modeled as fixed points $w_1, \ldots, w_M$ [79].

A collision region is also defined for agent i, $i \in \mathbf{n} := \{1, 2, \ldots, n\}$, such that any other agent or obstacle inside a circle of radius R_i around the agent is considered as an obstacle. The set of obstacles of agent i is defined as:

$$\mathcal{C}_i = \{\|p_i - p_j\| \leq R_i, \forall j \neq i\} \cup \{\|p_i - w_k\| \leq R_i, k = 1, \ldots, M\} \tag{5.1}$$

Assume for simplicity that the radius of the collision region is the same for different agents, i.e. $R_i = R_{col}$, where R_{col} is a given positive value. Fig. 5.1 shows an example of collision region of the i-th agent in the presence of obstacles.

Let $V = \{V_1, V_2, \ldots, V_n\}$ be the Voronoi partition of Q, where the agent positions are the generator points. Agents i and j are called neighbors if $V_i \cap V_j \neq 0$ (i.e., they share an edge). The set of all neighbors of agent i is denoted by $\mathcal{N}_i$. In this chapter, it is assumed that the cost of covering (or sensing) a point q by the i-th agent is $\|q - p_i\|^2$. As the distance of points from an agent increases, the cost of covering

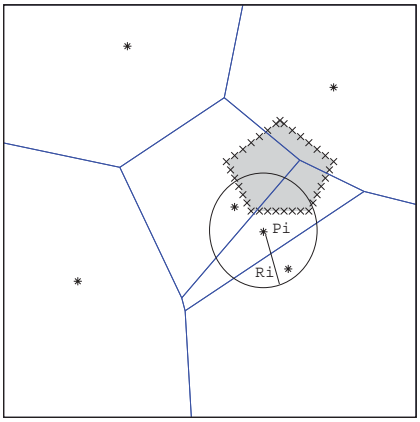


Figure 5.1: The collision region of the i-th agent in the presence of obstacles, and its Voronoi region.

of those points by the corresponding agent grows as well. It is also assumed that each agent is in charge of covering all points of its own Voronoi region. Since the whole field is partitioned by all Voronoi regions, the coverage performance function is defined as:

$$\mathcal{H}(\mathcal{P}) = \sum_{i=1}^{n} \int_{V_i} \|q - p_i\|^2 \, \varphi(q) dq \tag{5.2}$$

where $\mathcal{P}$ is the set of all agent positions ($\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$).

In Chapter 2, it shown that the centroidal Voronoi configuration, the configuration in which each agent is placed in the center of mass of its corresponding Voronoi region, is the optimal for the case when cost of coverage is a quadratic function. In other words, the centroidal Voronoi configuration is the optimal solution for the coverage function (5.2). Hence, this results will be used in the following section to develop a distributed control strategy for driving the agents such that inter-agent collision and obstacle avoidance are guaranteed.

5.2 Navigation Functions

The navigation function is widely used in the literature [43, 80, 81]. Define the following distributed navigation function $\phi_i : Q \to [0, 1]$ for agent i:

$$\phi_i = \frac{\gamma_i}{\left(\gamma_i^k + \beta_i\right)^{1/k}} \tag{5.3}$$

where $\gamma_i : Q \to R_+$ is a positive semi-definite scalar function called the *goal function* which vanishes only when the i-th agent reaches its desired position; $\beta_i : Q \to [0, 1]$ is called the *collision avoidance function* and vanishes only when the i-th agent is in contact with the obstacles or other agents, and $k \in R_+$ is a tuning parameter. The maximum value of the navigation function is 1 (achieved when $\beta_i = 0$) and the minimum value is 0 (achieved when $\gamma_i = 0$). The navigation function ϕ_i is required to be [82]:

- analytic on Q, (at least a $\mathcal{C}^2$ function);
- admissible on Q, i.e. it is uniformly maximum on the obstacles' boundary;
- polar on Q, i.e. reaches its unique minimum only if the i-th agent is at its desired position, and
- a Morse function, i.e. the critical points of ϕ_i are non-degenerate.

In what follow, the goal and collision avoidance functions are further elaborated upon.

5.2.1 The Goal Function

The goal function γ_i in (5.3) reflects the control objective of agent i, which is minimized once the desired objective with respect to this particular agent is fulfilled. As mentioned in Section 5.1, the optimal configuration to cover the most important area

of the field for the quadratic coverage cost is the centroidal Voronoi configuration. Then, at every time instant the desired destination of each agent is the centroid of its corresponding Voronoi region. Let the goal function of agent i be defined as:

$$\gamma_i = \|p_i - C_{V_i}\|^2 \tag{5.4}$$

It is clear that the above function has a unique minimum at its desired position.

5.2.2 The Collision Avoidance Function

The collision avoidance function β_i in (5.3) is defined as:

$$\beta_i = \prod_{l \in \mathcal{C}_i} \beta_{il} \tag{5.5}$$

where β_{il} is the mutual avoidance coefficient between agent i and agent or obstacle l. The collision avoidance function β_i guarantees collision avoidance for agent i by monitoring agents and obstacles located within its collision region at every time instant. Specifically, the collision avoidance function β_i is designed in such a way that it vanishes whenever agent i touches a fixed obstacle or another agent. The function β_{il} is chosen as a sigmoidal function below:

$$\beta_{il} = \frac{1}{1 + e^{-(\|p_{il}\| - \frac{R_{col}}{2})\frac{12}{R_{col}}}} \tag{5.6}$$

where $\|p_{il}\| = \|p_i - p_l\|$, and R_{col} is the radius of collision region of the i-th agent. This function ensures that agent i is repulsed from other agents or obstacles to prevent collision. As shown in Fig. 5.2, the function β_{il} has a unique minimum at collision ($\|p_{il}\| = 0$), and reaches its maximum 1 for $\|p_{il}\| > R_{col}$.

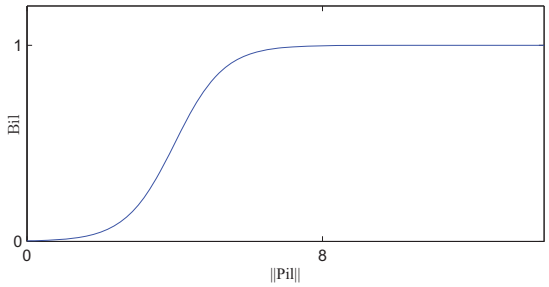


Figure 5.2: An example of the collision function $\beta_{il}(\|p_{il}\|)$, with $R_{col} = 8$.

The navigation function is used to control each agent to satisfy the desired objective and to avoid the obstacles. Every agent sends its position information to its neighboring agents and then constructs its Voronoi region at each time instant. Then, it calculates the mass center of its Voronoi region, and is driven towards it by its local control input. To coordinate the mobile agents for covering an environment while avoiding collision, the controller for the i-th agent is designed as:

$$u_i = -\alpha \nabla_{p_i} \phi_i \tag{5.7}$$

where α is a positive scalar gain, and $\nabla_{p_i}\phi_i$ is the gradient of ϕ_i with respect to p_i. In the next section, some important properties of the distributed controller and navigation function are presented.

5.3 Convergence Analysis

In what follows, the convergence of the overall system is investigated.

Consider a Lyapunov function candidate $\mathcal{V}(\mathbf{P}) = \sum_{i=1}^{N} \phi_i$, where $\mathbf{P} = [p_1, ..., p_N]^T$ is the states of all agents. Since dynamics of all agents are single-integrator and using

(5.7), the time derivative of $\mathcal{V}$ is computed as:

$$\dot{\mathcal{V}} = (\nabla \mathcal{V})^T \dot{\mathbf{P}} = -\alpha \sum_{i=1}^{N} \sum_{j=1}^{N} (\nabla_{p_i} \phi_i)^T (\nabla_{p_i} \phi_j)$$

Clearly, the largest invariant set for which $\dot{V} = 0$ is the set of critical points. To prove the stability, it is required to show that $\dot{V}$ is negative whenever at least one agent is not located in its critical point. Assume the i-th agent is not located in its critical point (i.e. $\nabla_{p_i} \phi_i \neq 0$). Then $\dot{V}$ can be rewritten as:

$$\dot{V} = -\alpha \sum_{i:\nabla_{p_i}\phi_i \neq 0} \left(\|\nabla_{p_i} \phi_i\|^2 + \sum_{j \neq i} (\nabla_{p_i} \phi_i)^T (\nabla_{p_i} \phi_j) \right) \tag{5.8}$$

It is straightforward to verify that $\dot{V}$ in (5.8) is negative definite when $\sum_{j \neq i} (\nabla_{p_i} \phi_i)^T (\nabla_{p_i} \phi_j) > 0$. The gradient of the navigation function (5.3) can be easily obtained as:

$$\nabla_{p_i} \phi_i = \frac{1}{(\gamma_i^k + \beta_i)^{\frac{1}{k}+1}} (\beta_i \nabla_{p_i} \gamma_i - \frac{\gamma_i}{k} \nabla_{p_i} \beta_i)^T \tag{5.9}$$

By substituting (5.9) into $\sum_{j \neq i} (\nabla_{p_i} \phi_i)^T (\nabla_{p_i} \phi_j)$ one arrives at:

$$\frac{(\beta_i \nabla_{p_i} \gamma_i - \frac{\gamma_i}{k} \nabla_{p_i} \beta_i)^T}{(\gamma_i^k + \beta_i)^{\frac{1}{k}+1}} \left(\sum_{j \neq i} \frac{\beta_j \nabla_{p_i} \gamma_j - \frac{\gamma_j}{k} \nabla_{p_i} \beta_j}{(\gamma_j^k + \beta_j)^{\frac{1}{k}+1}} \right) > 0 \tag{5.10}$$

Since γ_i and β_i are positive functions and cannot both be zero simultaneously, thus the above expression is positive if:

$$\left(\beta_i \nabla_{p_i} \gamma_i - \frac{\gamma_i}{k} \nabla_{p_i} \beta_i \right)^T \left(\sum_{j \neq i} (\beta_j \nabla_{p_i} \gamma_j - \frac{\gamma_j}{k} \nabla_{p_i} \beta_j) \right) > 0, \tag{5.11}$$

The above inequality can be rewritten as:

$$\frac{1}{k^2}a_1 + \frac{1}{k}a_2 + a_3 > 0 \tag{5.12}$$

where a_1, a_2 and a_3 are defined as follows:

$$\begin{aligned}
a_1 &= \gamma_i (\nabla_{p_i}\beta_i)^T \sum_{j\neq i} \gamma_j \nabla_{p_i}\beta_j \\
a_2 &= -\beta_i (\nabla_{p_i}\gamma_i)^T \sum_{j\neq i} \gamma_j \nabla_{p_i}\beta_j - \gamma_i (\nabla_{p_i}\beta_i)^T \sum_{j\neq i} \beta_j \nabla_{p_i}\gamma_j \\
a_3 &= \beta_i (\nabla_{p_i}\gamma_i)^T \sum_{j\neq i} \beta_j \nabla_{p_i}\gamma_j
\end{aligned} \tag{5.13}$$

In order to prove the inequality (5.12) holds, assume first that $a_3 < 0$. Hence, the polynomial (5.12) is positive for:

$$k > \frac{-a_2 + \sqrt{a_2^2 - 4a_1a_3}}{2a_3}$$

On the other hand, for any $a_3 \geq 0$ the inequality (5.12) is positive if $k > \frac{-a_1}{|a_2|}$. Finally, it can be concluded that if $k > \max\left\{\frac{-a_2+\sqrt{a_2^2-4a_1a_3}}{2a_3}, \frac{-a_1}{|a_2|}\right\}$, then the overall system converges to the set of critical points. This result is formally presented in the following theorem.

