AIRBORNE GROUND SURVEILLANCE WITH MULTI-HOP UAV NETWORKS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Abdulsamet Dağaşan

December 2020

Airborne Ground Surveillance With Multi-Hop UAV Networks
By Abdulsamet Dağaşan
December 2020

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Ezhan Karaşan(Advisor)

Nail Akar

Mehmet Köseoğlu

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

AIRBORNE GROUND SURVEILLANCE WITH MULTI-HOP UAV NETWORKS

Abdulsamet Dağaşan
M.S. in Electrical and Electronics Engineering
Advisor: Ezhan Karaşan
December 2020

Cooperative utilization of unmanned aerial vehicles (UAVs) in public and military surveillance applications has attracted significant attention in recent years. Most UAVs are equipped with sensors that have limited coverage and wireless communication equipment with limited range. Such limitations pose challenging problems to monitor mobile targets. The thesis examines fulfilling surveillance objectives to achieve better coverage while building a reliable network between UAVs. Area coverage and cooperative multiple target tracking problems are investigated and linear integer programming models are presented for these problems. For the area coverage problem, the optimal placement of UAVs for a given area is investigated with varying coverage ranges and the results are discussed. Coverage of the map achieved by the UAVs and the maximum possible coverage area are compared under different coverage and communications constraints. For the cooperative multiple target tracking, the optimal placement of UAVs to monitor mobile targets where their mobility is modeled with the random waypoint mobility model is studied. The multiple target tracking problem is further extended by assuming a relay UAV within the fleet whose trajectory is planned in order to achieve a reliable connected network among all UAVs. Optimization problems are established for single-hop and multi-hop communications. Three algorithms are proposed for multi-hop communications and their performances are evaluated. The effect of the time horizon considered in the optimization problem is also studied. Performance evaluation results show that the trajectories planned for relay UAV by the proposed algorithms generates network topologies that remain connected for more than 90% of the maximum possible duration that the UAVs can be connected by an ideal relay.

Keywords: Area Coverage, Cooperative Multi Target Tracking, Path Planning, Single-Hop and Multi-Hop Communication.

ÖZET

ÇOKLU-HOP İLETİŞİMLİ İHA İLE HAVADAN YER GÖZETİMİ

Abdulsamet Dağaşan
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Ezhan Karaşan
Aralık 2020

Kamu ve askeri gözetleme uygulamalarında insansız hava araçlarının (İHA) ortak kullanımı son yıllarda büyük ilgi görmektedir. Çoğu İHA, kapsama alanı ve iletişim menzili sınırlı olan sensörlerle donatılmıştır. Bu tür sınırlamalar, mobil hedeflerin izlenmesini zorlaştırmaktadır. Bu çalışmada, gözetleme görevlerini yerine getirirken güvenilir bir ağ kurmaya çalışan İHA'lar incelenmektedir. Alan kapsama problemi ve işbirlikli çoklu hedef takibi incelenmiş ve optimizasyon problemi olarak doğrusal tamsayılı programlama modelleri sunulmuştur. Alan kapsama problemi için, belirli bir alan içerisinde İHA'ların en uygun yerleşimi, değişen kapsama aralığında karşılaştırılıp sonuçlar tartışılmıştır. İHA'lar tarafından elde edilen haritanın kapsamı ve mümkün olan maksimum kapsama alanı, farklı kapsama ve iletişim kısıtlamaları altında karşılaştırılır. İşbirlikçi çoklu hedef takibi için, İHA'ların hareketliliklerinin rastgele yol noktası hareketlilik modeli ile modellendiği mobil hedefleri izlemek için optimum yerleşimi incelenmiştir. Tüm İHA'lar arasında güvenilir bir bağlı ağ elde etmek için filo içinde yörüngesi planlanan bir röle İHA varsayımı ile çoklu hedef izleme problemi daha da genişletilmiştir. Tek ve çok sekmeli iletişimler için optimizasyon problemleri oluşturulmuştur. Çok sekmeli iletişimler için üç ayrı algoritma önerilmiştir ve performansları değerlendirilmiştir. Zaman ufkunun optimizasyon problemine etkisi de incelenmiştir. Performans değerlendirmesi sonucunda önerilen algoritmalar tarafından İHA rölesi için planlanan yörüngelerin, İHA'ların ideal bir röle ile bağlanabileceği maksimum olası sürenin 90 % 'ından daha fazla bağlı kalan ağ topolojileri ürettiğini göstermektedir.

Anahtar sözcükler: Alan Kapsama, İşbirlikli Çoklu Hedef Takibi, Yol planlama, Tekli-Hop ve Çoklu-Hop İletişim.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Ezhan Karaşan. His guidance and encouragement significantly shaped my thesis. Without his support and patience, this thesis would not have been possible.

I would like thank to Prof. Nail Akar and Dr. Mehmet Köseoğlu their valuable contributions to my thesis committee and examining my thesis.

I would like to thank my roommates Abdullah Ömer Arol and Bilal Taşdelen for making the life easier throughout my graduate years. I would like to thank my department friends Mahmut Can Soydan, Erdem Aras, Ahmet Safa Öztürk, Muzaffer Özbey for their valuable discussions as well as the enjoyable times we had together. I am grateful to all my friends throughout my education life from high school to graduate school for their valuable friendship.

I would also like to express my gratitude to Ayşegül İlay Tunalı for her company. She is the joy of my life and I feel very fortunate to have her.

Finally, I am grateful to my dear family. They always motivated and encouraged me throughout my life as well as this study. This accomplishment would not have been possible without the support of my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Surveillance systems have recently received a significant attention due to the rapid increase in security and safety threats. Among the available surveillance methods, use of Unmanned Aerial Vehicles (UAV) has been rather widespread for mainly two reasons. Firstly, UAVs can operate where it might be too dangerous for humans to fulfill surveillance duties. Secondly, UAVs also can be operated autonomously with a minimum human resource allocation. Some of the surveillance applications include search and rescue operations, monitoring an environment and tracking mobile targets.

The objective of a target tracking with UAVs is to monitor a specific environment and acquire information on the mobile targets. This information is analyzed locally and shared with either a central node or between each UAVs. Since the communication range of an UAV is limited, connectivity of the UAVs becomes an issue with increasing distance between the nodes. Therefore, building a reliable and connected network between the cooperative UAVs is critical.

This thesis investigates building a reliable network between tracker UAVs and comprises two parts. In the first part of this thesis, the area coverage problem and cooperative target tracking are investigated. For the area coverage problem, the objective is to monitor an environment with cooperative UAVs. Depending on the number of UAVs used and the coverage abilities of each UAVs, full coverage is not always feasible. The objective then becomes to maximize the covered area in a given environment. For the target tracking, our objective is to cooperatively monitor the mobile targets while staying connected. Number of mobile targets are usually smaller than then number of UAVs. Therefore, it is not always possible to monitor each and every target. In such scenarios, the goal is to maximize the number of mobile targets monitored.

Optimal UAV placements for the area coverage and multiple target tracking with cooperative UAVs can be achieved with the integer linear programming model proposed in the thesis. Increasing coverage radius enlarges the monitored area and the number of mobile targets, but the ratio of monitored area to maximum area that can be monitored with multiple UAVs decreases leading to waste of resources. The trade-off between these two is used to find the optimum coverage radius in a given environment with a specific communication range.

In the second part of this thesis, we study ensuring a connected network topology between distributed cooperative target tracking UAVs with a relay UAV. Connected network topology means every UAV in the network has a way of communicating with each other. We study single-hop and multi-hop communication to perform path planning for the relay UAV. In the single-hop communication, relay UAV is directly connected to other drones creating a star topology. In the multi-hop communication, relay UAV can communicate with each UAV through other drones creating an arbitrary tree topology. The objective is to maximize the during which a connected network topology can be maintained among UAVs
.

The proposed algorithms for the path planning of a relay UAV can guarantee a reliable connected network more than 90% of the possible duration that the topology remains connected in ideal conditions.

1.1 Contributions of the Thesis

In this thesis, for the area coverage problem, we present an area decomposition. But instead of using the convex polygon partition that was introduced in [1], we use grids to partition the given field. Unlike the conventional methods, we use two different grids to partition the area. This helps us to transform the problem into an integer linear programming problem, which guarantees an optimal solution if the problem is feasible. We also model connectivity with an integer flow parameter as a constraint.

For the cooperative target tracking, we use the same area decomposition model. With some additions to the objective function, we formulate this problem with a linear integer programming model as well. We also, model the communication restrictions into our system. Although [2] uses linear integer programming model to solve a multi-target tracking problem with cooperative multi-robots, it does not study on the communication restraints.

For the trajectory planning of the relay UAV, we use Kalman filtering based estimator to get the UAV positions and we have a nonsmooth sampling function to build communication links between UAVs. The distance between the tracker UAVs and the relay UAV determines the existence of communication links. To model the inaccuracy in position estimation of the UAVs, the link existence probabilities are calculated using a uniformly distributed position error model. Both single and multi-hop communication among tracker UAVs and relay UAV are considered and corresponding optimization problems are formulated.

1.2 Thesis Outline

Chapter 2 provides a detailed literature review on the area coverage problem and the cooperative target tracking. We also focus on communication and coverage ranges of different UAV types.

Chapter 3 introduces the coverage problem model and the proposed solution method. We study the relation between communication range and coverage range to check the relevancy. We introduce the cooperative tracking model with its solution and investigate the cooperative tracking while having communication link between each drone. We present numerical results to examine the theoretical solutions.

Chapter 4 presents creating a connected network topology for UAVs on the missions. We introduce system model for the relay drone. We present the objective functions for single-hop and multi-hop communications. For multi-hop communication, we present three different algorithms.

Chapter 5 presents numerical results for the main components of the system model defined in Chapter 4. The results for different algorithms are compared throughout a number of simulations to analyze the best performing algorithm.

Chapter 6 presents the conclusions.

Chapter 2

Literature Review on Aerial Target Tracking and Path Planning

Surveillance, tracking and search of mobile targets using cooperative UAVs has become an important task with the rapid improvement of the UAV technology. There are many applications [3] such as search operations [4], sports coverage [5], wildlife research [6], [7] in the literature. There are examples of surveillance operations carried out by cooperative UAVs to search targets [8], to observe targets [9] to coordinate paths [10], [11].

The surveillance applications that are studied in this thesis are area coverage and cooperative target tracking problems. For the area coverage problem, the objective is to observe an environment with a given network. The idea of using UAVs for area coverage problem is fairly new, because most of the area coverage problems are static and solved with static nodes [12]. [13] formulates this problem as a decision problem and presents polynomial-time algorithms that can be converted to distributed protocols to solve it. In [14], a Coverage Configuration Protocol to dynamically configure a network is provided. Both cases work on the static coverage with wireless sensors. UAVs, on the other hand, are mobile making the coverage problem dynamic. The coverage algorithms become very challenging because of this dynamic coverage. Also there are many limitations for UAVs such as communication and coverage range, battery limitations and energy consumption. Compared to static networks, UAVs require less nodes to solve the area coverage problem because of the dynamic coverage, however, it has a worse coverage precision [12]. In [1] the authors suggests an area partition problem that uses polygon decomposition to solve the coverage problem with UAVs. It partitions the area to convex polygon subparts and presents a heuristic coverage algorithm. [15] also uses this partition and assigns individual areas for the coverage algorithm to cover the area of interest.

The objective of cooperative target tracking is to observe a set of targets in a given field with cooperative UAVs. The problem of cooperative target tracking can be categorized into four components: environment, targets, sensors and coordination [3]. Environment can be represented as a continuous plane [9] or discrete plane [16]. Discrete planes mostly comprises of well defined partitions of the environment. Environments are mostly geometrical rectangular regions [17], but there are examples of other geometrical shapes such as circle [18].

The target can be defined with type and mobility. Targets can be seperated into three groups as cooperative, non-cooperative or evasive. While cooperative targets shares movement information to UAVs [19], in most cases the movement of the non-cooperative targets should be estimated [9], [20]. In most cases, the mobility of the target is simulated with random walks [21] or linear motion models [22].

The field of view (FOV) of the sensors can cover only a small portion of the given environment. Coverage area is usually much smaller compared to the communication range as well. If the sensing process is deterministic, the target in the FOV of an UAV is considered to be observed. If the sensing process is to be probabilistic, the probability of observing the target in the FOV of an UAV should be calculated [23].

Coordination amongst the UAVs is critical to achieve successful cooperative target tracking. Centralized coordination means that UAVs transmit the information to a central station. Task distribution and path planning for every UAV is calculated in the central station and sent back to the on-duty UAVs [4]. Decentralized coordination means there are multiple UAVs that act as central nodes. The information from the other UAVs are received by the leader UAVs, then they cooperatively assign tasks and plan trajectories for the on-duty UAVs [24]. The UAVs can also coordinate in a distributed manner. Distributed coordination means each UAV decides its task and plans its trajectory independently [25], [26].

For the trajectory planning of relay UAV, the task is to plan the path that optimizes the proposed problem. [27] investigates path planning for a relay UAV in urban areas for airborne to ground (A2G) communications where the ground nodes are stationary. A Gaussian process model is formed and it is solved with nonlinear model predictive control (NMPC)-based trajectory planner. In [28], the same problem is studied with ground mobile nodes. NMPC-based trajectory planner is also used in their study, however, a discrete genetic algorithm is used to find the optimal control input. To find the optimal path, NMPC is combined with a finite time horizon controlled system. Time horizon determines the time window that the relay UAV movements will be optimized. [29] uses the same NMPC-based planner, but to calculate the objective function, it uses the concept of minimum spanning tree (MST). MST is used to choose the communication link with the highest expectation of a successful transmission. In [30] instead of planning trajectory, it suggests the selection of relay UAVs from a set of cooperative UAVs. A matching market-based optimization for UAV communication models are presented for different coordination of cooperative UAVs to choose relay UAVs.

In the next chapter, the area coverage and the multiple target tracking will be investigated.

Chapter 3

Autonomous Ground Target Tracking with Multiple UAVs

This chapter contains the coverage problem for a given place and the given targets inside the place with a limited number of drones that needs to stay connected. There are two types of coverage problems: area coverage and target coverage. In area coverage, the objective is to maximize the covered area in a given place with a limited number of drones. In target coverage, the objective is to maximize the number of targets covered with a limited number of drones.

3.1 Area Coverage Maximization through Optimum Drone Placements

We consider a coverage problem with N drones in a given rectangular-shaped area. Our objective is to place N drones with limited communication range in a way that it maximizes the monitored area while being connected with multi-hop routing.

In a given region, the objective is to spread drones in a way that it covers as much area as possible while maintaining the communication link. We assume a point is covered by a drone if the distance between them is less than the coverage range R_{cov} and two drones can communicate with each other if the distance between them is less than the communication range R_{com}. We assume each drone has its own Flying Ad-Hoc Networks (FANETs) to communicate and each can use multi-hop routing to communicate with each other. However, the number of hops h each drone can use to communicate is limited in order to decrease the interference and delay time. An illustration of the system is shown in Figure 3.1

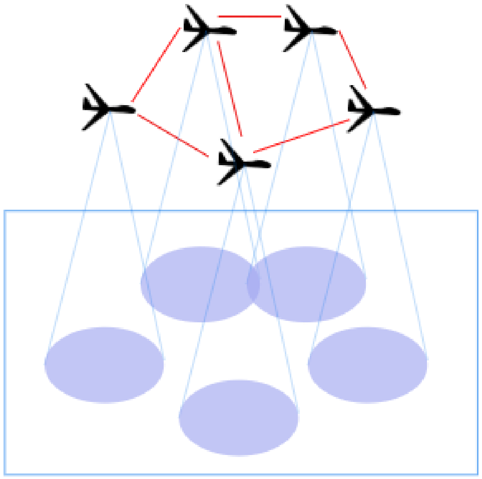


Figure 3.1: An example of coverage and communication scenario

Depending on the coverage range, communication range and number of hops, this problem might have no solution or infinitely many solutions. There are similar non-deterministic polynomial-time hard (NP-hard) problems with immobile sensors which have large time and space complexity. To solve the problem, we have quantized the map into two different gridlines. We have grids for possible drone placements and smaller grids to check the coverage. We formulated the problem as an integer linear programming problem.

3.1.1 Optimization Problem Formulation

The area is a rectangular map with width w and length l. For drone placement grid, we use a square with size $G \times G$ and for coverage grids we use a square with size $g \times g$. The set of possible location points for drones is denoted by S and the set of sensing points in the map is denoted by C. An illustration of the quantized map and random drone placements is shown in Figure 3.2 with $G = 5g$.

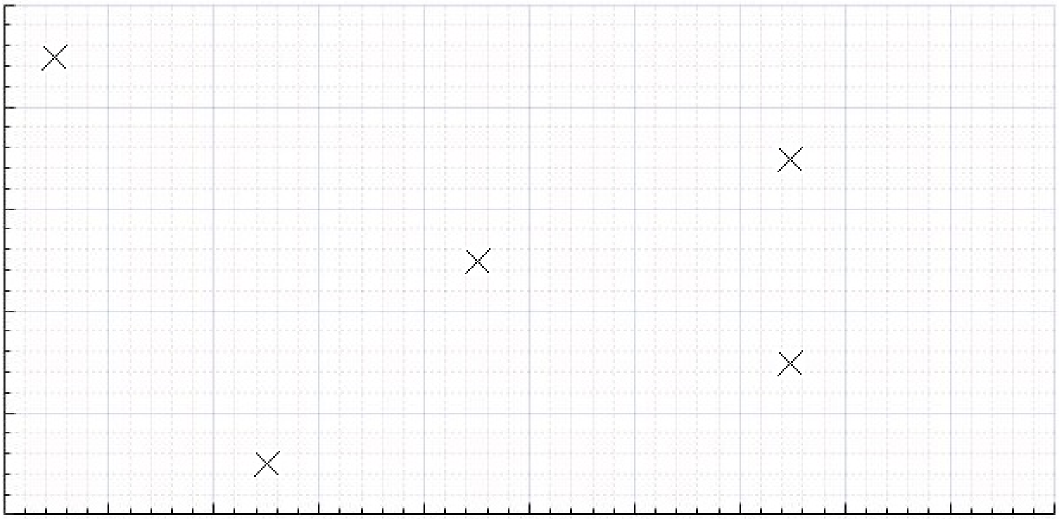


Figure 3.2: Example of a quantized system with $w, l = (25, 50)$ and $G, g = (5, 1)$ respectively

.

Model parameters for the optimization problem can be defined using the set S and set C. The binary parameter $A_{ij} = 1$ only if a drone in position $i \in S$ covers the sensing point $j \in C$. The binary parameter $E_{ij} = 1$ only if a drone in position $i \in S$ can communicate with a drone in position $j \in S$. These parameters are predetermined when R_{com} and R_{cov} are known. To determine the parameters, we use Euclidian distance between the centers of the grid points.

