FORMULATION OF CONTROL STRATEGIES FOR REQUIREMENT DEFINITION OF MULTI-AGENT SURVEILLANCE SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Derya Aksaray

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2014

FORMULATION OF CONTROL STRATEGIES FOR REQUIREMENT DEFINITION OF MULTI-AGENT SURVEILLANCE SYSTEMS

Approved by:

Prof. Dimitri N. Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Eric Feron
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Jeff S. Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Daniel P. Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kelly Griendling
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: *August 1, 2014*

To my mother,

Hatice Aksaray,

whose encouragement has been my strongest motivation.

ACKNOWLEDGEMENTS

It is my great pleasure to thank all the people who directly and indirectly played significant roles in the successful completion of my PhD. First of all, I would like to express my deepest sincere gratitude to my advisor, Prof. Dimitri Mavris, fondly known as *Doc*. Over the last six years, Doc has provided me with endless support and numerous opportunities to develop myself academically, professionally, and personally. His wisdom and insight have helped me to grow as a researcher and an engineer. I met him for the first time when he was visiting METU, Turkey in 2007; and that meeting had a big impact on the direction I have taken for my career. I am very grateful for all the opportunities given by him.

I also feel very privileged to have worked with Prof. Eric Feron, who has shared his invaluable knowledge and experience with me. Our discussions have had a deep impact on the rigor of my research, and his continuous support and profound belief in my abilities have greatly encouraged me to continue on my path. I would also like to thank Prof. Jeff Shamma for his invaluable guidance and support. His insightful comments and ingenious suggestions have greatly broadened my view of control theory.

I am very thankful to Dr. Kelly Griendling for her continuous support and helpful feedback throughout the development of this work. I very much appreciate her warm encouragement, thoughtful guidance, and critical comments. Also, I would like to thank Prof. Daniel Schrage for agreeing to be on my committee and providing me constructive criticism to improve my thesis.

My time at Georgia Tech was made enjoyable in large part due to the many friends and groups that became a part of my life. First of all, I am very happy to be a member of the ASDL family. I would like to acknowledge all my colleagues in ASDL including, but not limited to, Burak Bağdatlı, Mike Miller, Mike Ellis, Michael Balchanos, Bryan

Boling, Daniel Garmendia, Jonathan Sands, Gökçin Çınar, Elena Garcia, Simon Briceno, Olivia Pinon, and Stephane Dufresne for their friendship and valuable interaction. I also would like to recognize the numerous administrative help I received from Loretta Carroll throughout the last six years. Moreover, thanks should also go to all my friends in Atlanta, specifically the Turkish community in Georgia Tech, for their enjoyable company.

A special thanks goes to my fiancee, Yasin, for his constant love and unwavering support; and for our fruitful discussions about this thesis. I am extremely grateful to have him as a true friend and a great partner, who never lets me down and supports me continuously. Finally, I would like to express my deepest gratitude to my mother and my father for their eternal love, patience, understanding, and encouragement; my brother, my sister-in-law, and my nephew for their endless love and support. Although we have been very far from each other, I have always felt my family's steadfast encouragement and support, which have strongly motivated me to continue on my path. I could not have succeeded without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Over the last decade, advances in networking and computing technologies, along with new manufacturing techniques, have enabled a new paradigm shift towards multi-agent systems in engineering applications. This shift has facilitated a significant interest in the design and the control of such systems. Examples of multi-agent systems include, but are not limited to, a swarm of mobile robots, wireless sensor networks, satellite constellations, or a group of unmanned aerial vehicles (UAVs).

In a multi-agent system (MAS), the overall performance is greatly influenced by both the design and the control of the agents. The physical design determines the agent capabilities, and the control strategies drive the agents to pursue their objectives using the available capabilities. For example, in a multi-agent surveillance mission, the endurance of an agent is the amount of time it can operate in the mission area without refueling. On the other hand, an energy-aware control strategy enables the agents to efficiently use their limited fuel when monitoring the surveillance area. In this respect, how frequently an agent leaves the mission area for refueling (i.e. causing a degradation in the situational awareness) depends on both the endurance of the agent and the energy efficiency of the control strategy.

The objective of this thesis is to incorporate control strategies in the early conceptual design of an MAS. As such, this thesis proposes an additional component introduced to a generic design methodology of MAS, and it mainly explores the interdependency between the design variables of the agents and the control strategies used by the agents. The proposed methodology consists of two modules. In the control module, a set of candidate control strategies is generated based on the mission specifications. In the design module, the influential design variables are identified for each candidate strategy through a design

space exploration. Accordingly, the output of the proposed methodology, i.e. the interdependency between the design variables and the control strategies, can be utilized in the later design stages to optimize the overall system through some higher fidelity analyses.

