University of Nevada, Reno

Design Distributed Control and Learning Algorithms for a Team of UAVs for Optimal Field Coverage

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science and Engineering

by

Huy X. Pham

Dr. Hung M. La - Thesis Advisor
December 2018

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Huy Xuan Pham

Entitled

Design Distributed Control and Learning Algorithms for a Team of UAVs for Optimal Field Coverage

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Hung M. La, Ph.D., Advisor

David Feil-Seifer, Ph.D., Committee Member

Hao Xu, Ph.D., Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

December, 2018

Abstract

Optimal field coverage problem refers to an active research branch that studies how we can use a finite set of sensors, such as camera, to optimally cover a field with arbitrary shape that can either be static or dynamically change over time. The problem arises in a wide range of applications, notably wildfires and oil spill tracking, military surveillance, and agriculture monitoring. In these applications, it is of growing interest to send a team of Unmanned Aerial Vehicles (UAVs) acting as a mobile sensor network, as they can provide sensing information with low costs and high flexibility, compared with traditional static monitoring methods. In this thesis, we addressed the problem by two distinct approaches: one is model-based and the other is model-free.

In the first part, we proposed a model-based control framework for UAV teaming to monitor and track a dynamic field like wildfire spreading. Wildfire is well-known for their destructive ability to inflict massive damage and disruption. We characterized the optimal sensing coverage problem to work with a changing wildfire environment. We proposed a decentralized control algorithm for a team of UAVs that can autonomously and actively track the fire spreading boundary in a distributed manner. The UAV team can also effectively provide full coverage of the field and avoid in-flight collisions. Moreover, based on the proposed algorithm, some of the UAVs can automatically adjust their altitude to increase the image resolution of the border of the wildfire, while the whole team tries to maintain a complete view of it.

In the second part, we utilized a model-free learning algorithm to solve a problem of optimal coverage for a static field of arbitrary shape. The objective of the UAV team is not only to fully cover the field of interest, but also to minimize overlapping among field of views of the UAVs to increase image resolution and the efficiency of the team. Because the shape of the field is unknown, traditional approaches that rely on an accurate mathematical model of the field may fail. It is thus promising to address the problem with a model-free approach. We proposed a model-free Multi-Agent Reinforcement Learning (MARL) algorithm that combines Correlated Equilibrium strategy in Game Theory and Function approximation techniques to effectively overcome challenges in MARL such as the complex dynamics of the system and the curse of dimensionality.

From studying the two distinct approaches, we will draw some insights in solving optimal field coverage problems regarding each approach.

Dedication

This work is dedicated to Chi Pham & Quyen Dinh.

Acknowledgments

I am grateful to be supervised by my advisor, Dr. Hung La, who gave me wisdom, direction and instructions to help me learn and do research.

To my committee members, Dr. David Feil-Seifer who gave me advice and insight to improve the quality of my research and provided me with the wisdom I needed to achieve my goals and research, and Dr. Hao Xu, for the time he has taken to review this thesis and for his advice and encouragement many times.

Finally, I would like to thank my family, Quyen Dinh and Chi Pham, for their relentless support that helped me to earn my degree.

This work is supported by the National Aeronautics and Space Administration (NASA) Grant No. NNX15AI02H issued through the Nevada NASA Research Infrastructure Development Seed Grant, and the National Science Foundation #IIS-1528137.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Optimal field coverage problem refers to an active research branch that studies how we can use a finite set of sensors, such as camera, to optimally cover a field with arbitrary shape that can either be static or dynamically change over time. The problem arises in a wide range of applications, notably wildfires and oil spill tracking, military surveillance, and agriculture monitoring.

In these applications, it is of growing interest to send a team of Unmanned Aerial Vehicles (UAVs) acting as a mobile sensor network as they can provide sensing information with low costs and high flexibility, compared with traditional static monitoring methods [1] [2].

UAV technology continues to attract a huge amount of research [3,4]. Autonomous control algorithms for multirotor UAVs have been thoroughly studied [5, 6]. Researchers developed controllers for UAVs to help them attain stability and effectiveness in completing their tasks [7]. In [8–10], Wood et al. developed extended potential field controllers for a quadcopter that can track a dynamic target with smooth trajectory, while avoiding obstacles. A Model Predictive Control strategy was proposed in [11] for the same objective.

In the applications mentioned above, usually a team of UAVs could be deployed to increase the coverage range and reliability of the mission. Using multiple UAVs as a sensor network [12], especially in hazardous environment or disaster, is also well discussed. In [13,14], La et al. demonstrated how multiple UAVs can reach consensus to build a scalar field map of oil spills or fire. Maza et al. [15] provided a distributed decision framework for multi-UAV applications in disaster management. Specific applications, such as wildfire monitoring, involving multiple robots systems have been reported. In [16], multiple UAVs are commanded to track a spreading fire using checkpoints calculated based on visual images of the fire perimeter. Artificial potential field algorithms have been employed to control a team of UAVs in two separated tasks: track the boundary of a wildfire and suppress it [17]. A centralized optimal task allocation problem has been formulated in [18] to generate a set of waypoints for UAVs for shortest path planning. As with other multi-agent systems [19,20], the important challenges in designing an autonomous team of UAVs for field coverage include dealing with the dynamic complexity of the interaction between the UAVs so that they can coordinate to accomplish a common team goal.

1.2 Background on Optimal Coverage problem

Solutions for some instances of the Optimal Coverage problem have been proposed in the literature. Cortes et al. in [21] generalized the problem as a locational optimization problem. Schwager et al. in [22] presented a control law for a team of networked robots using Voronoi partitions for a generalized coverage problem. Subsequent works such as [23] and [24] expanded to work with non-convex environments with obstacles. Other solutions also used potential field methods [25, 26], or scalar field mapping [27, 28]. In [29], the coverage problem was expanded to also detect and track moving targets within a fixed environment.

Most of the aforementioned works only considered the coverage of a fixed, static environment. Also, in most of those works, authors made assumptions about the mathematical model of the environment, such as distribution models of the field or the predefined coverage path [30, 31]. In reality, however, it is very difficult to have an accurate model, because its data is normally limited or unavailable.

In this thesis, we address the problem by two distinct approaches. In the first part, we proposed a model-based control framework for UAV teaming to monitor and track a dynamic field like wildfire spreading. Wildfires are well-known for their destructive ability to inflict massive damage and disruption. We characterized the optimal sensing coverage problem to work with a changing wildfire environment. We proposed a decentralized control algorithm for a team of UAVs that can autonomously and actively track the fire spreading boundary in a distributed manner. The UAV team can also effectively provide full coverage of the field and avoid in-flight collision.

In the second part, we tackle a problem of optimal coverage with a model-free learning algorithm. The objective of the UAV team is not only to fully cover a static field of arbitrary shape, but also to minimize overlapping among field of views of the UAVs to increase image resolution and the efficiency of the team. We proposed a model-free Multi-Agent Reinforcement Learning (MARL) algorithm that combines Correlated Equilibrium strategy in Game Theory and Function approximation techniques to effectively overcome challenges in MARL such as the complex dynamics of the system and the curse of dimensionality.

From studying the two distinct approaches, we will draw some insights in solving optimal field coverage problems regarding each approach.

1.3 Content

The remainder of this thesis is organized as follows. In chapter 2, we proposed a model-based control framework for UAV teaming to monitor and track a dynamic field like wildfire spreading. We provide the problem statement in the first section 2.1. Section 2.2 discusses how wildfire spreading is modeled as an objective for this paper. In section 2.3, the wildfire tracking problem is formulated with clear objectives. In section 2.4, we propose a control design capable of solving the problem. A simulation scenario on MATLAB is provided in section 2.5. Section 2.6 provided a brief summary of the chapter.

In chapter 3, we present the context for a model-free approach and problem statement in section 3.1. Section 3.2 details on the optimal field coverage problem formulation. In section 3.3, we discuss our approach to solve the problem and the design of the learning algorithm. Basics in MARL will also be covered. We present our experimental result in section 3.4 with a comprehensive simulation, followed by an implementation with physical UAVs in a lab setting in section 3.5. A brief summary of the chapter is provided in section 3.6.

Finally, we draw a conclusion, and suggest directions for future work in chapter 4.

Chapter 2

Model-Based Approach

2.1 Problem Statement

Wildfire is well-known for their destructive ability to inflict massive damage and disruption. According to the U.S. Wildland Fire, an average of 70,000 wildfires burn around 7 million acres of land and destroy more than 2,600 structures annually [32]. Wildfire fighting is dangerous and time sensitive; lack of information about the current state and the dynamic evolution of fire contributes to many accidents [33]. Firefighters may easily lose their life if the fire unexpectedly propagates over them (Figure 2.1). Therefore, there is an urgent need to locate the wildfire correctly [34], and even more importantly to precisely observe the development of the fire to track its spreading boundaries [35]. The more information regarding the fire spreading areas collected, the better a scene commander can formulate a plan to evacuate people and property out of danger zones, as well as effectively prevent a fire from spreading to new areas.

Using Unmanned Aircraft Systems (UAS), also called Unmanned Aerial Vehicles (UAV) or drones, to assist wildfire fighting and other natural disaster relief is very promising. They can be used to assist humans for hazardous fire tracking tasks and replace the use of manned helicopters, conserving sizable operation costs in comparison with traditional methods [1] [2]. However, research that discusses the application of UAVs in assisting fire fighting remains limited [36].

Accurate UAV-based fire detection has been thoroughly demonstrated in current research. Merino et al. [1] proposed a cooperative perception system featuring infrared, visual camera, and fire detectors mounted on different UAV types. The system can precisely detect and estimate fire locations. Yuan et al. [37] developed a fire detection technique by analyzing fire segmentation in different color spaces. An efficient algorithm was proposed in [2] to work on UAV with low-cost cameras, using color index to distinguish fire from smoke, steam, and forest environment under fire, even in early stage. Merino et al. [38] utilized a team of UAVs to collaborate together to obtain fire front shape and position. In these works, cameras play a crucial role in capturing the raw information for higher level detection algorithms.

However, to the best of the authors' knowledge, most of the above mentioned work does not cover the behaviors of their system when the fire is spreading. Works in [16] and [18] centralized the decision making, thus potentially overloaded in computation and communication when the fire in large scale demands more UAVs. The team of UAVs in [17] can continuously track the boundary of the spreading fire but largely depends on the accuracy of the modeled shape function of the fire in control design.

In this chapter, we characterize the optimal sensing coverage problem to work with a changing environment. We propose a decentralized control algorithm for a team of UAVs that can autonomously and actively track the fire spreading boundaries in a distributed manner, without dependency on the wildfire modeling. The UAVs can effectively share the vision of the field, while maintaining safe distance in order to avoid in-flight collision. Moreover, during tracking, the proposed algorithm can allow

Figure 2.1: A wildfire outbreaks in California. Firefighting is really dangerous without continuous fire fronts growth information. Courtesy of USA Today.

the UAVs to increase image resolution captured on the border of the wildfire. This idea is greatly inspired by the work of Schwager et al. in [26], where a decentralized control strategy was developed for a team of robotic cameras to minimize the information loss over an environment. For safety reason, our proposed control algorithm also allows each UAV to maintain a certain height level to the ground to avoid getting caught by the fire [39].

2.2 Wildfire Model

Wildfire simulation has attracted significant research efforts over the past decades, due to the potential in predicting wildfire spreading. The core model of existing fire simulation systems is the fire spreading propagation [40]. Rothermel developed basic fire spread equations to mathematically and empirically calculate rate of speed and intensity in 1972 [41]. Richards [42] introduced a technique to estimate fire fronts growth using an elliptical model. These previous works were later developed further by Finney [43] and became a well-known fire growth model called Fire Area Simulator

(FARSITE). Among existing systems, FARSITE is the most reliable model [44], and widely used by federal land management agencies such as U.S. Department of Agriculture (USDA) Forest Service. However, in order to implement the model precisely, we need significant information regarding geography, topography, conditions of terrain, fuels, and weather. To focus on the scope of multi-UAV control rather than pursuing an accurate fire growth model, in this paper we modify the fire spreading propagation in FARSITE model to describe the fire front growth in a simplified model. We make the following assumptions:

- the model will be implemented for a discrete grid-based environment;
- the steady-state rate of spreading is already calculated for each grid;
- only the fire front points spread.

Originally, the equation for calculating the differentials of spreading fire front proposed in [42] and [43] as Equation (2.1):

$$\begin{aligned} X_t &= \frac{a^2 \cos\Theta(x_s \sin\Theta + y_s \cos\Theta) - b^2 \sin\Theta(x_s \cos\Theta - y_s \sin\Theta)}{\sqrt{b^2(x_s \cos\Theta + y_s \sin\Theta) - a^2(x_s \sin\Theta - y_s \cos\Theta}} + c\sin\Theta) \\ Y_t &= \frac{-a^2 \sin\Theta(x_s \sin\Theta + y_s \cos\Theta) - b^2 \cos\Theta(x_s \cos\Theta - y_s \sin\Theta)}{\sqrt{b^2(x_s \cos\Theta + y_s \sin\Theta) - a^2(x_s \sin\Theta - y_s \cos\Theta}} + c\cos\Theta), \end{aligned} \tag{2.1}$$

where X_t and Y_t are the differentials, Θ is the azimuth angle of the wind direction and y-axis ($0 \leq \Theta \leq 2\pi$). Θ increases following clock-wise direction. a and b are the length of semi-minor and semi-major axes of the elliptical fire shape growing from one fire front point, respectively. c is the distance from the fire source (ignition point) to the center of the ellipse. x_s and y_s are the orientation of the fire vertex. We simplify

Equation (2.1) to only retain the center of the new developed fire front as follows:

$$\begin{aligned} X_t &= c \sin \Theta \\ Y_t &= c \cos \Theta. \end{aligned} \tag{2.2}$$

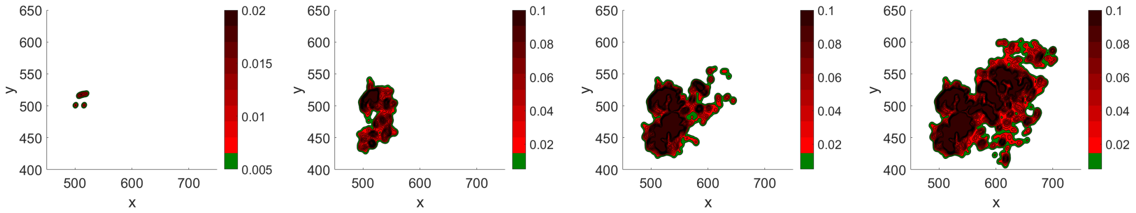
We use equation from Finney [43] to calculate c according to the set of equations (2.2) as follows:

$$\begin{aligned} c &= \frac{R - \frac{R}{HB}}{2} \\ HB &= \frac{LB + (LB^2 - 1)^{0.5}}{LB - (LB^2 - 1)^{0.5}} \\ LB &= 0.936e^{0.2566U} + 0.461e^{-0.1548U} - 0.397, \end{aligned} \tag{2.3}$$

where R is the steady-state rate of fire spreading. U is the scalar value of mid-flame wind speed, which is the wind speed at the ground. It can be calculated from actual wind speed value after taking account of the wind resistance by the forest. The new fire front location after time step δt is calculated as:

$$\begin{aligned} x_f(t + \Delta t) &= x_f(t) + \Delta t X_t(t) \\ y_f(t + \Delta t) &= y_f(t) + \Delta t Y_t(t). \end{aligned} \tag{2.4}$$

Additionally, in order to simulate the intensity caused by fire around each fire front source, we also assume that each fire front source would radiate energy to the surrounding environment resembling a multivariate normal distribution probability density function of its coordinates x and y. Assuming linearity, the intensity of each point in the field is a linear summation of intensity functions caused by multiple fire front sources. Therefore, we have the following equation describing the intensity of



(a) t = 0 (b) t = 1000 (c) t= 3000 (d) t = 6000

Figure 2.2: Simulation result showing a wildfire spreading at different time steps. The wildfire starts with a few heat sources around (500, 500), grows bigger and spreads around the field. The color bar indicates the intensity level of the fire. The darker the color, the higher the intensity.

each point in the wildfire caused by a number of k sources:

$$I(x,y) = \sum_{i=1}^{k} \frac{1}{2\pi\sigma_{x_i}\sigma_{y_i}} e^{-\frac{1}{2}[\frac{(x-x_f)^2}{\sigma_{x_i}^2}+\frac{(y-y_f)^2}{\sigma_{y_i}^2}]}, \tag{2.5}$$

where $I(x,y)$ is the intensity of the fire at a certain point $q(x,y)$, (x_f, y_f) is the location of the heat source i, and $(\sigma_{x_i}, \sigma_{y_i})$ are deviations. A point closer to the heat source has a higher level of intensity of the fire. Figure 2.2 represents the simulated wildfire spreading from original source (a) until $t = 6000$ time steps (d). The simulation assumes the wind flows north-east with direction is normally distributed ($\mu_\Theta = \frac{\pi}{8}, \sigma_\Theta = 1$), midflame adjusted wind speed is also normally distributed ($\mu_U = 5, \sigma_u = 2$). The green area depicts the boundary with forest field, while red area represents the fire. The brighter red color area illustrates the outer of the fire and regions near the boundary where the intensity is lower. The darker red colors show the area in fire with high intensity.