Decision variables for the optimization problem can also be defined using the set S and set C. We have x_i to check if there is drone located at $i \in S$. We have y_j to check if $j \in C$ is covered by a drone. We have a control parameter z_{ij} to check if there are two drones located at $i, j \in S$, respectively. We have a flow parameter f_{ij}^{kl} which helps us limit the number of maximum hops two drones can have to communicate. We end up with the following integer linear programming model:

$$\text{maximize} \sum_{m=1}^{N_g} y_m - \epsilon \sum_{k,l=1}^{N_G} \sum_{i,j=1}^{N_G} f_{ij}^{kl} \tag{3.1}$$

$$\text{subject to} \sum_{n=1}^{N_G} x_n = N \tag{3.2}$$

$$y_m \geq x_n A_{nm} \tag{3.3}$$

$$y_m \leq \sum_{n=1}^{N_G} x_n A_{nm} \tag{3.4}$$

$$z_{ij} \leq x_i \tag{3.5}$$

$$z_{ij} \leq x_j \tag{3.6}$$

$$z_{ij} \geq (x_i + x_j - 1) \tag{3.7}$$

$$f_{ij}^{kl} \leq z_{ij} E_{ij} \tag{3.8}$$

$$\sum_{i=1}^{N_G} (f_{ij}^{kl} - f_{ji}^{kl}) = \begin{cases} -z_{kl}, & \text{if } j = k \\ z_{kl}, & \text{if } j = l \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

$$\sum_{i,j=1}^{N_G} f_{ij}^{kl} \leq t \tag{3.10}$$

$$y_m, x_n, z_{ij}, f_{ij}^{kl} \in \{0, 1\} \tag{3.11}$$

(3.1) is a multi-objective optimization. The dominating part of objective function in (3.1) is the number of covered grids and we want to maximize it. The non-dominating part is total number of hops used in the connected network and we want to minimize it. To assert domination between objective functions we used scalarization [31] and multiplied the second objective function with a small $\epsilon > 0$. The constraint in (3.2) guarantees each drone is placed in a grid in set S. (3.3) makes sure the sensing grid y_m is covered if it is in a drone's coverage area. If a sensing grid is covered by multiple drones, it needs to be counted only once. (3.4) handles it since $0 \leq y_m \leq 1$. (3.5), (3.6) and (3.7) is to assign the control parameter z_{ij} to check if there are two drones located at $i, j \in S$. All of these constraints is enough to solve the area coverage problem.

We want our drones to be connected for the area coverage problem, so we need to check if there is a connected network. Therefore we need to check every connection link that can be established between the grids in set S. A connection link will be possible if the communication range is bigger than the distance between i, j in set S and there are drones in i, j grids as in the constraint (3.8). The following constraint (3.9) represents network connectivity problem that needs to be solved. Assume we have a network with k and l as source and sink nodes. We represent the flow from node i to node j with f_{ij}^{kl}. The problem then becomes similar to network flow problems. Lets assume we are investigating a directed graph between the source and sink nodes k and l. For the other nodes, the amount of a flow entering is the same that exits the node. Only the source node will have a flow exiting the node and only the sink node will have a flow entering as in (3.10). The constraint in (3.10) is to limit the number of hops a network graph from k to l. Lastly, (3.11) constrains the parameters to take values between 0 and 1.

Lets define $N_G = \frac{wl}{G^2}$ and $N_g = \frac{wl}{g^2}$ to specify the number of grids for drone placement and coverage respectively. Using these definitions we can specify the model variables as $m = 1, ..., N_g$ and $n, i, j, k, l = 1, ..., N_G$.

When $R_{com} \geq 2R_{cov}$, the problem becomes trivial because you can spread each drone in a way that their coverage areas do not intersect and can cover all the coverage area possible in the map. Therefore, we only studied the case of $R_{cov} \leq R_{com} \leq 2R_{cov}$ and the numerical results will be presented only for this case.

3.2 Multiple Target Tracking with Maximum Coverage

The target coverage problem is a special case of the area coverage problem. Here instead of maximizing the area covered in a given map, we focus on the maximization of the number of targets monitored.

We consider a target coverage problem of N drones with T targets in a given rectangular-shaped area. Our objective is to place N drones with limited communication ranges in such a way the number of targets monitored in the area is maximized while drones are connected with multi-hop routing. The difference here is instead of checking the coverage points, we have potential targets in each of those coverage points. We assume a target is covered by a drone if the distance between them is less than the coverage range R_{cov}. An illustration of the system is shown in Figure 3.3.

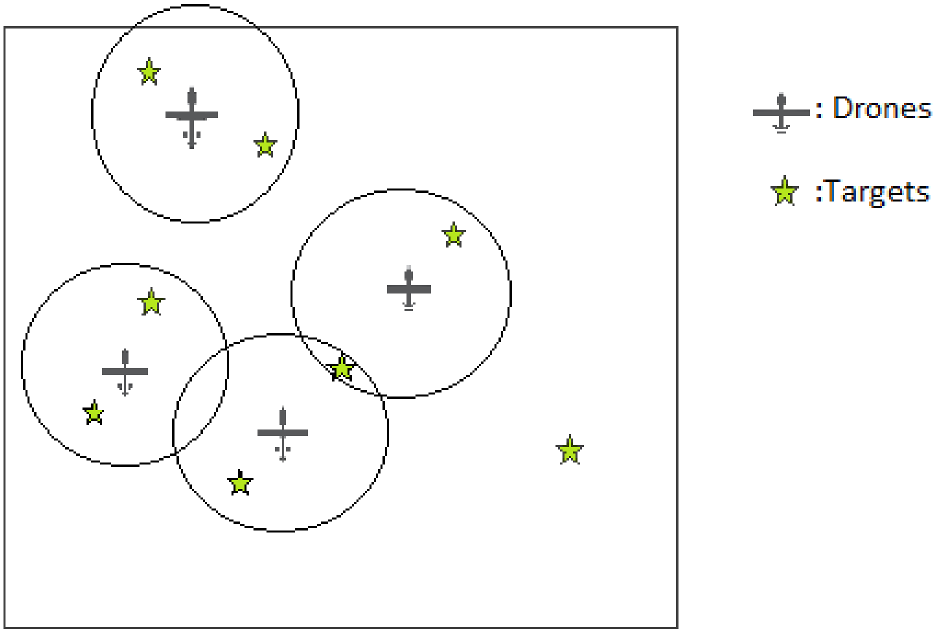


Figure 3.3: An example of target coverage problem.

For target movement, we used the Random Waypoint Model [32]. This model is a random model for the movement of mobile targets, and how their location, velocity and acceleration changes over time. Since the number of targets T is larger than the number of drones N also due to the network limitations there are some cases that where it is not always possible to track all the targets. For this optimization problem, we use the same method we used on the area coverage problem. We quantize the given map, and we define a linear integer programming problem to solve the optimization problem. We also track the targets in real time while solving the optimization problem for given time sequences. We assume that the target locations are available while solving the optimization problem.

3.2.1 Optimization Problem Formulation

The only difference from the integer linear programming model presented in Section 3.1 is that we have a predefined **w** array that is used to indicate which coverage grid has a target. If there is a target on the m^{th} grid then $w_m = 1$, otherwise $w_m = 0$. The resulting integer linear programming model is given as:

$$\begin{aligned}
\text{maximize} \quad & \sum_m y_m w_m - \epsilon \sum_{k,l} \sum_{i,j} f_{ij}^{kl} \\
\text{subject to} \quad & \sum_n x_n = N \\
& y_m \geq x_n A_{nm} \\
& y_m \leq \sum_n x_n A_{nm} \\
& z_{ij} \leq x_i \\
& z_{ij} \leq x_j \\
& z_{ij} \geq (x_i + x_j - 1) \\
& f_{ij}^{kl} \leq z_{ij} E_{ij} \\
& \sum_i (f_{ij}^{kl} - f_{ij}^{kl}) = \begin{cases} -z_{kl}, & \text{if } j = k \\ z_{kl}, & \text{if } j = l \\ 0, & \text{otherwise} \end{cases} \\
& \sum_{i,j} f_{ij}^{kl} \leq t \\
& y_m, x_n, z_{ij}, f_{ij}^{kl} \in \{0, 1\}
\end{aligned} \tag{3.12}$$

The predetermined **w** vector is added to the objective function in the optimization problem formulated in 3.1. Therefore, the objective of the optimization problem turns into maximizing the number of targets tracked.

In case there are more than one target on the same grid, weight parameter **w** can be increased for that particular grid. Also, if there are some high priority targets, their weights can be increased so that algorithm will try to monitor them first.

3.3 Numerical Results

3.3.1 Area Coverage based on different UAV Settings

In area coverage problem we are given a rectangular shaped map. We have a 50 km $\times$ 100 km map with 10 km $\times$ 10 km grids for possible drone positions and 2 km $\times$ 2 km km grids for coverage positions. We have $N = 4$ drones to cover the area with communication range of 25 km and three different coverage ranges of 7.5, 25, 15 km, respectively. The covered grids are shown with yellow color and uncovered grids are shown with purple color.

Table 3.1: Simulation 1 Parameters.

Parameter	Value
Number of Drones	4
Width	50 km
Length	100 km
Coverage Range	7.5 km
Communication Range	25 km
Grid Size for Drones	10 km
Grid Size for Coverage	2 km

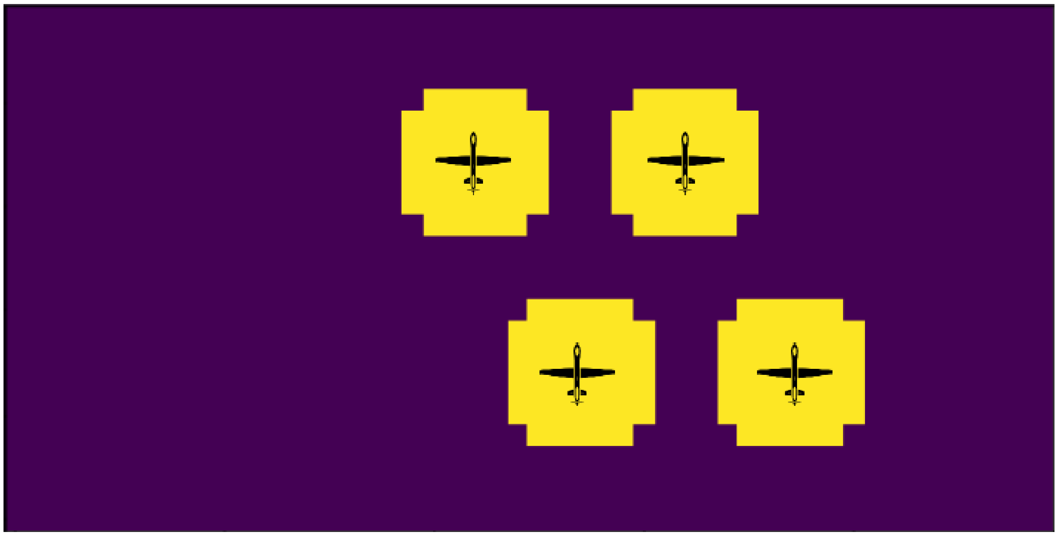


Figure 3.4: Simulation Results with the Parameters in Table 3.1.

In simulation 1, we can, at maximum, cover $N\pi R_{cov}^2 = 706.85$ km^2 area, and our drones can cover 100% of the maximum possible coverage area. The resulting placements are shown in Figure 3.4. This area coverage problem with $R_{com} > 2R_{cov}$ is trivial, because there is no intersection of the coverage areas hence it is enough for drones to stay connected.

Table 3.2: Simulation 2 Parameters.

Parameter	Value
Number of Drones	4
Width	50 km
Length	100 km
Coverage Range	25 km
Communication Range	25 km
Grid Size for Drones	10 km
Grid Size for Coverage	2 km

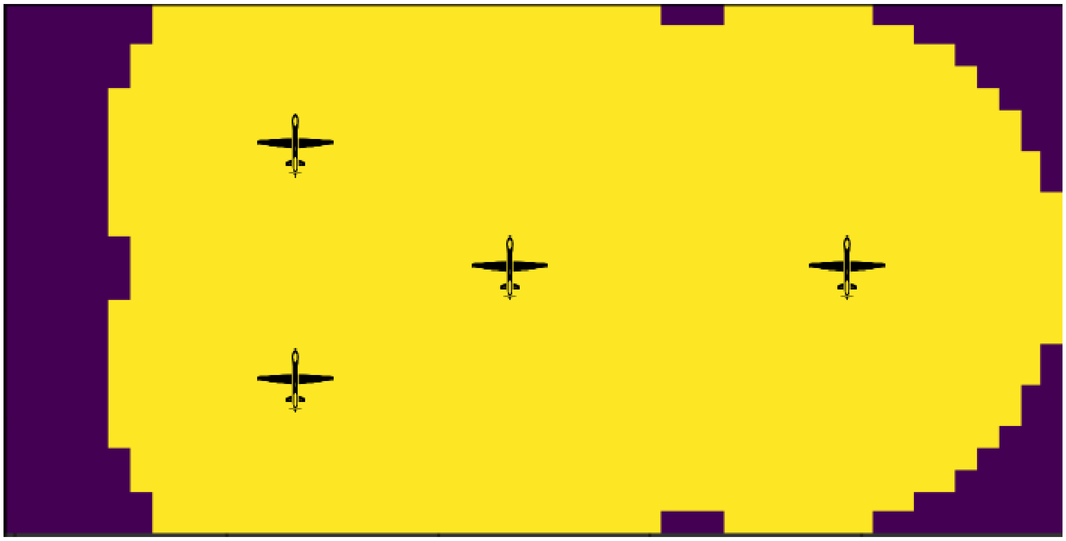


Figure 3.5: Simulation Results with the Parameters in Table 3.2.

In simulation 2, we can, at maximum, cover $N\pi R_{cov}^2 = 7853.98$ km^2 area. However, since the coverage size is comparable to map size, two coverage areas intersect. Four drones with the given coverage radius can cover all the map, but there are network limitations. Therefore they can only cover a portion of the map. The resulting placements are shown in Figure 3.6.

Table 3.3: Simulation 3 Parameters.

Parameter	Value
Number of Drones	4
Width	50 km
Length	100 km
Coverage Range	15 km
Communication Range	25 km
Grid Size for Drones	10 km
Grid Size for Coverage	2 km

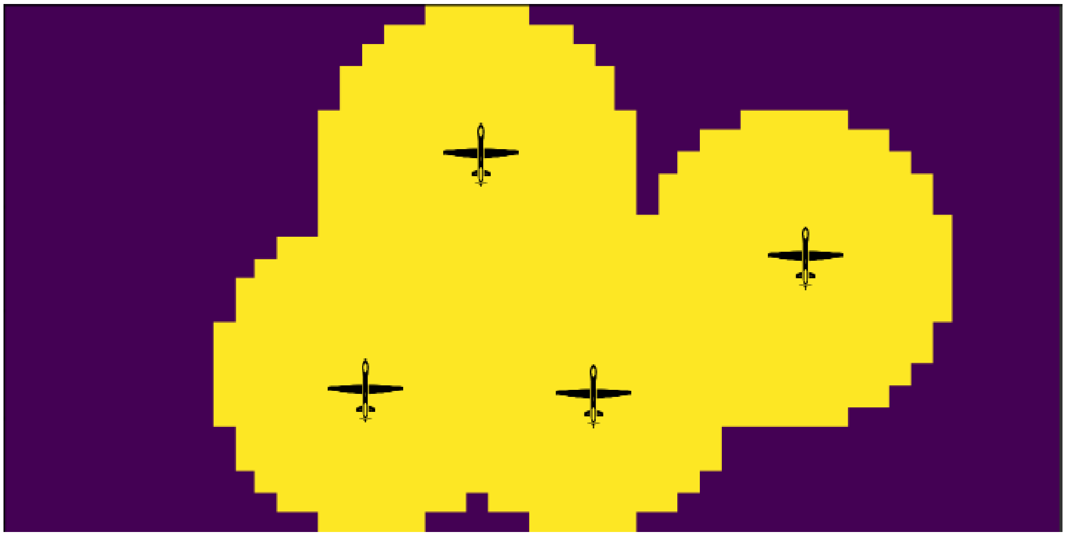


Figure 3.6: Simulation Results with the Parameters in Table 3.3.

In simulation 3, we can, at maximum, cover $N\pi R_{cov}^2 = 2827.43\ \text{km}^2$ area. This is the most interesting case to investigate because, we cannot span the map with only four drones. We also cannot sparsely put them into the areas because of the network limitations. The resulting placements are shown in Figure 3.6.

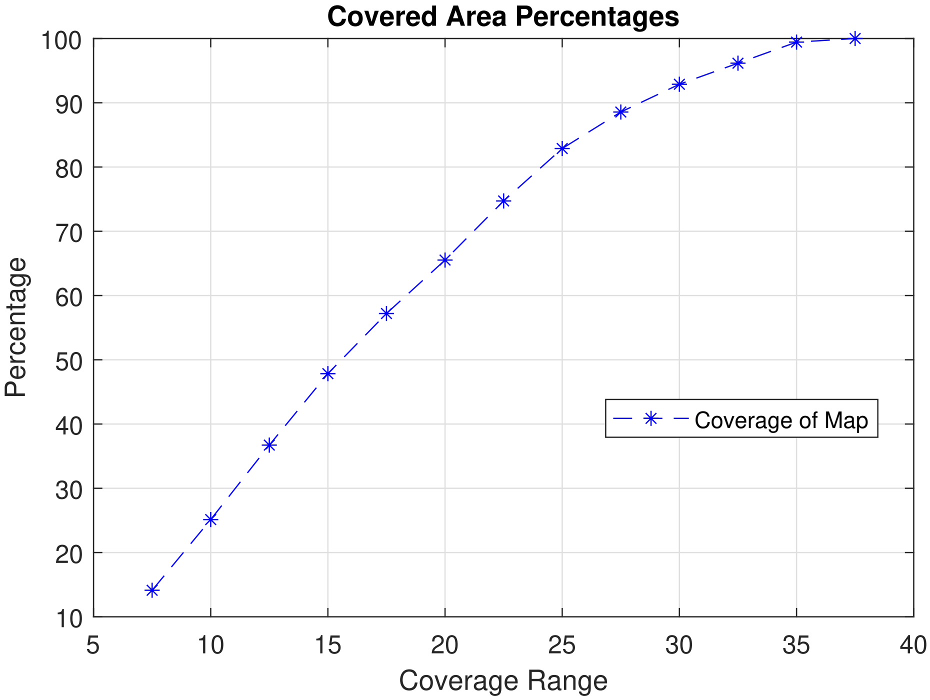


Figure 3.7: Coverage Performance of each coverage range.

For $R_{com} = 25$ km, we simulated different coverage ranges between [7.5-37.5] km with 2.5 km resolution which can be seen in Figure 3.7. Blue stars represent how much area of the rectangular area is covered. The field of view of a drone increases quadratically when coverage radius is increased. However, it can be seen that the increase of coverage of map is sublinear. Because of the communication restrictions and the rectangular shaped area, the FOVs of drones overlap or extend outside the rectangular area. Therefore increasing coverage range increases the total coverage area with a decreasing rate.

3.3.2 Area Coverage with Multiple Target Tracking

In the target coverage problem, we are given a square shaped map. We have a 500 km × 500 km map with 70 km × 70 km grids for possible drone positions and 10 km × 10 km grids for coverage positions. We have $N = 4$ drones to to track $T = 10$ targets with communication range of 150 km and coverage range of

100 km. Drone velocities differ between [50,70] km/h and target velocities differ between [5,60] km/h. Our knowledge on target locations gets updated every 10 minutes. It means in each time frame, a target can only move by one coverage grid. It also means a drone can change its coverage grid in approximately 1 hours. The covered grids are shown with yellow color, uncovered grids are shown with purple color, target grids are shown with green color.

Table 3.4: Simulation Parameters.

Parameter	Value
Number of Drones	4
Number of Targets	10
Drone Velocities	50-70 km/h
Target Velocities	5-60 km/h
Width	500 km
Length	500 km
Coverage Range	100 km
Communication Range	150 km
Grid Size for Drones	70 km
Grid Size for Coverage	10 km

Figure 3.8 shows 8 hours simulation of 10 targets movements in an area of 500 km × 500 km square originated at (0,0). Each movement is calculated with 10 minutes time frame and it is approximated to the closest coverage grid.