In this thesis, the proposed methodology is applied to a persistent multi-UAV surveillance problem. The main objective of this problem is to increase the situational awareness of a base that receives some instantaneous monitoring information from a group of UAVs. Each UAV has a limited energy capacity and a limited communication range. Accordingly, the connectivity of the communication network becomes essential for the information flow from the UAVs to the base. However, in long-run missions, the UAVs need to return to the base for refueling/recharging with certain frequencies depending on their endurance. Whenever a UAV leaves the surveillance area, the remaining UAVs may need relocation to mitigate the impact of its absence. Accordingly, the proposed methodology is applied to this problem as follows: In the control module of the proposed methodology, a set of energy-aware control strategies are developed for efficient multi-UAV surveillance operations. To this end, this thesis first proposes a decentralized strategy to recover the connectivity of the communication network, which maintains the instantaneous information flow from the UAVs to the base. Second, it presents two return policies for UAVs to achieve energy-aware persistent surveillance. In the design module of the proposed methodology, a design space exploration is performed to investigate the overall performance by varying a set of design variables and the candidate control strategies developed in the control module. Overall, it is shown that a control strategy used by an MAS affects the influence of the design variables (i.e. physical characteristics) on the mission performance. Furthermore, the proposed methodology identifies the preferable pairs of design variables and control strategies through low fidelity analysis in the early design stages.

CHAPTER I

INTRODUCTION

Over the last decade, advances in networking and computing technologies, along with new manufacturing techniques, have enabled a new paradigm shift towards multi-agent systems in engineering applications. This shift has facilitated a significant interest in the design and the control of such systems. Accordingly, the focus of this thesis is on the design and the control of multi-agent surveillance systems for efficient operations.

This chapter starts with introducing the terminology used throughout the thesis. Then, the next section depicts the motivation for multi-agent surveillance systems by giving examples from the literature. After that, the research questions of this thesis are introduced, and finally the last section presents the organization of this thesis.

1.1 Terminology

A multi-agent system (MAS) consists of a set of individuals called *agents*, each of which can be a robot, an aircraft, or a living organism, to name a few. Examples of MAS include, but are not limited to, ecosystems (e.g. [20, 31]), power grids (e.g. [44, 97]), air traffic management (e.g. [125, 132]), a swarm of mobile robots (e.g. [46, 135]), wireless sensor networks (e.g. [4, 6]), satellite constellations (e.g. [60, 114]), or a group of unmanned air vehicles (UAVs) (e.g. [18, 119]).

In an MAS, each agent works individually and interacts with only a subset of agents. Interaction among agents typically refers to *communication* between them. Accordingly, agents can achieve collective *objectives* by operating individually and communicating with each other. For example, some collective objectives in an MAS can be formation, coverage, or patrolling (e.g. [36, 54, 82]).

The research on MAS is an emerging area, and some major disciplines in this field are *analysis*, *design*, and *control*. In analysis, the main goal is to understand the nature of complex systems (e.g. [39, 78]). In design, it is aimed to provide principles for designing complex system-of-systems (e.g. [64, 96]). In control, the main goal is to develop distributed control algorithms for achieving collaborative objectives such as formation, exploration, or task allocation (e.g. [34, 59]). This thesis mainly focuses on the design and the control of multi-agent systems.

In an MAS, each agent can be represented by a set of design variables. A *design variable* of an agent in this thesis refers to a quantitative parameter that represents a system capability. For an agent representing a UAV, some examples of design variables are speed, endurance, or communication range. Thus, any variation in the design variables will potentially affect the mission performance. As such, a strategic *decision-making* for the set of design variables is essential to result in efficient agent operations. In this thesis, the design of an MAS pertains to *the identification of the influential design variables* on the mission performance.

In an MAS, an agent takes *actions* to pursue its objectives, and an agent can control its action selection by following some *rules* or *policies*. A set of rules denotes to a *control strategy*, which guides an agent to select the best set of actions to result in a desired performance. Assume that an agent represents a UAV patrolling a border (i.e. a line from point A to point B). Some examples of agent actions can be "turn right", "turn left", or "fly until a point". For a particular agent, an example rule can be "for any initial coordinate, turn right and fly until reaching point B". Furthermore, an example of a control strategy for an agent can be the set of following rules:

- "for any initial coordinate, turn right and fly until reaching point B",
- "if point B is reached, turn left and fly until point A",
- "if point A is reached, turn right and fly until point B".

Consequently, an agent utilizing the aforementioned control strategy travels between the points A and B continuously. In this thesis, the control of an MAS refers to *designing efficient control strategies for agents* to result in desired mission performance. Hence, *the design of an MAS* and *the control of an MAS* throughout this document will imply the following:

- **Design of an MAS:** Identifying the influential design variables of agents on the mission outcome.

- **Control of an MAS:** Developing effective control strategies for agents to achieve a desired mission objective.

1.2 Motivation

In complex missions, a multi-agent system offers some distinct advantages over a single agent system such as parallelism, robustness, scalability, simpler programming, geographic distribution, and cost effectiveness [120]. In particular, the advantages of MAS such as parallelism, robustness, and geographic distribution make it a good candidate for *reconnaissance and surveillance missions*, whose goal is to obtain information about the activities or the resources in a particular area by using various detection techniques.

In reconnaissance and surveillance missions, unmanned vehicles are also in high demand due to the presence of dangerous, dirty, or dull operations. Here, "dangerous" refers to threats posed by suppression of enemy defense, "dirty" corresponds to areas that may be contaminated, and "dull" implies prolonged surveillance or sentry duty (e.g. [40]). As a result, a group of unmanned vehicles is a promising multi-agent system for reconnaissance and surveillance operations.