Theorem 5.1. *Consider a group of n mobile agents whose dynamic models are single-integrator. Under the control law (5.7), the overall system is stable and converges to the largest set of critical points* $C = \{q|\nabla_{p_i}\phi_i|_q = 0\}$ *if the tuning parameter in (5.3) satisfies the inequality* $k > \max\left\{\frac{-a_2+\sqrt{a_2^2-4a_1a_3}}{2a_3}, \frac{-a_1}{|a_2|}\right\}$.

Proposition 5.1. *The navigation function is minimized at the centroid of the Voronoi regions.*

Proof. If the Hessian of ϕ_i, $i \in \mathbf{n}$, is positive definite at a critical point, then the

navigation function ϕ_i would be minimized at this point. The i-th goal function and its gradient at the centroid of the corresponding Voronoi region are zero ($\gamma_i|_{C_{V_i}} = 0; \nabla_{p_i}\gamma_i|_{C_{V_i}} = 0$). By substituting these equalities in the gradient of the navigation function (5.9), one can conclude that $\nabla_{p_i}\phi_i|_{C_{V_i}} = 0$. Thus, the centroid of the Voronoi region C_{V_i} is a critical point of ϕ_i. The Hessian of φ_i, on the other hand, is:

$$\begin{aligned}\nabla^2_{p_i}\phi_i = \frac{1}{k(\gamma_i^k+\beta_i)^{1/k+2}}\{&(\gamma_i^k+\beta_i)[k\nabla_{p_i}\beta_i\nabla_{p_i}{\gamma_i}^T]\} \\ &- \nabla_{p_i}\gamma_i(\nabla_{p_i}\beta_i)^T + k\beta_i\nabla^2_{p_i}\gamma_i - \gamma_i\nabla^2_{p_i}\beta_i \\ &- \left(\frac{1}{k}+1\right)[k\beta_i\nabla_{p_i}\gamma_i - \gamma_i\nabla_{p_i}\beta_i][k\gamma_i^{k-1}\nabla_{p_i}\gamma_i + \nabla_{p_i}\beta_i]^T\} \quad (5.14)\end{aligned}$$

At the centroid of the i-th Voronoi region the Hessian becomes $\nabla^2_{p_i}\phi_i|_{C_{V_i}} = \beta_i^{-\frac{1}{k}}\nabla^2_{p_i}\gamma_i$. Since the collision avoidance function and the Hessian of the goal function at centroid are both positive ($\beta_i > 0$ and $\nabla^2_{p_i}\gamma_i|_{C_{V_i}} = 2$), thus the Hessian matrix becomes $\nabla^2_{p_i}\phi_i|_{C_{V_i}} = 2\beta_i^{-\frac{1}{k}}$ which is clearly positive definite. Therefore, the navigation function is minimized at the centroidal Voronoi configuration. □

The following proposition follows directly from the *dual Lyapunov analysis* in [83].

Proposition 5.2. *The attractors of the undesirable critical points of the navigation function are sets of measure zero.*

Remark 5.1. *It was shown in Theorem 5.1 that the closed-loop system converges to the largest set of critical points. On the other hand, it is inferred from Propositions 5.1 and 5.2 that the set of centroids of the Voronoi regions is the only set of stable critical points. Then, the overall system under the control law (5.7) is asymptotically stable for almost all stable initial positions.*

5.4 Simulation Results

In this section, the effectiveness of the proposed distributed agent-deployment algorithm is demonstrated by simulations.

The environment used in this simulation is a 10m $\times$ 10m square. Two different scenarios are considered in this section. In the first scenario, a group of five agents with single-integrator dynamics starts moving from up-left corner of the environment. Consider the following Gaussian density function:

$$\varphi(q) = \frac{1}{\sigma\sqrt{2\pi}} \left(e^{-\frac{(q-\mu)^2}{2\sigma^2}} \right) \tag{5.15}$$

where $\mu = (7.5, 3.5)$ and $\sigma = 0.6$. The radius of the collision regions is $R_{col} = 0.8$, and the tuning parameter in the navigation function (5.3) is $k = 1$.

A fixed square obstacle also exists in this scenario, which is depicted in black in Figs. 5.3 and 5.4. Moreover, each agent is shown by a circle, and the Gaussian density function $\varphi(.)$ is shown in gray, with a color intensity proportional to the value of the function. The initial configuration of the agents is shown in Fig. 5.3. Each agent computes its Voronoi region and moves towards its center of mass under the control input (5.7). The trajectories of the agents along with their final positions are shown in Fig. 5.4. Furthermore, the coverage function $\mathcal{H}$ resulted from the proposed controller is depicted in Fig. 5.5, which demonstrates a rapid drop to a small neighborhood of zero.

One of the advantages of the proposed navigation function-based coordination algorithm is the inter-agent collision avoidance. This characteristic is very important in a real-world application when each mobile vehicle has a reasonable size and collision can be very expensive. The distance among all the agents throughout the simulation is depicted in Fig. 5.6. As it can be seen from this figure, these distances are always

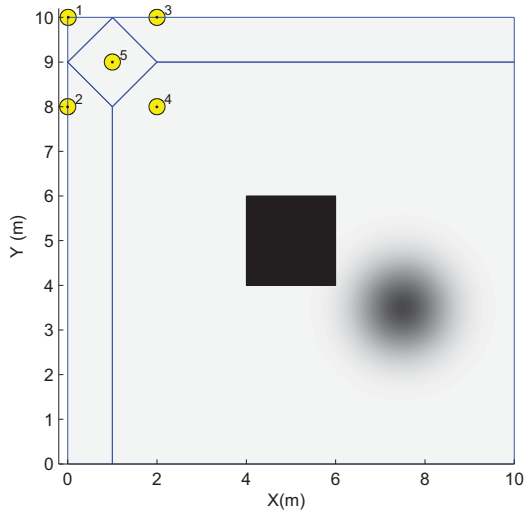


Figure 5.3: The initial positions of the five agents along with their corresponding Voronoi regions in the first scenario.

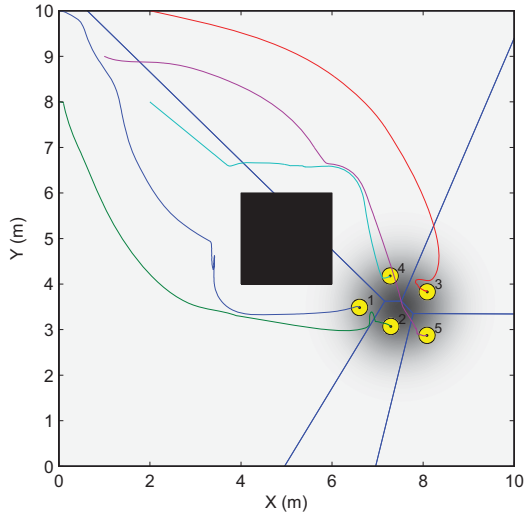


Figure 5.4: The trajectories and final positions of the agents along with their Voronoi regions in the first scenario.

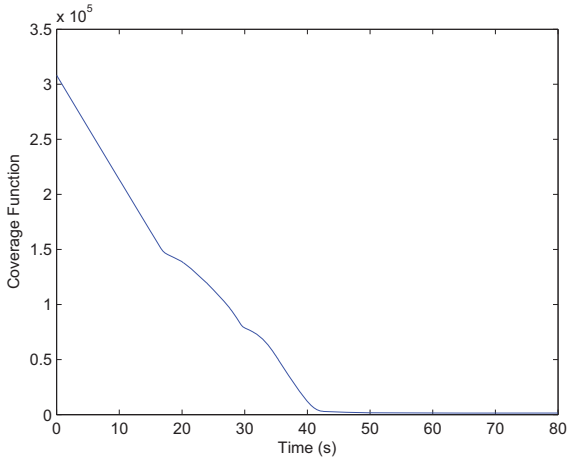


Figure 5.5: The coverage function $\mathcal{H}$ for the first scenario.

greater than the radius of the collision region of agents $R_{col} = 0.8$.

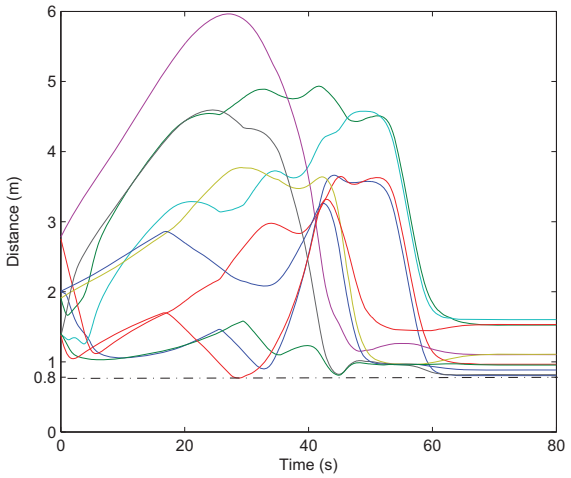


Figure 5.6: Distances between every pair of agents in the first scenario.

In the second scenario, a group of nine agents with random initial positions is considered in the same field as before. The priority function is the same as the first scenario (5.15) with center $\mu = (5, 5)$. Moreover, two fixed obstacles exist in the environment. The initial configuration of agents, their trajectories and final configuration are depicted in Figs. 5.7 and 5.8. As shown in Fig. 5.8, the agents cover the most important area while they avoid obstacles. For comparison, simulations

are also carried out using the method introduced in [70], where the visibility-aware multiplicatively-weighted Voronoi diagram (VMW-Voronoi) is used for partitioning the field in the presence of obstacles. The coverage function values in the second scenario resulted from the proposed controller and the VMW-Voronoi-based method are shown in Fig. 5.9. These curves show that, the coverage function resulted from the controller proposed in the present work drops more rapidly and has a smaller final value.

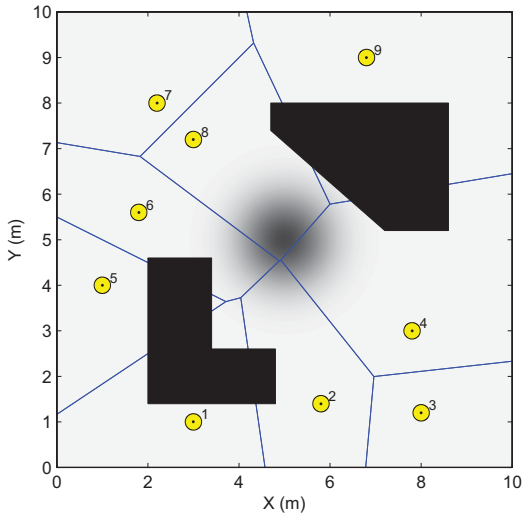


Figure 5.7: The initial positions of the nine agents along with their corresponding Voronoi regions in the second scenario.