It should be noted that in this paper, the accuracy of the model should not affect the performance of our distributed control algorithm, as explained in section IV, subsection A. In case a different model of wildfire spreading is used, for instance, by changing equation set (2.2) and (2.3), only the shape of the wildfire changes, but the controller should still work.

2.3 Problem Formulation

In this section, we translate our motivation into a formal problem formulation. Our objective is to control a team of multiple UAVs for collaboratively covering a wildfire and tracking the fire front propagation. By covering, we mean to let the UAVs take multiple sub-pictures of the affected area so that most of the entire field is captured. We assume that the fire happens in a known section of a forest, where the priori information regarding the location of any specific point are made available. Suppose that when a wildfire happens, its estimated location is notified to the UAVs. A command is then sent to the UAV team allowing them to start. The team needs to satisfy the following objectives:

- *Deployment objective*: The UAVs can take flight from the deployment depots to the initially estimated wildfire location.

- *Coverage and tracking objective*: Upon reaching the reported fire location, the team will spread out to cover the entire wildfire from a certain altitude. The UAVs then follow and track the development of the fire fronts. When following the expanding fire fronts of the wildfire, some of the UAV team may lower their altitude to increase the image resolution of the fire boundary, while the whole team tries to maintain a complete view of the wildfire.

- *Collision avoidance and safety objective*: Because the number of UAVs can be large (i.e. for sufficient coverage a large wildfire), it is important to ensure that the participating UAVs are able to avoid in-flight collisions with other UAVs. Moreover, a safety distance between the UAVs and the ground should be established to prevent the UAVs from catching the fire.

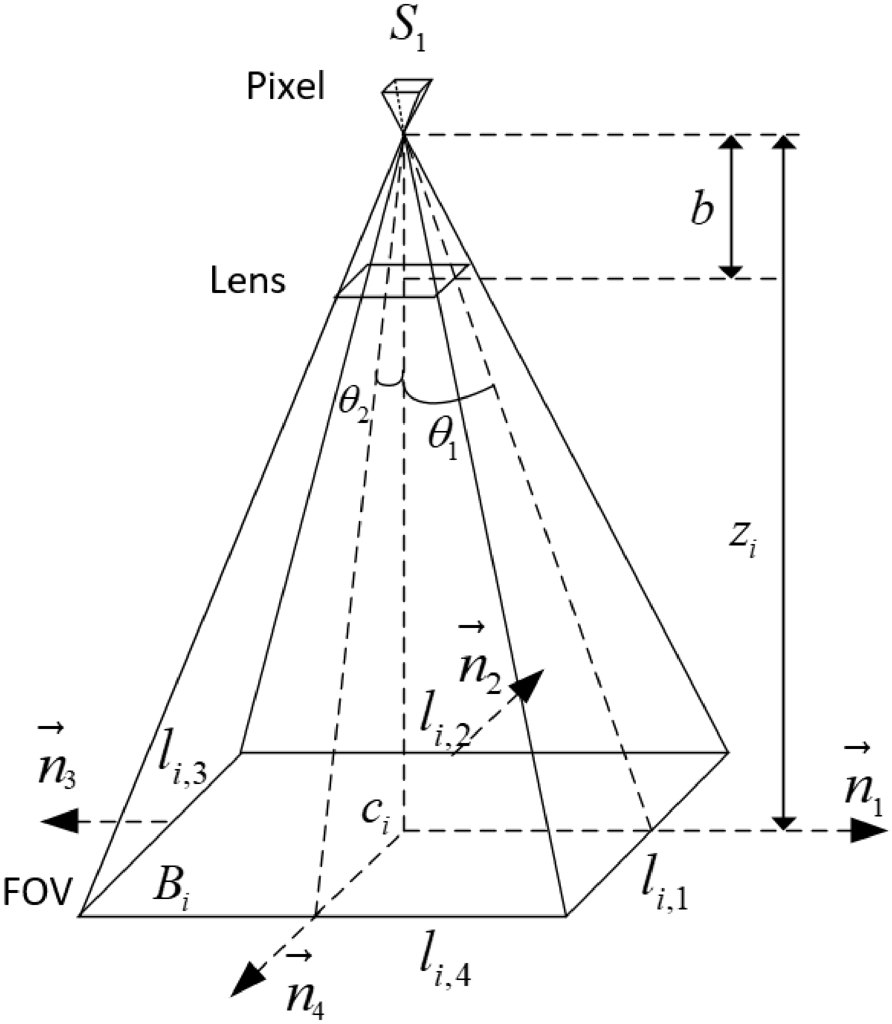


Figure 2.3: Rectangular FOV of a UAV, with half-angles θ_1, θ_2, composed of 4 lines $l_{i,1}, l_{i,2}, l_{i,3}, l_{i,4}$ and their respective normal vector n_1, n_2, n_3, n_4. Each UAV will capture the area under its field of view using its camera, and record the information into a number of pixels.

Assume that each UAV equipped with localization devices (such as GPS and IMU), and identical downward-facing cameras capable of detecting fire. Each camera has a rectangular *field of view* (FOV). When covering, the camera and its FOV form a pyramid with half-angles $\theta^T = [\theta_1, \theta_2]^T$ (see Figure 2.3). Each UAV will capture the area under its FOV using its camera, and record the information into an array of pixels. We also assume that a UAV can communicate and exchange information with other UAVs if it remains inside a communication sphere with radius r (see Figure 2.4).

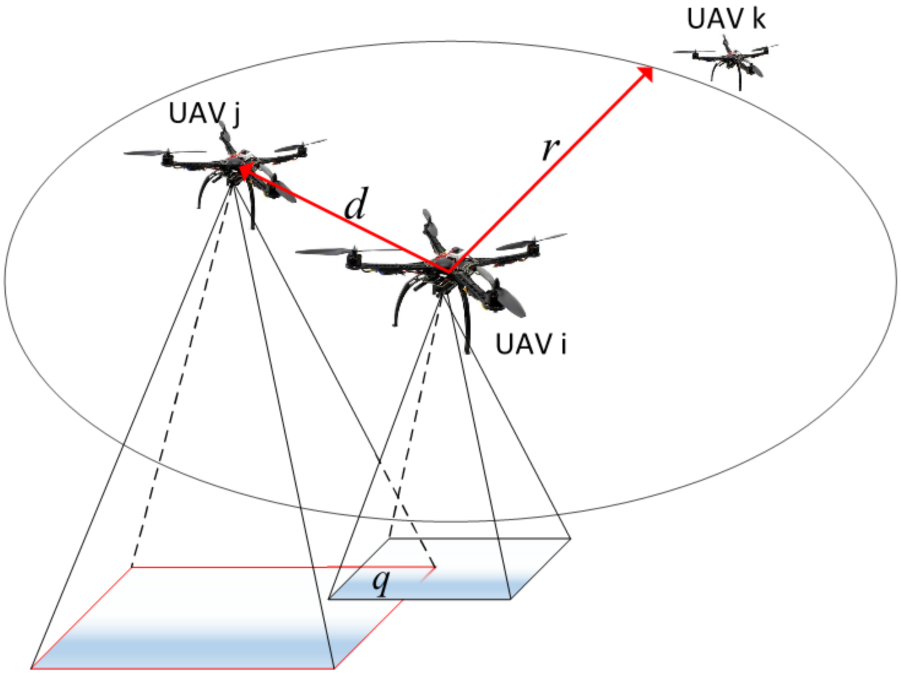


Figure 2.4: UAV i only communicates with a nearby UAV inside its communication range r (UAV j) (their *physical neighbor*). Each UAV would try to maintain a designed safe distance d to other UAVs in the team. If two physical neighboring UAVs cover one common point q, they are also *sensing neighbors*.

We define the following variables that will be used throughout this paper. Let N denote the set of the UAVs. Let $p_i = [c_i^T, z_i]^T$denote the pose of a UAV $i \in N$. In which, $c_i^T = [x_i, y_i]^T$ indicates the lateral coordination, and z_i indicates the altitude. Let B_i denote the set of points that lie inside the field of view of UAV i. Let $l_{k,i}, k = 1 : 4$ denotes each edge of the rectangular FOV of UAV i. Let $n_k, k = 1 : 4$ denotes the outward-facing normal vectors of each edge, where $n_1 = [1, 0]^T$, $n_2 = [0, 1]^T$, $n_3 = [-1, 0]^T$, $n_4 = [0, -1]^T$. We then define the objective function for each task of the UAV team.

2.3.1 Deployment objective

The UAVs can be deployed from depots distributed around the forest, or from a forest firefighting department center. Upon receiving the report of a wildfire, the UAVs are commanded to start and move to the point where the location of the fire was initially estimated. We call this point a rendezvous point $p_r = [p_x, p_y, p_z]^T$. The UAVs would keep moving toward this point until they can detect the wildfire inside their FOV.

2.3.2 Collision avoidance and safety objective

The team of UAVs must be able to avoid in-flight collisions. In order to do that, a UAV needs to identify its neighbors first. UAV i only communicates with a nearby UAV j that remains inside its communication range (Figure 2.4), and satisfies the following equation:

$$||p_j - p_i|| \leq r, \tag{2.6}$$

where $||.||$ denotes the Euclidean distance, and r is the communication range radius. If Equation (2.6) is satisfied, the two UAVs become *physical neighbors*. For UAV i to avoid collision with other neighbor UAV j, they must keep their distance not less than a designed distance d:

$$||p_j - p_i|| \geq d. \tag{2.7}$$

As we proposed earlier, during the implementation of the tracking and coverage task, the UAVs can lower their altitude to increase the resolution of the border of the wildfire. Since there is no obvious guarantee about the minimum altitude of the UAVs, they can keep lowering their altitude, and may be caught in the fire during their mission. Therefore, it is imperative that the UAVs maintain a safe distance to the ground. Suppose the safe altitude is z_{min}, and infer the position of the image of the UAV i as $p_{i'} = [c_i^T, 0]$, we have the safe altitude condition:

$$||p_i - p_{i'}|| \geq z_{min}. \tag{2.8}$$

2.3.3 Coverage and tracking objective

Let $Q(t)$ denote the wildfire varying over time t on a plane. The general optimal coverage problem is normally represented by a coverage objective function with the following form:

$$\min H(p_1, ..., p_n) = \int_{Q(t)} f(q, p_1, p_2, ..., p_n)\phi(q, t)dq, \tag{2.9}$$

where $f(q, p_1, p_2, ..., p_n)$ represents some cost to cover a certain point q of the environment. The function $\phi(q, t)$, which is known as distribution density function, level of interestingness, or strategic importance, indicates the specific weight of the point q in that environment at time t. In this paper, the cost we are interested in is the quality of images when covering a spreading fire with a limited number of cameras of the UAVs. This notion was first described in [26]. Since each camera has limited number of pixels to capture an image, it will provide one snapshot of the wildfire with lower resolution when covering it in a bigger FOV, and vice versa. By minimizing the information captured by the pixels of all the cameras, we could provide optimal-resolution images of the fire with respect to the developing fire fronts, while still fully covering the whole wildfire.

To quantify the cost, we first consider the image captured by one camera. Digital camera normally uses photosensitive electronics which can host a large number of pixels. The quality of recording an image by a single pixel can represent the quality of the image captured by that camera. From the relationship between object and image distance through a converging lens in classic optics, we can easily calculate the FOV area that a UAV covers (see figure 2.3) as follows:

$$f(q,p_i)=\frac{S_1}{b^2}(b-z_i)^2,\forall q\in B_i, \tag{2.10}$$

where $q^T=[q_x,q_y]^T$ is the coordination of a given point that belongs to $Q(t)$, S_1 is the area of one pixel of a camera, and b denotes the focal length. Note that, for a point q to lie on or inside the FOV of a UAV i, it must satisfy the following condition:

$$\frac{||q-c_i||}{z_i}\leq\tan\theta. \tag{2.11}$$

From Equation (2.10), it can be seen that the higher the altitude of the camera (z_i), the higher the cost the camera incurs, or the lower its image resolution is.

For multiple cameras covering a point q, Schwager et al. [26] formulated a cost to represent the coverage of a point q in a static field Q over total number of pixels from a multiple of n cameras as follows:

$$f_{N_q}(q,p_1,...,p_n)=(\sum_{i\in N_q}f(p_i,q)^{-1})^{-1}, \tag{2.12}$$

where $f(p_i,q)$ calculated as in equation (2.10), N_q is the set of UAVs that include the point q in their FOVs. However, in case the point q is not covered by any UAV, $f(p_i,q)=\infty$, the denominator in (2.12) can become zero. To avoid zero division, we need to introduce a constant m:

$$f_{N_q}(q,p_1,...,p_n)=(\sum_{i\in N_q}f(p_i,q)^{-1}+m)^{-1}. \tag{2.13}$$

The value of m should be very small, so that in such cases, the cost in (2.13) become very large, thus discouraging this case to happen. We further adapt the objective function (2.9) so that the UAVs will try to cover the field in the way that considers the region around the border of the fire more important. First, we consider that each fire front radiates a heat aura, as described in Equation (2.5). The border region of each fire front has the least heat energy, while the center of the fire front has the most intense level. We assume that the UAVs equipped with infrared camera allowing them to sense different color spectra with respect to the levels of fire heat intensity. Furthermore, the UAVs are assumed to have installed an on-board fire detection program to quantify the differences in color into varying levels of fire heat intensity [2]. Let $I(q)$ denote the varying levels of fire heat intensity at point q, and suppose that the cameras have the same detection range $[I_{min}, I_{max}]$. The desired objective function that weights the fire border region higher than at the center of the fire allows us to characterize the importance function as follows:

$$\phi(q) = \kappa(I_{max} - I(q)) \quad = \kappa \Delta I(q). \tag{2.14}$$

One may notice that the intensity $I(q)$ actually changes over time. This makes $\phi(q)$ depend on the time, and would complicate Equation (2.9) [29]. In this paper, we assume that the speed of the fire spreading is much less than the speed of the UAVs, therefore at a certain period of time, the intensity at a point can be considered constant. Also, note that some regions at the center of the wildfire may have $I = I_{max}$ now become not important. This makes sense because these regions likely burn out quickly, and they are not the goals for the UAV to track. We have the following objective function for wildfire coverage and tracking objective:

$$\min H = \int_{Q(t)} (\sum_{i \in N_q} f(p_i, q)^{-1} + m)^{-1} \kappa \Delta I(q) dq. \tag{2.15}$$

Note that when two UAVs have one or more points in common, they will become *sensing neighbors.* For a UAV to identify the set N_q of a point q inside its FOV, that UAV must know the pose of other UAVs as indicated by the condition (2.11). Therefore, in order to become sensing neighbors, the UAVs must first become physical neighbors, defined by (2.6). One should notice this condition to select the range radius r large enough to guarantee communication among the UAVs that have overlapping field of views. But we must also limit r so that communication overload does not occur as a result of having too many neighbors.