The physical design of agents plays an important role in achieving a desired surveillance performance. For instance, consider a group of UAVs monitoring some target points over a field. Some influential physical design variables on the situational awareness are the endurance and the velocity of each UAV. The endurance is the overall time a UAV can remain at the surveillance area without refueling or recharging. Accordingly, it limits the longest time the situational awareness can be maintained. The velocity of each UAV determines the time required to travel from one point to another on the surveillance area. On top of these, the number of UAVs is also a critical design variable for the overall MAS, which determines the maximum area that can be monitored during a mission. In other words, it bounds the maximum amount of information that can be gathered from monitoring.

Optimizing such design variables helps to obtain an efficient multi-agent surveillance system; however, only designing for the physical aspects of the systems is not sufficient to ensure a desired performance. The control strategies utilized by the agents also play an important role in the overall performance. For example, if a UAV leaves the surveillance area for refueling, an efficient control strategy is expected to relocate the remaining UAVs to compensate for the absence of the refueling UAV. Accordingly, utilizing such a reactive strategy may relax the design variables of UAVs (e.g. high velocity short endurance vehicles can be preferable over long endurance vehicles if a reactive control strategy is utilized). Hence, the control strategies must be considered as an integral part of the design process.

1.3 Multi-Agent Surveillance

Surveillance is the monitoring of activities and changing information on an area to provide situational awareness. For large surveillance areas, a promising multi-agent system candidate is a group of unmanned aerial vehicles that can fly over the regions of interest. In a multi-UAV system, one critical aspect pertains to the design of UAVs, which determines the vehicle capabilities by optimizing the parameters regarding the weight, the propulsion system performance, or the aerodynamics characteristics. Accordingly, the overall system capabilities can determine the answers of the following questions:

- *How long can a multi-UAV system operate in a mission area?*

- *How fast can a multi-UAV system travel over the surveillance area?*

- *How well can a multi-UAV system acquire information about the regions of interest?*

- *Is it possible for a UAV to share the monitoring information with the other UAVs?*

It is desirable to design a multi-UAV system that can achieve all desired capabilities. However, system design typically involves many compromises that result in some decision-making regarding the system specifications. For instance, increasing the fuel capacity of a UAV results in a longer duration of time it can remain above the surveillance area. Nonetheless, it also causes an increase in the total weight, which potentially requires a more powerful propulsion system. Accordingly, one major goal becomes to make a set of strategic decisions that maximize the system capabilities in accordance with the design compromises. In order to support a strategic decision-making, it is crucial to understand the *design trade-offs* in a multi-UAV system.

In addition to the influential design variables of UAVs on the mission performance, another critical aspect for multi-UAV surveillance pertains to the control strategies utilized by the UAVs. As depicted before, a control strategy results in the selection of the best set of actions that will drive UAVs to move efficiently above the surveillance area. Consequently, efficiently moving UAVs results in trajectories that increase situational awareness by visiting the regions of interest with the desired frequency. Some motivating questions to develop control strategies are as follows:

- *How can the trajectories of UAVs be designed to maximize situational awareness?*

- *How can UAVs perform autonomous operations in the surveillance area?*

This section continues to elaborate the essential design variables and the main aspects on the development of the control strategies for efficient multi-agent surveillance.

1.3.1 Essential Design Variables for Efficient Multi-Agent Surveillance

A surveillance mission usually requires monitoring for long periods of time. For instance, an unfortunate event happened on March 8, 2014, when the Malaysian Airlines Flight 370 disappeared in the southern Indian ocean, and a search of 4,638,670 square kilometers of ocean was conducted by 29 civilian and military aircraft and 14 ships [88]. Even in the third month of the search, no wreckage information has been identified. A major issue encountered in such long endurance missions pertains to the limited fuel/energy capacity of the vehicles, which causes frequent returns to the base for refueling/recharging. One way to reduce the frequency of returns to the base is enhancing vehicle endurance, i.e the maximum length of time an aircraft can continuously fly. The equations for the vehicle endurance differ for jet-driven (e.g. [21]), propeller-driven (e.g. [22]), and battery-powered aircraft (e.g. [124]). Moreover, different equations exist for different flight conditions for a given class of aircraft. In the following two sections, some commonly used endurance equations in UAV performance calculations will be discussed.

In addition to the endurance, some other essential design variables are the velocity and the communication capability of UAVs. In long-run missions, leaving surveillance area for refueling (or recharging) is essential for finite fuel (or energy) capacity vehicles. Thus, having high velocity is crucial to minimize the time required for traveling between the base and the surveillance area. However, it is not possible to increase velocity without compromising from other vehicle capabilities. Moreover, the communication capability of a UAV determines the way it shares monitoring information with the base. Due to the limited communication ranges of UAVs, it may not be always feasible to share instantaneous information with the base from any point on the surveillance area. Hence, the aspects regarding the velocity and the communication capability of a UAV are also discussed along this section.

1.3.1.1 Endurance of jet aircraft

A commonly used endurance equation for jet-driven aircraft with fixed altitude, fixed throttle setting, and fixed angle of attack is the *Breguet endurance equation*,

$$E = \frac{1}{TSFC}\Big(\frac{L}{D}\Big) ln\Big(\frac{W_0}{W_1}\Big), \tag{1}$$

where $TSFC$ denotes the thrust specific fuel consumption (i.e. propulsion system performance), L/D is the lift-to-drag ratio (i.e. aerodynamic performance), W_0 is the initial weight and W_1 is the final weight of the aircraft. As such, any change in the propulsion system, in the geometry, or in the fuel capacity greatly affect the endurance of a system. One intuitive solution to obtain longer endurance is increasing the overall fuel capacity of an aircraft. However, an increase in the weight will require a redesign of the vehicle.