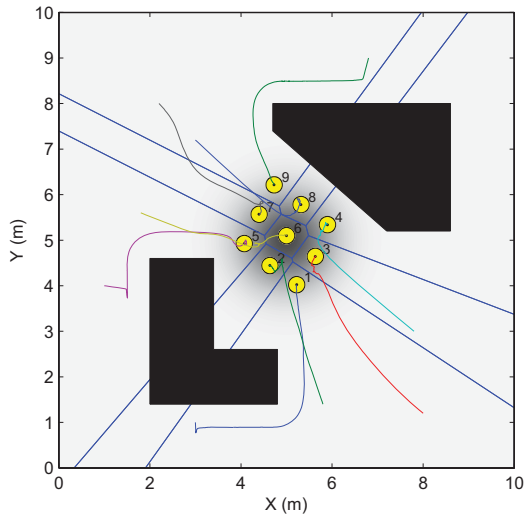


Figure 5.8: The trajectories and final positions of the agents along with their corresponding Voronoi regions in the second scenario.

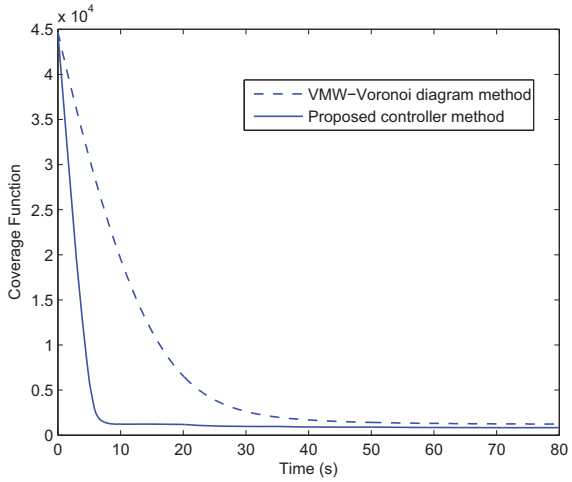


Figure 5.9: The coverage function $\mathcal{H}$ in the second scenario using the proposed control strategy and the VMW-Voronoi-based method.

Chapter 6

Coverage Control of Multi-Agent Systems in Uncertain Environments

In this chapter, it is considered the case in which some service vehicles are deployed to cover an uncertain environment. They are expected to spread out over the environment while aggregating in areas of high service needs. Furthermore, the service vehicles have uncertain information on the exact areas of service needs beforehand. In order to reduce the level of uncertainty, the environment is searched by some search vehicles which are equipped with sensors to detect the exact areas of service needs. As mission goes on, the service vehicles use the updated information provided by the search vehicles to change their position and cover the environment more efficiently [84].

Most of the previous works in the area of Voronoi-based coverage control assume the distribution of sensory information in the environment is required to be known *a priori* by all agents. However, the problem of the online learning of the priority

function, and estimation of priority function using neural networks while moving towards the optimal locations is addressed respectively in [15], [63]. In another work [85], local interpolations are used to represent spatial fields as they are measured by a mobile sensor network which are able to take point measurements. A nonparametric estimate of the field is provided by two interpolation methods, which are refined via a Kalman filter-like recursion. In all above mentioned studies, it is assumed that the unknown priority function can be measured by each agent at its position.

In this chapter, a new formulation is proposed that allows the study of search and coverage problems in uncertain environments. The uncertainty in the environment is captured by an unknown priority function. Unlike the aforementioned works, in this study the priority function is not directly measurable at each point. Motivated by real applications, a new priority function is introduced which is a function of position of some unknown targets in the environment. The information about the position of targets is updated by some search vehicles, using a decentralized decision making approach. A centroidal Voronoi method is then used to optimally deploy service vehicles over the environment. This new formulation is a reasonable and practical problem set-up in view of better efficiency and cost effectiveness for different applications such as forest fire monitoring and detection using separate search and fighting (service) UAVs, as well as search and rescue missions. To evaluate the performance of this method for real-time applications, some experiments are conducted on the real test-bed. Both simulation and experimental results confirm the effectiveness of the proposed approach.

6.1 Forest Fire Monitoring and Detection: A Practical Application

The cooperative multi-vehicles search and coverage approach is useful for many applications involving distributed sensing and distributed actuation. This framework can be used by groups of vehicles to carry out tasks such as environmental monitoring and clean-up, or search and rescue [5, 86]. For example, consider a team of UAVs tasked to detect and extinguish multiple fires in a partially known environment such as a forest. With their on-board sensors, the UAVs search the environment to find the center of fires. Then by using this information, the fire fighter UAVs aggregate in the perimeter of fires. Similarly, consider a group of water-borne vehicles which are in charge of monitoring and cleaning up an oil spill. The monitoring vehicles find the areas where the spill is most severe, while cleaning vehicles distribute themselves over the spill and concentrate their efforts on the areas mostly affected, without neglecting the areas where the spill is not much severe. In general, any application in which a group of automated mobile agents is required to provide collective sensing and actuation over an environment can be considered as an example of the use of the multi-vehicle search and coverage systems.

As mentioned, forest fire monitoring and detection is a practical application for the problem set-up that will investigate in details in this chapter. Hence, a few research work in the literature on this application is shortly surveyed in this section.

Despite the technological advances and substantial infrastructure dedicated to forest fire fighting, the average annual area burned in Canada is 2.5 million hectares based on the report of Natural Resources Canada [87]. Forest fires are highly complex, non-structured environments where the use of multiple sources of information at different locations is essential. Traditional fire protection methods use mechanical

devices or humans to monitor the surroundings which are very dangerous activities requiring extensive human resources [88]. However, due to rapid growth of the electronics, digital camera technologies, artificial vision and image processing techniques there has been a shift from conventional forest fire detection systems towards computer-vision-based systems for forest fire detection [88,89]. In general, the studies of forest fire detection can be divided into three different groups: ground systems, systems on aerial means, and satellite based systems [6]. These platforms have different technological and practical problems for their use in operational conditions. Therefore, UAVs with computer-vision based systems represent a natural and good option to fill in this gap by providing rapid and low-cost responses to the forest fires and accomplishing long missions which are way beyond human capabilities [90, 91].

Vision-based systems generally use three features of the fire: color, motion, and geometry. The image processing is one of the main techniques widely used and implemented for automatic fire detection and monitoring [92]. Fire detection estimations are obtained from the processing of both infrared and visual images. In [93], a novel method is proposed to detect the fire and flame by processing the video data captured by an ordinary camera, monitoring an open scene and combining fire flicker and color clues to reach a final decision. In addition, intelligent methods are widely used in the recent years for fire detection to reduce the false alarms rate and the cost of sensors [94].

Dynamics of forest fires are considered as one of the most important scientific challenges in the field of the environmental studies [95]. In [96], the feasibility of the application of a team of small (low altitude, short endurance) UAVs to cooperatively monitor and track the propagation of large forest fires is explored. This work provides simulations using a numerical propagation model for the forest fire monitoring and detection. The Airborne Wildfire Intelligence System (AWIS) includes wildfire

detection and mapping of the fire-front and burned area [97]. In general, the main objective of the forest fire monitoring system based on UAVs is to automatically monitor and detect forest fires in real time. The location and shape of the fire front, the rate of spread and the fire flame height are the most important elements in the UAVs-based forest fire monitoring [98].

6.2 Problem Statement

This chapter addresses the cooperative multi-vehicle search and coverage problem in an uncertain environment. Consider the scenario that some search vehicles are deployed to search and detect some targets in the terrain. There are also service vehicles that their duty is to spread out over the environment to provide coverage. The search vehicles broadcast their information about the environment to the service vehicles. This information allows the service vehicles to find where in the environment they are mostly needed and to aggregate in those areas.

For the search problem, the environment is discretized in cells that are described by a probability of target existence. There is an uncertainty region corresponding to each target. Each target is assumed to lie somewhere within its uncertainty region, but its exact position is unknown. Each search vehicle stores a probability map, which contains the probability of existence of all targets in each cell. During the mission, sensors of search vehicles can detect targets in their footprints. The probability map is updated during the mission based on whether or not the target is detected by the sensors. The objective of the cooperative search mission is to maximize the amount of information about the environment.

The objective of service vehicles is to spread out over the area to cover the entire environment. However, in most cases, all points in the environment do not have the same level of importance. A priority function which reflects the measure of relative

importance of different points in the environment is also considered. The priority of each point is a decreasing function of the distance between that point and position of targets. Therefore, points closer to the targets have more value and more level of importance in the environment. Since our information about the position of targets improves during the search mission, the priority function is changed and get more accurate as mission goes on.

The number of targets in the environment is known *a priori*. However their exact position is unknown. Each vehicle is assigned a unique altitude, therefore avoiding the need to consider collision avoidance, which is outside the scope of this chapter. Although search vehicles are at different altitudes, their sensor footprints can be assumed to have almost the same resolutions without loss of generality. It is also assumed that each search vehicle can communicate with all other vehicles in the team. Moreover, all service vehicles can communicate with their neighboring vehicles.

6.3 Search Problem

A decision making approach is proposed for the search problem. The search vehicles cooperatively try to choose the best paths to gain the maximum amount of information about the environment. Each search vehicle uses a limited look-ahead dynamic programming (DP) algorithm [99] to find its path. The search vehicles can share their sensor measurements and their locations at each time step with others over their communication channels. Therefore, search vehicles are always aware of the current information of the system when they have to decide their next action. An appropriate model must incorporate the influence of the current control action on future states. In general, the search vehicle model is of the form

$$x_{k+1}^i = f(x_k^i, u_k^i, w_k^i), \forall i \in \{1, 2, ..., n_s\} \tag{6.1}$$

where x_k^i is the state of the i-th search vehicle at the discrete time step k, u_k^i is the control input of the i-th search vehicle, w_k^i is a random variable that captures the stochastic elements in the system dynamics, and n_s is the number of search vehicles. The state of the search vehicle consists of the search status and the vehicle status. The probability of existence of targets and their level of uncertainty in location constitute the search status. The vehicle status comprises position and heading angle of the vehicle. The search vehicles can communicate with each other so they can form a comprehensive view of the state. The control input, u_k^i, comes from a set of possible assignments U such as: turn left, turn right, or go straight. Stochastic elements which are captured by w_k^i come from different sources, including unknown locations of search targets, unknown actions of other vehicles, and imperfect sensor information.

6.3.1 Updating the Probability Map

Search vehicles store a probability map, which contains the probability of existence of a target in any given cell. An initial map of the environment, which is uncertain and incomplete, is created based on the *a priori* knowledge of the environment. Search vehicles are equipped with sensors which are able to detect the targets in their footprints. During the mission, the probability map is updated based on whether or not the target is detected by the sensors. Therefore, the following events can be defined:

$E_{x,y}^l$: target l is in the cell (x, y)

D_Ω^l: target l is detected in the region Ω

where Ω is the collection of cells that are covered by the vehicle sensor during its last step.