2.4 Controller Design

Figure 2.5 shows our controller architecture for each UAV. Our controller consists of two components: the coverage and tracking component and the potential field component. The coverage and tracking component calculates the position of the UAV for wildfire coverage and tracking. The potential field component controls the UAV to move to desired positions, and to avoid collision with other UAVs, as well as maintain the safety distance to the ground, by using potential field method. Upon reaching the wildfire region, the coverage and tracking control component will update the desired position of the UAV to the potential field control component. Assume the UAVs are quadcopters, then the dynamics of each UAV is:

$$u_i = \dot{p}_i, \tag{2.16}$$

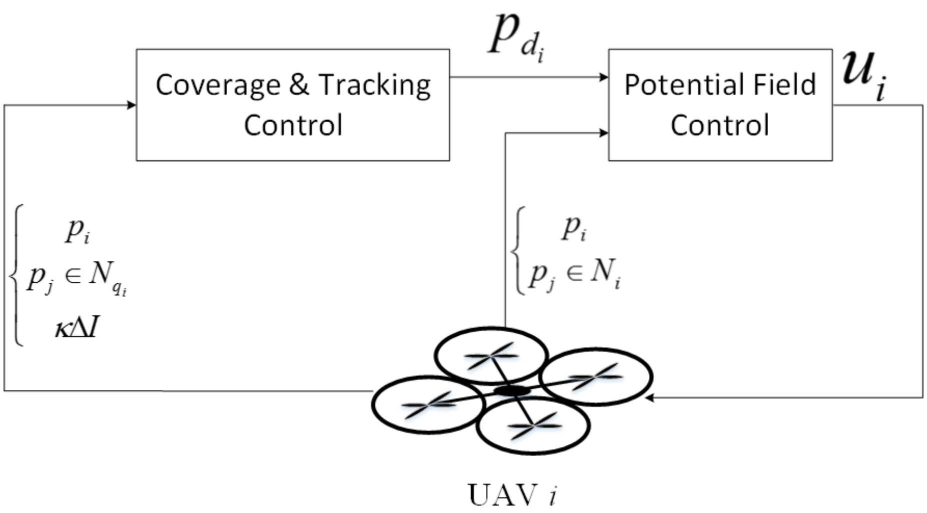


Figure 2.5: Controller architecture of UAV i, consisting of two components: the Coverage and Tracking component and the Potential Field component. The Coverage and Tracking component generates the desired position, p_{d_i}, for the UAV for wildfire coverage and tracking. The Potential Field component controls the UAV to move to the desired positions, which were generated by the Coverage & Tracking component, and to avoid collision with other UAVs and the ground.

we can then develop the control equation for each component in the upcoming subsections.

2.4.1 Coverage & tracking control

Based on the artificial potential field approach [26, 45, 46], each UAV is distributedly controlled by a negative gradient (gradient descent) of the objective function H in equation (2.15) with respect to its pose $p_i = [c_i, z_i]^T$ as follows:

$$u_i^{ct} = -k_s \frac{\partial H}{\partial p_i}, \tag{2.17}$$

where k_s is the proportional gain parameter. Taking the derivative with notation that $Q(t) = (Q(t) \cap B_i) \cup (\partial(Q(t) \cap B_i)) \cup (Q(t) \setminus B_i) \cup (\partial(Q(t) \setminus B_i))$ as in [26], where $\partial.$

denotes the boundaries of a set, we have:

$$\begin{aligned}\frac{\partial H}{\partial p_i} &= \frac{\partial}{\partial p_i}\int\limits_{Q(t)\cap B_i} f_{N_q}\Delta I dq + \frac{\partial}{\partial p_i}\int\limits_{\partial(Q(t)\cap B_i)} f_{N_q}\Delta I dq \\ &+ \frac{\partial}{\partial p_i}\int\limits_{\partial(Q(t)\setminus B_i)} f_{N_{q\setminus i}}\Delta I dq + \frac{\partial}{\partial p_i}\int\limits_{Q(t)\setminus B_i} f_{N_{q\setminus i}}\Delta I dq.\end{aligned} \tag{2.18}$$

In the last component, $Q(t)\setminus B_i$ does not depend on p_i so it is equal to zero. Then the lateral position and altitude of each UAV is controlled by taking the partial derivatives of the objective function H as follows:

$$\begin{aligned}\frac{\partial H}{\partial c_i} &= \sum_{k=1}^{4}\int\limits_{Q(t)\cap l_{k,i}} (f_{N_q} - f_{N_q\setminus i}) n_k \kappa \Delta I dq, \\ \frac{\partial H}{\partial z_i} &= \sum_{k=1}^{4}\int\limits_{Q(t)\cap l_{k,i}} (f_{N_q} - f_{N_q\setminus i}) \tan\theta^T n_k \kappa \Delta I dq, \\ &- \int\limits_{Q(t)\cap B_i} \frac{2 f_{N_q}^2}{\frac{S_1}{b^2}(b - z_i)^3}\kappa \Delta I dq,\end{aligned} \tag{2.19}$$

where f_{N_q} and $f_{N_{q\setminus i}}$ are calculated as in equation (2.13), $N_q \setminus i$ denotes the coverage neighbor set excludes the UAV i. In (2.19), the component $\frac{\partial H}{\partial c_i}$ allows the UAV to move along x-axis and y-axis of the wildfire area which has ΔI is larger, while reducing the coverage intersections with other UAVs. The component $\frac{\partial H}{\partial z_i}$ allows the UAV to change its altitude along the z-axis to trade off between cover larger FOV (the first component) over the wildfire and to have a better resolution of the fire fronts propagation (the second component). This set of equations is similar to the one proposed in [26], except that we extend them to work with an environment $Q(t)$, which now changes over the time, and the weight function $\phi(q)$ is characterized specifically to solve the dynamic wildfire tracking problem.

In order to compute these control inputs in (2.19), one needs to determine $Q(t) \cap B_i$ and $Q(t) \cap l_{k,i}$. This can be done through the discretization of B_i (i.e. area inside the FOV) and $l_{k,i}$ (i.e. the edges of the FOV) of a UAV i into discrete points, and checking if those points also belong to Q at time t. In other words, check the level of intensity of each point by using the fire detection system of the UAV. We need to assume the intensity model of the environment in (2.5) to hold true, hence our approach is still model-based. However, we would not need explicit information such as the accurate shape of the fire, as in [17], to implement the controller. This is an advantage, since it is more difficult to get an accurate shape model of the fire, comparing to the reasonable assumption of fire intensity model.

From (2.19), the desired virtual position p_{d_i} will be updated to the potential field control component (see Figure 2.5):

$$p_{d_i}(t + \Delta t) = p_{d_i}(t) - u_i^{ct} \Delta t, u_i^{ct} = (k_c \frac{\partial H}{\partial c_i}, k_z \frac{\partial H}{\partial z_i}). \tag{2.20}$$

2.4.2 Potential field control

The objective of the potential field control component is to control a UAV from the current position to a new position updated from the coverage and tracking control. Similarly, our approach is to create an artificial potential field to control each UAV to move to a desired position, and to avoid in-flight collision with other UAVs. We first create an attractive force to pull the UAVs to the initial rendezvous point p_r by using a quadratic function of distance as the potential field, and take the gradient of it to yield the attractive force:

$$\begin{aligned} U_r^{att} &= \frac{1}{2}k_r||p_r - p_i||^2 \\ u_i^r &= -\nabla U_r^{att} = -k_r(p_i - p_r). \end{aligned} \tag{2.21}$$

Similarly, the UAV moves to desired virtual position, p_{d_i}, passed from equation (2.20) in coverage & tracking component, by using this attractive force:

$$\begin{aligned} U_d^{att} &= \frac{1}{2}k_d||p_{d_i} - p_i||^2 \\ u_i^d &= -\nabla U_d^{att} = -k_d(p_i - p_{d_i}). \end{aligned} \tag{2.22}$$

In order to avoid collision with its neighboring UAVs, we create repulsive forces from neighbors to push a UAV away if their distances become less than a designed safe distance d. Define the potential field for each neighbor UAV j as:

$$U_j^{rep} = \begin{cases} \frac{1}{2}\nu(\frac{1}{||p_j - p_i||} - \frac{1}{d})^2, & if\ ||p_j - p_i|| < d \\ 0, & otherwise, \end{cases} \tag{2.23}$$

where ν is a constant. The repulsive force can be attained by taking the gradient of the sum of the potential fields created by all neighboring UAVs as follows:

$$\begin{aligned} u_i^{rep1} &= -\sum_{j \in N_i} a_{ij} \nabla U_j^{rep} \\ &= -\sum_{j \in N_i} \nu a_{ij} \Big(\frac{1}{||p_j - p_i||} - \frac{1}{d} \Big) \frac{1}{||p_j - p_i||^3}(p_i - p_j) \\ a_{ij} &= \begin{cases} 1, & if\ ||p_j - p_i|| < d \\ 0, & otherwise. \end{cases} \end{aligned} \tag{2.24}$$

Similarly, for maintaining a safe distance to the ground, we have:

$$\begin{aligned}
u_i^{rep2} &= -a_{ii'} \nabla U_{i'}^{rep} \\
&= -\nu' a_{ii'} \Big(\frac{1}{||p_{i'} - p_i||} - \frac{1}{z_{min}} \Big) \frac{1}{||p_{i'} - p_i||^3} (p_i - p_{i'}) \\
a_{ii'} &= \begin{cases} 1, & if \ ||p_{i'} - p_i|| < z_{min} \\ 0, & otherwise. \end{cases}
\end{aligned} \tag{2.25}$$

From (2.21), (2.22), (2.24), and (2.25), we have the general control law for the potential field control component:

$$\begin{aligned}
u_i = &- \sum_{j \in N_i} \nu a_{ij} \Big(\frac{1}{||p_j - p_i||} - \frac{1}{d} \Big) \frac{1}{||p_j - p_i||^3} (p_i - p_j) \\
&- \nu' a_{ii'} \Big(\frac{1}{||p_{i'} - p_i||} - \frac{1}{z_{min}} \Big) \frac{1}{||p_{i'} - p_i||^3} (p_i - p_{i'}) \\
&- (1 - \zeta_i) k_r (p_i - p_r) - \zeta_i k_d (p_i - p_{d_i}), \\
\zeta_i = &\begin{cases} 1, & if \ Q(t) \cap (B_i \cup l_{k,i}) \neq \varnothing \\ 0, & if \ otherwise. \end{cases}
\end{aligned} \tag{2.26}$$

Note that, during the time the UAVs travel to the wildfire region, the coverage control component would not work because the sets $Q(t) \cap B_i$ and $Q(t) \cap l_{k,i}$ are initially empty, so $\zeta_i = 0$. Upon reaching the waypoint region where the UAVs can sense the fire, $\zeta_i = 1$, that would cancel the potential forces that draw the UAVs to the rendezvous point and let the UAVs track the fire front growth. The final position of UAV i will be updated as follows:

$$p_i(t + \Delta t) = p_i(t) + u_i \Delta t. \tag{2.27}$$

2.4.3 Stability analysis

In this section, we study the stability of the proposed control framework. I assume that the two control components presented in subsection 2.4.1 and 2.4.2 are not coupled, so we can study the stability of the proposed controllers separately. The proof of the stability of the coverage and tracking controller (2.19) is similar to the proof in [26]. Choose a Lyapunov candidate function $V = H(p_1, p_2, ..., p_n)$, where H is the objective function in (2.15). Since H is the area under the FOVs of all UAVs multiplying with point-wise, positive importance index, H is positive definite for all $(p_1, p_2, ..., p_n)$. We have:

$$\begin{aligned}\dot{V} &= [\frac{\partial H}{\partial p_1}, \frac{\partial H}{\partial p_2}, ..., \frac{\partial H}{\partial p_n}]^T[\dot{p_1}, \dot{p_2}, ..., \dot{p_n}] \\ &= \sum_{i=1}^{n} \frac{\partial H}{\partial p_i}\dot{p_i} = \sum_{i=1}^{n} \frac{\partial H}{\partial p_i}(-k\frac{\partial H}{\partial p_i}) = -k\sum_{i=1}^{n}(\frac{\partial H}{\partial p_i})^2 \leq 0.\end{aligned} \tag{2.28}$$

Note that $\dot{V} = 0$ if and only if $p_i = p_i^*$ at local minima of H as in (2.15). Therefore, the equilibrium point $p_i = p_i^*$ is asymptotically stable according to the Lyapunov stability theorem. The potential field controller is a combination of repulsive and attractive artificial forces in two separatable phases. In the first phase, $\zeta_i = 0$, let $p = p_i - p_j$, $p' = p_i - p_{i'}$, $p_1 = p_i - p_r$, and choose a Lyapunov candidate function $V_1 = \frac{1}{2}p^2 + \frac{1}{2}p\prime^2 + \frac{1}{2}{p_1}^2$ which is positive definite, radially unbounded. We have:

$$\begin{aligned}\dot{V_1} &= p\dot{p} + p\prime\dot{p\prime} + p_1\dot{p_1} = pu_i^{rep1} + p\prime u_i^{rep2} + p_1 u_i^r \\ &= p_1(-kp_1) - \sum_{j\in N_i} \nu a_{ij}\Big(\frac{1}{||p||} - \frac{1}{d}\Big)\frac{1}{||p||^3}p^2 \\ &- \nu' a_{ii'}\Big(\frac{1}{||p\prime||} - \frac{1}{z_{min}}\Big)\frac{1}{||p\prime||^3}p\prime^2 \leq 0,\end{aligned} \tag{2.29}$$

since $\frac{1}{||p||} - \frac{1}{d} > 0$ and $\frac{1}{||p'||} - \frac{1}{z_{min}} > 0$. $V_1 = 0$ if and only if at equilibrium points. Therefore, the equilibrium points $p_i = p_r, p_i = p_j, p_i = p_{i'}$ are global asymptotically stable. The proof for second phase, $\zeta_i = 1$, is similar. In conclusion, the two controllers are asymptotically stable.

2.4.4 Overall algorithm for decentralized control of UAVs

We implemented the control strategy for the UAVs in a distributed manner as summarized in Algorithm 1. Each UAV needs to know its position from localization using means such as GPS+IMU sensor fusion with an Extended Kalman Filter (EKF) at each time step [47–49]. They can also communicate with other UAVs within the communication range to get their positions. Each UAV must also be able to read the heat intensity of any point under its FOV from the sensor. The coverage and tracking control component will calculate the new position for the UAV in each loop. To move to a new position, a UAV will use the potential field control component, which takes the new position as their input. To calculate the integrals in (2.19), we need to discretize the rectangular FOV of a UAV and its four edges in to a set of points, with Δq is either the length of a small line in each edge or the area of a small square. The integrals can then be transformed into the sum of all the small particles.