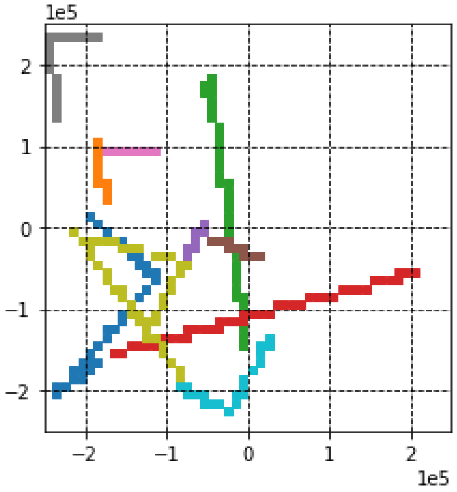


Figure 3.8: Movements of 10 Targets with Random Waypoint Model.

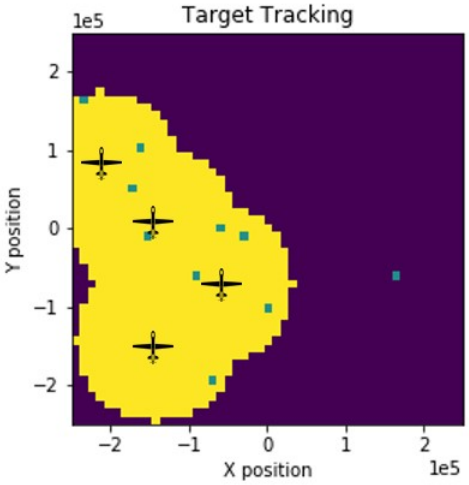
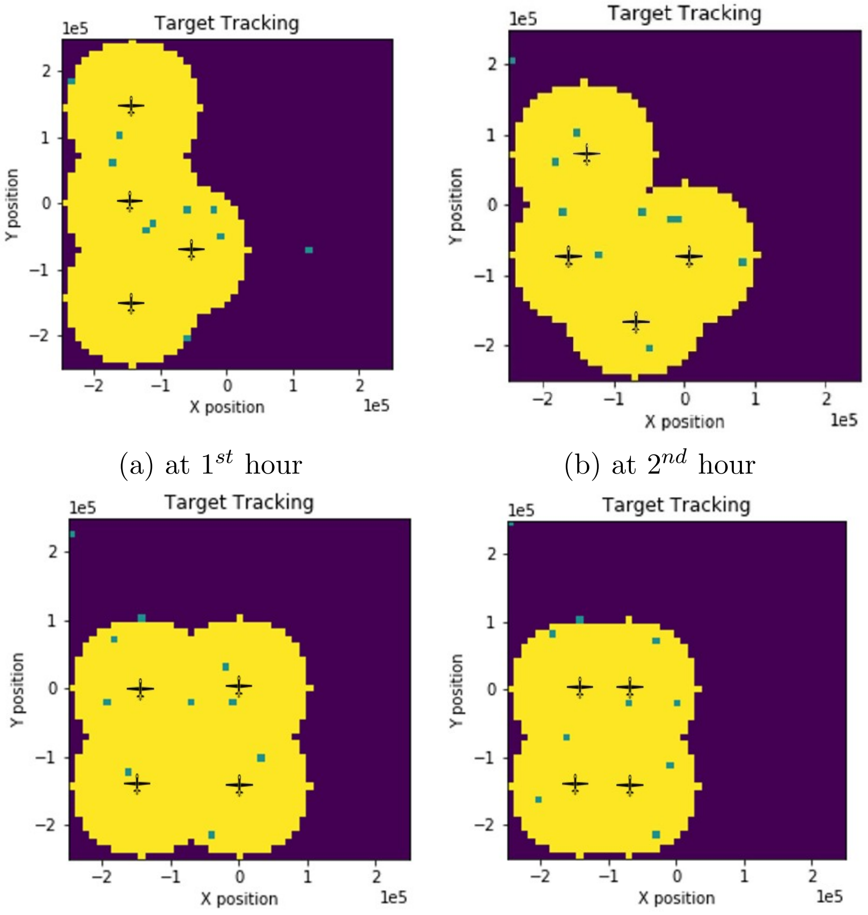


Figure 3.9: Drone positions at the start.

In the beginning the drones are positioned in a way that they can cover all the targets on the left side of the map. The target on the right side is not tracked because if a drone flies to that position, it will either lose connection or it will track less targets.



(a) at 1^{st} hour

(b) at 2^{nd} hour

(c) at 3^{rd} hour

(d) at 4^{th} hour

Figure 3.10: Drone positions at each hour.

The simulation results for the first 5 hours can be seen in Figures 3.9 and 3.10. As we can see, the drones are moving corresponding to target movements. For this case, the average number of targets tracked was 8.62. Although it depends on the target locations, average number of targets tracked increases with the coverage range.

Chapter 4

Autonomous Airborne Surveillance

In this chapter, our objective is to achieve connectivity between N drones that are tracking targets. In order to enhance the connectivity among drones, we use a relay drone which autonomously decides its path to be able to communicate with other drones. The relay drone has three components: its sensors, estimator and the path planner. The main purpose of the sensor and estimator components is to develop an optimization-based path planner.

4.1 Path Estimation for Tracker UAVs using a Relay Drone

Tracker drones each have their inertial measurement units (IMU) sensors and has a global navigation satellite system (GNSS). The relay drone has GNSS receiver and using the data from the sensors, it estimates the positions of the drones. We consider position and velocity information from the sensors to estimate the position and the path of the other drones. We used Kalman Filter to estimate the position and velocity over time.

4.1.1 Estimator

A Kalman filter is used to estimate linear dynamical systems. The process model defines the state transition from time $k-1$ to time k as:

$$\mathbf{x_k} = \mathbf{F}\mathbf{x_{k-1}} + \mathbf{B}\mathbf{u_{k-1}} + \mathbf{w_{k-1}}, \tag{4.1}$$

where $\mathbf{x_k}$ is the state vector, $\mathbf{F}$ is the state transition matrix, $\mathbf{u_k}$ is the control vector and $\mathbf{B}$ is the control input matrix.

$\mathbf{w_{k-1}}$ is the Gaussian process noise vector:

$$\mathbf{w_{k-1}} \sim N(0, \mathbf{Q}). \tag{4.2}$$

The process model and measurement model are used to define the relationship between state and the measurement at time step k:

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} + \mathbf{v_k}, \tag{4.3}$$

where $\mathbf{z_k}$ is the measurement vector and $\mathbf{H}$ is the measurement matrix.

$\mathbf{v_k}$ is the Gaussian measurement noise vector:

$$\mathbf{v_k} \sim N(0, \mathbf{R}). \tag{4.4}$$

The Kalman Filter Algorithm has prediction and update stages. Prediction stage equations are as follows:

$$\mathbf{\hat{x}_k^{'}} = \mathbf{F}\mathbf{\hat{x}_{k-1}} + \mathbf{B}\mathbf{u_{k-1}}, \tag{4.5}$$

$$\mathbf{P_k^{'}} = \mathbf{F}\mathbf{P_{k-1}}\mathbf{F}^T + \mathbf{Q}. \tag{4.6}$$

Update stage equation are as follows:

$$\mathbf{K_k} = \mathbf{P_k^{'}} \mathbf{H}^T (\mathbf{R} + \mathbf{H} \mathbf{P_k^{'}} \mathbf{H}^T)^{-1}, \tag{4.7}$$

$$\mathbf{\hat{x}_k} = \mathbf{\hat{x}_k^{'}} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H} \mathbf{\hat{x}_k^{'}}), \tag{4.8}$$

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k} \mathbf{H}) \mathbf{P_k^{'}}, \tag{4.9}$$

where $\hat{\mathbf{x_k}}$ is the estimate of the state vector $\mathbf{x_k}$ and $\hat{\mathbf{x_k}}'$ is the prior estimate of the state vector. $\mathbf{P_k}$ is the error covariance matrix. $\mathbf{P_k^{'}}$ is the prior estimate of the error covariance matrix. $\mathbf{K_k}$ is the Kalman gain that is used to update the predicted estimate $\mathbf{x_k}$ and the covariance matrix $\mathbf{P_k}$ [33].

4.1.2 System Model

We assume we have $\mathbf{pos} = [pos_x, pos_y, pos_z]^T$ position vector and $\mathbf{vel} = [vel_x, vel_y, vel_z]^T$ velocity vector as the state vector. We also have $\mathbf{acc} = [acc_x, acc_y, acc_z]^T$ as the acceleration vector that will be used as control vector. The state vector can be written as:

$$x = [\mathbf{pos^T}, \mathbf{vel^T}]^T. \tag{4.10}$$

The state prediction stage can be written as:

$$\mathbf{x_k} = \begin{bmatrix} \mathbf{pos_k} \\ \mathbf{vel_k} \end{bmatrix} = \begin{bmatrix} \mathbf{pos_{k-1}} + \mathbf{vel_{k-1}} \Delta t + \frac{1}{2} \widetilde{\mathbf{acc}}_\mathbf{k-1} \Delta t^2 \\ \mathbf{vel_{k-1}} + \widetilde{\mathbf{acc}}_\mathbf{k-1} \Delta t \end{bmatrix}. \tag{4.11}$$

We can rearrange the state prediction equation by using identity and zero matrices to form an equation similar to the 4.1:

$$\mathbf{x_k} = \begin{bmatrix} \mathbf{I_{3x3}} & \mathbf{I_{3x3}} \Delta t \\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}} \end{bmatrix} \mathbf{x_{k-1}} + \begin{bmatrix} \frac{1}{2} \mathbf{I_{3x3}} \Delta t^2 \\ \mathbf{I_{3x3}} \Delta t \end{bmatrix} \widetilde{\mathbf{acc}}_\mathbf{k-1}. \tag{4.12}$$

The process noise on the acceleration vector can be written with its zero mean Gaussian noise vector $\mathbf{e}$.

$$\mathbf{acc_{k-1}} = \widetilde{\mathbf{acc}}_{\mathbf{k-1}} + \mathbf{e_{k-1}}, \tag{4.13}$$

$$\mathbf{e_{k-1}} \sim N(0, \mathbf{I_{3x3}}\sigma_{\mathbf{e}}^{\mathbf{2}}). \tag{4.14}$$

We can use this noise to calculate the process noise covariance matrix.

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{2}\mathbf{I_{3x3}}\Delta t^2 \\ \mathbf{I_{3x3}}\Delta t \end{bmatrix} \mathbf{I_{3x3}}\sigma_{\mathbf{e}}^{\mathbf{2}} \begin{bmatrix} \frac{1}{2}\mathbf{I_{3x3}}\Delta t^2 \\ \mathbf{I_{3x3}}\Delta t \end{bmatrix}^T = \begin{bmatrix} \frac{1}{4}\mathbf{I_{3x3}}\Delta t^4 & \mathbf{0_{3x3}} \\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}}\Delta t^2 \end{bmatrix} \sigma_{\mathbf{e}}^{\mathbf{2}}. \tag{4.15}$$

We can use all the parameters to define our system model as shown in (4.1):

$$\mathbf{x_k} = \mathbf{F}\mathbf{x_{k-1}} + \mathbf{B}\mathbf{acc_{k-1}} + \mathbf{w_{k-1}}, \tag{4.16}$$

Using the model in (4.12) combined with the process noise in the (4.13) and covariance matrix in the (4.15), we can write our state transition matrix, control input matrix and process noise vector as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I_{3x3}} & \mathbf{I_{3x3}}\Delta t \\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}} \end{bmatrix}, \tag{4.17}$$

$$B = \begin{bmatrix} \frac{1}{2}\mathbf{I_{3x3}}\Delta t^2 \\ \mathbf{I_{3x3}}\Delta t \end{bmatrix}^T, \tag{4.18}$$

$$\mathbf{w_{k-1}} \sim N(0, \mathbf{Q}). \tag{4.19}$$

For the measurement vector $\mathbf{z_k}$, we can write the following equation:

$$\mathbf{z_k} = \begin{bmatrix} \mathbf{pos_k} \\ \mathbf{vel_k} \end{bmatrix} + \mathbf{v_k}, \tag{4.20}$$

where we defined $\mathbf{v_k}$ as:

$$\mathbf{v_k} \sim N(0, \mathbf{R}), \tag{4.21}$$

From (4.3) and (4.20), we see H matrix is an identity matrix with 6 × 6 size.

We have all the parameters of the Kalman Filter Algorithm to estimate the tracker drone positions. We will use these estimates with a probability parameter to calculate the cost function in the optimization problem.

4.2 Path Planning for the Relay Drone using Single/Multi-Hop Communication

After the estimation of the state vectors $\hat{\mathbf{x_t}}$ of the tracker drones, the relay drone needs to solve a dynamic optimization problem to create its own path autonomously in order to build a connected network between the drones. We would like to provide a connected network for a given time interval $[0, T]$. If we try to solve the optimization problem only once at the beginning, if T is large, because of the time and computational complexity, the optimization problem will become intractable. Also, we need to be aware of the environmental changes, target paths, and hence drone paths. Therefore, we will solve the optimization problem using a receding horizon.

We assume we have step size of t_s. We will estimate the drone positions for $[t, t + t_k]$ using our estimator where $t_k = K \times t_s$. Then we will create a path planning for $[t, t + t_f]$ where $t_f = F \times t_s$ which maximizes the duration that drones remain connected. K and F are chosen as $F \mid K$.

Considering the physical constraints of a UAV, we consider a constant-altitude kinematic model for the relay drone. The dynamics of the drone are:

$$V_x(t) = V_D(t) \cos(\theta(t)), \tag{4.22}$$

$$V_y(t) = V_D(t) \sin(\theta(t)), \tag{4.23}$$

$$|\theta(t) - \theta(t-1)| \leq \theta_{max}, \tag{4.24}$$

$$V_{min} \leq V_D(t) \leq V_{max}. \tag{4.25}$$

Our relay drone state parameter is $\mathbf{x(t)} = [\mathbf{p(t)}, \mathbf{v(t)}]$ and control input parameters $\mathbf{u(t)} = [V_D(t), \theta(t)]$ which are the velocity and angle of the drone, respectively. The velocity can have a minimum velocity of V_{min} and maximum velocity of V_{max}. The change in the drone angle for each step size is limited to $[-\theta_{max}, \theta_{max}]$.

The aim of the optimization problem is to find the optimal input vector $\mathbf{u}^*\mathbf{(t)} \in U$ which maximizes the probability that the network topology is connected at time t. U represents the set of possible inputs that is defined in (4.24) and (4.25). For arbitrary time t_0, given the variable $\mathbf{u}^*\mathbf{(t)}$ for the interval $t \in [t_0, t_0 + t_f]$ the optimal state parameter $\mathbf{x}^*\mathbf{(t)}$ can be found.

The dynamic optimization problem can be formulated similar to a Bolza-type optimal control problem (OCP) [34]. The Lagrange term can be defined as:

$$\Phi_L(t, u) = \rho(t, \mathbf{x(t)}, \mathbf{u(t)}), \tag{4.26}$$

where $\rho(t, \mathbf{x(t)}, \mathbf{u(t)})$ donates the probability of connected network topology with state $\mathbf{x_t}$ at time t. the Mayer term can be defined as:

$$\Phi_M(t) = d(\mathbf{x(t_0 + t_f)}, \mathbf{x_t(t_0 + t_f)}), \tag{4.27}$$

where $d(\mathbf{x(t_0 + t_f)}, \mathbf{x_t(t_0 + t_f)})$ is a function of distance between drones for at the end of the optimization horizon for $t \in [t_0, t_0 + t_f]$. The resulting optimization problem is given by:

$$\begin{aligned} \underset{u(t)}{\text{maximize}} \quad & \rho(t, \mathbf{x(t)}, \mathbf{u(t)}) + d(\mathbf{x(t_0 + t_f)}, \mathbf{x_t(t_0 + t_f)}) \\ \text{subject to} \quad & (4.22), (4.23), (4.24), (4.25) \\ & t \in [t_0, t_0 + t_f] \end{aligned} \tag{4.28}$$

Our aim is choosing the best $V_D(t)$ and $\theta(t)$ for $t \in [t_0, t_0 + t_f]$ to maximize the objective function. The objective function $\Phi_L(t, u) + \Phi_M(t)$ changes depending on the network type single-hop or multi-hop communication, respectively. We will represent details in the upcoming sections.

To check the impact of the optimization horizon, we will also solve the optimization problem with $F = 0$ thereby $t_f = 0$.

4.2.1 Single-Hop Communication

In this section, tracker drones and the relay drone should communicate with a single-hop communication. The velocity and position vector of i^{th} drone is defined as $\mathbf{vel_i(t)}$ and $\mathbf{pos_i(t)}$ at time t. The position of the relay drone is $\mathbf{p(t)}$ at time t.

All the tracker drones should be connected to the relay drone directly. To check the connectivity between relay drone and the tracker drones we will first define the distance parameter as:

$$d_i(t) = ||\mathbf{p(t)} - \mathbf{pos_i(t)}||_2, \tag{4.29}$$

where $d_i(t)$ is the distance between i^{th} tracker drone and relay drone at time t (4.29).

In order to model the inaccuracy in estimating the velocity and position of drones, we use the following probabilistic model as opposed to a deterministic model.

We assign a probable convex region to the estimated drone to guarantee the maximization of the connectivity between drones. In this case, we assign a circular region with radius $r_i(t) = k \cdot \mathbf{vel_i(t)}$ and center $\mathbf{pos_i(t)}$ to a tracker drone. Assuming a uniformly distributed position error, calculate the intersection area between the circular region and the area in the communication range to get the connection function. The visualization for an example scenario is shown in Figure 4.1.

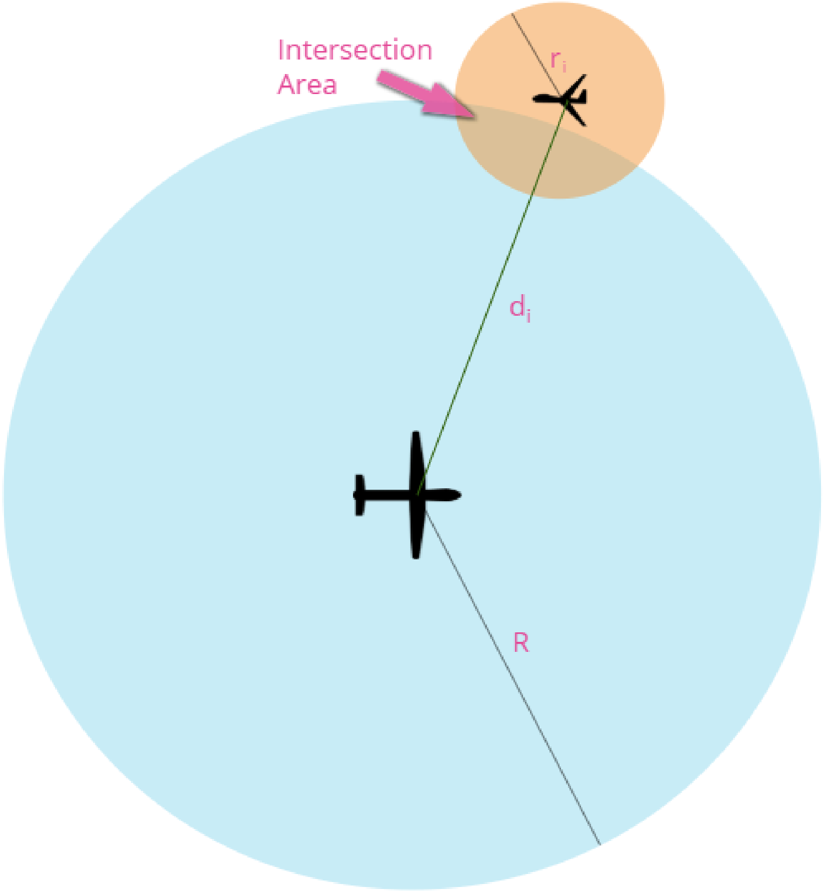


Figure 4.1: Blue circle is the communication area of the relay drone and the orange circular region is the estimated drone position area.