Another solution to increase endurance pertains to the fuel management of a vehicle such that the operations requiring lower fuel consumption can be preferred during a mission. For instance, an aircraft has two major flight phases, namely cruise and loiter phases. It can be theoretically shown that an aircraft loiters optimally at its speed for the maximum L/D, whereas it cruises optimally at a higher speed [102]. Thus, the change in speed from cruise to loiter changes the aircraft lift-to-drag ratio. An approximation method is proposed in [103], where the lift-to-drag ratios for cruise and loiter are related as $(L/D)_{cruise} = 0.866(L/D)_{loiter}$. As such, the cruise endurance of a particular aircraft (E_{cruise}) becomes less than its loiter endurance (E_{loiter}) due to the fact that $(L/D)_{cruise} < (L/D)_{loiter}$. Moreover, based on the proposed L/D relationship in [103], the approximation results for range and loiter endurance are compared with the real data as illustrated in Figure 1.

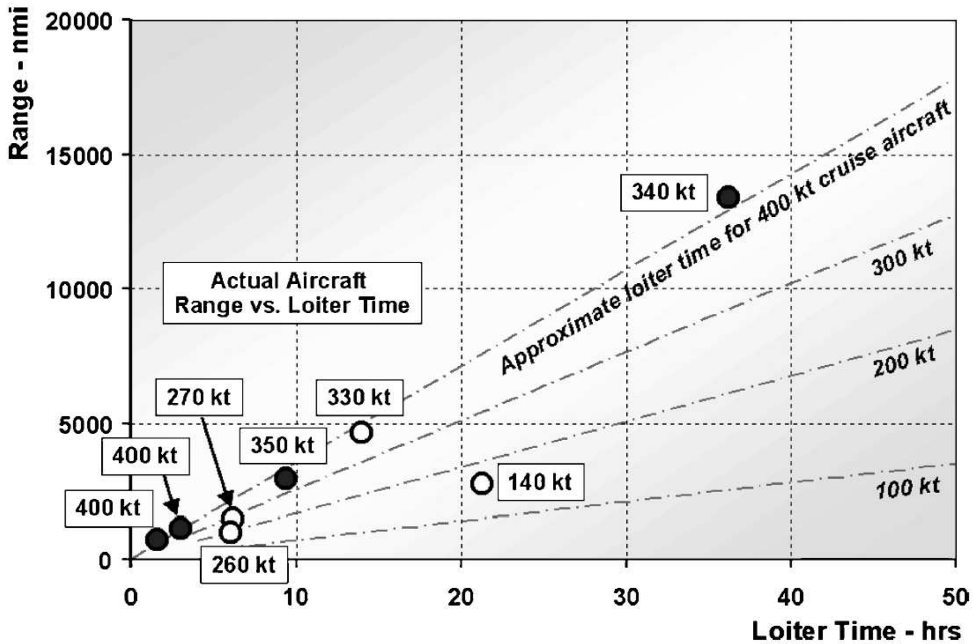


Figure 1: Comparing the approximation method in [103] for cruise range and loiter endurance with the operational data for several aircraft.

In Figure 1, the range of an aircraft is related with its loiter endurance for various speeds shown as dashed lines. Accordingly, any point on a dashed line represents an aircraft with a corresponding cruise speed. For any point on this plot, the y-coordinate indicates the maximum possible range of the aircraft whereas the x-coordinate shows the maximum possible loiter endurance of the aircraft. Most of the points on the dashed line of Figure 1 correspond to a shorter cruise endurance than the loiter endurance. For instance, consider the point corresponding to 30 hours loiter time on the 200 kt dashed line. This point in Figure 1 represents an aircraft having a speed of 200 kt, a loiter endurance of 30 hours, and the cruise range of 5000 nmi. Based on this data, the cruise endurance can be approximated as 25 hours (i.e. $Range/Speed$), which is less than the loiter endurance. Such a result suggests that the difference between the cruise and loiter endurance is due to the fact that the fuel consumption in cruise is slightly higher than the fuel consumption at loiter (i.e. the aircraft is subjected to more drag in cruise, which can be overcome by generating more thrust). Supporting this conclusion, the actual data of the Hunter RQ-5A UAV indicates that the fuel consumption rates of this UAV for cruise and loiter are 4.2 gal/hr and 3.5 gal/hr, respectively [58]. Without loss of generality, loiter can be interpreted as a more efficient flight phase than cruise.

1.3.1.2 Endurance of battery-powered aircraft

Jet-driven UAVs such as Global Hawk and Predator are generally large, heavy, and expensive (e.g. [56]). Recently, there is a huge interest in designing smaller and cheaper UAVs powered by batteries. A battery-powered aircraft mainly differs from a jet aircraft by maintaining a constant weight during its operation. The overall endurance equation of such aircraft at steady level flight is derived in [124] as follows:

$$E = Rt^{1-n}\left[\frac{\mu_{tot}V \times C}{\frac{1}{2}\rho U^3 SC_{D0} + (2W^2k/\rho US)}\right]^n, \quad (2)$$

where E is the endurance in hours, Rt is the battery hour rating (in hours), n is a discharge parameter dependent on the battery type, μ_{tot} denotes the total efficiency of the propulsion system, V is volts, and C is the battery capacity in ampere hours. Moreover, in (2), ρ is the air density, U is the flight velocity, S is the reference area, c_{D0} is the zero lift drag coefficient, W is the weight of the aircraft, and k denotes an aerodynamic parameter related to the drag polar.