When activated, the UAV will first discretize its rectangular FOV into sets of points of a grid (line 3). These points will be classified into sets of edges $\hat{l_k}, k = 1:4$, and a set for the area inside the FOV $\hat{B}_i$, together with the value of Δq associated with each set. Then the UAV would read the intensity level of each point, $I(q)$, of these sets to determine if the point is currently in the fire or not, and form the set $Q(t) \cap \hat{B}_i$ and $Q(t) \cap \hat{l_{k,i}}$. If the sets $Q(t) \cap \hat{B}_i$ and $Q(t) \cap \hat{l_{k,i}}$ are not empty, then it would go to the rendezvous point (line 6). This will help the UAV to go to the right place in the initialization phase, as well as help the UAVs not to venture completely

```
Algorithm 1: Distributed wildfire tracking control.
  Input: Real-time localization the UAV $p_i$ and other neighbor UAVs $p_j, j \in N$.
         Heat intensity of each point $I(q)$ under the FOV
  Output: New position $p_i$
  for $i = 1 : N$ do
      Locate FOV of UAV $i$ and discretize them to get the set of points $\hat{B}_i$ and
        its four edges $\hat{l_{k,i}}, k = 1 : 4$
      Check if this point is on the fire $Q(t)$ to compute $Q(t) \cap \hat{B}_i$ and $Q(t) \cap \hat{l_{k,i}}$
      if $Q(t) \cap B_i = \varnothing$ then
          Calculate $u_i^{rep1}$, $u_i^{rep2}$ according to (2.24) and (2.25)
          Calculate $u_i$ according to (2.26):

              $u_i = u_i^{rep1} + u_i^{rep2} - k_r(p_i - p_r)$

            Update: $p_i(t + \Delta t) = p_i(t) - u_i \Delta t$
      else
          for $q \in Q(t) \cap \hat{B}_i \& Q(t) \cap \hat{l_{k,i}}$ do
              Compute $f_{N_q}$ and $f_{N_{q \backslash i}}$
              Estimate $\Delta I(q) = I_{max} - I(q)$
          Compute:

              $\frac{\Delta H}{\Delta c_i} = \sum_{k=1}^{4} \sum_{q \in Q(t) \hat{\cap} l_{k,i}} (f_{N_q} - f_{N_q \backslash i}) n_k \kappa \Delta I(q) \Delta q$
              $\frac{\Delta H}{\Delta z_i} = \sum_{k=1}^{4} \sum_{q \in Q(t) \hat{\cap} l_{k,i}} (f_{N_q} - f_{N_q \backslash i}) tan\theta^T n_k$
                  $\kappa \Delta I(q) \Delta q$
                  $- \sum_{q \in Q(t) \hat{\cap} B_i} \frac{2 f_{N_q}^2}{\frac{S_1}{b^2}(b - z_i)^3} \kappa \Delta I(q) \Delta q$

          Update: $p_{d_i}(t + \Delta t) = p_{d_i}(t) - (k_c \frac{\Delta H}{\Delta c_i}, k_z \frac{\Delta H}{\Delta z_i}) \Delta t$ Calculate $u_i^{rep1}$, $u_i^{rep2}$
            according to (2.24) and (2.25)
          Go to desired position $p_{d_i}$ according to (2.26):

              $u_i = u_i^{rep1} + u_i^{rep2} - k_d(p_i - p_{d_i})$

            Update: $p_i(t + \Delta t) = p_i(t) - u_i \Delta t$
```

out of the fire. If at least one set is not empty, it will then identify the set N_q and $N_{q\backslash i}$ by testing with equation (2.11), and compute $\Delta I(q)$, f_{N_q} and $f_{N_{q\backslash i}}$ as in (2.13), for every point in $Q(t) \cap \hat{B}_i$ and $Q(t) \cap \widehat{l_{k,i}})$. The integrals in (2.19) then can be calculated, and the new position p_{d_i} is then updated as in (2.20).

2.5 Simulation Result

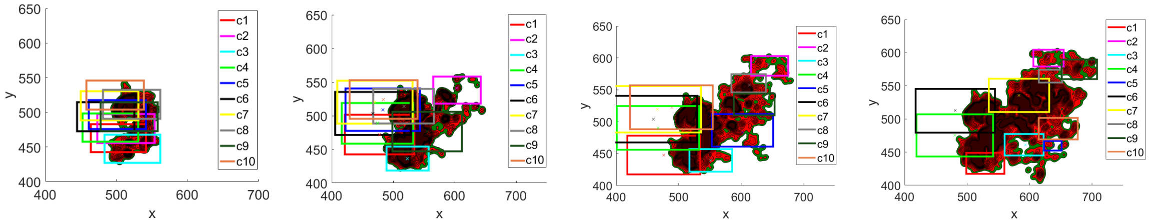
Our simulation was conducted in a Matlab environment. We started with 10 UAVs on the ground ($z_i = 0$) from a fire fighting center with initial location arbitrarily generated around $[300, 300]^T$. The safe distance was $d = 10$, and the safe altitude was $z_{min} = 15$. The UAVs were equipped with identical cameras with focal length $b = 10$, area of one pixel $S_1 = 10^{-4}$, half-angles $\theta_1 = 30°$, $\theta_2 = 45°$. We chose parameter $m = 1.5^{-5}$ to avoid zero division as in (2.13). The intensity sensitivity range of each camera was $[0.005, 0.1]^T$, and $\kappa = 1$. The wildfire started with five initial fire front points near $[500, 500]^T$. The regulated mid-flame wind speed magnitude followed a Gaussian distribution with $\mu = 5mph$ and $\sigma = 2$. The wind direction azimuth angle Θ also followed a Gaussian distribution with $\mu = \frac{\pi}{8}$ and $\sigma = 1$. The UAVs had a communication range $r = 500$.

We conducted tests in two different scenarios. In the first test, the UAVs performed wildfire coverage with specific focus on border of the fire, while in the second one, the UAVs have no specific focus on the border. In both of the two scenarios, the coverage and tracking controller parameters were $k_c = 10^{-9}$, $k_z = 2^{-10}$, while the potential field controller parameters were $k_r = k_d = 0.06$, $\nu = 2.1$ and $\nu\prime = 10^3$. The simulation parameters, presented in Table 2.1 and Table 2.2, were selected after some experiments.

Table 2.1: Simulation parameters for wildfire coverage with specific focus on border of the fire

Wind direction angle Θ		Wind speed magnitude U	
$\mu = \frac{\pi}{8} rad$	$\sigma = 1$	$\mu = 5mph$	$\sigma = 2$
Camera and sensing parameters			
$b = 10$	$S_1 = 10^{-4}$	$\theta_1 = 30°$	$\theta_2 = 45°$
$m = 1.5^{-5}$	$I_{min} = 0.005$	$I_{max} = 0.1$	$\kappa = 1$
Coverage & tracking		Safe distance	
$k_c = 10^{-9}$	$k_z = 2^{-10}$	$d = 10$	$z_{min} = 15$
Potential field controller's parameters			
$k_r = 0.06$	$k_d = 0.06$	$\nu = 2.1$	$\nu\prime = 10^3$

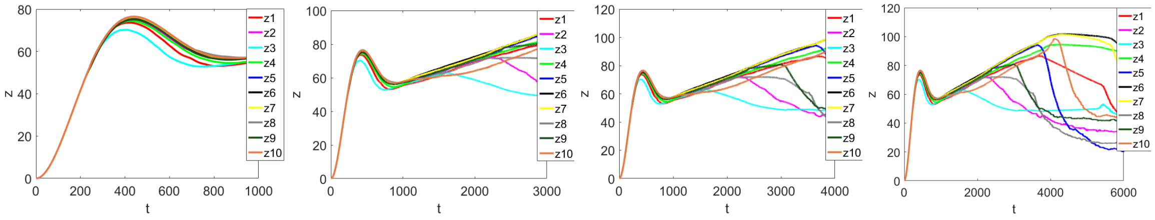
2.5.1 Scenario: Wildfire coverage with specific focus on border of the fire



(a) t = 1000 (b) t = 3000 (c) t= 4000 (d) t = 6000

Figure 2.6: Simulation result showing the FOV of each UAV on the ground in a) t = 1000, b) t = 3000, c) t= 4000, and d) t = 6000. The UAVs followed the newly developed fire front propagation.

The main parameters for the simulation were given in Table 2.1. We ran simulations in MATLAB for 6000 time steps which yielded the result as shown in Figures 2.6 and 2.7. The UAVs came from the ground at $t = 0$ (Figure 2.7), and drove toward the wildfire region. The initial rendezvous point was $p_r = [500, 500, 60]^T$. Upon reaching the region near the initial rendezvous point at $[500, 500]^T$, the UAVs spread out to cover the entire wildfire (Figure 2.6-a). As the wildfire expanded, the UAVs fragment



(a) t = 1000 (b) t = 3000 (c) t= 4000 (d) t = 6000

Figure 2.7: Plots showing the altitude of each UAV from the ground in a) t = 1000, b) t = 3000, c) t= 4000, and d) t = 6000. The UAVs change altitude from $z_i \approx 60$ to different altitudes, making the area of the FOV of each UAV is different.

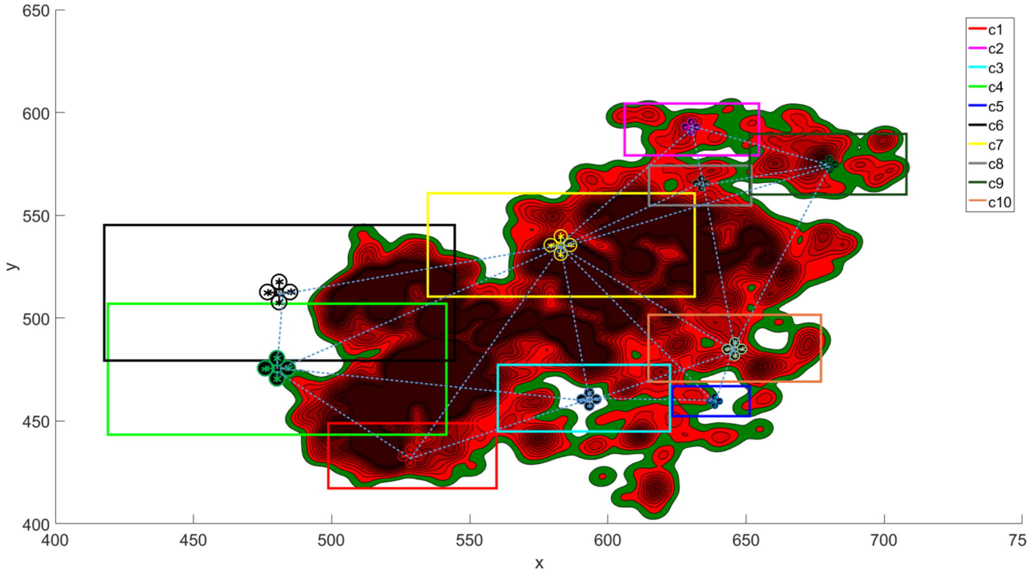


Figure 2.8: Rendering showing UAV positions and FOV during wildfire tracking, showing that the UAVs attempted to follow the fire front propagation, with greater focus on newly developed fire front.

and follow the fire border regions (Figure 2.6-b, c, d). Note that the UAVs may not cover some regions with intensity $I = I_{max}$ (represented by black-shade color). Some UAVs may have low altitude if they cover regions with small intensity I (for example, UAV 5 in this simulation). The UAVs change altitude from $z_i \approx 60$ (Figure 2.7-a) to different altitudes (Figure 2.7-b, c, d), hence the area of the FOV of each UAV is different. It is noticeable that the UAVs attempted to follow the fire front propagation, hence satisfying the tracking objective. Figure 2.8 indicates the position of

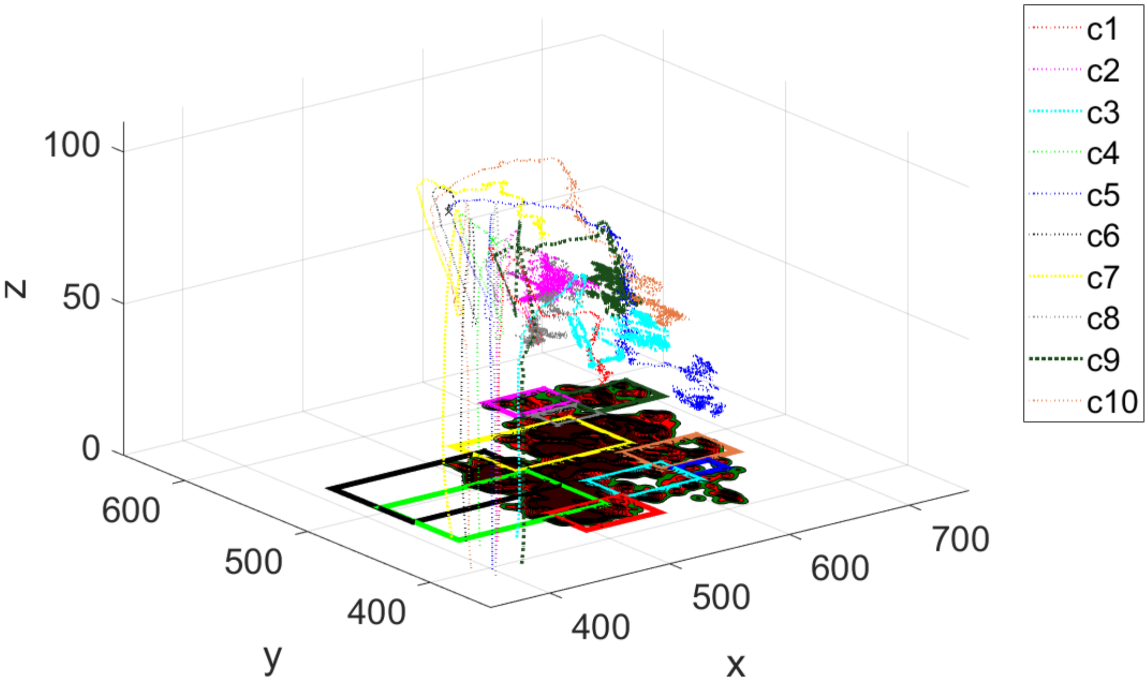


Figure 2.9: 3D representation of the UAVs showing the trajectory of each UAV in 3-dimensions while tracking the wildfire spreading north-east, and their current FOV on the ground.

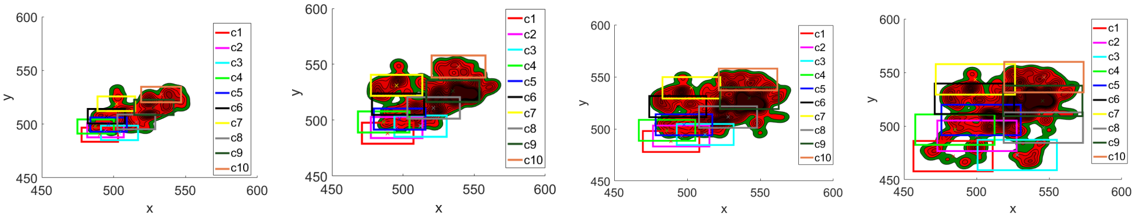
each UAV and its respective FOV in the last stage $t = 6000$. UAVs that are physical neighbors are connected with a dashed blue line. We can see that most UAVs have sensing neighbors. Figure 2.9 shows the trajectory of each UAV in 3-dimensions while tracking the wildfire spreading north-east, and their current FOV on the ground.

2.5.2 Scenario: Normal wildfire coverage

In this simulation scenario, we demonstrate the ability of the group of UAVs to cover the spreading fire with no specific focus. The main simulation parameters for this scenario were given in Table 2.2. The control strategy and parameters were the same as in the previous scenario, except there was no special interest in providing higher-resolution images of the fire border, therefore, equation 2.14 became $\phi(q) = \kappa$. The

Table 2.2: Simulation parameters for normal wildfire coverage with no specific focus

Wind direction angle Θ		Wind speed magnitude U	
$\mu = \frac{\pi}{8} rad$	$\sigma = 1$	$\mu = 5mph$	$\sigma = 2$
Camera and sensing parameters			
$b = 10$	$S_1 = 10^{-4}$	$\theta_1 = 30°$	$\theta_2 = 45°$
$m = 1.5^{-5}$	$I_{min} = N/A$	$I_{max} = N/A$	$\kappa = 10^{-3}$
Coverage & tracking		Safe distance	
$k_c = 10^{-9}$	$k_z = 2^{-10}$	$d = 10$	$z_{min} = 15$
Potential field controller's parameters			
$k_r = 0.06$	$k_d = 0.06$	$\nu = 2.1$	$\nu\prime = 10^3$



(a) t = 1000

(b) t = 3000

(c) t= 4000

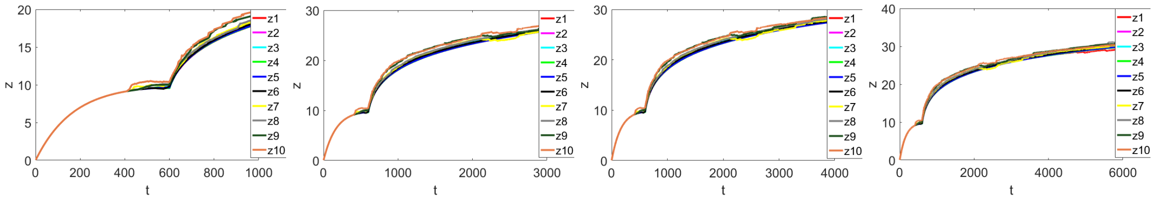
(d) t = 6000

Figure 2.10: Simulation results showing the FOV of each UAV on the ground in a) t = 1000, b) t = 3000, c) t= 4000, and d) t = 6000. The whole wildfire got covered, with no specific focus.

initial rendezvous point was $p_r = [500, 500, 10]^T$. As we can see in Figures 2.10 and 2.11, the UAVs covered the fire spreading very well, with no space uncovered. Since they were not focusing on the border of the fire, the altitudes of the UAVs were almost equal.