To find the probabilistic connectivity function ρ, we need to find the intersection area, and the probability of the drone being in the intersection area. Our assumption on the two circle radii is $R \gg r_i(t)$, therefore we know that to have an intersection $d_i(t) < R + r_i(t)$ should be satisfied, otherwise the area is equal to zero. For $d_i(t) < R + r_i(t)$, the intersection area between two circles can be found as:

$$I_i(t) = R^2 \arccos\left(\frac{R^2 - r_i(t)^2 + d_i(t)^2}{2d_i(t)R}\right) + r_i(t)^2 \arccos\left(\frac{r_i(t)^2 - R^2 + d_i(t)^2}{2d_i(t)r_i(t)}\right)$$
$$-\frac{1}{2}\sqrt{(R + r_i(t) + d_i(t))(R - r_i(t) + d_i(t))(-R + r_i(t) + d_i)(R + r_i(t) - d_i(t))}. \tag{4.30}$$

We can write the probability of connection ρ_i as a rate of intersection area to overall circular region of tracker drone. The circular area of the i^{th} drone is $\pi r_i(t)^2$. So the probability of connection for each drone becomes:

$$\rho_i(t, \mathbf{x(t)}, \mathbf{u(t)}) = \begin{cases} \frac{I_i}{\pi r_i(t)^2}, & \text{if } d_i(t) \leq R + r_i(t) \\ 0, & \text{otherwise.} \end{cases} \tag{4.31}$$

To calculate the probabilistic connectivity function ρ, we multiply each ρ_i, assuming that position errors of different UAVs are independent. We can write the probability that all tracker drones are connected to the relay UAV as:

$$\rho(t, \mathbf{x(t)}, \mathbf{u(t)}) = \prod_{i=1}^{N-1} \rho_i(t, \mathbf{x(t)}, \mathbf{u(t)}), \tag{4.32}$$

where N is the number of drones, including the relay drone.

Single-hop communication means relay drone needs to be directly connected to the tracker drones. Our approach here is to stay close to each tracker drone to adjust any changes that might happen in their movements. Hence, we use a function $d(\mathbf{x(t)}, \mathbf{x_t(t)})$ that try to minimize the maximum of the distances from relay drone to tracker drones. We do not want this part to dominate the objective function so we will multiply it with a small $\epsilon > 0$ value. We can write the function as follows:

$$d(\mathbf{x(t)}, \mathbf{x_t(t)}) = -\epsilon \cdot \max\{d_i(t)\}. \tag{4.33}$$

The objective function of the optimization problem can be defined as:

$$\begin{aligned} \underset{u(t)}{\text{maximize}} \quad & \prod_{i}^{N-1} \rho_i(t, \mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t})) - \epsilon \cdot \max\{d_i(t_0 + t_f)\} \\ \text{subject to} \quad & (4.22), (4.23), (4.24), (4.25) \\ & t \in [t_0, t_0 + t_f] \end{aligned} \tag{4.34}$$

The path planner component solves this optimization problem with the position estimates of the other drones in a chosen horizon window and creates the trajectory.

4.2.2 Multi-Hop Communication

In this section, tracker drones and the relay drone can communicate with multi-hop communication, i.e. tracker drones do not have to be directly connected to the relay drone. Instead, we are looking for a connected network topology among all N drones. We will use the objective function that is established in the (4.28) for the dynamic optimization problem. The Lagrange term $\rho(t, \mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$ and the Mayer term $d(\mathbf{x}(\mathbf{t_0} + \mathbf{t_f}), \mathbf{x_t}(\mathbf{t_0} + \mathbf{t_f}))$ of the objective function $\Phi_L(t, u) + \Phi_M(t)$ will be different for the multi-hop communication.

For the multi-hop communication, we create a network graph with N drones, including the relay drone. We assume each drone as a node in the graph. Then for all the possible $\binom{N}{2}$ edges, we define a state matrix $\mathbf{S}$ to calculate the probabilistic connectivity function ρ. State matrix contains all the possible connection combinations of edges. Therefore, we have a state matrix $\mathbf{S}$ of size $\binom{N}{2} \times 2^{\binom{N}{2}}$. To calculate the probability of having a connected network, we will calculate the connection probability of each edges. Then we will multiply it with the state matrix to get the value of the probabilistic connectivity function ρ.

To find the connection probability for each edge, we calculate the average intersection area between each drone. We will use the same model which is visualized in Figure 4.1. For the edges between the relay drone and the other drones, we use the probability of connection that we calculated in 4.31. However, for the network edges between two tracker drones we will use a different probabilistic model. Because when we calculate the probability of connection ρ_i, we do not use a inaccuracy model for the position of the relay drone. Positions and velocities of the both drones are estimated. Both of them are represented with a inaccuracy model. Hence, we calculate an average intersection area for the two tracker drones i and j which is used to calculate the probability of having a link between UAVs i and j. The average intersection area can be found as follows:

$$I_{ij}(t) = \int_{d_{ij}(t)-r_i(t)}^{d_{ij}(t)+r_i(t)} I_j(x,t) f_{d_{ij}(t),r_j(t)}(x,t)dx \tag{4.35}$$

where $I_j(x,t)$ is the intersection area between drone j with communication radius R_c and drone i with location region with radius r_i at time t. The assumption here is that all tracker drones have the same communication radius. x is a variable for the distance between drone i and drone j. We integrate over the distance between two drones to calculate the average intersection area. $f_{d_{ij},r_i}(x)$ is the probability density function (PDF) of distance between two drones.

$$\begin{aligned} I_j(x,t) = {} & R_c^2 \arccos\left(\frac{R_c^2 - r_i(t)^2 + x^2}{2xR_c}\right) + r_i(t)^2 \arccos\left(\frac{r_i(t)^2 - R_c^2 + x^2}{2xr_i(t)}\right) \\ & -\frac{1}{2}\sqrt{(R_c + r_i(t) + x)(R_c - r_i(t) + x)(-R_c + r_i(t) + x)(R_c + r_i(t) - x)}. \end{aligned} \tag{4.36}$$

Geometrically we can calculate the $f_{d_{ij}(t),r_j(t)}(x,t)$ as:

$$f_{d_{ij}(t),r_j(t)}(x,t) = \frac{2\arccos\left(\frac{x^2 - r_j(t)^2 + d_{ij}(t)^2}{2d_{ij}x}\right)x}{\pi r_j(t)^2}, d_{ij}(t) - r_j(t) \leq x \leq d_{ij}(t) + r_j(t). \tag{4.37}$$

Using (4.35) for each pair of vertices in the matrix, we will get a $1\times\binom{N}{2}$ size $\mathbf{I(t)}$ vector. By multiplying these average intersection area probabilities with the state matrix $\boldsymbol{S}$ we get a $1\times 2^{\binom{N}{2}}$ size vector with probabilities for every possible network configuration. Multiplying the transpose of it with the one vector $\mathbb{1}_{1\times 2\binom{N}{2}}$ will give us the probabilistic connectivity function.

$$\rho(t, \mathbf{x(t)}, \mathbf{u(t)}) = \mathbb{1}(\mathbf{I(t)S})^T \tag{4.38}$$

Multi-hop communication means that the relay drone needs to provide a network that is connected. We have three possible approaches here such that the relay drone will position itself to enable connectivity for longer periods in the future as the positions of the drones evolve.

Nearest Point Algorithm:

The first of our approaches is to divide the drones into two sets S_1 and S_2, respectively. We set the drone which is the furthest away from the other drones as S_1. Rest of the drones are on S_2. From each set, we measure the distances of all drones to the relay drone and select the ones that are closest to the relay drone from both sets. Lets assume the drone k from S_1 and the drone l from S_2 are the closest to the relay drone with distances $d_k(t)$ and $d_l(t)$, respectively. We compare the distances of these two drones to the relay drone to find the longest. In each iteration, we choose the control input that minimizes the longest distance. We do not want this part to dominate the objective function so we will multiply it with a small $\epsilon > 0$ value. We can use this as the Meyer part of the OCP as follows:

$$d(\mathbf{x(t)}, \mathbf{x_t(t)}) = -\epsilon \cdot \max\{d_k(t), d_l(t)\} \tag{4.39}$$

The objective function of the resulting optimization problem can be written as:

$$\begin{aligned} \text{maximize} \quad & \mathbb{1}(\mathbf{I}(\mathbf{t})\mathbf{S})^T - \epsilon \cdot \max\{d_k(t_0+t_f), d_l(t_0+t_f)\} \\ \text{subject to} \quad & (4.22), (4.23), (4.24), (4.25) \\ & t \in [t_0, t_0+t_f] \end{aligned} \tag{4.40}$$

Mid-Point Algorithm:

A similar approach to the Nearest Point algorithm is, instead of taking only one drone position, we take the average of the drone positions from S_2 and set it as a new variable $\mathbf{pos_{mean}}(\mathbf{t})$. Then we calculate the distance between the relay drone and $\mathbf{pos_{mean}}(\mathbf{t})$ as $d_{mean}(t)$. Again in this approach, we compare the distances $d_k(t)$ and $d_{mean}(t)$ and in each iteration, we choose the control input that minimizes the longest distance among them. We do not want this part to dominate the objective function so we will multiply it with a small $\epsilon > 0$ value.

where $d_{mean}(t)$ and $\mathbf{pos_{mean}}(\mathbf{t})$ is defined as:

$$pos_{mean}(t) = \frac{\sum_{i \in S_2} pos_i(t)}{N-1} \tag{4.41}$$

$$d_{mean}(t) = ||p(t) - pos_{mean}(t)||_2 \tag{4.42}$$

We can use this as the Meyer part of the OCP as follows:

$$d(\mathbf{x}(\mathbf{t}), \mathbf{x_t}(\mathbf{t})) = -\epsilon \cdot \max\{d_k(t), d_{mean}(t)\} \tag{4.43}$$

The objective function of the optimization problem can be expressed as:

$$\begin{aligned} \text{maximize} \quad & \mathbb{1}(\mathbf{I}(\mathbf{t})\mathbf{S})^T - \epsilon \cdot \max\{d_k(t_0+t_f), d_{mean}(t_0+t_f)\} \\ \text{subject to} \quad & (4.22), (4.23), (4.24), (4.25) \\ & t \in [t_0, t_0+t_f] \end{aligned} \tag{4.44}$$

Hybrid Algorithm:

Our last approach will be a hybrid approach and we will see its results in the next chapter. For the multi-hop communication, we will first try to optimize the single-hop communication optimization problem in (4.34) to check if we can create a single-hop communication, because compared to multi-hop networks, this will have less average delay time. If we cannot create a single-hop communication, then we use the Mid-Point algorithm to create the trajectory.

Chapter 5

Numerical Results

In this section, we evaluate the performance of the estimator and the path planner. We also show the simulation results for the single-hop and multi-hop communication. We have 3 tracker drones and 1 relay drone, i.e, N = 4.

5.1 Analysis of Estimated Paths of Tracker Drones

In this section, we will present numerical examples to assess the performance of the estimator. A Monte Carlo simulation is conducted to check the robustness of the estimator.

We have three sets of available target models based on their speeds: slow targets, normal targets and fast targets. Therefore we defined a set of velocities for each drone to choose. Each drone have a velocity from the set $[25, 30, 35]$ m/s. Their initial positions are $\mathbf{pos_1} = [0, 0]$ m, $\mathbf{pos_2} = [500, 500]$ m, $\mathbf{pos_3} = [1000, 0]$ m. We run 3 hours of simulation.

For Kalman Filter, we used $\sigma_{\mathbf{acc}} = [0.3, 0.3]$, $\sigma_{\mathbf{gps}} = [3, 3]$ and $\sigma_{\mathbf{vel}} = [0.3, 0.3]$.

Our initial guesses are the following:

$$\mathbf{\hat{x}(0)} = \begin{bmatrix} 0 & 0 & 25\cos(\pi/4) & 25\sin(\pi/4) \\ 500 & 500 & 30 & 0 \\ 1000 & 0 & 35\cos(-\pi/2) & 35\sin(-\pi/2) \end{bmatrix} + \mathbf{w(0)}, \tag{5.1}$$

$$\mathbf{P(0)} = 4^2 \begin{bmatrix} \mathbf{I_{2x2}} & \mathbf{0_{2x2}} \\ \mathbf{0_{2x2}} & 0.1\mathbf{I_{2x2}}, \end{bmatrix} \tag{5.2}$$

$$\mathbf{Q} = \hat{\sigma}^{\mathbf{2}}_{\mathbf{acc}} \begin{bmatrix} \mathbf{I_{2x2}} & \mathbf{0_{2x2}} \\ \mathbf{0_{2x2}} & 0.1\mathbf{I_{2x2}}, \end{bmatrix} \tag{5.3}$$

$$\mathbf{R} = \begin{bmatrix} \hat{\sigma}^{\mathbf{2}}_{\mathbf{gps}}\mathbf{I_{2x2}} & \mathbf{0_{2x2}} \\ \mathbf{0_{2x2}} & \hat{\sigma}^{\mathbf{2}}_{\mathbf{vel}}\mathbf{I_{2x2}}, \end{bmatrix} \tag{5.4}$$

where $\mathbf{w(0)} \sim N(0, R)$, $\hat{\sigma}_{\mathbf{acc}} = 10\sigma_{\mathbf{acc}}$, $\hat{\sigma}_{\mathbf{gps}} = 10\sigma_{\mathbf{gps}}$, $\hat{\sigma}_{\mathbf{vel}} = 10\sigma_{\mathbf{vel}}$.

The trajectories of the drones that we are trying to estimate can be seen in Figure 5.1. The tracker drones change their velocity and angle approximately every 5 minutes throughout the simulation. The drone positions are specified with data points for each 30 minutes to visualize the trajectories better.

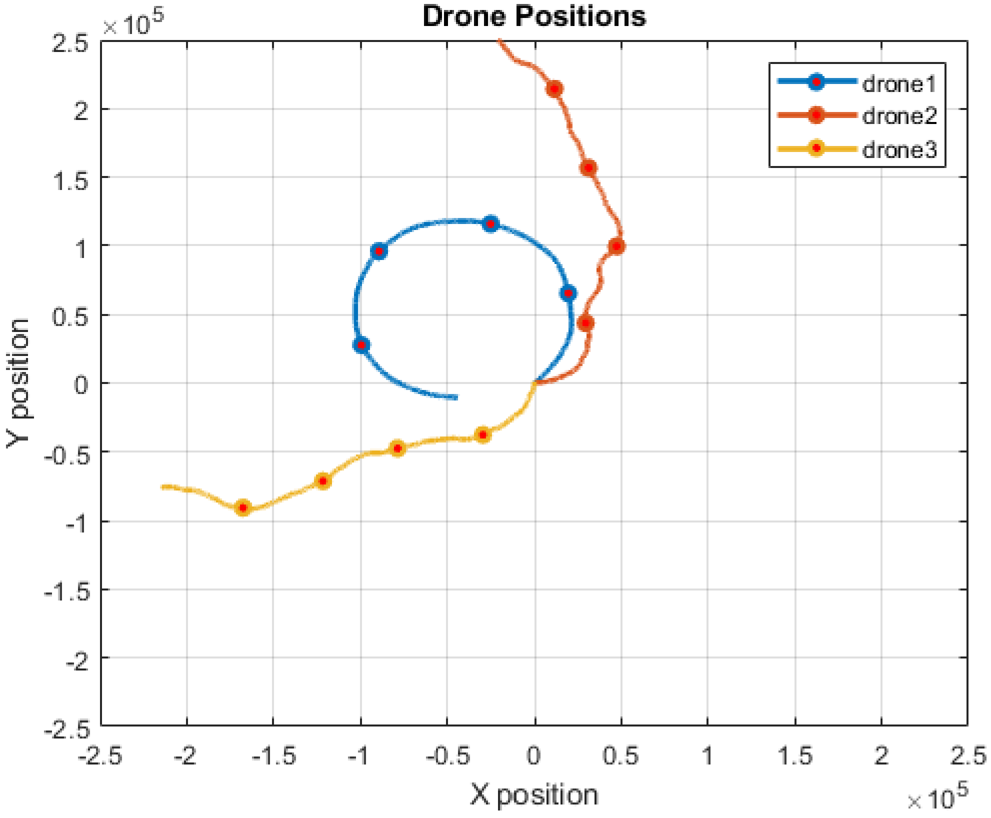


Figure 5.1: Drone Trajectories.

In Figure 5.2 and Figure 5.3, we can see the true position and velocity values for drone$_1$ on the x and y planes. We observe that the true values and the estimated values matches. The error and Root Mean Square Error (RMSE) values for the position and the velocity that we get from estimation can be seen in Figures 5.4 and 5.5. We also run a Monte Carlo simulation with $M = 100$ to see the robustness of the filter. It can be seen from Figures 5.6 and 5.7 that average RMSE values for the estimated values are very small compared to the actual values.

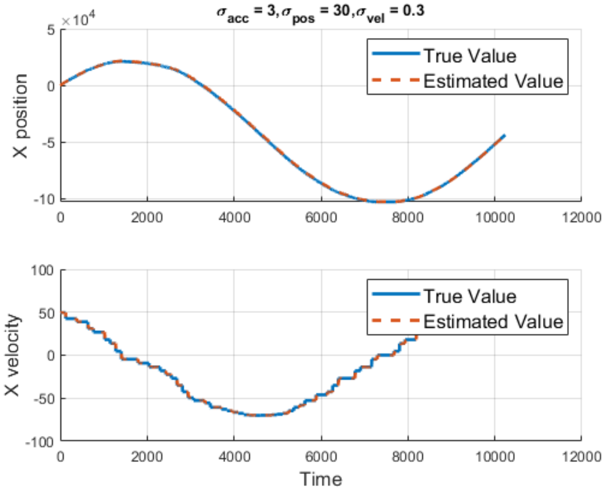


Figure 5.2: Comparison of true values and the estimation values for the pos_{1x} and vel_{1x}.

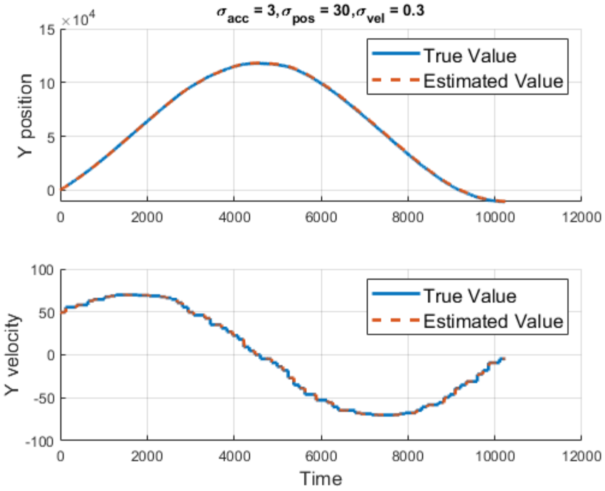


Figure 5.3: Comparison of true values and the estimation values for the pos_{1y} and vel_{1y}.