Target not detected

When the sensor has not detected the target during the last step of the mission, the probability of existence of the target for each cell can be updated by Bayes' rule as follows:

$$P(E^l_{x,y}|\bar{D}^l_\Omega) = \frac{P(E^l_{x,y})P(\bar{D}^l_\Omega|E^l_{x,y})}{P(\bar{D}^l_\Omega)}$$

where the overbar on the events represents the complement of the events. The probabilities of true positive and false positive measurements of sensors are assumed to be as follows:

$$\gamma \triangleq P(D^l_\Omega|E^l_\Omega)$$

$$\varepsilon \triangleq P(D^l_\Omega|\bar{E}^l_\Omega)$$

These two parameters are obtained from technical specifications on the sensors, and are considered to be known *a priori*. Therefore, the probability map can be updated as follows:

$$P(E^l_{x,y}|\bar{D}^l_\Omega) = \begin{cases} \dfrac{P(E^l_{x,y})\bar{\gamma}}{P(\bar{D}^l_\Omega)} & (x,y) \in \Omega \\ \dfrac{P(E^l_{x,y})\bar{\varepsilon}}{P(\bar{D}^l_\Omega)} & (x,y) \notin \Omega \end{cases}$$

In the derivation of second equation, the fact is used that the existence of target l in a cell outside of coverage region Ω means that there is no target inside that region. The probability that target l is not detected in the last step of the mission, $P(\bar{D}^l_\Omega)$, can be calculated as follows:

$$\begin{aligned} P(\bar{D}^l_\Omega) &= P(\bar{D}^l_\Omega|E^l_{x,y})P(E^l_{x,y}) + P(\bar{D}^l_\Omega|\bar{E}^l_{x,y})P(\bar{E}^l_{x,y}) \\ &= \bar{\gamma}P(E^l_\Omega) + \bar{\varepsilon}(1 - P(E^l_\Omega)) \\ &= \bar{\gamma}(\sum_{\forall(x,y)\in\Omega} P(E^l_{x,y})) + \bar{\varepsilon}(1 - \sum_{\forall(x,y)\in\Omega} P(E^l_{x,y})) \end{aligned}$$

Therefore, if the total probability of finding target l in the collection of cells Ω is defined as:

$$N_{\Omega}^{l} = \sum_{\forall (x,y) \in \Omega} P(E_{x,y}^{l})$$

the *posterior* probability of existence of target l in each cell can be computed as follows:

$$P(E_{x,y}^{l} | \bar{D}_{\Omega}^{l}) = \begin{cases} \dfrac{P(E_{x,y}^{l})\bar{\gamma}}{\bar{\gamma} N_{\Omega}^{l} + \bar{\varepsilon}(1 - N_{\Omega}^{l})} & (x,y) \in \Omega \\ \dfrac{P(E_{x,y}^{l})\bar{\varepsilon}}{\bar{\gamma} N_{\Omega}^{l} + \bar{\varepsilon}(1 - N_{\Omega}^{l})} & (x,y) \notin \Omega \end{cases}$$

It is always assumed that the probability of true positive measurement of sensors is more than 0.5 and the probability of false positive measurement of sensors is less than 0.5. In this case, it is easy to show that when a target is not detected, the probability of existence of that target in the cells inside the footprint of the sensor will be decreased, while the probability of the cells outside the sensor footprint will be increased.

Target detected

Using a similar procedure, when the sensor has detected the target, the *posterior* probability of existence of target l in each cell can be computed as follows:

$$P(E_{x,y}^{l} | D_{\Omega}^{i}) = \begin{cases} \dfrac{P(E_{x,y}^{l})\gamma}{\gamma N_{\Omega}^{l} + \varepsilon(1 - N_{\Omega}^{l})} & (x,y) \in \Omega \\ \dfrac{P(E_{x,y}^{l})\varepsilon}{\gamma N_{\Omega}^{l} + \varepsilon(1 - N_{\Omega}^{l})} & (x,y) \notin \Omega \end{cases}$$

6.3.2 Dynamic Programming Formulation

The search vehicles must choose a control signal such that it results in the best possible paths, in the sense that the team of search vehicles gathers maximum information about position of targets. In other words, each search vehicle attempts to optimize the

number of expected found targets over the planning horizon of the decision process. This leads naturally to the idea of applying DP techniques [99]. However, as the dimension of the problem grows so does the computation time. The dimension of the problem is given by the possible states to be examined over the planning horizon of the entire mission. To make the problem tractable, and solvable in real-time, a rolling horizon limited look-ahead policy can be utilized [54, 99]. The price to pay for such an approximation is a loss in performance (near optimality). This rolling horizon approximation defines a horizon of time steps T, and then replaces the value of final gain with the gain at T-steps ahead. Define $J_k^i(x_k^i)$ as the "gain" of i-th search vehicle at decision time step k which represents the time-discounted expected number of targets identified by the vehicle as it travels from time step k to T-steps ahead. The rolling horizon approximate dynamic programming (ADP) equation for this problem can be expressed as:

$$J_k^i(x_k^i) = \max_{u_k^i \in \Omega} \{E_{w_k^i}\{g(x_k^i, u_k^i, w_k^i)\} + J_{k+1}^i(f(x_k^i, u_k^i, w_k^i))\} \tag{6.2}$$

where the term $g(x_k^i, u_k^i, w_k^i)$ is the single step gain. The optimal decisions can be found by taking the arguments of the maximization of the DP recursion. To calculate the gain function, one needs to find the expected value of single step gain or the gain that i-th search vehicle will receive at one time step (specifically at time step k). This value can be written as:

$$g(x_k^i, u_k^i, w_k^i) = \lambda^k \delta_k^i \sigma_k^i \tag{6.3}$$

where σ_k^i is the search gain for the i-th search vehicle at time step k which is the expected value of the number of targets detected during the mission from time step k to time step $k+1$, δ_k^i is the probability that the planning vehicle is operational at time k, and $\lambda \in [0, 1]$ is the time discount factor. Let Ω_k^i be the collection of cells that

the i-th search vehicle covers during its mission from time step k to time step $k+1$. The search gain σ_k^i can be calculated by adding up the probabilities of existence of targets over Ω_k^i, i.e. $\sum_{\forall l} \sum_{\forall (x,y) \in \Omega_k^i} p(E_{x,y}^l)$. The term δ_k^i is normally a decreasing function of time which means the probability that the vehicle is operational decreases as time goes on. With the time discount less than one, it is typically desirable to find the objects as soon as possible.

6.3.3 Cooperative Decision Making

The objective of search mission is to search the terrain to gain as much information about the environment as possible. To achieve this goal, a decentralized method is used where each vehicle makes a decision about its next action individually. Each vehicle is viewed as a self-interested decision maker. The proposed approach in this section consists of optimizing a global objective function through autonomous vehicles that are capable of making individual decisions to optimize their own objective functions. In non-cooperative decision making, it is possible that two or more search vehicles decide to search the same area. Although the decision of each vehicle may be individually optimal, the overall gain will be less than the case if the vehicles search completely different areas addressing a team goal. It is desired to obtain localized objective function for each vehicle that aligns with the global objective function. In the proposed approach, each vehicle uses a look-up table to estimate the probability of different actions of other vehicles. These probabilities are utilized to modify the objective function of the vehicle to comply with the global objectives. Therefore, when the search vehicles want to make decisions on their next actions, they must simply optimize their own objective functions which also optimize the global objective.

It has been shown that when a vehicle searches an area and does not find any targets in it, the probability of target existence in that area is reduced. Therefore

the gain σ of searching that area in the future decreases. If all vehicles know the future position of other vehicles, a similar method can be used to prevent them from searching the same areas in the future. Assume that each vehicle knows the probability of presence of other vehicles in each cell over the future look-ahead horizon. Then the modified search gain of the vehicles can be defined as follows:

$$\hat{\sigma}_k^i = \sum_{\forall l} \sum_{\forall (x,y) \in \Omega_k^i} \rho_{x,y}^k p(E_{x,y}^l) \tag{6.4}$$

where $\rho_{x,y}^k \in [0,1]$ is the interference discount factor which is a decreasing function of the probability that other vehicles also decide to search the cell (x,y) in the next k steps. Now, one can modify the single step gain of the i-th vehicle in (6.3) by replacing σ_k^i with $\hat{\sigma}_k^i$. Evaluating the probability of presence of other vehicles exactly requires each vehicle to expand the planning tree of every other vehicle as shown in Fig. 6.1.

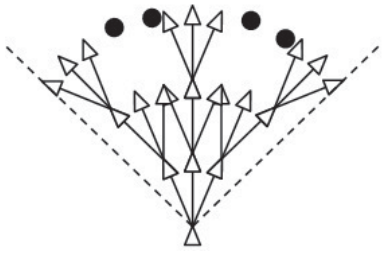


Figure 6.1: Future position of a vehicle for three steps look-ahead.

Although this method reduces the computational complexity of cooperative decision making compared to a centralized approach, it is still impractical when the number of vehicles or the search horizon increases. To reduce the computational burden, we provide all search vehicles with a look-up table that contains the probability of presence of a search vehicle in each cell at the next k steps, given its current position and heading, i.e. $p(x,y,k|x_0,y_0,\theta_0)$, where (x,y) is the position of vehicle at k-step ahead and (x_0,y_0), and θ_0 are its current position and heading. This table

is made off-line and can be produced analytically by using different estimation algorithms or simply through reasonable amount of simulations where the vehicle chooses its next action randomly. It is obvious that since the search vehicles neglect the probability map of targets in the approximation of the next positions of other vehicles, the method is not optimal, but it dramatically decreases the computational demand and processing time. By using the probability of presence of the other vehicles, the interference discount factor for the i-th search vehicle can be defined as:

$$\rho_{x,y}^{k} = e^{-\sum\limits_{\forall j \neq i} \sum\limits_{t=1}^{k} p_j(x,y,t)}$$

where $p_j(x, y, t)$ is the probability of j-th search vehicle being in the cell (x, y) at t step ahead.

6.4 Coverage Problem

Analogous to the coverage problem in Chapter 2, every arbitrary point in the environment Q is denoted as q, the position of the i-th service vehicle is p_i, and the set of all service vehicle positions is denoted as $\mathcal{P} = \{p_1, p_2, \ldots, p_{n_c}\}$. The function $\varphi : Q \to R_+$ is a priority function that defines a weight for each point. This function may reflect knowledge of the probability of occurrence of events in different regions, or simply a measure of relative importance of different regions in Q. Therefore, the higher the value of $\varphi(q)$ the more attention the group has to pay to q. Let $V = \{V_1, V_2, \ldots, V_n\}$ be the Voronoi partition of Q, for which the service vehicle positions are the generator points. Moreover, two service vehicles V_i and V_j are (Voronoi) neighbors if $V_i \cap V_j \neq 0$.

Similar to previous chapters, as a measure of the system performance, the coverage

function is defined as:

$$\mathcal{H}(\mathcal{P}) = \sum_{i=1}^{n_c} \int_{V_i} \frac{1}{2} \|q - p_i\|^2 \varphi(q) dq \tag{6.5}$$

where it is assumed that the i-th service vehicle is responsible for its Voronoi region V_i. Note that the function (6.5) measures the ability of the coverage provided by the network of service vehicles in Q. Qualitatively, a low value of function (6.5) corresponds to a good configuration for coverage of the environment Q.

The above coverage problem is completely investigated in Chapter 2, and it is shown that the Centroidal Voronoi configuration (i.e., every vehicle is at the centroid of its Voronoi region, $p_i = C_{V_i},\ \forall i \in \{1, 2, \ldots, n_c\}$) is the optimal solution for this problem. This result will be used in following subsections to propose a coverage controller in uncertain environments.