Based on (2), the endurance of an aircraft can be increased by mainly three ways. Minimizing the losses in the propulsion system improves the total efficiency (i.e. μ_{tot}) of the system, which increases the numerator of (2). On the other hand, the denominator of (2) can be decreased by slower flight velocity (due to the third order of U) or lighter weight (due to the second order of W). However, slow flight velocity may not be practical for UAV operations due to causing long amount of time to travel from one point to another. On the other hand, light weight limits the payload capacity that impacts the sensors and cameras that can be carried by the aircraft.

1.3.1.3 Velocity limitations in a surveillance mission

High velocity vehicles are desired in most missions (including surveillance) due to providing quick transition from one coordinate to another. However, high values of velocity bring other considerations into the other vehicle capabilities. For instance, if a UAV increases its speed, it is subjected to more drag, which causes a reduction in the L/D ratio. Thus, the endurance of a UAV gets shorter as it operates at higher speeds. Furthermore, the capabilities of the onboard sensors also result in another limitation in the maximum velocity. The data acquisition of the onboard sensors may not necessarily be the same at any speed. Generally, better quality information (e.g. image or video) can be obtained by UAVs flying at slower speeds. Consequently, slow velocity is desirable in UAV surveillance as long as it does not hinder the quick response capability during monitoring.

1.3.1.4 Communication capability of an aircraft

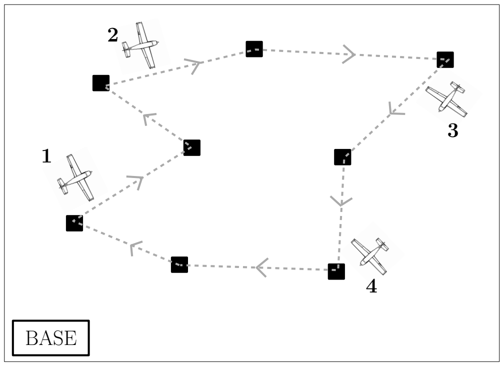
As a UAV gathers some information, it needs a way to share this information with the base (or the ground station). One way for a UAV to share the monitoring information is via recording the data in on-board units. Accordingly, the monitoring information becomes reachable whenever a UAV returns to the base. Note that the surveillance data can be time-critical in some cases, and sharing the information instantly becomes more desirable in such cases. To this end, a UAV needs to communicate with the base during its operation. Communication among UAVs and the base can be via single- or multi-hop. Single-hop communication implies that a UAV can communicate directly with the base. In other words, its communication range should be longer than its distance to the base. Nonetheless, if the UAV is far from the base, then the single-hop communication may not be always feasible and reliable due to adverse weather effects, jamming attacks, or service delays [110]. Alternatively, a UAV can communicate with the base through multi-hop communication, which implies that the communication range of the UAV is shorter than its distance to the base. In this case, information can propagate through a sequence of messages forwarded among the UAVs and the base.

1.3.2 Control Strategies for Efficient Multi-Agent Surveillance

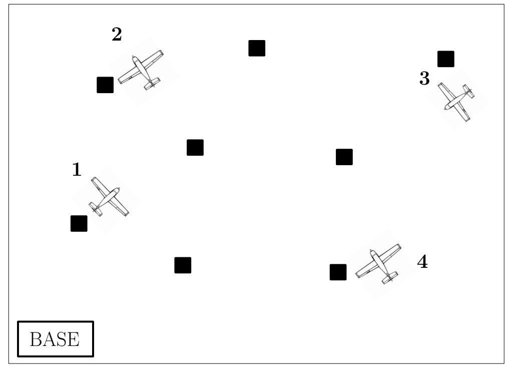
One way to achieve surveillance is to patrol over a field. Patrolling is defined as "the act of walking or traveling around an area, at regular intervals, in order to protect or supervise it" [3]. A major objective in patrolling problems is to schedule the trajectories of agents to optimize a certain performance criteria. One of the most frequently used criteria is minimizing the age, i.e. the time lag between any two visits of the same region [76].

In general, three major strategies exist in the literature for designing the agent movements during patrolling. *Cyclic-based strategies* (e.g. [35,48,57]) rely on creating a closed-route through the viewpoints of the desired region, and agents travel repeatedly such route at maximum speed. *Randomized strategies* (e.g. [48, 72, 76]) are based on random movements of agents in a desired region, and the nature of randomization results in almost uniform distribution of visits to each viewpoint in the desired region within a given sufficient time. *Partition-based strategies* (e.g. [35, 48, 94]) rely on space decomposition such that a desired region is divided into sub-areas, and each agent is assigned to one of the partitioned areas.