2.6 Summary

In this chapter, I formulated a dynamic optimal coverage problem in which a team of UAVs try to collaboratively cover a wildfire spreading and track its development.



(a) t = 1000 (b) t = 3000 (c) t= 4000 (d) t = 6000

Figure 2.11: Plots showing the altitude of each UAV on the ground in a) t = 1000, b) t = 3000, c) t= 4000, and d) t = 6000. Since they were not focusing on the border of the fire, the altitudes of the UAVs were almost equal.

I also described how a spreading wildfire is modeled. I designed a distributed control algorithm based on gradient descent and potential field methods for the UAVs to follow the border region of the wildfire as it keeps expanding, while still trying to maintain coverage of the whole wildfire. The UAVs are also capable of avoiding collision, maintaining safe distance to fire level, and flexible in deployment. The system is validated by a simulation in MATLAB.

Chapter 3

Cooperative Reinforcement Learning Algorithm for a team of UAVs for Static Field Coverage

3.1 Problem Statement

In the previous chapter, we showed that although model-based method is popular and efficient in solving the Optimal sensing coverage problem, the performance of the controller will certainly depend on the accuracy of the model. In reality, however, it is very difficult to have an accurate model, because its data is normally limited or unavailable.

Model-free learning algorithms, such as Reinforcement learning (RL), would be a natural approach to address the aforementioned challenges relating to the required accurate mathematical models for the environment, and the complex behaviors of the system. These algorithms will allow each agent in the team to learn new behavior, or reach consensus with others [7], without depending on a model of the environment [50]. Among them, RL is popular because it is relatively generic to address a wide range of problems, while it is simple to implement.

Classic individual RL algorithms have already been extensively researched in UAV applications. Previous papers focus on applying RL algorithm into UAV control to achieve desired trajectory tracking/following [51], or discussion of using RL to improve the performance in UAV application [52]. Multi-Agent Reinforcement Learning (MARL) is also an active field of research. In multi-agent systems (MAS), agents' behaviors cannot be fully designed as *priori* due to the complicated nature, therefore, the ability to learn appropriate behaviors and interactions will provide a huge advantage for the system. This particularly benefits the system when new agents are introduced, or the environment is changed [53]. Recent publications concerned the possibility of applying MARL into a variety of applications, such as in autonomous driving [54], or traffic control [55].

In robotics, efforts have been focused on robotic system coordination and collaboration [56], transfer learning [57], or multi-target observation [58]. For robot path planning and control, most prior research focuses on classic problems, such as navigation and collision avoidance [59], object carrying by robot teams [60], or pursuing prey/avoiding predators [61, 62]. Many other papers in multi-robotic systems even simplified the dynamic nature of the system to use individual agent learning such as classic RL algorithms [63], or the actor-critic model [64]. To our best knowledge, not so many works available addressed the complexity of MARL in a multi-UAV system and their daily missions such as optimal sensing coverage. In this chapter, we propose how a MARL algorithm can be applied to solve an optimal coverage problem. We address two challenges in MARL: (1) the complex dynamic of the joint-actions of the UAV team, that will be solved using game-theoric correlated equilibrium, and (2) the challenge in huge dimensional state space will be tackled with an efficient space-reduced representation of the value function.

3.2 Problem Formulation

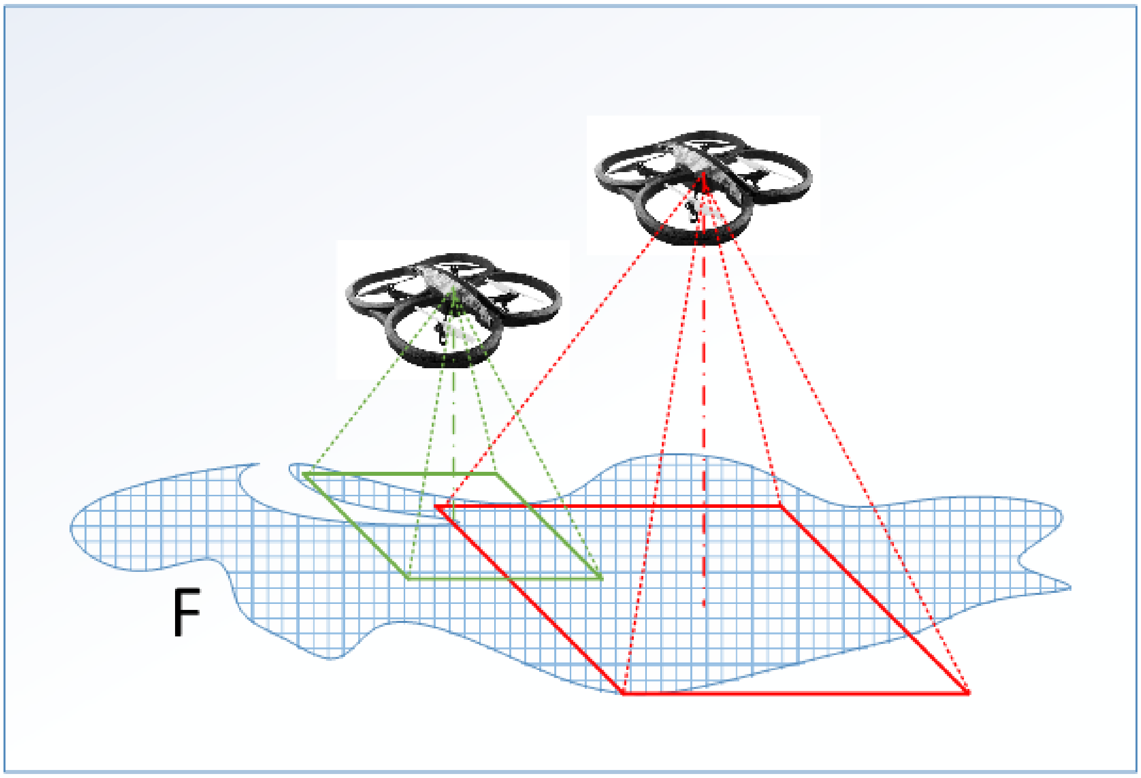


Figure 3.1: A team of UAVs to cover a field of interest F. A UAV can enlarge the FOV by flying higher, but risk in getting overlapped with other UAVs in the system. Minimizing overlap will increase the field coverage and resolution.

In a mission like exploring a new environment such as monitoring an oil spill or a wildfire area, it is growing interest to send out a fleet of UAVs acting as a mobile sensor network, as it provides many advantages comparing to traditional static monitoring methods [27]. In such a mission, the UAV team needs to surround the field of interest to get more information, for example, visual data. We made an assumption that the field could be discretized and represented by a finite number of grids. Suppose that we have a team of quadrotor-type UAVs (Figure 3.1). Each UAV is an independent decision maker, thus the system is distributed. They can localize themselves using on-board localization systems, such as using GPS. They can also exchange information with other UAVs through communication links. Each UAV equipped with identical downward facing cameras provides it a square field of view (FOV). The camera of each UAV and its FOV form a pyramid with half-angles $\theta^T = [\theta_1, \theta_2]^T$ (Figure 3.2). A point q is covered by the FOV of UAV i if it satisfies the following equations:

$$\frac{||q - c_i||}{z_i} \leq \tan\theta^T, \tag{3.1}$$

where c_i is the lateral-projected position, and z_i is the altitude of the UAV i, respectively. The objective of the team is not only to provide a full coverage over the shape of the field of interest F under their UAVs' FOV, but also to minimize overlapping other UAVs' FOV to improve the efficiency of the team (e.g., minimizing overlap can increase resolution of field coverage). A UAV covers F by trying to put a section of it under its FOV. It can enlarge the FOV to cover a larger area by increasing the altitude z_i according to (3.1), however it may risk overlapping other UAVs' FOV in doing so. Formally speaking, let us consider a field F of arbitrary shapes. Let $p_1, p_2, ..., p_m$ denote the positions of m UAV $1, 2, ..., m$, respectively. Each UAV i has a square FOV projected on the environment plane, denoted as B_i. Let $f(q, p_1, p_2, ..., p_m)$ represent a combined area under the FOVs of the UAVs. The team has a cost function H represented by:

$$\begin{aligned} H = & \int_{q\in F} f(q, p_1, p_2, ..., p_m)\Phi(q)dq \\ & - \int_{q\in B_i\cap B_j, \forall i,j\in m} f(q, p_1, p_2, ..., p_m)\Phi(q)dq, \end{aligned} \tag{3.2}$$

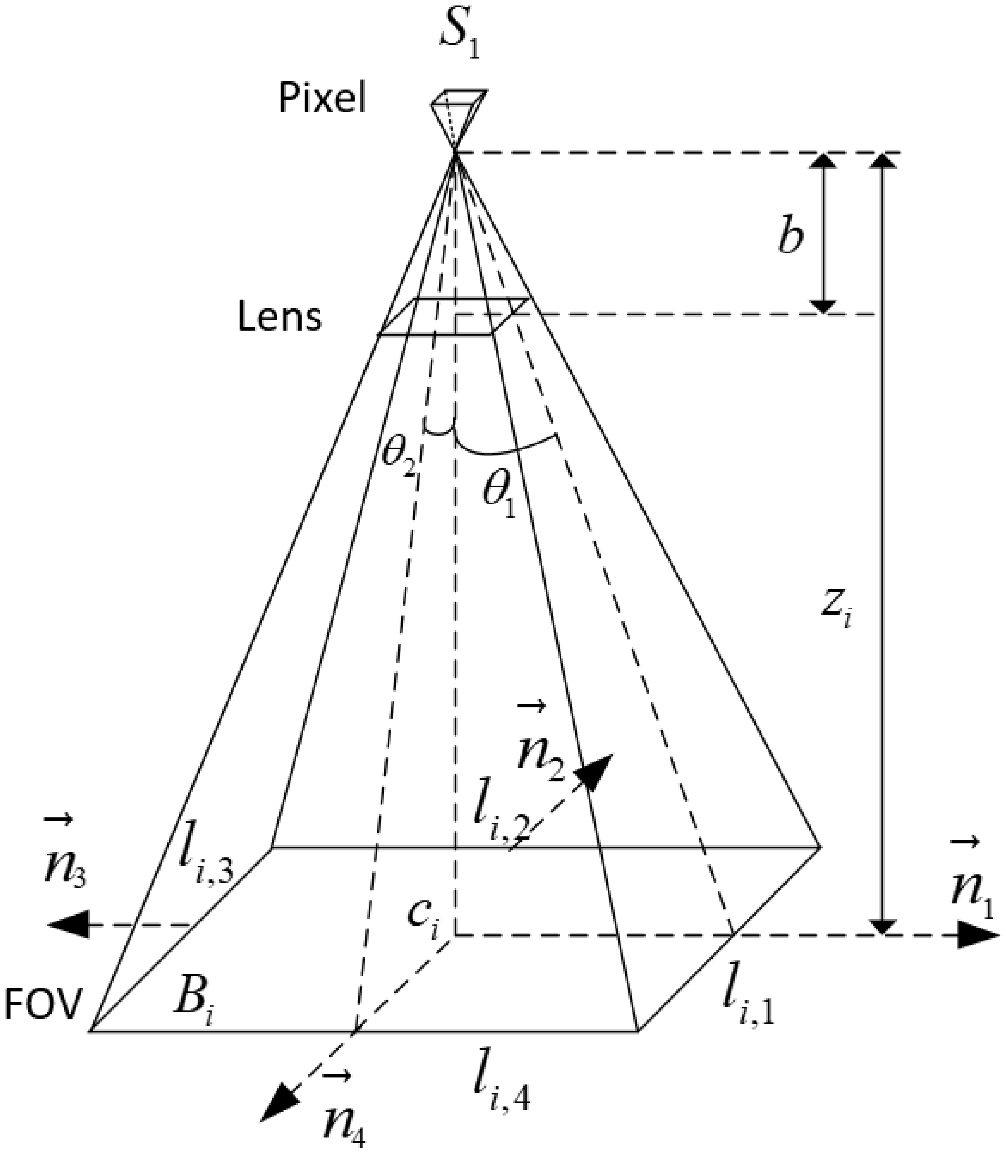


Figure 3.2: Field of view of each UAV.

where $\Phi(q)$ measures the importance of a specific area. In a plain field of interest, $\Phi(q)$ is constant.

The problem can be solved using traditional methods, such as, using Voronoi partitions [22,23], or using potential field methods [25,26]. Most of these works proposed model-based approaches, where authors made assumptions about the mathematical model of the environment, such as the shape of the target [30,31]. In reality, however, it is very difficult to obtain an accurate model, because the data of the environment is normally insufficient or unavailable. This can be problematic, as the systems may fail if using incorrect models. On the other hand, many learning algorithms, such as RL algorithms, rely only on the data obtained directly from the system, would be a natural option to address the problem.

3.3 Algorithm Design

3.3.1 Reinforcement Learning and Multi-Agent Reinforcement Learning

Classic RL defines the learning process happens when a decision maker, or an agent, interacts with the environment. During the learning process, the agent will select the appropriate actions when presented a situation at each state according to a policy π, to maximize a numerical reward signal, that measures the performance of the agent, feedback from the environment. In MAS, the agents interact with not only the environment, but also with other agents in the system, making their interactions more complex. The state transition of the system is more complicated, resulting from a join action containing all the actions of all agents taking at a time step. The agents in the system now must also consider other agents states and actions to coordinate and/or compete with. Assuming that the environment has Markovian property, where the next state and reward of an agent only depends on the current state, the Multi-Agent Learning model can be generalized as a Markov game $< m, \{S\}, \{A\}, T, R >$, where:

- m is the number of agents in the system.
- $\{S\}$ is the joint state space $\{S\} = \mathbf{S_1} \times \mathbf{S_2} \times ... \times \mathbf{S_m}$, where $\mathbf{S_i}, i = 1, ..., m$ is the individual state space of an agent i. At time step k, the individual

state of agent i is denoted as $s_{i,k}$. The joint state at time step k is denoted as $S_k = \{s_{1,k}, s_{2,k}, ..., s_{m,k}\}$.

- $\{A\}$ is the joint action space, $\{A\} = \mathbf{A_1} \times \mathbf{A_2} \times ... \times \mathbf{A_m}$, where $\mathbf{A_i}, i = 1, ..., m$ is the individual action space of an agent i. Each joint action at time k is denoted as $A_k \in \{A\}$ while the individual action of agent i is denoted as $a_{i,k}$. We have: $A_k = \{a_{1,k}, a_{2,k}, ..., a_{m,k}\}$.

- T is the transition probability function, $T : S \times \mathbf{A} \times \mathbf{S} \to [0, 1]$, is the probability of agent i that takes action $a_{i,k}$ to move from state $s_{i,k}$ to state $s_{i,k+1}$. Generally, it is represented by a probability: $T(s_{i,k}, a_{i,k}) = P(s_{i,k+1}|s_{i,k}, a_{i,k}) = P_i(a_k)$.

- R is the individual reward function: $R : \mathbf{S} \times \mathbf{A} \to \mathbb{R}$ that specifies the immediate reward of the agent i for getting from $s_{i,k}$ at time step k to state $s_{i,k+1}$ at time step $k + 1$ after taking action $a_{i,k}$. In MARL, the team has a global reward $GR : \{S\} \times \{A\} \to \mathbb{R}$ in achieving the team's objective. We have: $GR(S_k, A_k) = r_{k+1}$.