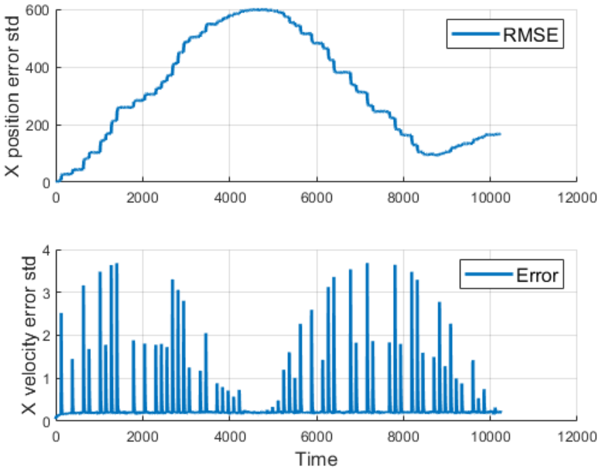


Figure 5.4: The RMSE values for the pos_{1y} and vel_{1y} in the simulation.

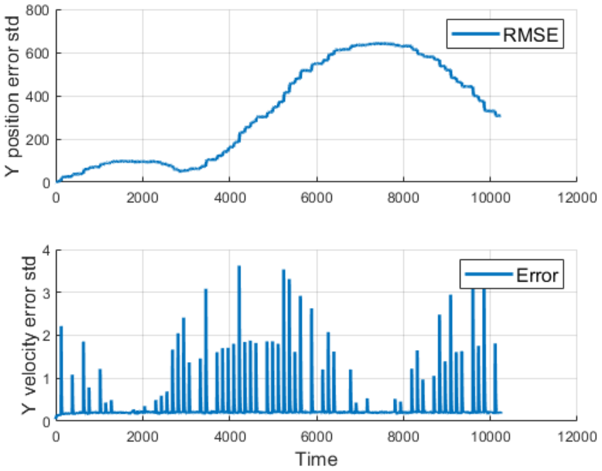


Figure 5.5: The RMSE values for the pos_{1x} and vel_{1x} in the simulation.

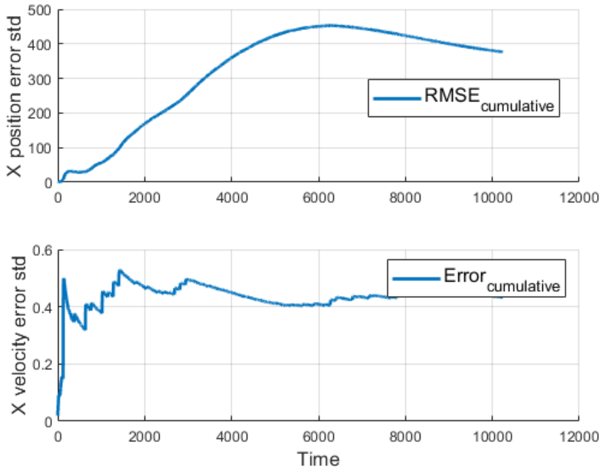


Figure 5.6: Average RMSE values for the pos_{1x} and vel_{1x} obtained by Monte Carlo simulations.

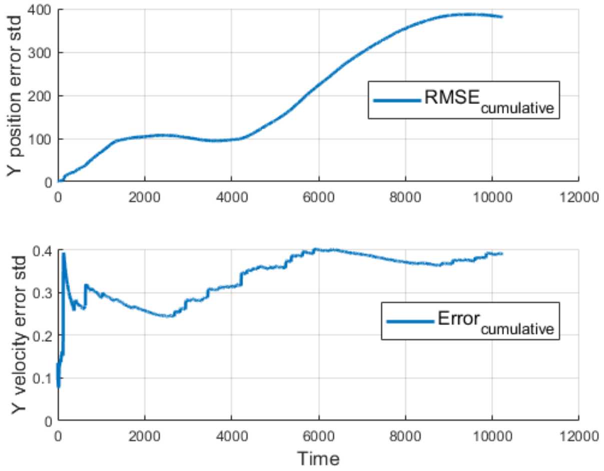
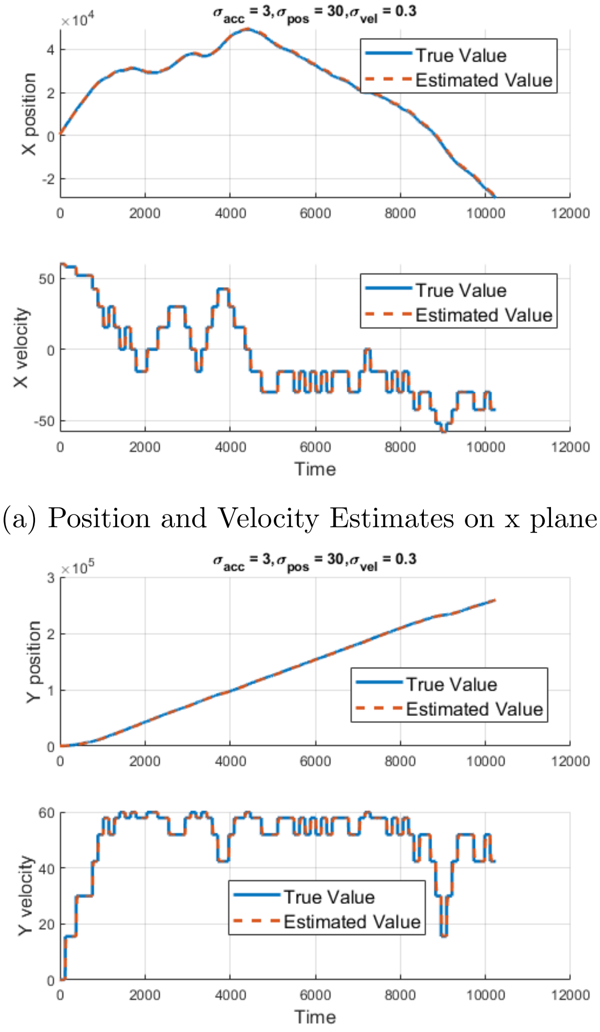


Figure 5.7: Average RMSE values for the pos_{1y} and vel_{1y} obtained by Monte Carlo simulations.

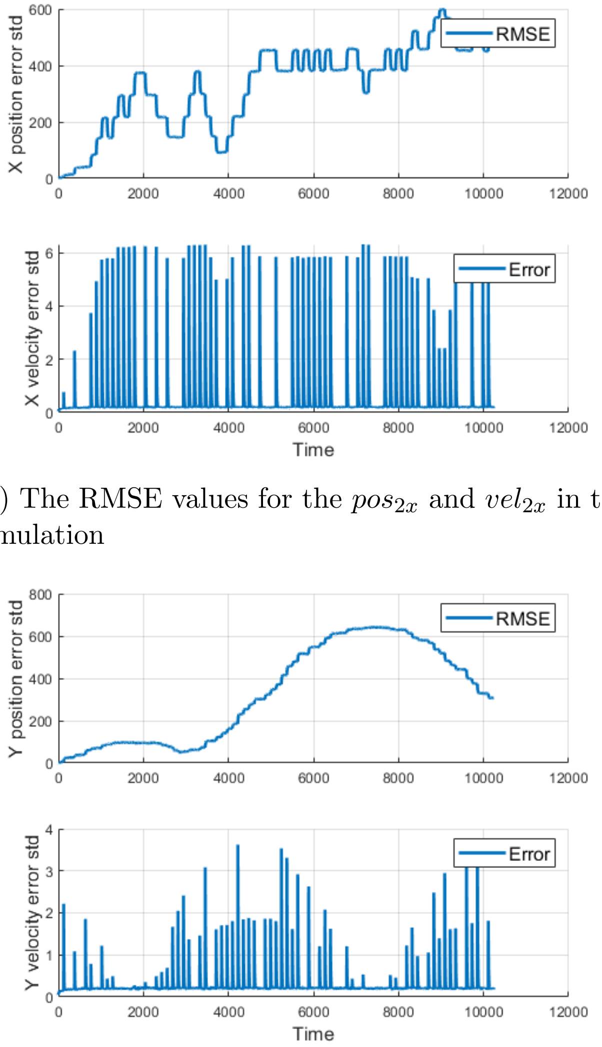
The estimation results for the other drones can be found in Figures 5.8-5.13.



(a) Position and Velocity Estimates on x plane.

(b) Position and Velocity Estimates on y plane.

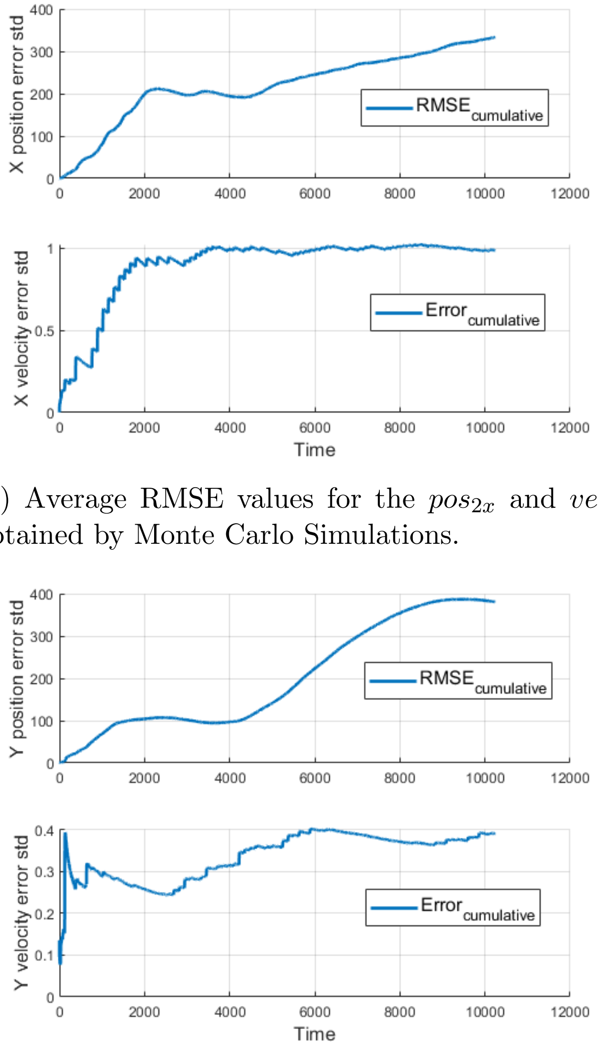
Figure 5.8: Comparison of true values and the estimation values.



(a) The RMSE values for the pos_{2x} and vel_{2x} in the simulation

(b) The RMSE values for the pos_{2y} and vel_{2y} in the simulation

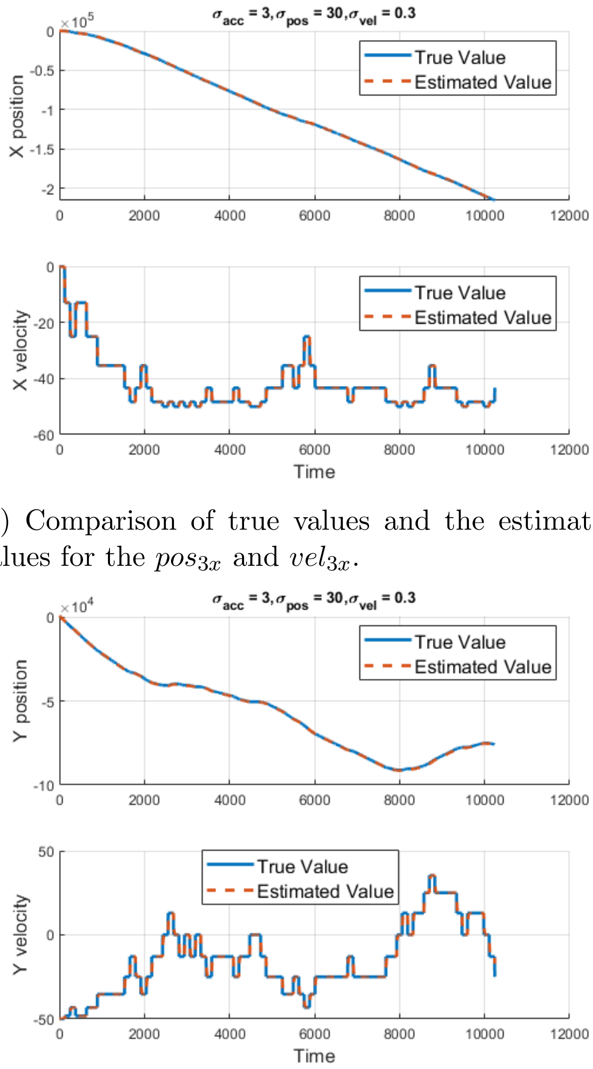
Figure 5.9: The RMSE values for estimated values.



(a) Average RMSE values for the pos_{2x} and vel_{2x} obtained by Monte Carlo Simulations.

(b) Average RMSE values for the pos_{2y} and vel_{2y} obtained by Monte Carlo Simulations.

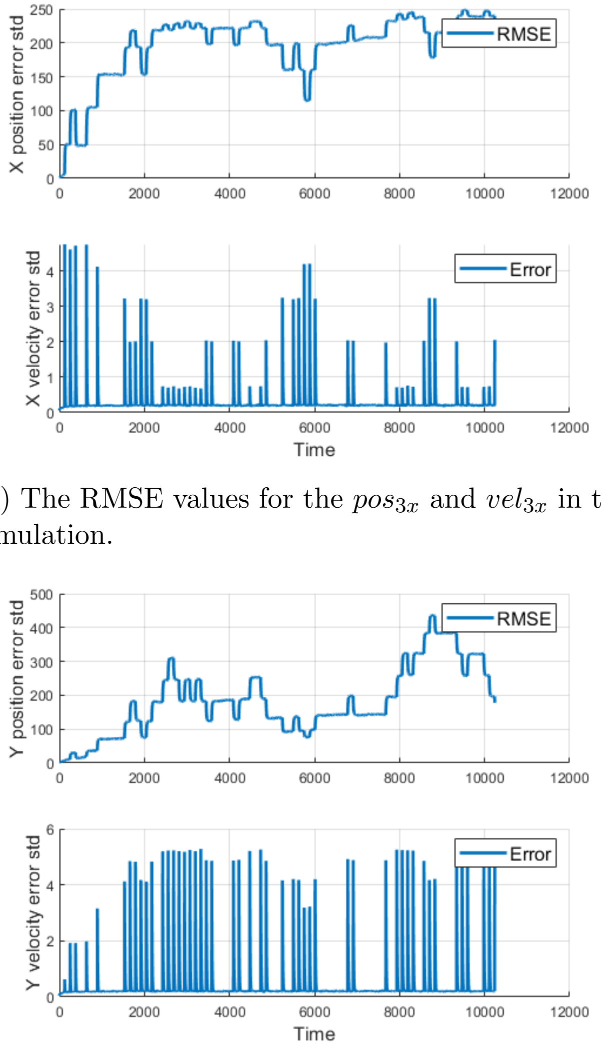
Figure 5.10: Average RMSE values obtained by Monte Carlo simulations.



(a) Comparison of true values and the estimation values for the pos_{3x} and vel_{3x}.

(b) Comparison of true values and the estimation values for the pos_{3y} and vel_{3y}.

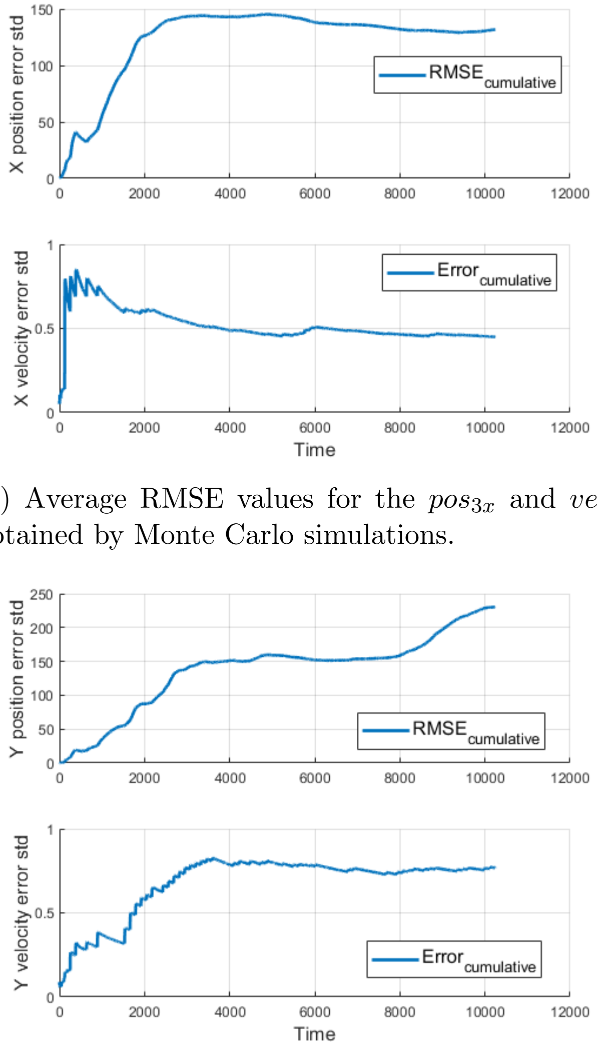
Figure 5.11: Comparison of true values and the estimation values.



(a) The RMSE values for the pos_{3x} and vel_{3x} in the simulation.

(b) The RMSE values for the pos_{3y} and vel_{3y} in the simulation.

Figure 5.12: The RMSE values for estimated values.



(a) Average RMSE values for the pos_{3x} and vel_{3x} obtained by Monte Carlo simulations.

(b) Average RMSE values for the pos_{3y} and vel_{3y} obtained by Monte Carlo Simulations.

Figure 5.13: Average RMSE values obtained by Monte Carlo Simulations.

From the values and plots we see that we can estimate the trajectories and the velocities of the drones accurately.

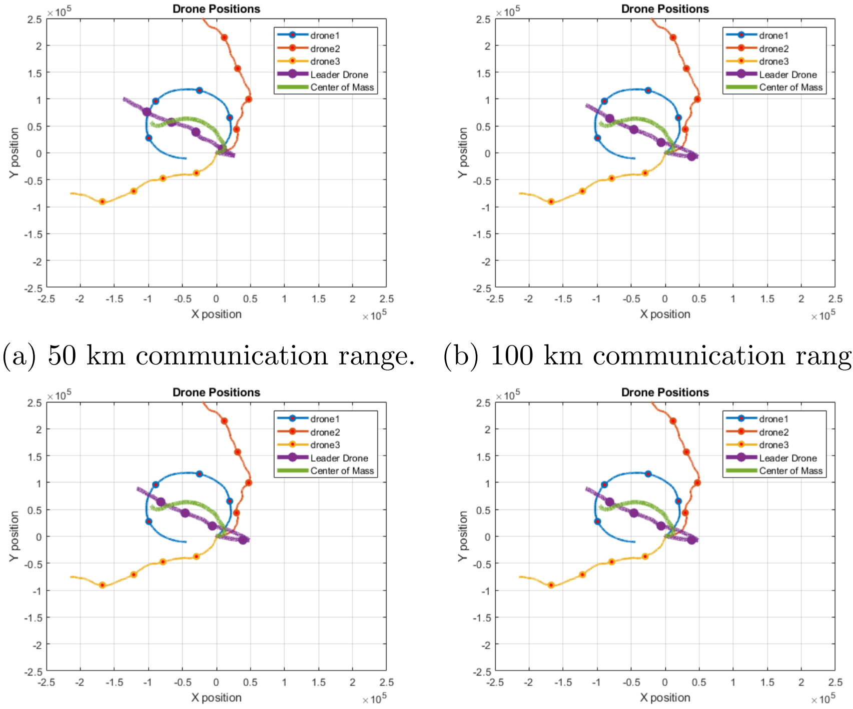
5.2 Path Planning using Single/ Multi-Hop Communication

We simulated 20 different scenarios to assess the performance of the path planner for both single and multi-hop communication. We use 4 different communication ranges [50, 100, 150, 200] km. Our simulation lasts 171 minutes, with each time step being 2 seconds. The relay drone estimates the drone positions and velocities every 30 seconds. We solve the optimization problem under two cases: with no horizon and 8 seconds optimization horizon window. To solve the optimization problem, we put some constraints to the velocity and the angle parameters. At each time step, the path planner chooses its velocity from the set [20, 30, 40] m/s and its bank angle either stays same, or it increases or decreases by 30°.