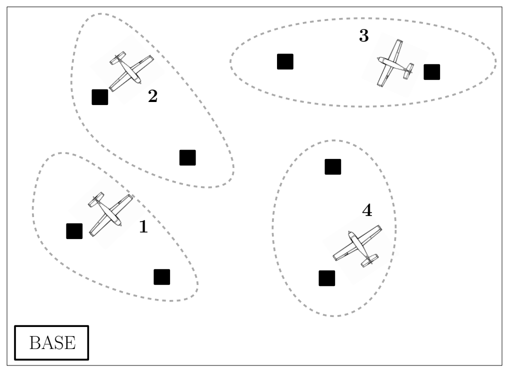
The aforementioned strategies exhibit various advantages and disadvantages under different scenario assumptions and objectives. For example, the authors of [35] compare cyclic-based and partition-based strategies. They show that the cyclic-based strategies perform better whenever the ratio of the longest to the shortest distance of any two neighboring viewpoints is small, while the partition-based strategies exhibit better, otherwise. Another survey, [9], offers an empirical comparison between different approaches of patrolling with regards to the idleness criteria, which is defined as the amount of time elapsed since a particular region was visited by an agent. The authors of [9] observed that cycle-based approaches perform better in optimizing idleness.



(a) A cyclic-based strategy.



(b) A randomized strategy.



(c) A partition-based strategy.

Figure 2: Four UAVs employing different patrolling strategies a surveillance area.

For maximizing the coverage during multi-agent patrolling, the authors of [57] address robustness and efficiency issues in a family of algorithms based on spanning-tree coverage of approximate cell decomposition, and they propose an algorithm generating a closed route for the agents to travel on continuously. Similarly, [48] is also based on the idea of Spanning-Tree Patrolling (STP) such that a Hamiltonian cycle is generated, and some robots are placed in equidistant positions to pursue traveling. In this work, the authors show that their proposed algorithm guarantees the maximal uniform frequency such that each point in the target area is visited at the same optimal frequency.

The design of an agent trajectory can also rely on the use of pheromone traces to mark the visited regions. In this manner, (virtual) pheromone traces can act as potential fields to guide each agent towards areas that have not been visited for a long time (e.g. [53]).

Note that when the global representation of an environment is unavailable and there exist severe communication constraints, these techniques can be effective; however, they do not explicitly deal with the optimal patrolling trajectories.

Multi-agent patrolling can also be achieved in a decentralized fashion, where the agents have limited sensing and communication capabilities. For this, the authors of [77] introduce a fully decentralized patrolling algorithm for perimeter patrol by using a behavior-based approach and creating a Finite State Automata for the proper action selection. On the other hand, the authors of [63] study a cooperative surveillance problem and introduce a decentralized solution that can handle the perimeter growth and the insertion/deletion of some agents. Their proposed solution is based on perimeters represented as a line, and they claim that an arbitrary connected perimeter can be reduced to a linear perimeter. Finally, a very recent study, [94], focuses on optimizing the time gap between any two visits to the same region (i.e. age), and the time necessary to inform every agent about an event occurred in the environment (referred to as latency). In this study, the authors also study the computational complexity of the patrolling problem as a function of the environment topology.

1.4 Primary Research Questions

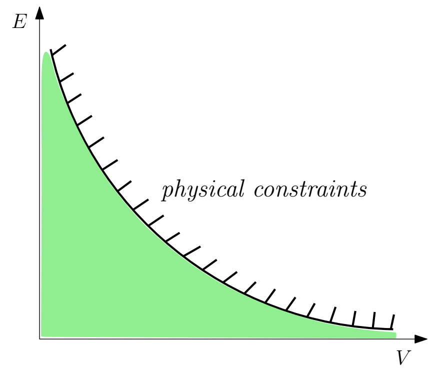
This thesis addresses how to consider control strategies in the early design stages of multi-agent surveillance systems. As discussed, the performance of an MAS is greatly influenced by the physical design variables of the agents and the control strategies utilized by them. For any given system, the physical design variables indicate the capabilities of each agent. On the other hand, the control strategies guide agents to achieve a task with the given capabilities.

In a design space of a vehicle (e.g. UAV), there are typically some physical constraints limiting the feasible space. For instance, consider a two dimensional design space depicting the endurance and the velocity of an aircraft. Due to the inverse relationship between the endurance and the velocity, e.g. (1) or (2), the feasible design space is illustrated in Figure 3a by the filled green region. Note that any point on the feasible space corresponds to a vehicle with a particular pair of an endurance and a velocity. Generally, there might exist some control strategies such that none of the vehicles in the design space accomplish the mission by using one of them. Let CS1 denote such an inefficient control strategy. Then the feasible space disappears as illustrated in Figure 3b due to the mission failure caused by CS1. Similarly, let CS2 denote to a control strategy, which is a more efficient strategy than CS1. Then, some points on the design space may fail and some others can accomplish the mission by using CS2. Accordingly, the design space may reduce as in Figure 3c. Consequently, the first research question of this thesis is as follows:

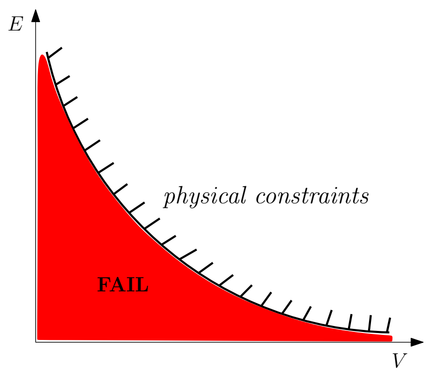
***Research Question* 1.** *Can a control strategy influence the selection of the physical design variables to result in a desired MAS performance?*