The agents seek to optimize expected rewards in an *episode* by determining which action to take that will have the highest return in the long run. In single agent learning, a value function $Q(s_k, a_k)$, $\mathbf{A} \times \mathbf{S} \to \mathbb{R}$, helps quantify strategically how good the agent will be if it takes an action a_k at state s_k, by calculating its expected return obtained over an episode. In MARL, the action-state value function of each agent also depends on the joint state and joint action [65], represented as:

$$Q(s_{i,k}, a_{i,k}, s_{-i,k}, a_{-i,k}) = Q(S_k, A_k) = E\{\sum_{k}^{\infty} \gamma r_{i,k+1}\}, \tag{3.3}$$

where $0 < \gamma \leq 1$ is the discount factor of the learning. This function is also called Q-function. It is obvious that the state space and action space, as well as the value function in MARL is much larger than in individual RL, therefore MARL would require much larger memory space, that will be a huge challenge concerning the scalability of the problem.

3.3.2 Correlated Equilibrium

In order to accomplish the team's goal, the agents must reach consensus in selecting actions. The set of actions that they agreed to choose is called a joint action, $A_k \in \{A\}$. Such an agreement can be evaluated at equilibrium, such as Nash equilibrium (NE) [59] or Correlated equilibrium (CE) [66]. Unlike NE, CE can be solved with the help of linear programming (LP) [67]. Inspired by [66] and [60], in this work we use a strategy that computes the optimal policy by finding the CE equilibrium for the agents in the systems. From the general problem of finding CE in game theory [67], we formulate a LP to help find the stable action for each agent as follows:

$$\begin{aligned}
&\pi(A_k) = \arg\max_{A_k}\{\sum_{i=1}^{m} Q_{i,k}(S_k, A_k))P_i(a_k)\}. \\
&\text{subject to:} \\
&\sum_{a_k \in \mathbf{A_i}} P_i(a_k) = 1, \forall i \in \{m\} \\
&P_i(a_k) \geq 0, \forall i \in \{m\}, \forall a_k \in \mathbf{A_i} \\
&\sum_{a'_k \in \mathbf{A_i}} [Q_{i,k}(S_k, a_k, A_{k,-i}) - Q_{i,k}(S_k, a'_k, A_{k,-i})]P_i(a_k) \\
&\geq 0, \forall i \in \{m\}.
\end{aligned} \tag{3.4}$$

Here, $P_i(a_k)$ is the probability of UAV i selecting action a at time k, and A_{-i} denotes the rest of the actions of other agents. Solving LP has long been researched by the optimization community. In this work, we use a state-of-the-art program from the community to help us solve the above LP.

3.3.3 Learning Design

In this section, we design a MARL algorithm to solve our problem formulated in section 3.2. We assume that the system is fully observable. We also assume the UAVs are identical, and operated in the same environment, and have identical sets of states and actions: $\mathbf{S_1} = \mathbf{S_2} = ... = \mathbf{S_m}$, and $\mathbf{A_1} = \mathbf{A_2} = ... = \mathbf{A_m}$.

The state space and action space set of each agent should be represented as discrete finite sets approximately, to guarantee the convergence of the RL algorithm [68]. We consider the environment as a 3-D grid, containing a finite set of cubes, with the center of each cube represents a discrete location of the environment. The state of an UAV i is defined as its approximate position in the environment, $s_{i,k} \triangleq [x_c, y_c, z_c] \in \mathbf{S}$, where x_c, y_c, z_c are the coordinates of the center of a cube c at time step k. The objective equation (3.2) now becomes:

$$\max_{S_k \in \{S\}} H = \arg\max_{S_k \in \{S\}} \{\sum_i f_i(S_k) - \sum_i o_i(S_k)\}, \tag{3.5}$$

where $f_i : \{S\} \to \mathbb{R}$ is the count of squares, or cells, approximating the field F under the FOV of UAV i, and $o_i : \{S\} \to \mathbb{R}$ is the total number of cells overlapped with other UAVs.

To navigate, each UAV i can take an action $a_{i,k}$ out of a set of six possible actions A: heading North, West, South or East in lateral direction, or go Up or Down to change the altitude. Note that if the UAV stays in a state near the border of the environment, and selects an action that takes it out of the space, it should stay still in the current state. Certainly, the action $a_{i,k}$ belongs to an optimal joint-action strategy A_k resulted from (3.4). Note that in case multiple equilibrium exists, since each UAV is an independent agent, they can choose different equilibrium, making their respective actions deviate from the optimal joint action to a sub-optimal joint action. To overcome this, we employ a mechanism called *social conventions* [69], where the UAVs take turn to carry out an action. Each UAV is assigned with a specific ranking order. When considering the optimal joint action sets, the one with higher order will have priority to choose its action first, and let the subsequent one know its action. The other UAVs then can match their actions with respect to the selected action. To ensure collision avoidance, lower-ranking UAVs cannot take an action that will lead to the newly occupied states of higher-ranking UAVs in the system. By this, at a time step k, only one unique joint action will be agreed among the UAV's.

Defining the reward in MARL is another open problem due to the dynamic nature of the system [53]. In this paper, the individual reward that each agent receives can be considered as the total number of cells it covered, minus the cells overlapping with other agents. However, a global team goal would help the team to accomplish the task quicker, and also speed up the learning process to converge faster [60]. We define the global team's reward is a function $GR : \{S\} \times \{A\} \to \mathbb{R}$ that weights the entire team's joint state S_k and joint action A_k at time step k in achieving (3.5). The agent only receives reward if the team's goal reached:

$$GR(S_k, A_k) = \begin{cases} r, & if \ \sum_i f_i(S_k) \geq fb, \sum_i o_i(S_k) \leq 0 \\ 0, & otherwise. \end{cases} \tag{3.6}$$

where $fb \in \mathbb{R}$ is an acceptable bound of the field being covered. During the course of learning, the state - action value function $Q_{i,k}(s_i, a_i)$ for each agent i at time k can be iteratively updated as in Multi-Agent Q - learning algorithm, similar to those proposed in [60, 65]:

$$\begin{aligned} Q_{i,k+1}(S_k, A_k) \leftarrow (1-\alpha)Q_{i,k}(S_k, A_k) + \alpha[GR(S_k, A_k) \\ + \gamma \max_{A' in\{A\}} Q_{i,k}(S_{k+1}, A')], \end{aligned} \tag{3.7}$$

where $0 < \alpha \leq 1$ is the learning rate, and γ is the discount rate of the RL algorithm. The term $\max\limits_{A' in\{A\}} Q_{i,k}(S_{k+1}, A')$ derived from (3.4) at joint state S_{k+1}.

3.3.4 Approximate Multi-Agent Q-learning

In MARL, each agent updates its value function with respect to other agents' state and action, therefore the state and action variable dimensions can grow exponentially if we increase the number of agent in the system. This makes value function representation a challenge. Consider the value function $Q_{i,k+1}(S_k, A_k)$ in (3.3), the space needed to store all the possible state - action pairs is $|\mathbf{S_1}| \cdot |\mathbf{S_2}| ... \cdot |\mathbf{S_m}| \cdot |\mathbf{A_1}| \cdot |\mathbf{A_2}| ... |\mathbf{A_m}| = |\mathbf{S_i}|^m |\mathbf{A_i}|^m$.

Works have been proposed in the literature to tackle the problem: using graph theory [70] to decompose the global Q-function into a local function concerning only a subset of the agents, reducing dimension of Q-table [71], or eliminating other agents to reduce the space [72]. However, most previous approaches require additional step to reduce the space, that may place more pressure on the already-intense calculation time. In this work, we employ simple approximation techniques [73]: Fixed Sparse Representation (FSR) and Radial Basis Function (RBF) to map the original Q to a parameter vector θ by using state and action - dependent basis functions $\phi : \{S\} \times \{A\} \to \mathbb{R}$:

$$\hat{Q}_{i,k}(S_k, A_k) = \sum_l \phi_l(S_k, A_k)\theta_{i,l} = \phi^T(S_k, A_k)\theta_i, \tag{3.8}$$

The FSR scheme uses a column vector $\phi(S, A)$ of the size $D \cdot |\{A\}|$, where D is the sum of *dimensions* of the state space. For example, if the state space is a 3-D space: $X \times Y \times Z$, then $D = X + Y + Z$. Each element in ϕ is defined as follows:

$$\phi(x, y) = \begin{cases} 1, & if\ x = S_k, y = A_k; \\ 0, & otherwise. \end{cases} \tag{3.9}$$

In RBF scheme, we can use a column vector ϕ of $l \cdot |\{A\}|$ element, each can be calculated as:

$$\phi(l, y) = \begin{cases} e^{-\frac{S_k - c_l}{2\mu_l^2}}, & ify = A_k; \\ 0, & otherwise, \end{cases} \tag{3.10}$$

where c_l is the center and μ_l is the radius of l pre-defined basis functions that have the shape of a Gaussian bell.

The $\phi(S, A)$ and θ_i in FSR and RBF schemes are column vectors of the size $D \cdot |\{A\}|$ and $l \cdot |\{A\}|$, respectively, which is much less than the space required in the original Q-value function. For instance, if we deploy 3 agents on a space of $7 \times 7 \times 4$, and each agent has 6 actions, the original Q-table size would have $(7 \cdot 7 \cdot 7 \cdot 6)^3 = 1.6 \cdot 10^9$ numbers in it, while approximated parameter vectors in FSR scheme is $3 \cdot (7 + 7 + 4) \cdot 6^3) = 3.8 \cdot 10^3$, and in RBF scheme is just $3 \cdot 8 \cdot 6 = 144$ numbers. If we use a 8-byte memory to represent a number, the FSR scheme needs 29.69 Kilobytes, the RBF scheme requires only 1.13 Kilobytes, compare to the total space required by a Q-table is 12.8 Gigabytes, the required space is hugely saved.

After approximation, the update rule in (3.7) for Q-function becomes the update rule for the parameter [68] set of each UAV i:

$$\theta_{i,k+1} \leftarrow \theta_{i,k} + \alpha[GR(S_k, A_k) + \gamma \max_{A' in\{A\}} (\phi^T(S_{k+1}, A')\theta_{i,k}) - (\phi^T(S_k, A_k)\theta_{i,k}]\phi(S_k, A_k). \tag{3.11}$$

3.3.5 Algorithm

Algorithm 2: Multi-Agent Approximated Equilibrium-based Q-Learning.

```
Input: Learning parameters: Discount factor $\gamma$, learning rate $\alpha$, schedule $\{\epsilon^k\}$,
       number of step per episode $L$
Input: Basis Function vector $\phi(S, A)$, $\forall s_{i,0} \in S_i$, $\forall a_{i,0} \in A_i$
Initialize $\theta_{i,0} \leftarrow 0$, $i = 1, ..., m$;
for $episode = 1, 2, ...$ do
    Randomly initialize state $s_{i,0}$, $\forall i$
    for $k = 0, 1, 2, ...$ do
        for $i = 0, 1, 2, ..., m$ do
            Exchange information with other UAVs to obtain their state $s_{j,k}$
              and parameters $\theta_j$, $j \neq i, j = 1...m$
            $\pi(A_k) = \begin{cases} \text{Find an optimal joint-action by (3.4)}, & w.\ prob.\ 1 - \epsilon_k \\ \text{Take a random joint action}, & otherwise. \end{cases}$
            Decide unique joint action $A_k$, and take individual joint action
            according to social conventions rule
            Receive other UAVs' new states $s_{j,k+1} | j \neq i, j - 1, ..., m$
            Observe global reward $r_{k+1} = GR(S_k, A_k)$
            Update:
                $\theta_{i,k+1} \leftarrow \theta_{i,k} + \alpha[GR(S_k, A_k) + \gamma \max_{A' in\{A\}} (\phi^T(S_{k+1}, A')\theta_{i,k})$
                        $- (\phi^T(S_k, A_k)\theta_{i,k}]\phi(S_k, A_k).$
Output: parameter vector $\theta_i, i = 1...m$ and policy $\pi$
```

We propose our learning process as Algorithm 2. The algorithm required learning rate α, discount factor γ, and a schedule $\{\epsilon^k\}$. The learning process is divided into

Table 3.1: Simulation parameters for Multi-Agent Reinforcement Learning

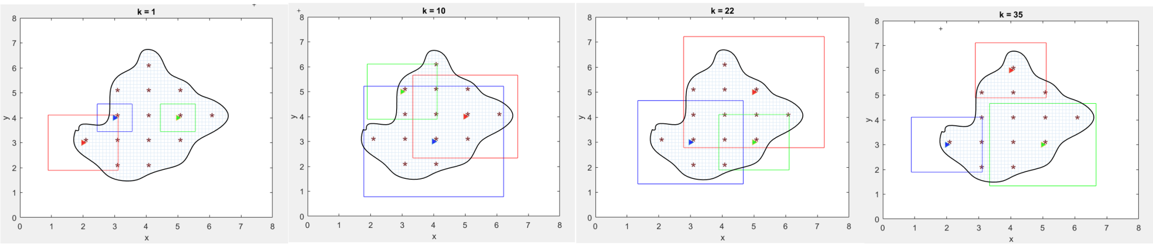
$\alpha = 0.1$	$\gamma = 0.9$	$\epsilon = 0.9$
number of agents $m = 3$	number of episodes = 700	number of steps = 2000

episodes, with arbitrarily-initialized UAVs' states in each episode. We use a greedy policy π with a big initial ϵ to increase the exploration actions in the early stages, but it will be diminished over time to focus on finding optimal joint action according to (3.4). Each UAV will evaluate their performance based on a global reward function in (3.6), and update the approximated value function of their states and action using the law (3.11) in a distributed manner.

3.4 Simulation Results

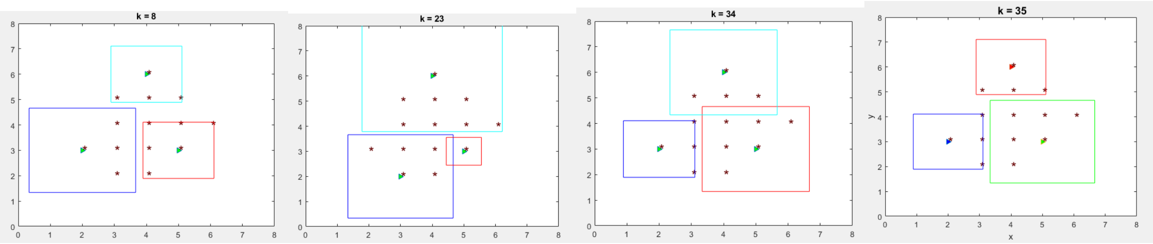
We set up a simulation on MATLAB environment to prove the effectiveness of our proposed algorithm. Consider our environment space as a $7 \times 7 \times 5$ discrete 3-D space, and a field of interest F on a grid board with an unknown shape (Figure 3.3). The system has $m = 3$ UAVs, each UAV can take six possible actions to navigate: forward, backward, go left, go right, go up or go down. Each UAV in the team will have a positive reward $r = 0.1$ if the team covers the whole field F with no overlapping, otherwise it receives $r = 0$.

We implement the proposed algorithm 2 with both approximation schemes: FSR and RBF, and compare their performance with a baseline algorithm. For the baseline algorithm, the agents seek to solve the problem by optimizing individual performance, that is to maximize their own coverage of the field F, and stay away from overlapping others to avoid a penalty of -0.01 for each overlapping square. For the proposed algorithm, both schemes use learning rate $\alpha = 0.1$, discount rate $\gamma = 0.9$, and $\epsilon = 0.9$ for the greedy policy which is diminished over time. To find CE for the agents in



(a) k = 1 (b) k = 10 (c) k = 22 (d) k = 35

Figure 3.3: 2-D result showing the FOV of 3 UAVs collaborate in the last learning episode to provide a full coverage of the unknown field F with discrete points denoted by $*$ mark, while avoiding overlapping others.



(a) (b) (c) (d)

Figure 3.4: Different optimal solutions showing the configuration of the FOV of 3 UAVs with a full coverage and no discrete point ($*$ mark) overlapped.

(3.4), we utilize an optimization package for MATLAB from CVX [74].

Our simulation on MATLAB shows that, in both FSR and RBF schemes after some training episodes the proposed algorithm allows UAV team to organize in several optimal configurations that fully cover the field while having no overlapping, while the baseline algorithm fails in most episodes. Figure 3.3 shows how the UAVs coordinated to cover the field F in the last learning episode in 2D. Figure 3.4 shows a result of different solutions of the 3 UAV's FOV configuration with no overlapping. For a clearer view, Figure 3.5 shows the UAVs team and their FOVs in 3D environment in the last episode of the FSR scheme.