5.2.1 Single-Hop Communication Results

For the sake of consistency, we will show the results for the drone trajectories shown in Figure 5.1. As a baseline, we will show the trajectory for the center of mass as well of the tracker drones.

No Optimization Horizon



(a) 50 km communication range.

(b) 100 km communication range.

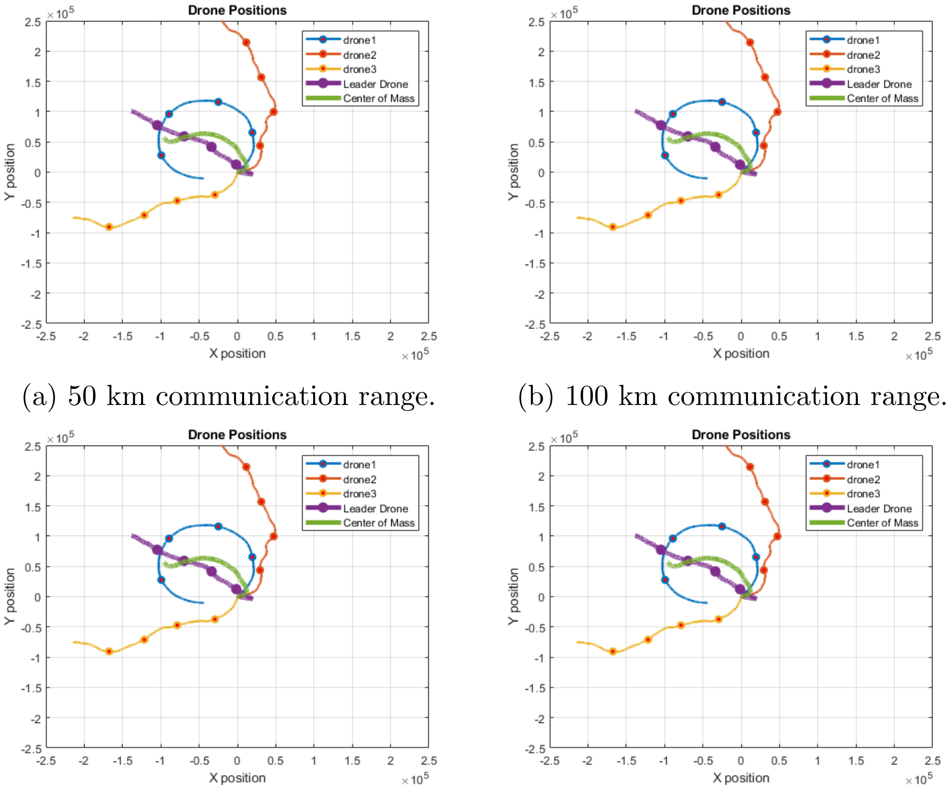
(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.14: Optimal trajectory of path planner with no optimization horizon.

The purple trajectory marked as leader drone in Figure 5.14 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 24 minutes. For $R_{com} = 100$ km, it ensures single-hop communication for 70 minutes. For $R_{com} = 150$ km, it ensures single-hop communication for 113 minutes. For $R_{com} = 200$ km, it stays connected to all the other drones throughout all the simulation. In comparison, center of mass stays connected for 23 minutes, 52 minutes, 110 minutes, 157 minutes, respectively. Connection time of each simulation for each communication range can be found in Table 5.1.

Optimization Horizon



(a) 50 km communication range.

(b) 100 km communication range.

(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.15: Optimal trajectory of path planner with optimization horizon.

The purple trajectory marked as leader drone in Figure 5.15 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 29 minutes. For R_{com} = 100 km, it ensures single-hop communication for 75 minutes. For R_{com} = 150 km, it ensures single-hop communication for 120 minutes. For R_{com} = 200 km, it stays connected to all the other drones throughout all the simulation. In comparison, center of mass stays connected for 23 minutes, 52 minutes, 110 minutes, 157 minutes respectively. Connection time of each simulation for each communication range can be found in Table 5.1.

Table 5.1: The duration of network connectivity established for each algorithms. NH: No time horizon, H: with time horizon, CM: Center of Mass.

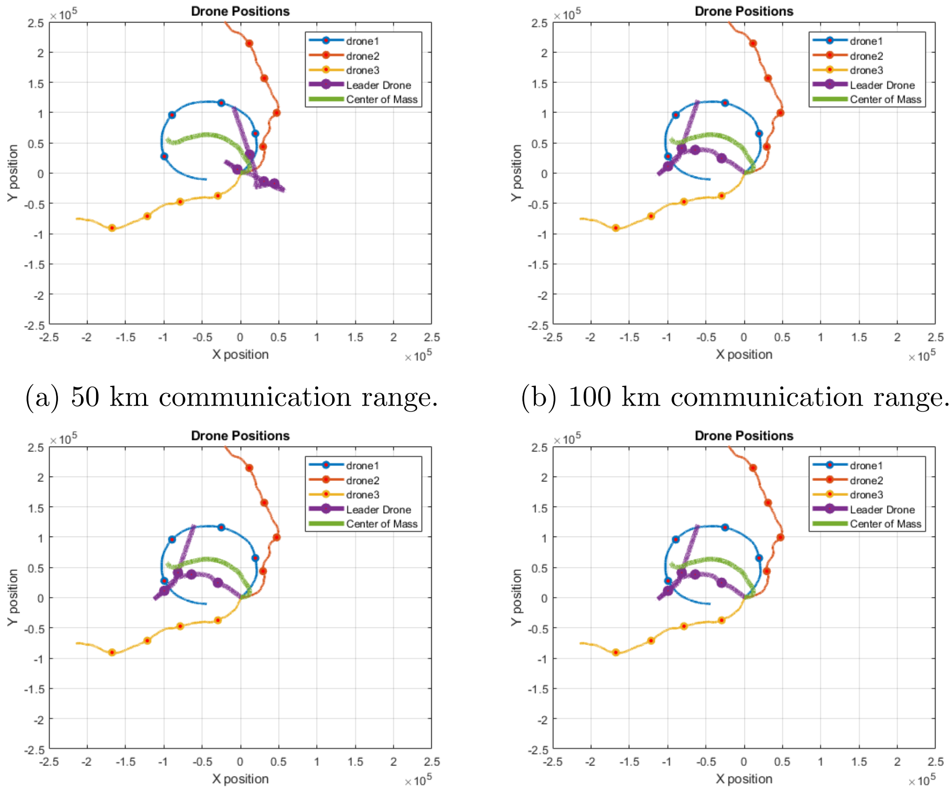
R_{com}	50 km			100 km			150 km			200 km		
Sim #	N	M	H	N	M	H	N	M	H	N	M	H
1	38	39	37	102	102	75	161	165	144	171	171	171
2	24	25	24	51	62	47	94	88	73	121	130	101
3	35	35	25	81	80	61	111	117	99	134	143	131
4	32	32	29	66	66	58	123	123	93	151	151	144
5	28	28	24	51	58	48	74	90	71	111	123	96
6	33	33	25	76	77	63	105	106	110	147	155	140
7	36	36	28	71	72	64	113	133	88	171	171	159
8	52	52	49	82	85	70	107	123	95	157	152	139
9	31	31	26	77	79	57	169	171	130	171	171	171
10	33	33	33	124	123	126	171	171	171	171	171	171
11	26	28	25	120	121	68	171	171	171	171	171	171
12	30	29	25	58	51	47	81	94	72	171	171	125
13	25	27	27	63	62	59	98	97	83	167	166	113
14	39	39	34	74	74	65	98	98	101	171	171	171
15	24	29	23	70	68	53	113	110	111	171	171	147
16	32	33	30	86	88	58	151	151	137	171	171	165
17	27	28	25	54	57	46	109	109	73	171	171	115
18	27	29	26	49	58	49	100	100	77	171	171	171
19	23	30	23	65	60	48	98	104	80	171	171	171
20	26	26	28	60	58	56	86	97	89	161	161	134

5.2.2 Multi-Hop Communication Results

For the sake of consistency, I will show the results for the drone trajectories shown in Figure 5.1. We get the simulation results for 4 different communication ranges. For multi-hop communication we used 3 different algorithms and 3 different objective functions that we mentioned in Chapter 4. We call the algorithms nearest point algorithm, mid point algorithm and hybrid algorithm respectively. We calculate the time that relay drone stays connected throughout the simulation. For each algorithm we have a table showing the amount of time each simulation stays connected.

No Optimization Horizon

Here is the optimal trajectory for the nearest point algorithm:



(a) 50 km communication range.

(b) 100 km communication range.

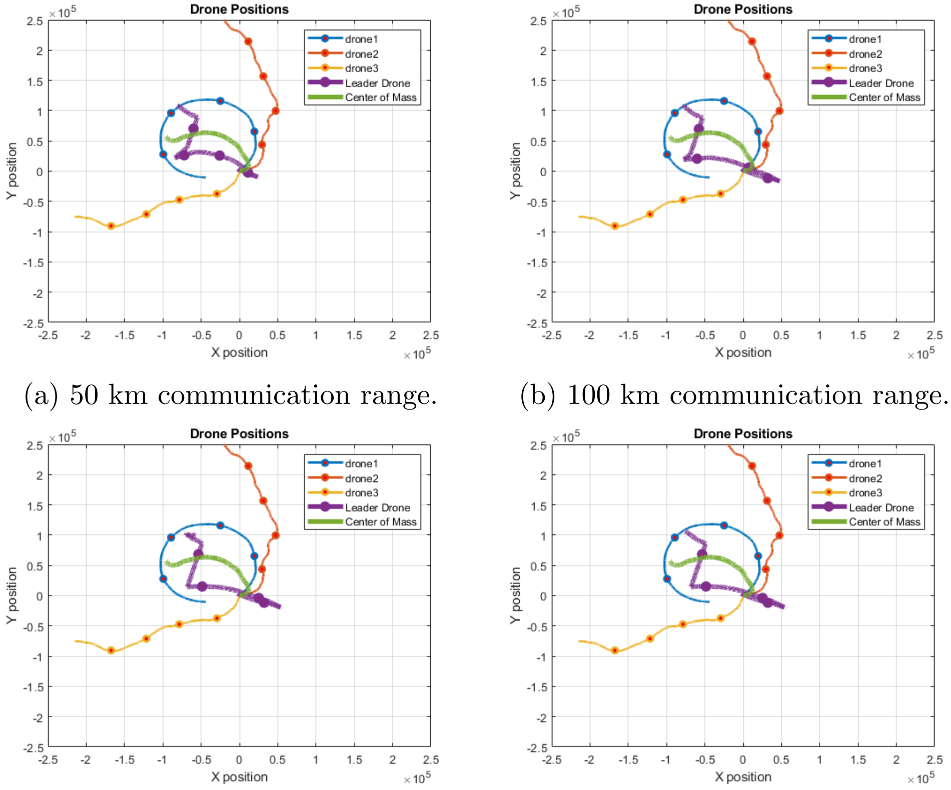
(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.16: Optimal trajectory of path planner with no optimization horizon.

The purple trajectory marked as leader drone in Figure 5.16 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 34 minutes. For R_{com} = 100 km, it ensures single-hop communication for 83 minutes. For R_{com} = 150 km, it ensures single-hop communication for 109 minutes. For R_{com} = 200 km, it stays connected to all the other drones for 167 minutes. Connection time of each simulation for each communication range can be found in Table 5.2.

Here is the optimal trajectory for the mid point algorithm:



(a) 50 km communication range.

(b) 100 km communication range.

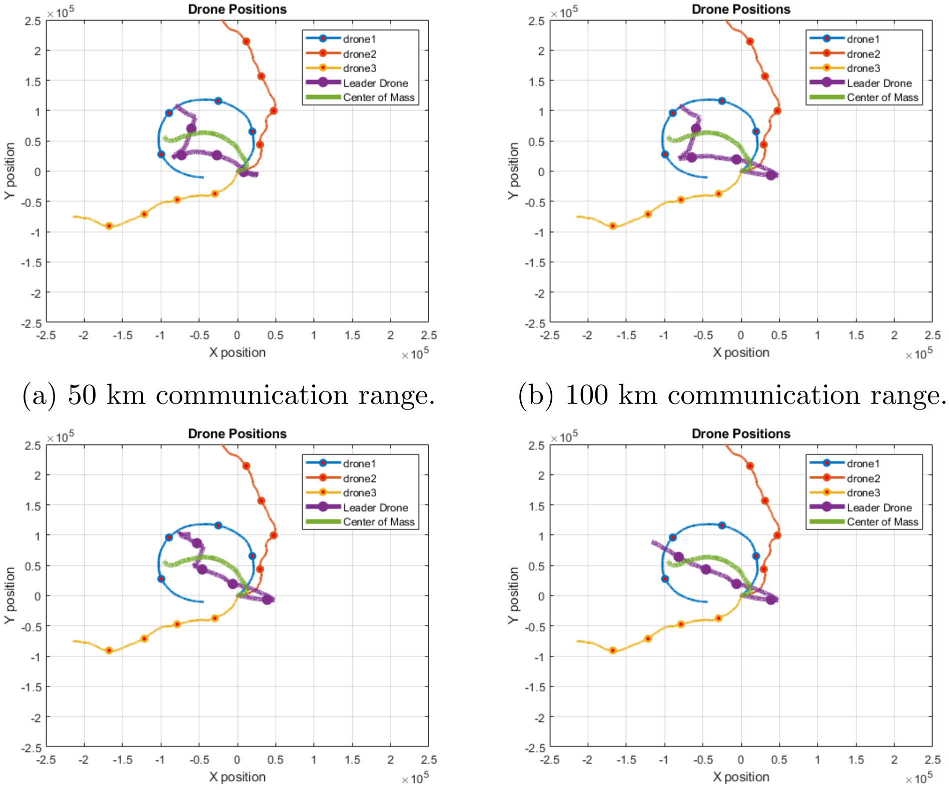
(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.17: Optimal trajectory of path planner with no optimization horizon.

The purple trajectory marked as leader drone in Figure 5.17 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 33 minutes. For R_{com} = 100 km, it ensures single-hop communication for 67 minutes. For R_{com} = 150 km, it ensures single-hop communication for 125 minutes. For R_{com} = 200 km, it stays connected to all the other drones throughout all the simulation. Connection time of each simulation for each communication range can be found in Table 5.2.

Here is the optimal trajectory for the hybrid algorithm:



(a) 50 km communication range.

(b) 100 km communication range.

(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.18: Optimal trajectory of path planner with no optimization horizon.

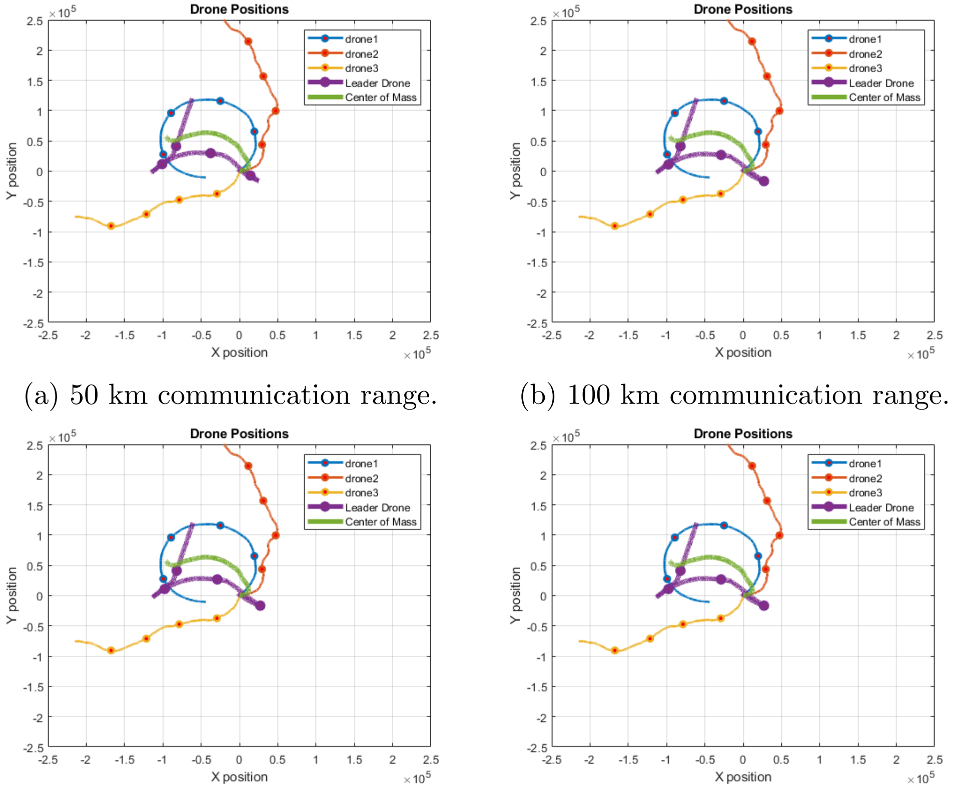
The purple trajectory marked as leader drone in Figure 5.18 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 32 minutes. For R_{com} = 100 km, it ensures single-hop communication for 83 minutes. For R_{com} = 150 km, it ensures single-hop communication for 151 minutes. For R_{com} = 200 km, it stays connected to all the other drones throughout all the simulation. Connection time of each simulation for each communication range can be found in Table 5.2.

Table 5.2: The duration of network connectivity established for each algorithms with no time horizon. N: Nearest Point Algorithm, M: Midpoint Algorithm, H: Hybrid single/multi-hop Algorithm

R_{com}	50 km			100 km			150 km			200 km		
Sim #	N	M	H	N	M	H	N	M	H	N	M	H
1	51	51	51	135	135	134	171	171	170	171	171	171
2	31	31	31	64	65	64	98	94	95	130	127	129
3	38	38	37	92	97	97	142	142	141	156	156	156
4	35	35	35	89	88	88	146	146	146	160	160	160
5	34	34	33	63	66	66	92	100	99	139	139	138
6	39	39	39	94	94	93	108	108	140	168	171	170
7	36	36	36	77	79	77	171	171	170	171	171	171
8	52	52	52	82	84	85	119	106	107	164	164	164
9	37	37	37	111	91	89	171	171	169	171	171	171
10	45	34	68	145	155	169	171	171	171	171	171	171
11	25	30	30	81	88	120	171	171	171	171	171	171
12	26	32	32	63	59	59	94	135	135	143	153	153
13	22	29	25	39	66	63	140	119	118	171	171	170
14	46	46	42	92	92	91	114	114	113	136	156	171
15	34	33	32	83	67	83	109	125	151	167	171	171
16	41	42	41	78	78	86	171	171	170	171	171	171
17	26	33	31	61	58	59	100	119	120	171	171	171
18	27	28	27	57	57	55	81	94	100	171	171	171
19	27	31	31	64	65	65	141	141	140	171	171	171
20	23	23	26	61	61	60	99	99	86	157	149	161

Optimization Horizon

Here is the optimal trajectory for the nearest point algorithm:



(a) 50 km communication range.