In an MAS, the overall performance can be improved by concurrently considering the design variables of the agents and the control strategies used by the agents. Due to the potential impact of the control strategies on the mission accomplishment, it is likely to observe some interdependency between the design variables and the control strategies. However, such interdependency may not be easily identified in the early design stages due to the limited knowledge about the design variables of the agents. Accordingly, the second research question of this thesis is as follows:

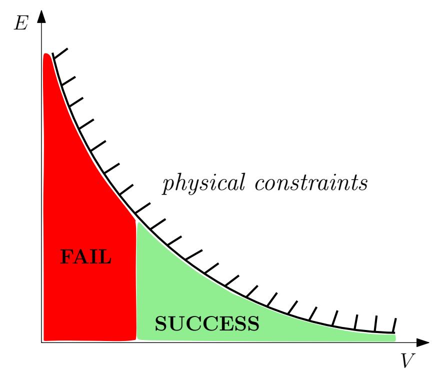
***Research Question* 2.** *How is it possible to explore the interdependency between the control strategies and the design variables of an MAS at early design stages?*



(a) Feasible design space.



(b) Feasible space disappears by using CS1.



(c) Feasible space shrinks by using CS2.

Figure 3: The potential impact of a control strategy on the design space of an agent.

1.5 Thesis Outline

The chapters of this thesis are organized as follows:

Chapter 2 presents a survey of the literature pertaining to the design and the control of multi-agent systems. In the design literature survey, more emphasis is put on the design of aerospace system and system-of-systems. In the control literature survey, first a brief review of centralized and decentralized control is presented. Then, two major topics are discussed, namely the control strategies for multi-agent surveillance and the control strategies for the recovery of network connectivity.

Chapter 3 introduces the main methodology proposed in this thesis. The chapter starts with an overview of a generic design methodology. Then some observations are presented for the control consideration in the generic design methodology. Based on that, the primary hypothesis of this thesis is constructed. Later, the methodology and its design and control modules are detailed throughout the chapter.

Chapter 4 introduces the multi-agent persistent surveillance problem, which is used to apply the proposed methodology. In the considered problem, there exist a group of UAVs, each of which has limited energy and communication capability. The main objective in the problem is to increase situational awareness of a base, which can receive instantaneous monitoring information from the UAVs. Accordingly, this chapter details the scenario assumptions as well as the overall objective of the problem.

Chapter 5 presents a decentralized connectivity maintenance strategy that is contained in the control module of the proposed methodology. The proposed connectivity maintenance scheme, called the *Message Passing Strategy* (MPS), is applicable to any initially connected networked system. The MPS is based on a sequence of replacements, each of which occurs between an agent and one of its immediate neighbors. Accordingly, the replacements always end with the relocation of an agent, whose removal from its previous position does not cause a disconnection. It has been shown that such an agent can be reached by a decision mechanism utilizing only some local information available in agents' immediate neighborhoods. Furthermore, in order to improve the optimality (i.e. minimum number of replacements) of MPS, a derivative strategy is proposed by incorporating the criticality of an agent in its local neighborhood. In this chapter, the proposed strategies are demonstrated through some simulations.

Chapter 6 presents an improved version of the MPS, which is more applicable to persistent surveillance missions. Accordingly, the proposed strategy incorporates an agent constraint that causes the agent not to move from its location other than a removal. Such a constraint represents an agent that directly communicates with the base. In this respect, this chapter introduces a decentralized strategy, named the constrained MPS, that is similar to the MPS by recovering connectivity through a sequence of replacements. The proposed strategy differs from the MPS by resulting in a sequence of replacements without violating the agent constraints. Consequently, the constrained MPS is applicable for multi-agent surveillance systems tham maintain connectivity with the base all the time. In this chapter, some simulations are conducted to compare the resulting network topologies via the MPS and the constrained MPS.

Chapter 7 discusses the importance of energy-aware strategies in persistent surveillance, and it presents some centralized and decentralized strategies (i.e, part of the control module of the proposed methodology). As a centralized solution, an approximate dynamic programming formulation is presented. As a decentralized solution, some locally applicable rules are introduced for agents to be assigned with monitoring locations, to decide when to recharge, and to relocate on the surveillance area. Finally, the depicted approaches are compared with each other through simulations.

Chapter 8 presents a design space exploration for multi-agent surveillance systems, which is contained in the design module of the proposed methodology. Mainly, a Design of Experiments study and Monte Carlo simulations are used to investigate the influence of the design variables and the control strategies on the mission performance. In this chapter, two scenarios are considered to conduct some experiments, and in each scenario an agent represents a notional Raven RQ-11B. In the first scenario, the number of agents is assumed to be equal to the number of monitoring zones, whereas the second scenario considers less agents than the number of zones. Finally, the results of these experiments show that there exist some interdependency between the design variables of the agents and the control strategies used in an MAS, and the proposed methodology can explore such interdependency in the early design stages.

Chapter 9 concludes the thesis by summarizing the contributions and identifying the possible future research.