6.4.1 Priority Function

The definition of the priority function $\varphi(q)$ depends on the desired application. It defines a weight for each point in the environment which is a measure of relative importance of that point. In many applications, there are some critical points and the level of importance of each point in the terrain is inversely proportional to the distance between the point and the critical points. For instance, the critical points can be hotspots in a forest fire or the source of gushing in the oil spill. Let $\varphi(q) = \sum_{i=1}^{n_t} \phi(q, q_c^i)$ where q_c^i is the i-th critical point and n_t is the number of critical points. Function $\phi(q, q_c^i)$ is known *a priori* and it has a maximum at the i-th critical point q_c^i. Therefore, knowing the exact location of critical points, one can find the weight of all points $\varphi(q)$.

In many cases, the locations of critical points are not known precisely but it is

known that they are lied somewhere inside some uncertainty regions. Knowing the probability distribution of each critical point i in its uncertainty region, $P(q_c^i)$, priority function $\varphi(q)$ can be obtained as follows:

$$\varphi(q) = \sum_{i=1}^{n_t} \int_{\Lambda_i} \phi(q, q_c^i) P(q_c^i) dq_c^i \tag{6.6}$$

where Λ_i is the uncertainty region of the i-th critical point. Indeed, $\int_{\Lambda_i} \phi(q, q_c^i) P(q_c^i) dq_c^i$ is the expected value of function $\phi(q, q_c^i)$ with respect to q_c^i.

These critical points are in fact the targets of search problem. Since the search is done in a discrete environment, the probability of all points inside a cell is assumed to be equal. Therefore, (6.6) can be modified as follows:

$$\varphi(q) = \sum_{i=1}^{n_t} \sum_{\forall (x,y) \in \Lambda_i} P(E_{x,y}^i) \int_{q_c^i \in (x,y)} \phi(q, q_c^i) dq_c^i \tag{6.7}$$

At the beginning of the mission, service vehicles have *a priori* information of probability distribution of critical points. They use this information to compute priority function $\varphi(q)$. Then, service vehicles spread out over the environment based on this distribution. During the mission, search vehicles update the probability maps of critical points and transmit this information to the service vehicles. Using these updated probability maps, service vehicles modify their configurations and change their positions in the environment.

6.4.2 Distributed Coverage Controllers

In this section, the coverage control for a group of service vehicles is investigated. Each service vehicle is modeled as a double-integrator point mass moving on a two-dimensional (2-D) plane (i.e. $\ddot{p}_i = u_i$ where u_i is the control input of the i-th service

vehicle). Following assumptions are used for derivation of distributed coverage controllers in this chapter:

Assumption 6.1. *The service vehicles have the ability to compute their own Voronoi regions in a distributed manner.*

Assumption 6.2. *Each service vehicle can communicate with other service vehicles in its neighboring Voronoi regions.*

Assumption 6.3. *The i-th service vehicle has a* receiving region *which is assumed to be a circle of radius R_{r_i}, centered at p_i. The i-th service vehicle can receive the information of search vehicles if they are inside of its receiving region.*

For the purpose of coordinating multiple-service vehicles to cover a planar environment, the position controller based on the Centroidal Voronoi configuration is designed. Consider that the position of the i-th service vehicle is denoted by p_i, and C_{V_i} is the center of Voronoi that corresponds to the i-th service vehicle. The position control law is proposed for the i-th service vehicle as:

$$u_i = k_1^i M_{V_i}(C_{V_i} - p_i) - k_2^i \dot{p}_i, \ \forall i \in \{1, 2, ..., n_c\} \tag{6.8}$$

where k_1^i and k_2^i are the positive gains. The following results can be inferred with the analogous analysis in Theorem 2.1.

Theorem 6.1. *Consider a group of n_c service vehicles whose dynamic models are described as a double-integrator. Let the Assumptions 6.1 through 6.3 hold. Under the control law (6.8), it is guaranteed that the whole system is asymptotically stable and the planar positions of service vehicles converge to a centroidal Voronoi configuration.*

The service vehicles applying the control law (6.8) will move towards the centroid of its Voronoi region. Due to the convexity of region, the centroid is always inside the

Voronoi region. Therefore, the Voronoi approach has implicit collision avoidance. In addition, by designing a suitable controller for heights of multiple vehicles, they can fly at different levels. Then the collision avoidance can be guaranteed in the entire mission even for the large dimension vehicles.

6.5 Simulation Results

The proposed distributed search and coverage algorithm has been demonstrated first via numerical simulations in the MATLAB® environment. The environment used in simulation is a 1 km $\times$ 1 km square. Since the search problem has discrete nature, the environment is divided into 10000 cells which make a 100 $\times$ 100 square grid. There exist three targets known to be in the squared areas as shown in Fig. 6.2, but their exact positions are unknown. The *a priori* probability of existence of these targets is uniformly distributed in their uncertainty region while their real positions are marked by the $*$ marker. It is also considered that a virtual target exists in the environment and its uncertainty region is the whole terrain. Considering this target enforces search UAVs to search the unexplored area of the environment.

A group of three fixed-wing search UAVs ($n_s = 3$) and ten quad-rotor service UAVs ($n_c = 10$) are deployed to search and cover the environment. Each search UAV is equipped with a sensor that can detect targets in its footprint which is assumed to be equal to the size of a cell. All three search UAVs start their mission from the south west corner of the terrain, while all service UAVs start their mission from their individual bases which are located on the border of the environment as shown in Fig. 6.2. The radius of the receiving region of all service vehicles are 150 m.

At each decision time step, search UAVs must decide to go straight, turn 15 degrees left or 15 degrees right. It is assumed that once a search UAV has made a decision about its next action, that action can be performed immediately and then the search

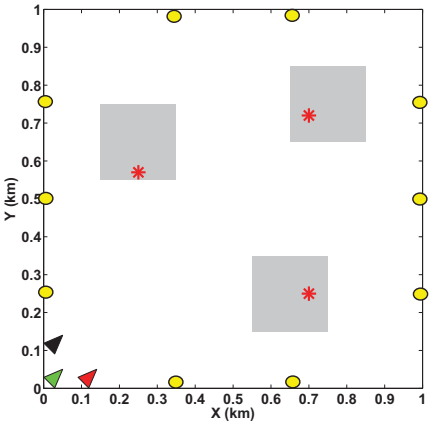


Figure 6.2: The problem environment; the grey rectangles are the uncertainty regions of different objects and * denotes the actual position of objects. Search UAVs and service UAVs are shown by ▷ and ○ markers respectively.

UAV continues its mission in a straight path until the next decision time step. The speeds of search UAVs are assumed to be constant. In order to execute the simulation in a reasonable amount of time, the look-ahead horizon of the DP algorithm is set to 5 time steps. Probabilities of true positive and false positive measurements of sensors are set to be 0.9 and 0.1 respectively. In this simulation, the following Gaussian density function is used:

$$\phi(q, q_c^i) = \frac{1}{\sigma\sqrt{2\pi}} \left(e^{-\frac{(q-q_c^i)^2}{2\sigma^2}} \right) \tag{6.9}$$

where $\sigma = 70$ m. The initial and final probability maps and their corresponding priority functions are shown in Fig. 6.3.

Following scenario has been considered in the simulation. At the beginning, coverage vehicles spread over the terrain based on the imprecise initial priority function which is derived from the *a priori* probability maps. The final configuration of planar positions, the trajectories of all UAVs, and the exact priority function are shown in Fig. 6.4(a). This distribution is calculated based on the actual position of critical points (targets). The color intensity is proportional to the value of priority function

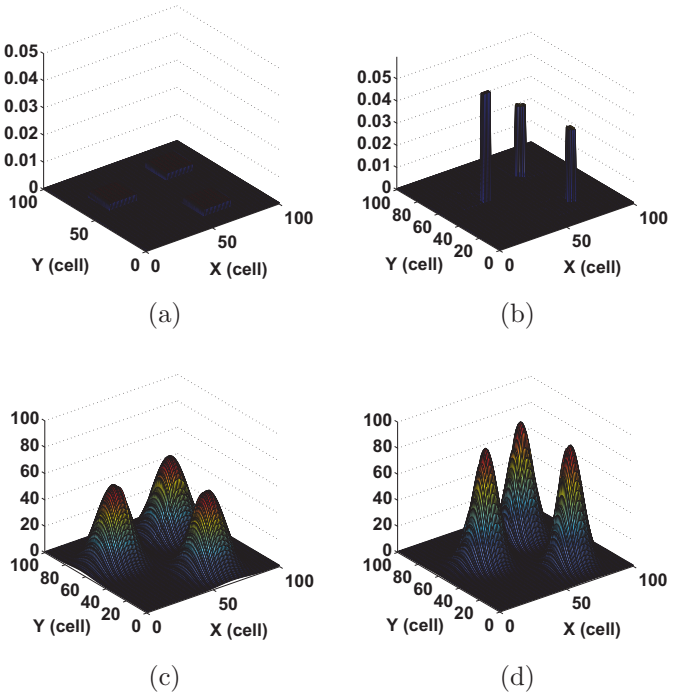


Figure 6.3: (a) the initial probability map, (b) the final probability map, (c) the initial priority function, and (d) the final priority function.

at each point. Corresponding priority function based on the probability maps is also depicted in Fig. 6.4(b). It can be seen from this figure that the configuration of service UAVs in the environment is optimal according to available priority function.

Next, search UAVs start their mission to explore the terrain. During the mission, they update the probability maps of targets and transmit these updated maps to the service vehicles on a regular basis. The position and trajectory of all UAVs and the exact priority function are shown in the Figs. 6.4(c), 6.4(e), and 6.4(g) for three different time steps, respectively. The positions of service UAVs and the corresponding priority function based on the most updated probability maps are shown in the Figs. 6.4(d), 6.4(f), and 6.4(h), respectively. It is noteworthy that as search UAVs explore the environment, the probability maps become more precise, and therefore the current priority function gets more similar to the exact one. Especially in Figs. 6.4(g) and 6.4(h), the priority functions are almost the same in the both figures. As expected,

deployment of search UAVs helps service vehicles to improve their performance to cover the most needed areas.

In the proposed algorithm, it is assumed that there is no limit on communication between neighboring service vehicles. To evaluate the effectiveness of the proposed method in more realistic situations where the communication is limited, the above simulation has been repeated with different communication ranges. The value of coverage function which is a measure of the system performance is reported in the Table 2 for five different times, using the exact priority function. It can be seen that the coverage performance is improved about 25% by using this approach in the case of no limit on communication ranges. As expected, the coverage performance degrades when the communication between service vehicles is limited. However, it can be seen that the performance degradation is insignificant and the proposed method still improves the coverage performance considerably.

To evaluate the average performance of the proposed approach, different simulations have been carried out 25 times. The average number of detected targets and the value of coverage function are depicted in Fig. 6.5. The uncertainty regions and actual positions of targets are randomly chosen for each repetition of simulation. As the number of detected targets increases, the value of coverage function decreases and the coverage performance improves.