(b) 100 km communication range.

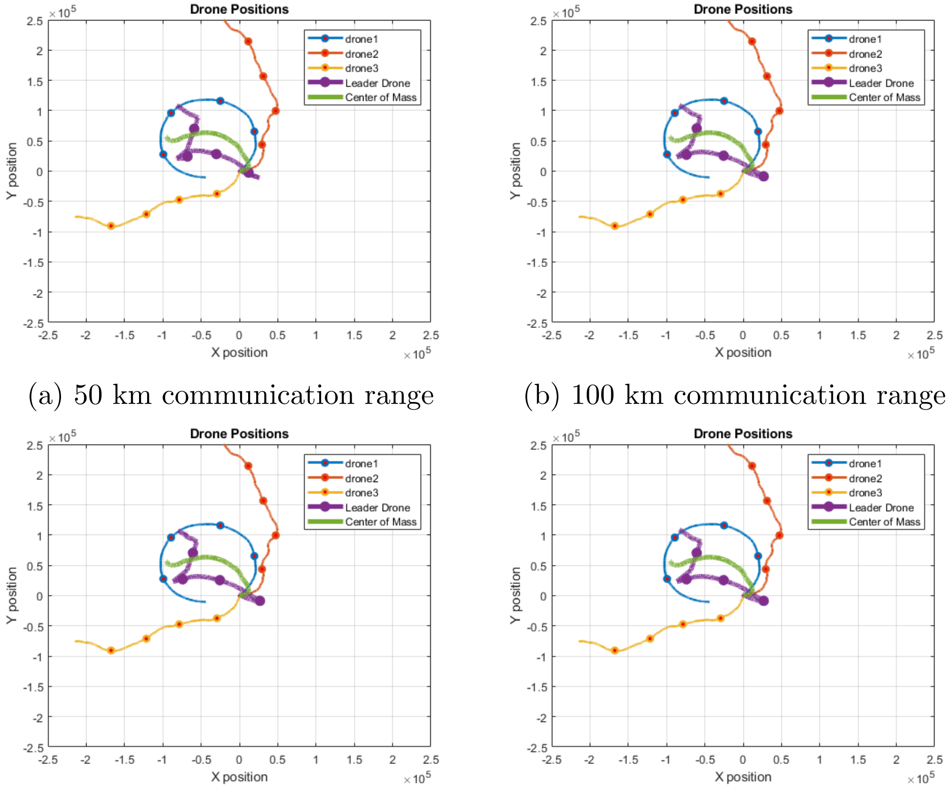
(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.19: Optimal trajectory of path planner with optimization horizon.

The purple trajectory marked as leader drone in Figure 5.19 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 29 minutes. For R_{com} = 100 km, it ensures single-hop communication for 83 minutes. For R_{com} = 150 km, it ensures single-hop communication for 109 minutes. For R_{com} = 200 km, it stays connected to all the other drones for 167 minutes. Connection time of each simulation for each communication range can be found in Table 5.3.

Here is the optimal trajectory for the mid point algorithm:



(a) 50 km communication range

(b) 100 km communication range

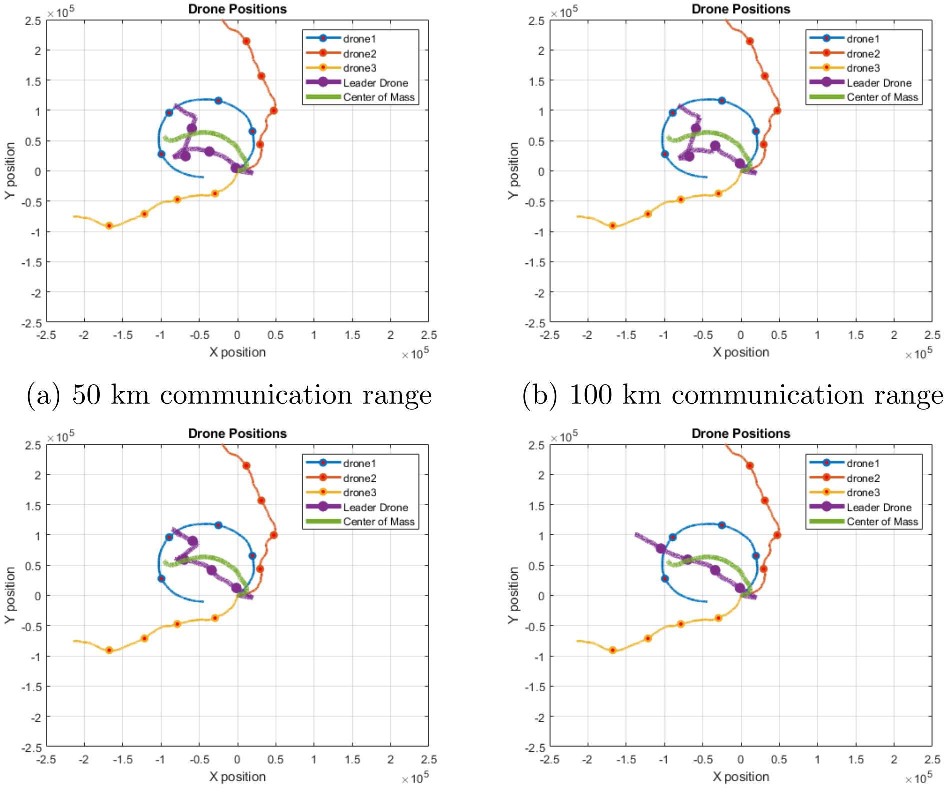
(c) 150 km communication range

(d) 200 km communication range

Figure 5.20: Optimal trajectory of path planner with optimization horizon

The purple trajectory marked as leader drone in Figure 5.20 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 33 minutes. For R_{com} = 100 km, it ensures single-hop communication for 83 minutes. For R_{com} = 150 km, it ensures single-hop communication for 125 minutes. For R_{com} = 200 km, it stays connected to all the other drones throughout all the simulation. Connection time of each simulation for each communication range can be found in Table 5.3.

Here is the optimal trajectory for the hybrid algorithm:



(a) 50 km communication range

(b) 100 km communication range

(c) 150 km communication range

(d) 200 km communication range

Figure 5.21: Optimal trajectory of path planner with optimization horizon

The purple trajectory marked as leader drone in Figure 5.21 is the optimal trajectory for the single-hop communication problem with no optimization horizon. When the communication range is 50 km, the relay drone stays connected for 34 minutes. For $R_{com} = 100$ km, it ensures single-hop communication for 83 minutes. For $R_{com} = 150$ km, it ensures single-hop communication for 148 minutes. For $R_{com} = 200$ km, it stays connected to all the other drones throughout all the simulation. Connection time of each simulation for each communication range can be found in Table 5.3.

Table 5.3: The duration of network connectivity established for each algorithms with time horizon. N: Nearest Point Algorithm, M: Midpoint Algorithm, H: Hybrid single/multi-hop Algorithm

R_{com}	50 km			100 km			150 km			200 km		
Sim #	N	M	H	N	M	H	N	M	H	N	M	H
1	51	51	51	136	136	136	171	171	162	171	171	171
2	28	31	32	64	64	63	93	93	93	125	126	142
3	37	38	38	97	98	98	142	142	142	156	156	156
4	35	35	35	89	89	89	146	146	146	160	160	160
5	32	33	33	61	65	65	99	99	91	137	137	136
6	38	39	39	94	94	94	108	126	141	169	171	170
7	36	36	36	77	77	77	171	171	171	171	171	171
8	46	52	52	83	84	86	120	107	147	164	164	164
9	37	37	37	101	84	90	171	171	171	171	171	171
10	45	52	68	144	156	165	171	171	171	171	171	171
11	30	30	30	81	81	121	171	171	171	171	171	171
12	32	32	32	64	64	64	135	135	135	153	153	153
13	22	29	27	39	66	62	145	118	100	171	171	170
14	47	46	42	92	92	92	114	114	114	134	146	171
15	29	33	34	83	83	83	109	125	148	167	171	171
16	40	42	42	78	88	88	171	171	170	171	171	171
17	33	33	32	55	58	63	107	121	120	171	171	171
18	28	28	29	57	57	58	89	100	100	171	171	171
19	31	32	32	66	66	66	141	141	140	171	171	171
20	23	23	26	61	61	58	99	99	97	157	149	161

5.3 Performance Analysis

For this part we compare the simulation results. For each simulation and each communication range, we have calculated theoretically the maximum amount of time the network can stay connected. We will compare our each algorithm with the maximum amount of time it can stay connected to assess our performance. We find the ratio between the time relay drone ensures network connectivity, and the maximum amount of time the network can stay connected. For both communication models, we will compare the results we get from no optimization horizon and optimization horizon.

5.3.1 Theoretical Calculation of the Amount of Time The Network Can Remain Connected

At a given time instance t, we checked if there is any feasible point in the given area that the relay drone can be placed so that the resulting network topology will be connected. A geometrical problem is formulated for both communication types.

Single-Hop Communication

To have a connected single-hop network, all tracker drones have to communicate with relay drone directly. At time t, let $d_i^{(x,y)}(t)$ denote the distance between point (x, y) and the tracker drone i. If the distance $d_i^{(x,y)}(t) \geq R_{com}$, the connection link cannot be formed. To call a topology feasible, there at least needs to be one point (x, y) in the map that satisfies $d_i^{(x,y)}(t) \leq R_{com}$ for $\forall i = 1, ..., N$.

For each time instance t, we form the circles centered at tracker drone positions with radius R_{com} and check if there is an intersection area or not between N circles. Then we sum them over the simulation time to get the maximum amount of time the network can remain connected. An illustration of a feasible area for

$N = 3$ can be seen in Figure 5.22.

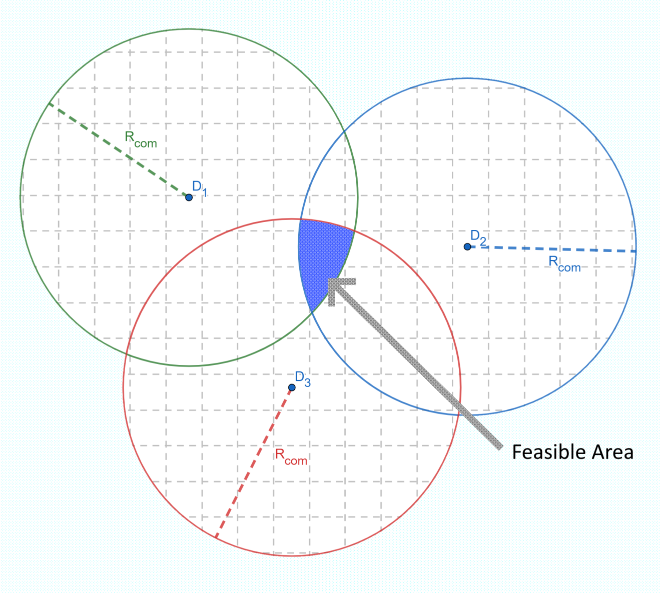


Figure 5.22: Feasible Area for Relay Drone to form a Single-Hop Communication.

Multi-Hop Communication

Having a multi-hop network means having a connected network graph between the drones. Checking if there is a feasible placement for the relay drone to have a connected network topology is more involved for multi-hop communications. For example, two subsets of tracker drones might be already communicating within the subsets and we might need the relay drone to be the bridge between these two subsets. We study below the case with $N = 3$ tracker drones to determine if there is a feasible area for the relay drone so that the resulting topology is connected. We have multiple topology configurations that can cover all the possible cases that we might have a feasible area.

We draw the communicable area for each drone as circular area with radius R_{com} and center with $\mathbf{pos_i(t)}$. The assumption is, a point in the circular area can communicate with the given drone i. Let d_{ij} donate the distance between the tracker drone i and j. Without loss of generality, we assume that $d_{12} \leq d_{13} \leq d_{23}$. Comparing the distance between drones to the communication range, we can find if there is a feasible area or not. The feasible points should be in the union of the communicable areas, otherwise the relay drone cannot communicate with any drones.

We have three possible cases with 3 drones: All 3 drones are communicating, only 2 drones can communicate or no drones are communicating. For the case where 3 drones are communicating, $d_{13} \leq R_{com}$ and any point in the union of the communicable area for each drone is a feasible point for the relay drone. d_{23} does not impact the communication in this case because three drones are connected event without the relay drone. An illustration of the configuration of this case can be seen in Figure 5.23.

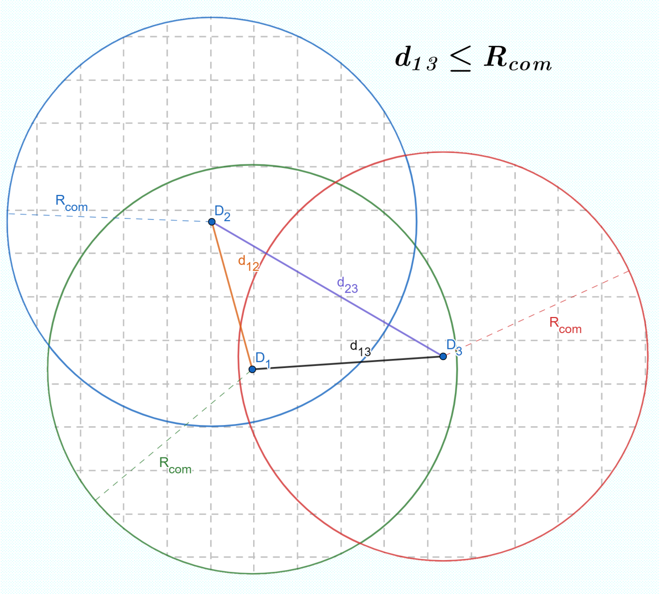
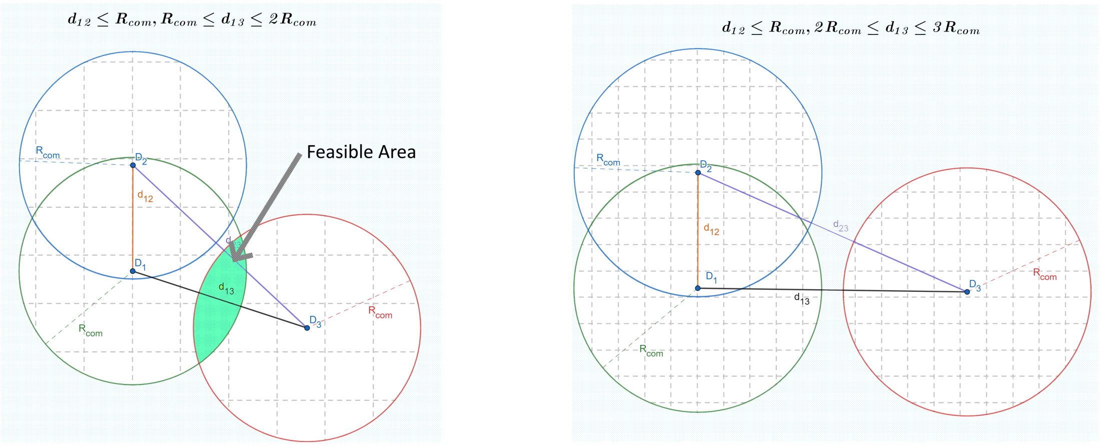


Figure 5.23: Illustration of the case with $d_{13} \leq R_{com}$.

If only 2 drones can communicate, it means that $d_{12} \leq R_{com}$ and $d_{13} \geq R_{com}$. We can divide this case into two subcases as $R_{com} \leq d_{13} \leq 2R_{com}$ and $2R_{com} \leq d_{13}$. For the first case, we can put the relay drone to the intersection area of drone 1 and drone 3 and it can communicate with both of them. Because drone 1 and drone 2 are already connected, we will have a connected network topology. For the second case, there is no area to put the relay drone to create a connected network topology because we cannot connect drone 1 and drone 3 with a relay drone. Illustrations for the configuration of these cases are shown in Figure 5.24.



(a) $d_{12} \leq R_{com} \leq d_{13} \leq 2R_{com}$.

(b) $d_{12} \leq R_{com}$ and $2R_{com} \leq d_{13}$.

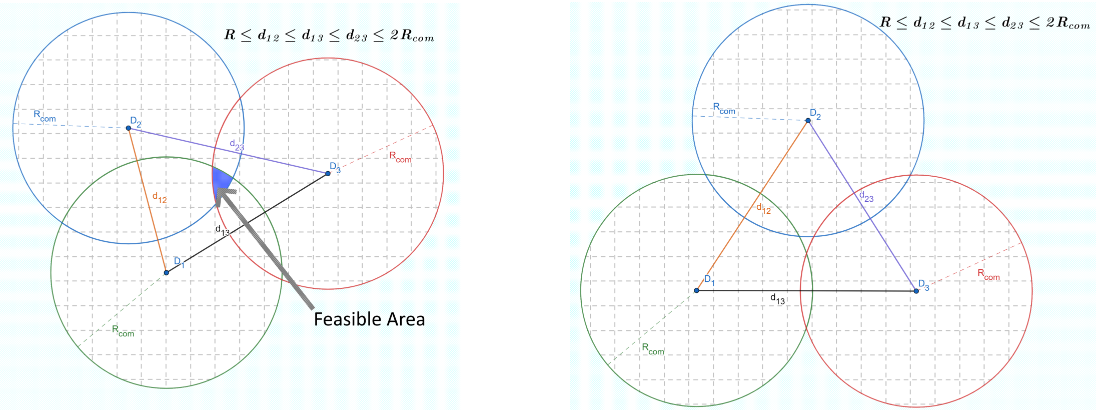
Figure 5.24: Illustration of the cases with $d_{12} \leq R_{com}$ and $d_{13} \geq R_{com}$.

If there is no connection between the drones, we have $d_{12} \geq R_{com}$. In this case, the relay drone needs to communicate with all 3 drones to create a connected network. This is the same as checking whether a single-hop network can be formed between the drones or not which was discussed earlier. For two circles to have an intersection area, the distance between their centers should be smaller than the summation of their radii. To have an intersection area between 3 circles, the distance between the circles should satisfy the following inequality $R_{com} \leq d_{12} \leq d_{13} \leq d_{23} \leq 2R_{com}$. There are two possible cases for this configuration with intersection area and no intersection area. Illustrations for the configuration of these cases can be seen in Figure 5.25.

We execute a feasibility check for each t and sum it through the simulation time to get the maximum possible duration that the topology can remain connected.

5.3.2 Performance Evaluation of Single-hop Communication

For single-hop communication, we also compare the results of the path planner with the center of mass results.



(a) There is an intersection area between circles.

(b) There is no intersection area between circles.

Figure 5.25: Illustration of the cases with $R_{com} \leq d_{12} \leq d_{13} \leq d_{23} \leq 2R_{com}$.