Figure 3.6 shows the number of steps per episode the team took to converge to optimal solution. The baseline algorithm fails to converge, so it took maximum number of steps (2000), while the two schemes using proposed algorithm converged

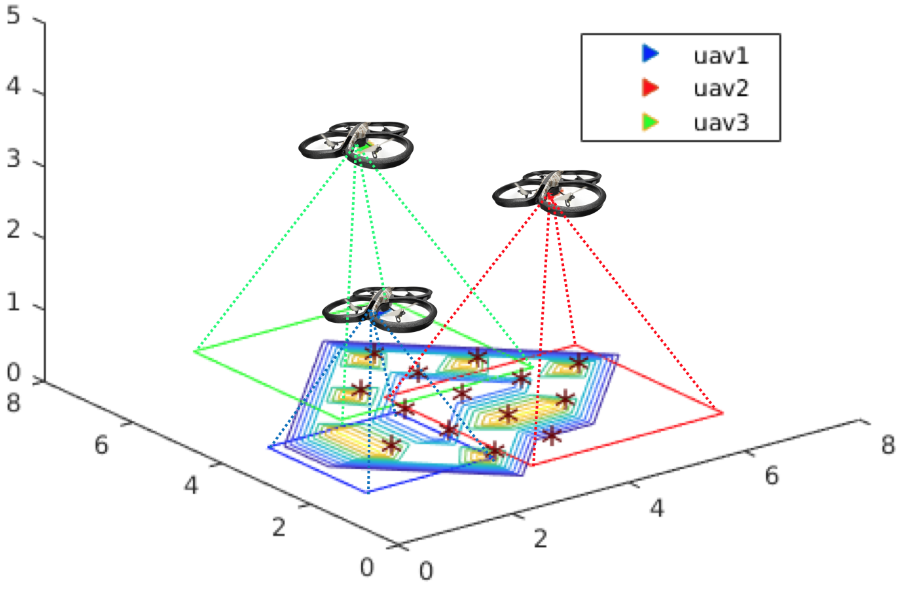


Figure 3.5: 3-D representation of the UAV team covering the field.

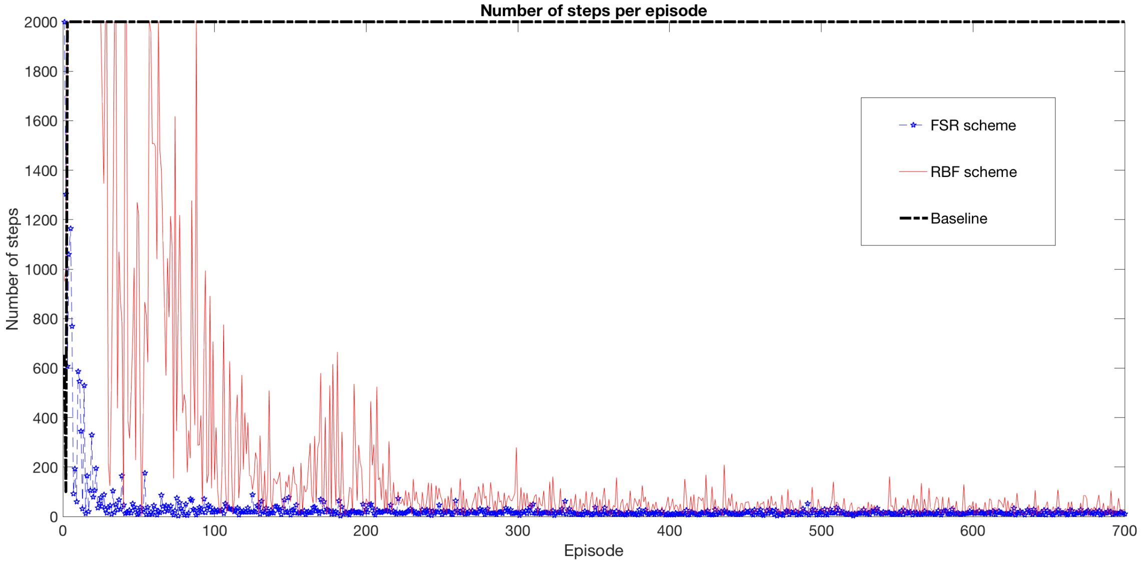


Figure 3.6: Number of steps the team UAV took over episodes to derive the optimal solution.

nicely. Interestingly, it took longer for the RBF scheme to converge, compare to the FSR scheme. It is likely due to the difference in accuracy of the approximation techniques, where RBF scheme has worse accuracy.

3.5 Implementation

3.5.1 Design the PID controller for a UAV

This subsection describes how I design a position controller for quadrotor-type UAVs. To carry out the proposed algorithm, the UAV should be able to transit from one state to another, and stay there before taking new action. We implemented the PID controller to help the UAV carry out its action. Suppose at time t, an UAV need to take action a_k to translate from current location s_k to new location s_{k+1} and stay hovering over the new state within a small error radius d. Define p_t is the real-time position of the UAV at time t, we start with a simple proportional gain controller:

$$u(t) = K_p(p(t) - s_{k+1}) = K_p e(t), \tag{3.12}$$

where $u(t)$ is the control input, K_p is the proportional control gain, and $e(t)$ is the tracking error between real-time position $p(t)$ and desired location s_{k+1}. Due to the nonlinear dynamics of the quadrotor [8], we experienced excessive overshoots of the UAV when it navigates from one state to another, making the UAV unstable after reaching a state. To overcome this, we used a standard PID controller [75]. Although the controller cannot effectively regulate the nonlinearity of the system, work such as [76, 77] indicated that using PID controller could still yield relatively good stabilization during hovering.

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de}{dt}. \tag{3.13}$$

Generally, the derivative component can help decrease the overshoot and the settling time, while the integral component can help decrease the steady-state error, but

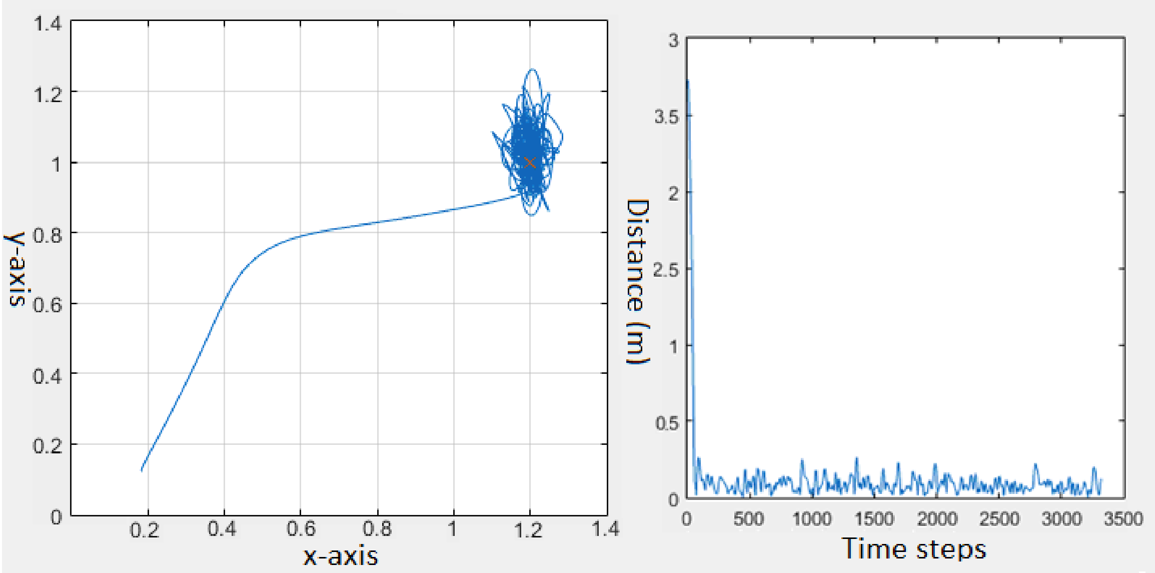


Figure 3.7: Distance error between the UAV and the target after tuning.

Table 3.2: Drones Implementation parameters for Multi-Agent Reinforcement Learning

$\alpha = 0.1$	$\gamma = 0.9$	$\epsilon = 0.9$
number of agents $m = 2$	number of episodes $= 700$	number of steps $= 2000$
$K_p = 0.8$	$K_i = 0$	$K_d = 0.9$

can cause increasing overshoot. During the tuning process, we increased the Derivative gain and eliminated the Integral component of the PID control to achieve stable trajectory. Note that $u(t)$ is calculated in the Inertial frame, and should be transformed to the UAV's Body frame before feeding to the propellers controller as linear speed [8]. Figure 3.7 shows the result after tuning. The UAV can remain within a radius of $d = 0.3$m from the desired state. The values of control parameters are provided in Table 3.2.

3.5.2 Implementation of Multiple UAVs Learning

We implemented a lab-setting experiment for 2 UAVs to cover the field of interest F with the similar specification as of the simulation (Table 3.2), in an environment space as a $7 \times 7 \times 4$ discrete 3-D space. We used two quadrotor Parrot AR.Drones 2.0, and the Motion Capture System from Motion Analysis [78] installed in our Advanced Robotics and Automation (ARA) lab. The UAV could be controlled by altering the linear/angular speed, and the motion capture system provides the UAV's relative position inside the room. We carried out the experiment using the FSR scheme, with

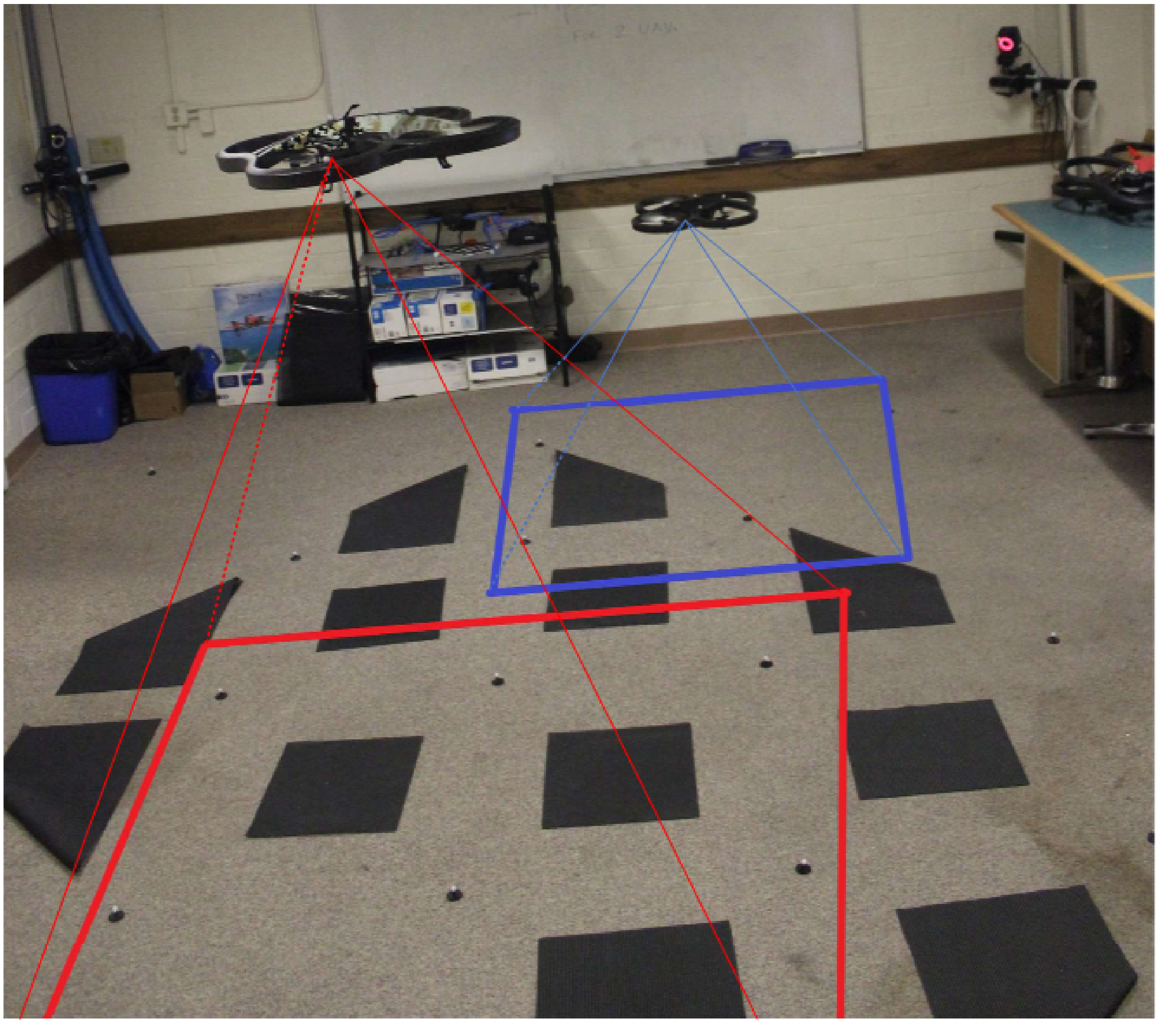


Figure 3.8: Physical implementation with 2 ARdrones. The UAVs cooperate to cover a field which consists of black markers, while avoid overlapping with each other.

similar parameters to the simulation, but now for only 2 UAVs. Each would have a positive reward $r = 0.1$ if the team covers the whole field F with no overlapping, and $r = 0$ otherwise. The learning rate was $\alpha = 0.1$, and discount rate $\gamma = 0.9$,

$\epsilon = 0.9$, which was diminished over time. Similar to the simulation result, the UAV team also accomplished the mission, with two UAVs coordinated to cover the whole field without overlapping each other, as showed in (Figure 3.8).

3.6 Summary

In this chapter, I formulated a problem in which a UAV team needs to surround the field of interest to get more information. The objective of the team is to provide a full coverage of the field of interest and minimize overlapping with each other to improve the efficiency of the team. I proposed a MARL algorithm that enable the UAVs to cooperatively learn to provide full coverage of an unknown field of interest, while minimizing the overlapping sections among their field of views. The complex dynamic of the joint-actions of the UAV team has been solved using game-theoretic correlated equilibrium. The challenge in huge dimensional state space has been also tackled with FSR and RBF approximation techniques that significantly reduce the space required to store the variables. The correctness of the solution was validated by experimental results of both simulation and physical implementations.

Chapter 4

Conclusion

4.1 Conclusion

In this thesis, we proposed two approaches for solving the Optimal Coverage Problem.

In the first approach, we presented a distributed control design for a team of UAVs that can collaboratively track a dynamic environment in the case of wildfire spreading. The UAVs can follow the border region of the wildfire as it keeps expanding, while still trying to maintain coverage of the whole wildfire. The UAVs are also capable of avoiding collision, maintaining safe distance to fire level, and flexible in deployment. A simulation implementing Algorithm 1 validates our approach. The application could certainly go beyond the scope of wildfire tracking, as the system can work with any dynamic environment, for instance, oil spilling or water flooding.

In the second approach, we proposed a MARL algorithm that can be applied to a team of UAVs that enable them to cooperatively learn to provide full coverage of an unknown field of interest, while minimizing the overlapping sections among their field of views. The complex dynamic of the joint-actions of the UAV team has been solved using game-theoretic correlated equilibrium. The challenge in huge dimensional state space has been also tackled with FSR and RBF approximation techniques that significantly reduce the space required to store the variables. We also provide our experimental results with both simulation and physical implementation of Algorithm 2 to show that the UAVs can successfully learn to accomplish the task without the need of a mathematical model.

We showed that each approach has its own advantages and disadvantages. While the model-based approach can perform well, it required a good, reliable, mathematical model of the environment, and thus, more a priori information. The model-free approach, on the other hand, required less priori information, but did require a period of extensive training to learn appropriate behaviors. This finding is aligned with other works comparing the performance of the two approaches, for example, [79–81]. The selection of each approach, thus, depends on the availability of prior data, the reliability of the models, and the problem domain.