Table 2: The coverage function $\mathcal{H}$ at different times for the scenario with different communication ranges

Communication Ranges \ T(sec)	0	120	160	200	240
No limit	2.4339	0.6771	0.6235	0.5948	0.5443
200 m	2.4339	0.6884	0.6294	0.5989	0.5468
100 m	2.4339	0.7801	0.6433	0.6127	0.5578
75 m	2.4339	1.4026	0.8925	0.7127	0.5872
Heterogeneous in $[75, 200]$	2.4339	0.8810	0.6728	0.6321	0.5692

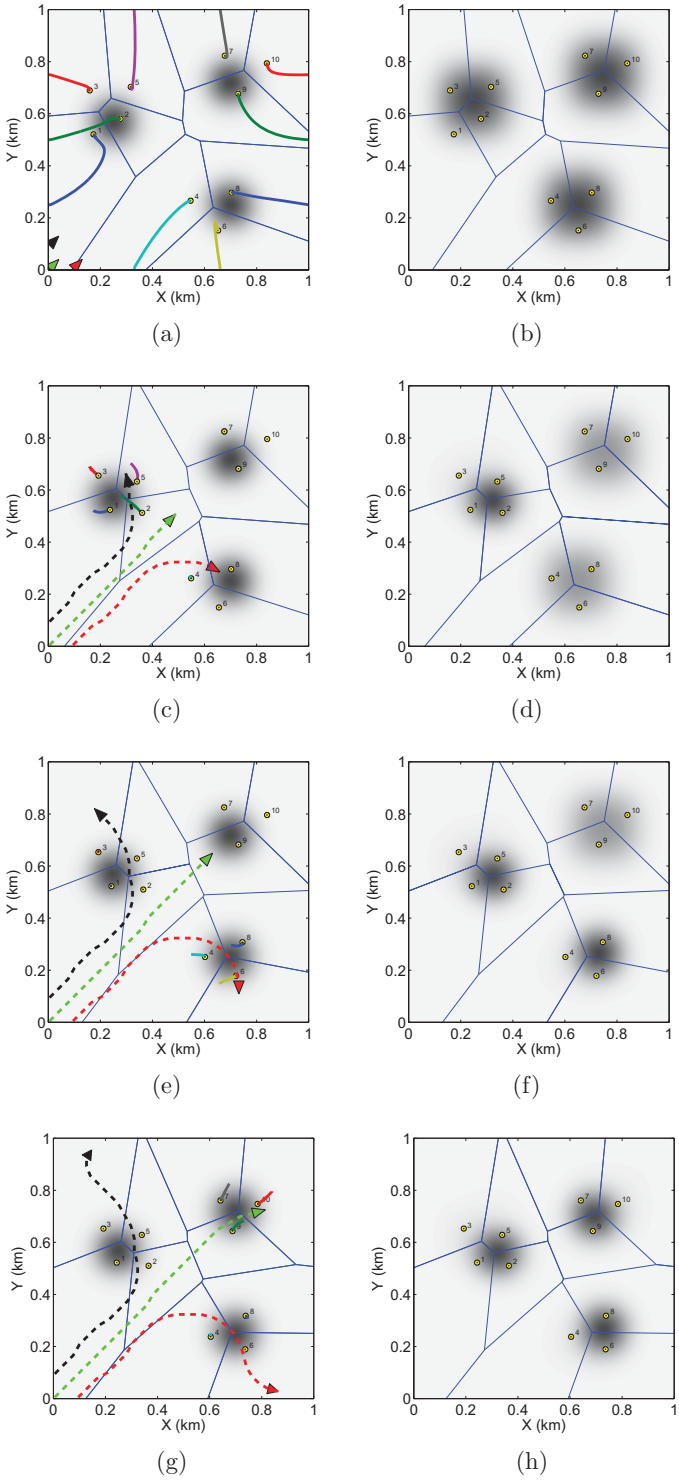


Figure 6.4: Left: The configuration and the trajectories of all UAVs and the exact priority function. The color intensity is proportional to the value of priority function. Search UAVs and service UAVs are shown by ▷ and ⊙ respectively. Right: The configuration of service UAVs and the corresponding priority function based on the probability maps. (a) and (b) t=100 sec, (c) and (d) t=160 sec, (e) and (f) t=200 sec, (g) and (h) t= 240 sec

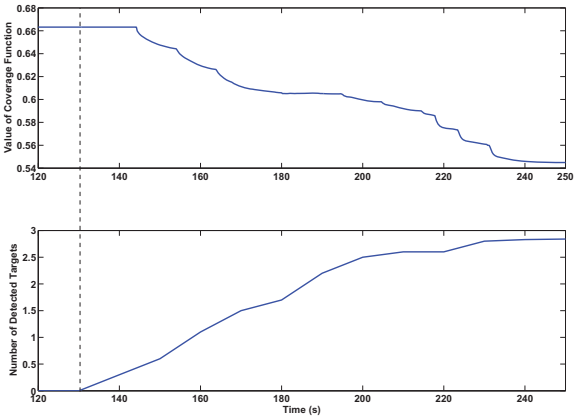


Figure 6.5: The average number of detected targets and the value of coverage function for 25 simulations.

6.6 Experimental Results

To demonstrate the effectiveness of the presented theoretical developments, an experiment is conducted on a group of UGVs available at the Networked Autonomous Vehicles Lab (NAVL) of Concordia University, which are provided by Quanser Inc. In this experiment, it is considered that the search mission is still carried out by simulation due to the difficulty for flying fixed-wing UAVs in the indoor testing environment, and the service problem is performed using a network of virtual robots and three available physical UGVs (Qbot and QGVs which are introduced in Subection 3.5.2).

As shown in Subsection 2.4.2, the kinematic model of UGVs can be described by (2.14). The nonholonomic kinematic model of UGVs can be transformed into a linear controllable system using *Dynamic* (i.e., time-variant) state feedback [100]. This results in a fully linearized model which can be described by a double-integral model as follows:

$$\ddot{x} = u_x$$

$$\ddot{y} = u_y$$

where (x, y) is the coordinate of a point located at the mid-axis of the rear wheels of the robot, u_x and u_y are the control inputs. The resulting dynamic compensator is

$$\begin{aligned} v &= \xi \\ \omega &= \frac{u_x cos(\theta) - u_y sin(\theta)}{\xi} \\ \dot{\xi} &= u_x cos(\theta) + u_y sin(\theta) \end{aligned} \tag{6.10}$$

where v and ω are the linear and angular velocities of the center of mass of the robot. Therefore, as Theorem 6.1 suggests, using the following control law for each UGV guarantees that the whole system is asymptotically stable and the planar positions of all service vehicles converge to a centroidal Voronoi configuration:

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} = k_1 M_V \Big(C_V - \begin{pmatrix} x \\ y \end{pmatrix} \Big) - k_2 \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$$

Inputs v and ω therefore can be calculated using (6.10). It is worth to mention that the model of virtual service vehicles is still a double-integrator and their control law is given by (6.8).

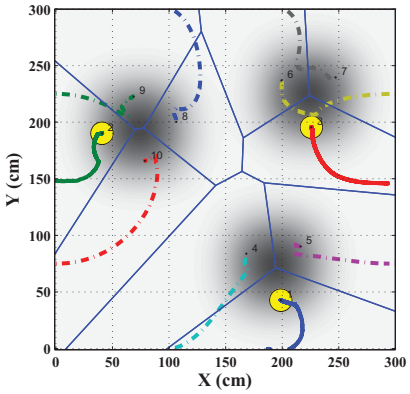
6.6.1 Experimental Tests

The environment in the experimental setup is similar to the simulation case. The terrain is a 3 m $\times$ 3 m square which is divided to 10000 cells to make a 100×100 square grid. There still exist three targets known to be in their 60 cm $\times$ 60 cm uncertainty region but their exact positions are initially unknown. The *a priori* probability of existence of these targets is uniformly distributed in their uncertainty region. It is also considered that a virtual target exists in the environment and its uncertainty region is the entire terrain.

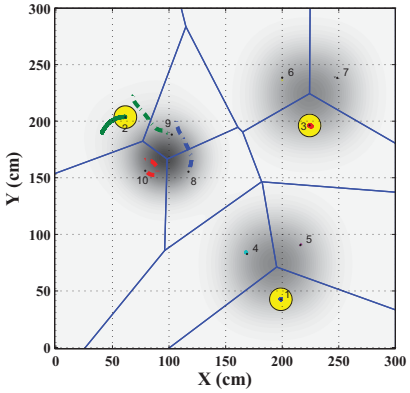
The search mission is performed by a group of three virtual UAVs. At each decision time step, search vehicles must decide to go straight ahead, turn 15 degrees left, or turn 15 degrees right. The velocity of each search vehicle is equal to 20 cm/s. The look-ahead horizon of the DP algorithm is set to 5 time steps. Simulation of search mission for all three UAVs is performed on the host computer which has a dual core 3.2 GHz processor. Service vehicles include seven virtual vehicles and three physical UGVs. Simulation of virtual vehicles is also performed on the host computer. The positions of vehicles are measured using the network of OptiTrack cameras. Host computer then sends the positions of all service vehicles (virtual and real) to the UGVs via the wireless communication channel. The Gaussian density function $\phi(q, q_i^c)$ is similar to the priority function in simulation (6.9) and the standard deviation is equal to $\sigma = 20$ cm.

At the beginning, service vehicles spread over the terrain based on the imprecise initial priority function which is derived from the *a priori* probability maps. After 30 sec, the search mission is commenced. The updated probability map is transmitted to the service vehicles every 5 sec by the host computer. The final configuration of planar position and the trajectories of all service vehicles for different time steps, 30 sec, 50 sec, 65 sec, and 80 sec are shown in the Fig. 6.5. The UGVs are shown by marker and their trajectories are shown by solid lines. The trajectory of virtual service vehicles are shown by dashed lines. The priority function based on the most updated probability maps are also shown in the figures. The color intensity is proportional to the value of priority function at each point. It can be seen from this figure that the configuration of service vehicles in the environment is optimal according to available priority function.

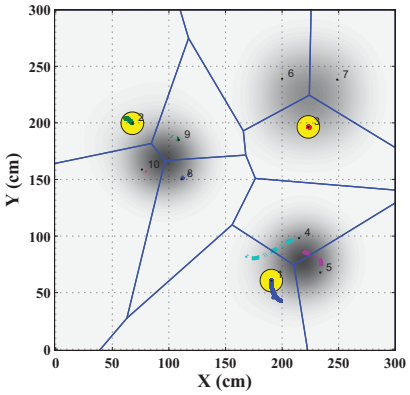
The value of coverage function $\mathcal{H}$, using the exact priority function, is shown in Fig. 6.6. As expected, the value of coverage function decreases dramatically at time


Y (cm)
X (cm)
0
50
100
150
200
250
300

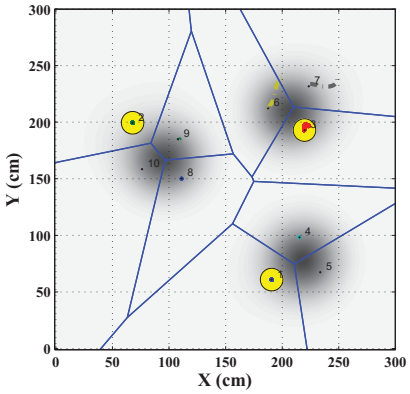
(a)


Y (cm)
X (cm)
0
50
100
150
200
250
300

(b)


Y (cm)
X (cm)
0
50
100
150
200
250
300

(c)



(d)

Figure 6.5: Experimental results: The configuration and the trajectories of all service vehicles and the corresponding priority function based on the probability maps. The color intensity is proportional to the value of priority function. The UGVs are shown by $\odot$ marker and their trajectories are shown by solid lines. (a) t=30 sec, (b) t=50 sec, (c) t=65 sec. (d) t= 80 sec

steps 45 sec, 60 sec, and 75 sec when the probability map is considerably improved due to the detection of a new target.