No horizon single-hop communication path planner achieves 92.98%, 94.60%, 93.22%, 98.10% of the maximum possible duration that the topology remains connected for R_{com} = 50, 100, 150, 200 km, respectively. On the other hand, optimization horizon single-hop communication path planner has 96.16%, 96.25%, 96.96%, 99.33% of the maximum possible duration that the topology remains connected for R_{com} = 50, 100, 150, 200 km, respectively. Center of mass, the algorithm we took as a baseline achieves 85.1%, 78.80%, 82.54%, 89.03% of the maximum possible duration that the topology remains connected for R_{com} = 50, 100, 150, 200 km, respectively. We can see the results in Figure 5.26.

We get the best results for the optimization horizon problem. That makes sense because we optimize the trajectory with more knowledge regarding the tracker drones. The reason why 200 km communication range gives the best results is because the position of the relay drone is not critical due to the extended range and both algorithms achieve best results.

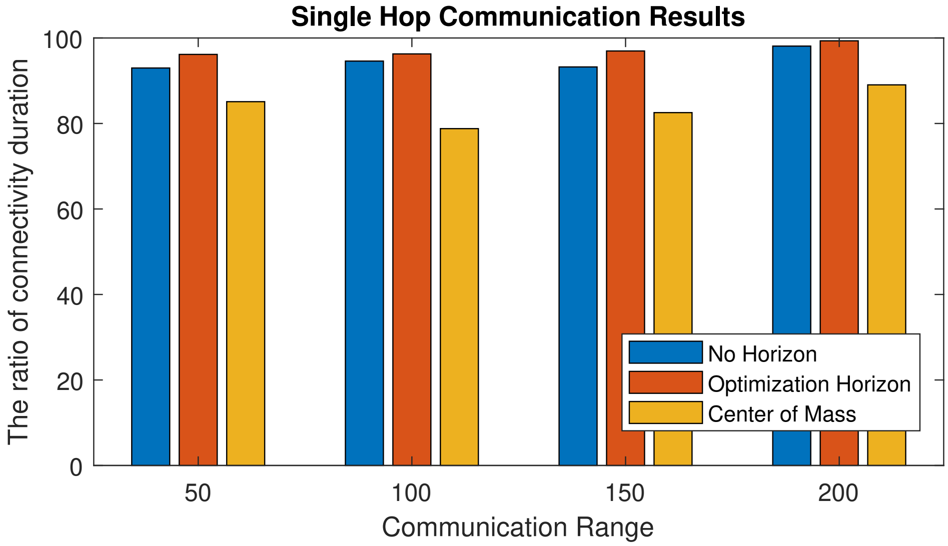


Figure 5.26: Performance of the Single-Hop Communication.

5.3.3 The Performance Evaluation of Multi-hop Communication

For multi-hop communication, we also compare the results of each algorithm and compare them to see which has the best performance.

No horizon multi-hop communication path planner results for each algorithm with R_{com} = 50, 100, 150, 200 km can be found in Table 5.4. The Optimization horizon multi-hop communication path planner results for each algorithm with R_{com} = 50, 100, 150, 200 km can be found in Table 5.5.

We can see the results in Figure 5.27. Optimization horizon hybrid algorithm gives the best results. Similar to the single-hop case, all three algorithms achieve the best performance for 200 km communication range.

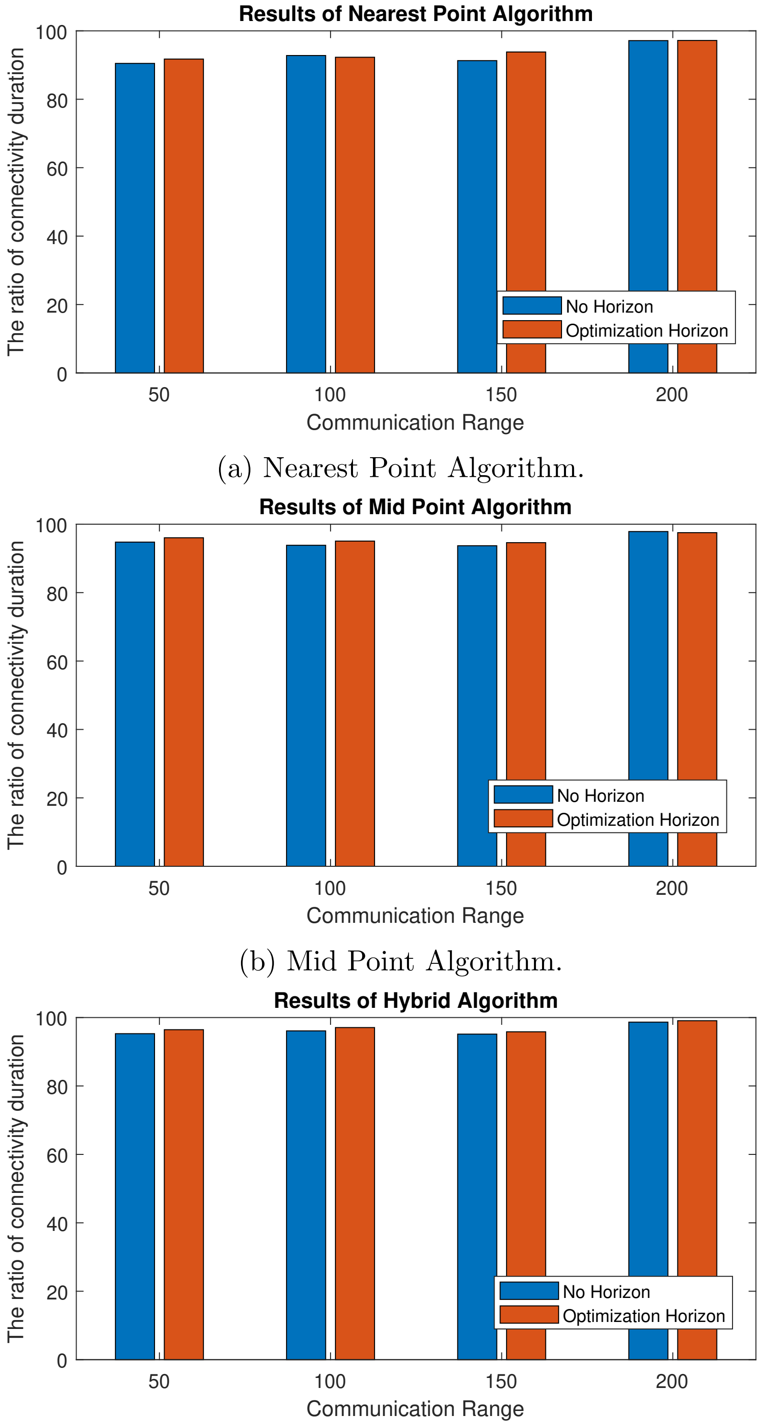
We can see the robustness of the algorithms with the box plots given in Figure 5.28. We see that the hybrid algorithm achieves the most robust performance among the three algorithms considered.

Table 5.4: The ratio of average connected duration and the maximum possible duration that the topology remains connected for Multi-hop communication with no optimization horizon.

	R_{com}			
Algorithms	50 km	100 km	150 km	200 km
Nearest Point	90.48	92.78	91.29	97.14
Mid Point	94.76	93.83	93.69	97.85
Hybrid	95.28	96.09	95.16	98.65

Table 5.5: The ratio of average connected duration and the maximum possible duration that the topology remains connected for Multi-hop communication with optimization horizon.

	R_{com}			
Algorithms	50 km	100 km	150 km	200 km
Nearest Point	91.74	92.27	93.81	97.18
Mid Point	96.02	95.06	94.60	97.52
Hybrid	96.45	97.07	95.83	99.07

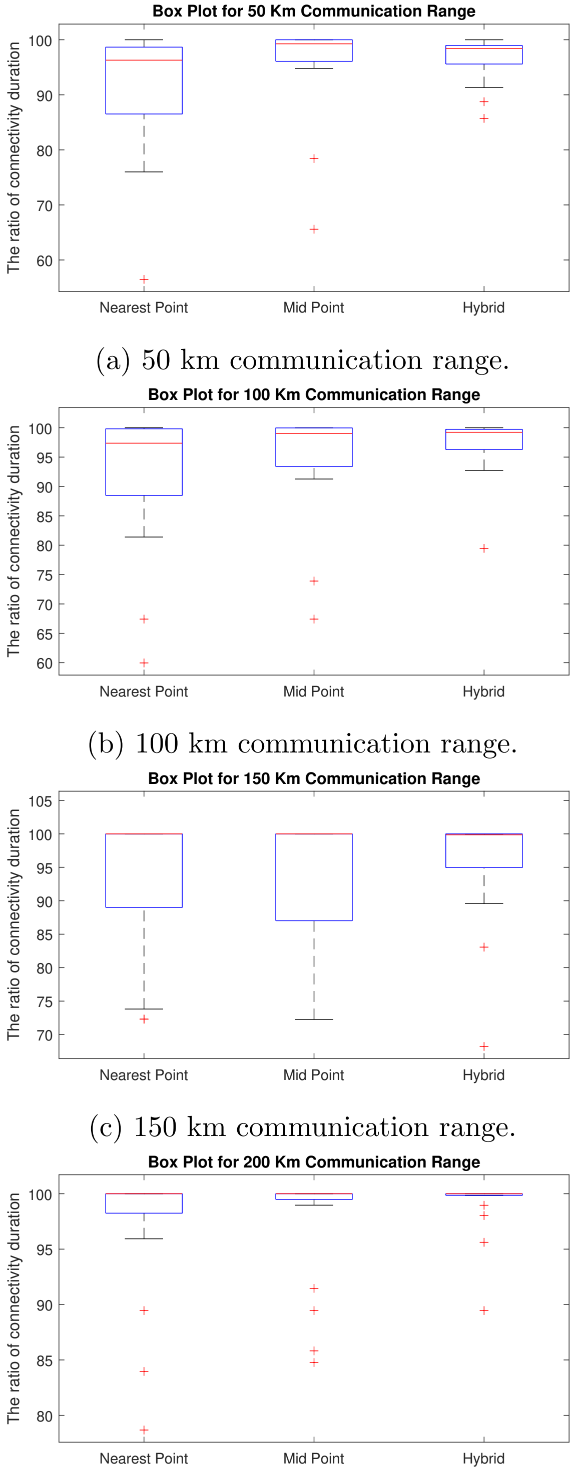


(a) Nearest Point Algorithm.

(b) Mid Point Algorithm.

(c) Hybrid Algorithm.

Figure 5.27: The ratio of average connected duration and the maximum possible duration that the topology remains connected for single-hop communication.



(a) 50 km communication range.

(b) 100 km communication range.

(c) 150 km communication range.

(d) 200 km communication range.

Figure 5.28: Box plot display of the each algorithm with the ratio of average connection duration

Chapter 6

Conclusions and Future Works

In this thesis, we studied the area coverage problem, cooperative target tracking and trajectory planning for a relay UAV with multi-hop communications. For the area coverage problem, our objective is to observe an environment with cooperative UAVs. We discretize the problem using gridization, and we formed a linear integer programming model. For a constant communication range, we compared the total covered area for different coverage radii. The results showed that there is an optimal range for coverage radius to utilize the UAVs.

For the cooperative target tracking, our task is to cooperatively monitor the mobile targets through connectivity limitations. Using gridization, we discretize the problem and give a linear programming model to solve it. The objective of the linear programming model is maximizing the number of targets monitored. Number of targets are generally bigger than number of UAVs, hence a limited number of targets can be tracked in a given time. Also, the restriction on number of hops each UAV can communicate decreases the number of targets tracked. The performance improves with increasing the coverage radius, but the algorithm works slower.

We studied motion planning for a relay UAV to create a connected network topology. Connected network topology means each UAV in the network has a way of communicating with each other. We have two means of communication, single-hop and multi-hop. In single-hop communication, there is only one direct link from tracking UAVs to relay UAV. This means a reliable and low delay communication is preferred in the surveillance task. In multi-hop communication, relay UAV does not have to be directly linked to the other UAVs. This increases the amount of time the UAVs stay connected. For single-hop and multi-hop communication we have an optimization problem defined. The objective is to maximize the amount of time the topology is connected. We compare the single-hop communication with center of mass, where the relay UAV trajectory is chosen as the center point of the three targets at time t. We treat its results as baseline to compare with our optimization problem results. There is 13% improvement for the solution with no time horizon and 16% improvement for the solution with time horizon. For the multi-hop communication, we compare three algorithms and the trajectories planned for relay UAV by the proposed algorithms can remain connected more than 90% of the maximum possible duration of having a connected topology. The best working algorithm is the hybrid algorithm achieving more than 95% of the total connected time.

In future works, heterogeneous UAV systems with different coverage and communication ranges can be used for the coverage and the target tracking problems to further improve the quality surveillance. Also, multiple relay UAVs can be implemented in order to improve the connectivity.

Bibliography

[1] Howie Choset. “Coverage for robotics–a survey of recent results”. In: *Annals of mathematics and artificial intelligence* 31.1-4 (2001), pp. 113–126.

[2] Jacopo Banfi et al. “Fair multi-target tracking in cooperative multi-robot systems”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 5411–5418.

[3] A. Khan, B. Rinner, and A. Cavallaro. “Cooperative Robots to Observe Moving Targets: Review”. In: *IEEE Transactions on Cybernetics* 48.1 (2018), pp. 187–198. DOI: `10.1109/TCYB.2016.2628161`.

[4] Tomonari Furukawa et al. “Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets”. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE. 2006, pp. 2521–2526.

[5] Colin Lecher. “How will drones change sports”. In: *Popular Science* (2014).

[6] Airam Rodrıguez et al. “The eye in the sky: combined use of unmanned aerial systems and GPS data loggers for ecological research and conservation of small birds”. In: *PLoS One* 7.12 (2012), e50336.

[7] Natallia Katenka, Elizaveta Levina, and George Michailidis. “Tracking multiple targets using binary decisions from wireless sensor networks”. In: *Journal of the American Statistical Association* 108.502 (2013), pp. 398–410.

[8] Ryan R Pitre, X Rong Li, and R Delbalzo. “UAV route planning for joint search and track missions—An information-value approach”. In: *IEEE Transactions on Aerospace and Electronic Systems* 48.3 (2012), pp. 2551–2565.

[9] Lynne E Parker and Brad A Emmons. "Cooperative multi-robot observation of multiple moving targets". In: *Proceedings of International Conference on Robotics and Automation.* Vol. 3. IEEE. 1997, pp. 2082–2089.

[10] Thierry Siméon, Stéphane Leroy, and J-P Lauumond. "Path coordination for multiple mobile robots: A resolution-complete algorithm". In: *IEEE Transactions on Robotics and Automation* 18.1 (2002), pp. 42–49.

[11] C. Luo et al. "UAV Position Estimation and Collision Avoidance Using the Extended Kalman Filter". In: *IEEE Transactions on Vehicular Technology* 62.6 (2013), pp. 2749–2762. DOI: 10.1109/TVT.2013.2243480.

[12] Mohammad Islam, Fariha Jaigirdar, and Mohammad Manzurul Islam. "Maximizing Network Interruption in Wireless Sensor Network: An Intruder's Perspective". In: *International Journal of Computer Networks and Communications* 7 (Mar. 2015). DOI: 10.5121/ijcnc.2015.7209.

[13] Chi-Fu Huang and Yu-Chee Tseng. "The coverage problem in a wireless sensor network". In: *Mobile networks and Applications* 10.4 (2005), pp. 519–528.

[14] Xiaorui Wang et al. "Integrated coverage and connectivity configuration in wireless sensor networks". In: *Proceedings of the 1st international conference on Embedded networked sensor systems.* 2003, pp. 28–39.

[15] Ryan R Pitre, X Rong Li, and R Delbalzo. "UAV route planning for joint search and track missions—An information-value approach". In: *IEEE Transactions on Aerospace and Electronic Systems* 48.3 (2012), pp. 2551–2565.

[16] Asif Khan, Bernhard Rinner, and Andrea Cavallaro. "Multiscale observation of multiple moving targets using micro aerial vehicles". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2015, pp. 4642–4649.

[17] Kyle Hollins Wray and Benjamin B Thompson. "An application of multiagent learning in highly dynamic environments". In: *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence.* Citeseer. 2014.

[18] Lynne E Parker. "Distributed algorithms for multi-robot observation of multiple moving targets". In: *Autonomous robots* 12.3 (2002), pp. 231–255.

[19] Daniel J Pack et al. "Cooperative control of UAVs for localization of intermittently emitting mobile targets". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.4 (2009), pp. 959–970.

[20] Andreas Kolling and Stefano Carpin. "Multirobot cooperation for surveillance of multiple moving targets-a new behavioral approach". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE. 2006, pp. 1311–1316.

[21] Timothy H Chung et al. "On a decentralized active sensing strategy using mobile sensor platforms in a network". In: *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601).* Vol. 2. IEEE. 2004, pp. 1914–1919.

[22] Jonathan P How et al. "Increasing autonomy of UAVs". In: *IEEE Robotics & Automation Magazine* 16.2 (2009), pp. 43–51.

[23] Mark P Kolba, Waymond R Scott, and Leslie M Collins. "A framework for information-based sensor management for the detection of static targets". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41.1 (2010), pp. 105–120.

[24] Hung Manh La and Weihua Sheng. "Dynamic target tracking and observing in a mobile sensor network". In: *Robotics and Autonomous Systems* 60.7 (2012), pp. 996–1009.

[25] Boyoon Jung and Gaurav S Sukhatme. "A region-based approach for cooperative multi-target tracking in a structured environment". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Vol. 3. IEEE. 2002, pp. 2764–2769.

[26] Yinfei Fu, Qing Ling, and Zhi Tian. "Distributed sensor allocation for multi-target tracking in wireless sensor networks". In: *IEEE Transactions on Aerospace and Electronic Systems* 48.4 (2012), pp. 3538–3553.

[27] P. Ladosz, H. Oh, and W. Chen. "Prediction of air-to-ground communication strength for relay UAV trajectory planner in urban environments". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 6831–6836. DOI: `10.1109/IROS.2017.8206603`.

[28] Pawel Ladosz, Hyondong Oh, and Wen-Hua Chen. "Trajectory planning for communication relay unmanned aerial vehicles in urban dynamic environments". In: *Journal of Intelligent & Robotic Systems* 89.1-2 (2018), pp. 7–25.

[29] Seungkeun Kim et al. "Coordinated trajectory planning for efficient communication relay using multiple UAVs". In: *Control Engineering Practice* 29 (2014), pp. 42–49. ISSN: 0967-0661. DOI: `https://doi.org/10.1016/j.conengprac.2014.04.003`. URL: `http://www.sciencedirect.com/science/article/pii/S0967066114001282`.

[30] Dianxiong Liu et al. "Self-organizing relay selection in UAV communication networks: A matching game perspective". In: *IEEE Wireless Communications* 26.6 (2019), pp. 102–110.

[31] Youngjoo Kim and Hyochoong Bang. "Introduction to Kalman filter and its applications". In: *Introduction and Implementations of the Kalman Filter*. IntechOpen, 2018.

[32] C-L Hwang and Abu Syed Md Masud. *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Vol. 164. Springer Science & Business Media, 2012.

[33] E. Hyytiä, P. Lassila, and J. Virtamo. "Spatial Node Distribution of the Random Waypoint Mobility Model with Applications". In: *IEEE Transactions on Mobile Computing* 5.6 (June 2006), pp. 680–694. URL: `http://www.netlab.hut.fi/tutkimus/naps/publ/rwp_with_applications.pdf`.

[34] Lorenz T Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.