4.2 Future Work

For future work, for the first approach, more should be focused on researching about the hardware implementation of the proposed controller. For example, we should pay attention to the communication between the UAVs under the condition of constantly changing topology of the networks, and the sensing endurance problem in hazardous environment. Also, we would like to investigate the relation between the speed of the UAVs and the spreading rate of the wildfire, and attempt to synchronize them. Multi-drone cooperative sensing [14, 82, 83], cooperative control [84–86], cooperative learning [87, 88], and user interface design [89] for wildland fire mapping will be also considered.

For the second approach, we are interested in using Neural network and Deep Learning to reduce computation time, especially in finding CE. An initial result simulating the algorithm using multi-layer neural network-based approximation can be seen in Figure 4.1. We will also consider to work in more important application where the dynamic of the field presents, such as in wildfire monitoring.

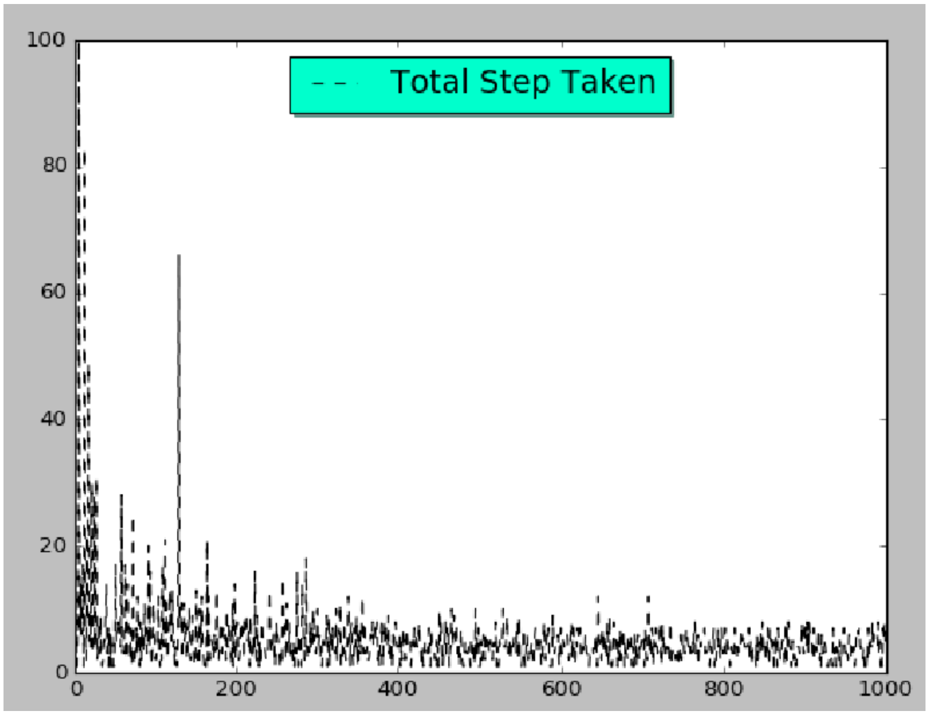


Figure 4.1: Learning Algorithm using Neural networks.

Bibliography

[1] L. Merino, F. Caballero, J. Martnez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, 2006.

[2] H. Cruz, M. Eckert, J. Meneses, and J.-F. Martínez, "Efficient forest fire detection index for application in unmanned aerial systems (uass)," *Sensors*, vol. 16, no. 6, p. 893, 2016.

[3] M. Jafari, S. Sengupta, and H. M. La, "Adaptive flocking control of multiple unmanned ground vehicles by using a uav," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, and G. Weber, Eds. Cham: Springer International Publishing, 2015, pp. 628–637.

[4] A. Singandhupe, H. M. La, D. Feil-Seifer, P. Huang, L. Guo, and M. Li, "Securing a uav using individual characteristics from an eeg signal," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2017, pp. 2748–2753.

[5] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, 2011.

[6] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice.* Princeton university press, 2012.

[7] F. Muoz, E. S. Espinoza Quesada, H. M. La, S. Salazar, S. Commuri, and L. R. Garcia Carrillo, "Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 9, pp. 1566–1588, 2017, rnc.3687. [Online]. Available: http://dx.doi.org/10.1002/rnc.3687

[8] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[9] A. C. Woods, H. M. La, and Q. P. Ha, "A novel extended potential field controller for use on aerial robots," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2016, pp. 286–291.

[10] A. C. Woods and H. M. La, *Dynamic Target Tracking and Obstacle Avoidance using a Drone.* Cham: Springer International Publishing, 2015, pp. 857–866. [Online]. Available: https://doi.org/10.1007/978-3-319-27857-5_76

[11] A. Bemporad, C. A. Pascucci, and C. Rocchi, "Hierarchical and hybrid model predictive control of quadcopter air vehicles," *IFAC Proceedings Volumes*, vol. 42, no. 17, pp. 14–19, 2009.

[12] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt^*;," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, July 2016.

[13] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2013, pp. 831–836.

[14] ——, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 1–12, Jan 2015.

[15] I. Maza, F. Caballero, J. Capitan, J. Martinez-de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1-4, pp. 563–585, 2011.

[16] D. Casbeer, R. Beard, T. McLain, S.-M. Li, and R. Mehra, "Forest fire monitoring with multiple small uavs," in *American Control Conference, 2005. Proceedings of the 2005*, June 2005, pp. 3530–3535 vol. 5.

[17] 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*, vol. 8, no. 1, pp. 1–16, 2011.

[18] C. Phan and H. H. Liu, "A cooperative uav/ugv platform for wildfire detection and fighting," in *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on.* IEEE, 2008, pp. 494–498.

[19] T. Nguyen, H. M. La, T. D. Le, and M. Jafari, "Formation control and obstacle avoidance of multiple rectangular agents with limited communication ranges,"

IEEE Transactions on Control of Network Systems, vol. 4, no. 4, pp. 680–691, Dec 2017.

[20] H. M. La and W. Sheng, “Dynamic target tracking and observing in a mobile sensor network,” *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 996 – 1009, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889012000565

[21] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[22] M. Schwager, D. Rus, and J.-J. Slotine, “Decentralized, adaptive coverage control for networked robots,” *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[23] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, “Voronoi coverage of non-convex environments with a group of networked robots,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 4982–4989.

[24] M. Adibi, H. Talebi, and K. Nikravesh, “Adaptive coverage control in non-convex environments with unknown obstacles,” in *Electrical Engineering (ICEE), 2013 21st Iranian Conference on.* IEEE, 2013, pp. 1–6.

[25] S. S. Ge and Y. J. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Autonomous robots*, vol. 13, no. 3, pp. 207–222, 2002.

[26] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, “Eyes in the sky: Decentralized control for the deployment of robotic camera networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[27] H. M. La and W. Sheng, “Distributed sensor fusion for scalar field mapping using mobile sensor networks,” *IEEE Transactions on cybernetics*, vol. 43, no. 2, pp. 766–778, 2013.

[28] M. T. Nguyen, H. M. La, and K. A. Teague, “Collaborative and compressed mobile sensing for data collection in distributed robotic networks,” *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[29] L. C. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. Pereira, “Simultaneous coverage and tracking (scat) of moving targets with robot networks,” in *Algorithmic foundation of robotics VIII.* Springer, 2009, pp. 85–99.

[30] H. M. La, “Multi-robot swarm for cooperative scalar field mapping,” *Handbook of Research on Design, Control, and Modeling of Swarm Robotics*, p. 383, 2015.

[31] H. M. La, W. Sheng, and J. Chen, “Cooperative and active sensing in mobile sensor networks for scalar field mapping,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 1–12, 2015.

[32] “National interagency fire center.” [Online]. Available: https://www.nifc.gov/fireInfo/fireInfo_statistics.html

[33] J. R. Martinez-de Dios, B. C. Arrue, A. Ollero, L. Merino, and F. Gómez-Rodríguez, “Computer vision techniques for forest fire perception,” *Image and vision computing*, vol. 26, no. 4, pp. 550–562, 2008.

[34] D. Stipaničev, M. Štula, D. Krstinić, L. Šerić, T. Jakovčević, and M. Bugarić, “Advanced automatic wildfire surveillance and monitoring network,” in *6th International Conference on Forest Fire Research, Coimbra, Portugal.(Ed. D. Viegas)*, 2010.

[35] P. Sujit, D. Kingston, and R. Beard, "Cooperative forest fire monitoring using multiple uavs," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 4875–4880.

[36] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian journal of forest research*, vol. 45, no. 7, pp. 783–792, 2015.

[37] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 639–643.

[38] L. Merino, F. Caballero, J. R. Martínez-de Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 533–548, 2012.

[39] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 6648–6653.

[40] J. Glasa and L. Halada, "A note on mathematical modelling of elliptical fire propagation." *Computing & Informatics*, vol. 30, no. 6, 2011.

[41] R. C. Rothermel, "A mathematical model for predicting fire spread in wildland fuels," 1972.

[42] G. D. Richards, "An elliptical growth model of forest fire fronts and its numerical solution," *International Journal for Numerical Methods in Engineering*, vol. 30, no. 6, pp. 1163–1179, 1990.

[43] M. A. Finney *et al.*, *FARSITE: Fire area simulator: model development and evaluation.* US Department of Agriculture, Forest Service, Rocky Mountain Research Station Ogden, UT, 2004.

[44] T. M. Williams, B. J. Williams, and B. Song, "Modeling a historic forest fire using gis and farsite," *Mathematical & Computational Forestry & Natural Resource Sciences*, vol. 6, no. 2, 2014.

[45] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.

[46] H. M. La and W. Sheng, "Dynamic target tracking and observing in a mobile sensor network," *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 996 – 1009, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889012000565

[47] H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero, and H. Parvardeh, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1655–1664, 2013.

[48] H. M. La, N. Gucunski, S.-H. Kee, J. Yi, T. Senlet, and L. Nguyen, "Autonomous robotic system for bridge deck data collection and analysis," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on.* IEEE, 2014, pp. 1950–1955.

[49] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, "Development of an autonomous bridge deck inspection robotic system," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1489 –1504, December 2017.

[50] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press Cambridge, 1998, vol. 1, no. 1.

[51] H. Bou-Ammar, H. Voos, and W. Ertel, "Controller design for quadrotor uavs using reinforcement learning," in *Control Applications (CCA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 2130–2135.

[52] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 4902–4909.

[53] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," in *Innovations in multi-agent systems and applications-1.* Springer, 2010, pp. 183–221.

[54] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[55] B. Bakker, S. Whiteson, L. Kester, and F. C. Groen, "Traffic light control by multiagent reinforcement learning systems," in *Interactive Collaborative Information Systems.* Springer, 2010, pp. 475–510.

[56] J. Ma and S. Cameron, "Combining policy search with planning in multi-agent cooperation," in *Robot Soccer World Cup.* Springer, 2008, pp. 532–543.

[57] M. K. Helwa and A. P. Schoellig, "Multi-robot transfer learning: A dynamical system perspective," *arXiv preprint arXiv:1707.08689*, 2017.

[58] F. Fernandez and L. E. Parker, "Learning in large cooperative multi-robot domains," 2001.

[59] J. Hu and M. P. Wellman, “Nash q-learning for general-sum stochastic games,” *Journal of machine learning research*, vol. 4, no. Nov, pp. 1039–1069, 2003.

[60] A. K. Sadhu and A. Konar, “Improving the speed of convergence of multi-agent q-learning for cooperative task-planning by a robot-team,” *Robotics and Autonomous Systems*, vol. 92, pp. 66–80, 2017.

[61] Y. Ishiwaka, T. Sato, and Y. Kakazu, “An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning,” *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 245–256, 2003.

[62] H. M. La, R. Lim, and W. Sheng, “Multirobot cooperative learning for predator avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, 2015.

[63] S.-M. Hung and S. N. Givigi, “A q-learning approach to flocking with uavs in a stochastic environment,” *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 186–197, 2017.

[64] A. A. Adepegba, M. S. Miah, and D. Spinello, “Multi-agent area coverage control using reinforcement learning.” in *FLAIRS Conference*, 2016, pp. 368–373.

[65] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, “Game theory and multi-agent reinforcement learning,” in *Reinforcement Learning.* Springer, 2012, pp. 441–470.

[66] A. Greenwald, K. Hall, and R. Serrano, “Correlated q-learning,” in *ICML*, vol. 3, 2003, pp. 242–249.

[67] C. H. Papadimitriou and T. Roughgarden, “Computing correlated equilibria in multi-player games,” *Journal of the ACM (JACM)*, vol. 55, no. 3, p. 14, 2008.

[68] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators.* CRC press, 2010, vol. 39.

[69] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 38, no. 2, pp. 156–172, 2008.

[70] C. Guestrin, M. Lagoudakis, and R. Parr, “Coordinated reinforcement learning,” in *ICML*, vol. 2, 2002, pp. 227–234.

[71] Z. Zhang, D. Zhao, J. Gao, D. Wang, and Y. Dai, “Fmrqa multiagent reinforcement learning algorithm for fully cooperative tasks,” *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1367–1379, 2017.

[72] D. Borrajo, L. E. Parker, *et al.*, “A reinforcement learning algorithm in cooperative multi-robot domains,” *Journal of Intelligent and Robotic Systems*, vol. 43, no. 2-4, pp. 161–174, 2005.

[73] A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, J. P. How, *et al.*, “A tutorial on linear function approximators for dynamic programming and reinforcement learning,” *Foundations and Trends® in Machine Learning*, vol. 6, no. 4, pp. 375–451, 2013.

[74] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.

[75] R. C. Dorf and R. H. Bishop, *Modern control systems.* Pearson, 2011.

[76] J. Li and Y. Li, “Dynamic analysis and pid control for a quadrotor,” in *Mechatronics and Automation (ICMA), 2011 International Conference on.* IEEE, 2011, pp. 573–578.

[77] K. U. Lee, H. S. Kim, J. B. Park, and Y. H. Choi, "Hovering control of a quadrotor," in *Control, Automation and Systems (ICCAS), 2012 12th International Conference on.* IEEE, 2012, pp. 162–167.

[78] "Motion analysis corporation." [Online]. Available: https://www.motionanalysis.com/

[79] B. Depraetere, M. Liu, G. Pinte, I. Grondman, and R. Babuka, "Comparison of model-free and model-based methods for time optimal hit control of a badminton robot," *Mechatronics*, vol. 24, no. 8, pp. 1021 – 1030, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415814001226

[80] G. Binetti, G. Leonetti, D. Naso, and B. Turchiano, "Comparison of model-free and model-based control techniques for a positioning actuator based on magnetic shape memory alloys," in *2015 IEEE International Conference on Mechatronics (ICM)*, March 2015, pp. 284–289.

[81] S. Gay, J. van den Kieboom, J. Santos-Victor, and A. Ijspeert, "Model-based and model-free approaches for postural control of a compliant humanoid robot using optical flow," in *IEEE Conference on Humanoids Robots*, no. EPFL-CONF-191285, 2013.

[82] M. T. Nguyen, H. M. La, and K. A. Teague, "Collaborative and compressed mobile sensing for data collection in distributed robotic networks," *IEEE Transactions on Control of Network Systems*, 2017.

[83] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 766–778, April 2013.

[84] T. Nguyen, H. M. La, T. D. Le, and M. Jafari, “Formation control and obstacle avoidance of multiple rectangular agents with limited communication ranges,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 680–691, 2017.

[85] H. M. La and W. Sheng, “Flocking control of multiple agents in noisy environments,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 4964–4969.

[86] H. M. La, T. H. Nguyen, C. H. Nguyen, and H. N. Nguyen, “Optimal flocking control for a mobile sensor network based a moving target tracking,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2009, pp. 4801–4806.

[87] H. M. La, R. Lim, and W. Sheng, “Multirobot cooperative learning for predator avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, Jan 2015.

[88] H. M. La, R. S. Lim, W. Sheng, and J. Chen, “Cooperative flocking and learning in multi-robot systems for predator avoidance,” in *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, May 2013, pp. 337–342.

[89] K. T. A. Siddiqui, D. Feil-Seifer, T. Yang, S. Jose, S. Liu, and S. Louis, “Development of a swarm uav simulator integrating realistic motion control models for disaster operations,” in *Proceedings of the ASME Dynamic Systems and Controls Conference*, Tysons Corner, Virginia, October 2017.