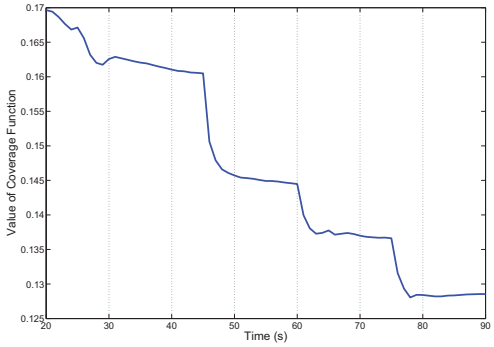


Figure 6.6: The value of coverage function.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this work, the cooperative coverage control problem for multi-agent systems is investigated from different aspects. There are many different challenges that need to be addressed in the coverage control problem, especially when considering real world applications. In Chapter 2, first the coverage problem which is the main framework of this study is stated, and the Voronoi diagram is then introduces as an optimal partitioning technique. Unlike the common assumption in the literature that the dynamics of agents are single-integrator, non-trivial dynamics are considered for agents. Therefore a planar position controller based on centroidal Voronoi configuration is proposed for group of quadrotor UAVs and wheeled robot UGVs. The Lyapunov-based stability analysis showed that the vehicles finally converge to the optimal configuration and the whole system is stable.

In Chapter 3, the problem of providing a prescribed service (such as coverage) using a group of heterogeneous mobile agents is investigated. It is assumed that the serving capabilities of different agents are nonidentical. The problem of service cost optimization is defined and the notion of center multiplicatively-weighted Voronoi

(CMWV) configuration is subsequently introduced to solve the problem. A motion coordination strategy is then proposed for group of agents. The effectiveness of the proposed technique is confirmed via numerical simulations and experiments on a testbed with a group of unmanned aerial and ground vehicles.

The inter-agent communication delay is unavoidable in multi-agent systems and neglecting the effect of delay in the analysis and design of multi-agent systems can lead to poor performance as well as unsafe behaviors such as inter-agent collision. In Chapter 4 a spatial partitioning technique is considered to address the problem of coverage control subject to inter-agent communication delays induced by certain communication faults. The above problem is also extended for the case sensors of agents may have variable effectiveness (health). The sensor effectiveness factor of each agent is incorporated in the locational optimization problem, and a novel partitioning technique is also introduced to address this problem. The notion of GMW-Voronoi diagram is introduced which divides the field based on the latest information received by the agents. Therefore, a distributed deployment algorithm is proposed for a network of mobile agents where the sensor effectiveness of agents may change with time and the inter-agent information exchange is subject to delay. One of the important properties of the proposed algorithm is inter-agent collision avoidance. Simulation results demonstrate the effectiveness of the proposed technique.

In Chapter 5, an effective coverage technique is developed in multi-agent systems with both collision avoidance and obstacle avoidance properties. It is desired to improve coverage using navigation functions. A distributed control law is proposed to drive the agents in the field in such a way that the coverage cost function is minimized. In addition, agent-to-agent and agent-to-obstacle collision avoidance are guaranteed under the proposed control strategy. The convergence of the closed-loop system is analyzed and the results show that the overall system is stable and asymptotically

convergent to the centroidal Voronoi configuration.

Most of the previous works in the area of Voronoi-based coverage control assume the distribution of sensory information in the environment is required to be known *a priori* by all agents. In Chapter 6, a new formulation is proposed that allows the study of search and coverage problems in uncertain environments. A group of vehicles called service vehicles are deployed to service the points or areas where they are mostly needed in the environment based on the Voronoi configuration. Since the high service areas are not known beforehand, a group of search vehicles are used to explore the environment based on the limited look-ahead dynamic programming method. The proposed approach leads to covering an uncertain environment more effective and improving the coverage performance. The proposed approach has been successfully verified by both numerical simulations and experimental tests.

7.2 Future Work

A framework for cooperative coverage control of multi-agent systems is presented in details in this thesis. However, there are several directions and possible related research areas in which future work can be carried out.

In Chapter 3, the optimal partitioning and configuration of agents introduced when the service (coverage) cost of agents are non-identical and in form of (3.2). However, this realistic assumption significantly outperform existing techniques, it can be made more general. In the general case, the conventional Voronoi-based diagram is not effective for agents deployment in the network. The problem of finding the optimal partitioning in this case can be very cumbersome, in general. It is also needed to find the optimal configuration of agents in this case.

To further generalize the results, a priority function is assumed to be given to prioritize the importance of giving service to (or covering of) different points in the

field. In this thesis, it is assumed that this priority function is stationary function like the Gaussian function in the simulation parts. Future work will extend the analysis of these algorithms to more general classes of problems such as dynamic priority function or moving targets in the environment.

The obstacle and collision avoidance feature in coverage control problem presented in Chapter 5. Throughout this chapter, it was assumed obstacles are fixed and has arbitrary shapes. Future work will utilize the same framework introduced here to develop the distributed control algorithms in the presence of moving obstacles in the environment.

In Chapter 2, the low level controller for vehicles with different dynamics provided based on the locational optimization problem. Although the problem of varying sensor effectiveness of vehicles considered in Chapter 4, the effect of uncertainty on actuator of vehicles did not address in this work. Therefore, uncertainty in the outcomes of vehicle actions and the degradation of the solution in this case can be considered as a future work.

In Chapter 6, two types of vehicles with different tasks used for search and coverage problems in uncertain environments. In order to design the technique provided here for more different applications and scenarios, the capabilities of search and coverage vehicles can be combined on one vehicle. This modification needs to redesign all algorithms but can reduce the operational cost and will be effective in more realistic applications like forest fire monitoring and detection.

Bibliography

[1] W. Dong and J. A. Farrell. Decentralized cooperative control of multiple nonholonomic systems. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 1486–1491, New Orleans, USA, 2007.

[2] D. Cruz, J. Mcclintock, B. Perteet, O. A. A. Orqueda, Y. Cao, and R. Fierro. Decentralized cooperative controla multivehicle platform for research in network embedded systems. *IEEE Control Systems Magazine*, 27(3):58–78, 2007.

[3] L. A. Weitz, J. E. Hurtado, and A. J. Sinclair. Decentralized cooperative-control design for multivehicle formations. *Journal of Guidance, Control, and Dynamics*, 31(4):970–979, 2008.

[4] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.

[5] M. Kumar, K. Cohen, and B. Homchaudhuri. Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires. *Journal of Aerospace Computing, Information, and Communication*, 8:1–16, 2011.

[6] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. *Proceedings of the IEEE*, 94(7):1306–1324, 2006.

[7] J. M. Bradley and C. N. Taylor. Georeferenced mosaics for tracking fires using unmanned miniature air vehicles. *Journal of Aerospace Computing, Information, and Communication*, 8:295–309, 2011.

[8] H. Dieter, Z. Werner, S. Gunter, and S. Peter. Monitoring of gas pipelines - a civil UAV application. *Aircraft Engineering and Aerospace Technology: An International Journal*, 77(5):352–360, 2005.

[9] S. Susca, S. Martínez, and F. Bullo. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2):288–296, 2008.

[10] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[11] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[12] J. Cortes, S. Martinez, and F. Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11:691–719, 2005.

[13] A. Howard, M. J. Matarić, and G. S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots*, 13:113–126, 2002.

[14] W. Li and C. G. Cassandras. Distributed cooperative coverage control of sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2542–2547, Seville, Spain, 2005.

[15] M. Schwager, D. Rus, and J. J. Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[16] Z. Drezner. *Facility Location: A Survey of Applications and Methods*. Springer-Verlag, 1995.

[17] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.

[18] J. Stergiopoulos and A. Tzes. Voronoi-based coverage optimization for mobile networks with limited sensing range a directional search approach. In *Proceedings of the American Control Conference*, pages 2642–2647, 2009.

[19] A. Boukerche and X. Fei. A voronoi approach for coverage protocols in wireless sensor networks. In *Proceedings of IEEE Global Communications Conference*, pages 5190–5194, Washington, D.C., USA, 2007.

[20] S. Salapaka, A. Khalak, and M. A. Dahlleh. Constraints on locational optimization problems. In *Proceedings of the 42th IEEE Conference on Decision and Control*, pages 1741–1746, 2003.

[21] G. Wang, G. Cao, and T. La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.

[22] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira. Sensing and coverage for a network of heterogeneous robots. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3947–3952, Cancun, Mexico, 2008.

[23] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo. Equitable partitioning policies for robotic networks. In *Proceedings of the International Conference on Robotics and Automation*, pages 2356–2361, 2009.

[24] L. C. A. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. S. Pereira. Simultaneous coverage and tracking (scat) of moving targets with robot networks. In *8th International Workshop on the Algorithmic Foundations of Robotics (WAFR'08)*, pages 85–99, 2008.

[25] J. Enright, K. Savla, and E. Frazzoli. Coverage control for nonholonomic agents. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4250–4256, Cancun, Mexico, 2008.

[26] F. Bullo, J. Cortés, and S. Martinez. *Distributed Control of Robotic Networks*. Applied Mathematics. 2009. Available at http://www.coordinationbook.info.

[27] S. Martinez, J. Cortes, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4):75–88, 2007.

[28] S. Bhattacharya, N. Michael, and V. Kumar. Distributed coverage and exploration in unknown non-convex environments. In *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, 2013.

[29] A. Breitenmoser, M. Schwager, J. Metzger, R. Siegwart, and D. Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4982–4989, Anchorage, Alaska, USA, 2010.

[30] C. H. Caicedo-Nunez and M. Zefran. A coverage algorithm for a class of non-convex regions. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4244–4249, Cancun, Mexico, 2008.

[31] M. Pavone, S. L. Smith, F. Bullo, and E. Frazzoli. Dynamic multi-vehicle routing with multiple classes of demands. In *Proceedings of American Control Conference*, pages 604–609, 2009.

[32] J. W. Durham, R. Carli, P. Frasca, and F. Bullo. Discrete partitioning and coverage control for gossiping robots. *IEEE Transactions on Robotics*, 28(2):364–378, 2012.

[33] A. Kwok and S. Martínez. Deployment algorithms for a power constrained mobile sensor network. *International Journal of Robust and Nonlinear Control*, 20(7):745–763, 2010.

[34] C. Nowzari and J. Cortés. Self-triggered coordination of robotic networks for optimal deployment. *Automatica*, 48(6):1077–1087, 2012.

[35] Y. Wang and I. I. Hussein. Awareness coverage control over large-scale domains with intermittent communications. *IEEE Transactions on Automatic Control*, 55(8):1850–1859, 2012.

[36] J. Stergiopoulos and A. Tzes. Decentralized communication range adjustment issues in multi-agent mobile networks. In *Proceedings of the American Control Conference*, pages 1629–1634, Baltimore, USA, 2010.

[37] A. Kwok and S. Martinez. Unicycle coverage control via hybrid modeling. *IEEE Transactions on Automatic Control*, 55(2):528–532, 2008.

[38] B. Bethke, J. How, and J. Vian. Group health management of UAV teams with applications to persistent surveillance. In *Proceedings of the American Control Conference*, pages 3145–3150, Cancun, Mexico, 2008.

[39] J. Cortés. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control*, 55(3):749–754, 2010.

[40] N. Bartolini, T. Calamoneri, T. La Porta, and S. Silvestri. Autonomous deployment of heterogeneous mobile sensors. *IEEE Transactions on Mobile Computing*, 10(6):753–766, 2011.

[41] Y. Stergiopoulos and A. Tzes. Convex voronoi-inspired space partitioning for heterogeneous networks: a coverage-oriented approach. *IET Control Theory and Applications*, 4(12):2802–2812, 2010.

[42] K. D. Do. Formation stabilization and tracking control of mobile agents using local potential functions. In *Proceedings of the American Control Conference*, number 2, pages 2148–2153, 2006.

[43] M. C. De Gennaro and A. Jadbabaie. Formation control for a cooperative multi-agent system using decentralized navigation functions. In *Proceedings of the American Control Conference*, pages 1346–1351, 2006.

[44] D. V. Dimarogonas and K. J. Kyriakopoulos. Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 84–89, 2005.

[45] I. I. Hussein and D. M. Stipanovic. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Transactions on Control Systems Technology*, 15(4):642–657, 2007.

[46] A. Dirafzoon, M. B. Menhaj, and A. Afshar. Voronoi based coverage control for nonholonomic mobile robots with collision avoidance. In *Proceedings of the Multi-Conference on Systems and Control*, pages 1755–1760, 2010.

[47] H. Tan, Y. Wang, X. Hao, Q. S. Hua, and F. C. M. Lau. Arbitrary obstacles constrained full coverage in wireless sensor networks. In *Proceeding of 5th International Conference of Wireless Algorithms, Systems and Applications*, pages 1–10, 2010.

[48] C. Y. Chang, C. T. Chang, Y. C. Chen, and H. R. Chang. Obstacle-resistant deployment algorithms for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 58(6):2925–2941, 2009.

[49] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems*, pages 299–308, 2002.

[50] A. Ahmadzadeh, G. Buchman, P. Cheng, A. Jadbabaie, J. Keller, V. Kumar, and G. Pappas. Cooperative control of UAVs for search and coverage. In *Proceedings of the AUVSI Conference on Unmanned Systems*, 2006.

[51] D. Wang, J. Liu, and Q. Zhang. Mobility-assisted sensor networking for field coverage. In *Proceedings of IEEE Global Communications Conference*, pages 1190–1194, Washington, D.C., USA, 2007.

[52] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Osaka, Japan, 1991.

[53] M. Li, W. Cheng, K. Liu, Y. He, X. Li, and X. Liao. Sweep coverage with mobile sensors. *IEEE Transactions on Mobile Computing*, 10(11):1534–1545, 2011.

[54] M. Flint and E. Fernandez. Approximate dynamic programming methods for cooperative UAV search. In *Proceedings of the 16th IFAC World Congress*, pages 59–64, Prague, Czech Republic, 2005.

[55] X. Tian, Y. Bar-Shalom, and K. R. Pattipati. Multi-step look-ahead policy for autonomous cooperative surveillance by UAVs in hostile environments. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 2438–2443, Cancun, Mexico, 2008.

[56] P. Chandler. Decentralized control for an autonomous team. In *Proceedings of AIAA Unmanned Unlimited Systems,Technologies, and Operations*, 2003.

[57] Y. Yang, M. M. Polycarpou, and A. A. Minai. Multi-UAV cooperative search using an opportunistic learning method. *ASME Journal of Dynamic Systems, Measurement, and Control*, 129:716–728, 2007.

[58] P. B. Sujit and D. Ghose. Negotiation schemes for multi-agent cooperative search. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 223(6):791–813, 2009.

[59] Y. Wang and I. I. Hussein. Bayesian-based decision-making for object search and classification. *IEEE Transactions on Control Systems Technology*, 19(6):1639–1647, 2011.

[60] K. R. Guruprasad and D. Ghose. Multi-agent search strategy based on centroidal voronoi configuration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3550–3555, Anchorage, USA, 2010.

[61] T. P. Lambrou and C. G. Panayiotou. Collaborative area monitoring using wireless sensor networks with stationary and mobile nodes. *EURASIP Journal on Advances in Signal Processing*, 2009:1–16, 2009.

[62] M. Schwager, J. McLurkin, and D. Rus. Distributed coverage control with sensory feedback for networked robots. In *Proceedings of Robotics: Science and Systems*, 2006.

[63] A. Dirafzoon, S. M. A. Salehizadeh, S. Emrani, M. B. Menhaj, and A. Afshar. Coverage control for mobile sensing robots in unknown environments using neural network. In *IEEE International Symposium on Intelligent Control*, pages 1482–1487, 2010.

[64] J. S. Marier, C. A. Rabbath, and N. Léchevin. Placement of a team of surveillance vehicles subject to navigation failures. In *Proceeding of the American Institute of Aeronautics and Astronautics - Guidance, Control and Navigation Conference*, Portland, Oregon, USA, 2011.

[65] A. Gusrialdi, S. Hirche, D. Asikin, T. Hatanaka, and M. Fujita. Voronoi-based coverage control with anisotropic sensors and experimental case study. *Intelligent Service Robotics*, 2(4):195–204, 2009.

[66] J. Cortés. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control*, 55(3):749–754, 2010.

[67] F. Sharifi, M. Mirzaei, B. W. Gordon, and Y. M. Zhang. Fault tolerant control of a quadrotor UAV using sliding mode control. In *Proceedings of the Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 239–244, Nice, France, 2010.

[68] G. Homan, H. Huang, C. J. Tomlin, and S. L. Waslander. Quadrotor helicopter flight dynamics and control: theory and experiment. In *Proceeding of the American Institute of Aeronautics and Astronautics - Guidance, Control and Navigation Conference*, Hilton Head, USA, 2007.

[69] E. Altug, J. P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *Proceedings ofthe 2002 IEEE International conference on Robotics and Automation*, pages 72–77, Washington, DC, 2002.

[70] H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. M. Zhang. Distributed coordination of multi-agent systems for coverage problem in presence of obstacles. In *Proceedings of the American Control Conference*, pages 5252–5257, Montreal, Canada, 2012.

[71] A. V. Akopyan and A. A. Zaslavsky. *Geometry of Conics*. American Mathematical Society, 2007.

[72] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh. Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing range. In *Proceedings of IEEE Global Communications Conference*, pages 1–6, Miami, Florida, USA, 2010.

[73] Y. M. Zhang and A. Chamseddine. *Fault Tolerant Flight Control Techniques with Application to a Quadrotor UAV Testbed*, pages 119–150. InTech Open Access Publisher, 2011.

[74] F. Sharifi, A. Chamseddine, H. Mahboubi, Y. M. Zhang, and A. G. Aghdam. Distributed coordination of multi-agent systems. Technical report, Concordia University, 2012. http://users.encs.concordia.ca/∼ymzhang/UAVs.htm.

[75] U. Munz, A. Papachristodoulou, and F. Allgower. Delay-dependent rendezvous and flocking of large scale multi-agent systems with communication delays. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 2038–2043, Cancun, Mexico, 2008.

[76] F. Sharifi, B. W. Gordon, and Y. M. Zhang. Decentralized sliding control of cooperative multi-agent systems subject to communication delay. In *Proceeding of the American Institute of Aeronautics and Astronautics - Guidance, Control and Navigation Conference*, Toronto, Canada, 2010.

[77] H. A. Izadi, B. W. Gordon, and Y. M. Zhang. Decentralized receding horizon control of multiple vehicles subject to communication failure. In *Proceedings of the American Control Conference*, pages 3531–3536, St. Louis, USA, 2009.

[78] J. Sember and W. Evans. Guaranteed voronoi diagrams of uncertain sites. In *20th Canadian Conference on Computational Geometry*, Montreal, Canada, 2008.

[79] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos. Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation*, 19(1):53–64, 2003.

[80] Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon. Ensuring network connectivity during formation control using a decentralized navigation function. *Military Communications Conference*, pages 954–959, 2010.

[81] D. V. Dimarogonas and K. J. Kyriakopoulos. Decentralized navigation functions for multiple robotic agents with limited sensing capabilities. *Journal of Intelligent and Robotic Systems*, 48(3):411–433, 2007.

[82] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[83] D. V. Dimarogonas and E. Frazzoli. Analysis of decentralized potential field based multi-agent navigation via primal-dual lyapunov theory. In *Proceedings of IEEE Conference on Decision and Control*, pages 1215–1220, 2010.

[84] M. Mirzaei, F. Sharifi, B. W. Gordon, C. A. Rabbath, and Y. M. Zhang. Cooperative multi-vehicle search and coverage problem in an uncertain environment. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 4140–4145, Orlando, Florida, USA, 2011.

[85] S. Martinez. Distributed interpolation schemes for field estimation by mobile sensor networks. *IEEE Transactions on Control Systems Technology*, 18(2):491–500, 2010.

[86] P. B. Sujit, D. Kingston, and R. Beard. Cooperative forest fire monitoring using multiple UAVs. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 4875–4880, New Orleans, Louisiana, USA, 2007.

[87] http://www.nrcan.gc.ca/forests/canada/sustainable-forest-management/criteria-indicators/13241.

[88] V. Vipin. Image processing based forest fire detection. *International Journal of Emerging Technology and Advanced Engineering*, 2(2):87–95, 2012.

[89] B.C. Arrue, A. Ollero, and J. R. Martinez de Dios. An intelligent system for false alarm reduction in infrared forest-fire detection. *IEEE Intelligent Systems*, 15(3):64–73, 2000.

[90] J. M. Bradley and C. N. Taylor. Geo-referenced mosaics for tracking fires using unmanned miniature air vehicles. *Journal of Aerospace Computing, Information, and Communication*, 8(10):295–309, 2011.

[91] J. Everaerts. The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:1187–1192, 2008.

[92] J. R. Martinez, L. Merino, F. Caballero, A. Ollero, and D. X. Viegas. Experimental results of automatic fire detection and monitoring with UAVs. In *International Conference on Forest Fire Research*, 2006.

[93] H. Jin and R. B. Zhang. A fire and flame detecting method based on video. In *International Conference on Machine Learning and Cybernetics*, pages 2347–2352, 2009.

[94] Q. Guo, J. Dai, and J. Wang. Study on fire detection model based on fuzzy neural network. In *2nd International Workshop on Intelligent Systems and Applications*, pages 1–4, 2010.

[95] R. Lasaponara, A. Santulli, and L. Telesca. Time-clustering analysis of forest-fire sequences in southern Italy. *Chaos, Solitons & Fractals*, 24(1):139–149, 2005.

[96] D. W. Casbeer, D. B. Kingston, A. W. Bear, T. W. McLain, S. Li, and R. Mehra. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.

[97] D. Campbell, W. G. Born, J. Beck, B. Bereska, K. Frederick, and S. Hua. The airborne wildfire intelligence system: A decision support tool for wild land fire

managers in Alberta. In *AeroSense 2002, International Society for Optics and Photonics*, pages 159–170, 2002.

[98] L. Merino, F. Caballero, J. R. Martinez de Dios, I. Maza, and A. Ollero. Automatic forest fire monitoring and measurement using unmanned aerial vehicles. In *Proceedings of the VI International Congress on Forest Fire Research*, 2010.

[99] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 2nd edition.

[100] G. Oriolo, A. De Luca, and M. Vendittelli. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.