CHAPTER II

BACKGROUND

In a multi-agent system, the overall performance depends on the design and the control of agents. Therefore, this chapter presents a survey of the literature pertaining to the design methodologies and the control strategies for an MAS in the following four sections. Since this thesis particularly focuses on aerospace applications, the first section of this chapter describes the existing studies in the literature for the design of the aerospace systems. While the general design methodology is discussed for single vehicle design, the state of the art design techniques are introduced for system-of-systems.

The second section of this chapter is on the control strategies for the agents. Mainly, centralized and decentralized strategies are discussed with respect to their advantages and disadvantages. Following that, the third section of this chapter is on the existing control strategies that enable efficient multi-agent surveillance. In addition to the patrolling strategies discussed previously, this section also presents some studies related to the refuel schedules for the agents in long endurance missions.

Finally, the last section of this chapter discusses the existing control strategies for the recovery of network connectivity. Note that after an agent leaves the surveillance area for some reason (e.g. refueling/recharging), there is no guarantee that the communication network will maintain its connectivity. In case of a disconnection, some of the other agents may not send information back to the base although they keep monitoring some regions of interest. As such, the overall situational awareness degrades significantly, if connectivity is not recovered. This section presents a survey of literature from various disciplines, which focus on the connectivity maintenance problem in networked systems.

2.1 *Design of Aerospace Systems*

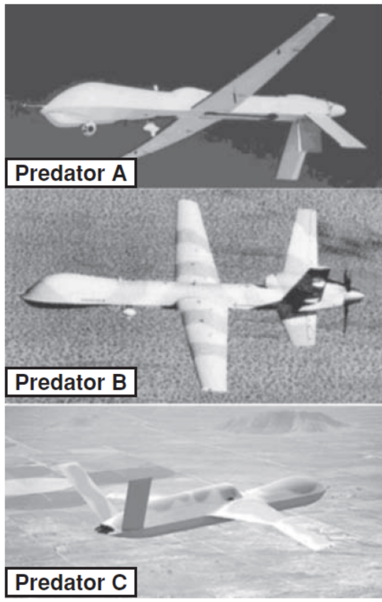
2.1.1 Single Vehicle Design

In aerospace system design, the primary focus has mostly been on individual vehicle designs to improve the performance- and economics-related specifications by using available technologies of the time. Starting in the 1990s, the general availability of Global Positioning System (GPS) and satellite communications have enabled UAVs to operate out to great ranges with positional accuracy. In this manner, the medium- and long-range systems have been developed as medium altitude long endurance (MALE) and high altitude long endurance (HALE) systems. Some examples to such vehicles are the General Atomics Gnat, the Predator MALE, and the Northrop-Grumman Global Hawk HALE unmanned aerial systems. Furthermore, some specifications of the different types of unmanned aerial vehicles are illustrated in Table 1.

Table 1: Specifications of some UAVs that are currently in operation (Reproduced from [95])

Group	UAV Name	Length x Wingspan (m^2)	Maximum Payload (kg)	Range (km)	Altitude (km)	Endurance (hr)
Micro	AirRobot AR100B	1.0×1.0	0.2	0.5-1.4	0.9	0.2-0.5
	Draganflyer X6	0.9×0.9	0.5	0.5	2.4	0.2-0.3
Small	AeroVironment Raven	1.1×1.3	0.2	10.0	4.6	1.3
	AAI Shadow 600	4.8×6.8	41.3	200	4.9	12-14
MALE	TAI Anka	10.1×17.3	441	200	9.1	24
	IAI Heron I	8.5×16.6	550	300	9.1	20-45
HALE	General Atomics Reaper	11.0×20.1	386-1361	5926	15.3	30
	Northrop Grumman Global Hawk	14.5×39.9	1361	22772	18.3	36

As the technology develops and the new operational capabilities are desired, an individual vehicle also evolves through time. Figure 4 illustrates the specifications of the different versions of the Predator (i.e. A, B, and C), which has possibly evolved based on the advances in technology (e.g. the trend going from piston to jet engines). Moreover, elevating the former MALE system to a HALE system as the Predator C was relied on the possible competition with the Global Hawk system [13]. As it is seen, the design specifications of an aerospace system is greatly affected by infusing available technologies and competing with the alternative designs.



Predator	A	B	C
Wing span (m)	14.83	20	20
Length (m)	8.13	10.6	13.1
MTOM (kg)	1,020	4,536	?
Engine Type	Piston	Turbo Prop	Turbo Fan
Engine Power/Thrust	78.3 (kW)	500 (kW)	18 (kN)
Max Speed (km/hr)	217	440	740
Endurance (hr)	>20	32	?
Ceiling (m)	7,920	12,000	20,000
Payload mass (kg)	204	385 int 1360 ext	1,363
Payload Type	EO,IR, SAR, SIGINT	EO & I.R.TV. SAR weapons	As B + internal weapons

Figure 4: Predator evolution [13].

In general, the life-cycle design stages of an aircraft mainly include conceptual design, preliminary design, detail design, production, and operation. Specifically, the conceptual design identifies the general layout of the vehicle through some low fidelity analysis. The preliminary design makes only minor changes to the configuration (generated in the conceptual design) and investigates the performance of the selected concept through high fidelity analysis. In the detail design, all performance computations are completed, and the main focus is on the precise design of any vehicle part for manufacturing.

As an aircraft concept progresses through the design stages, the variation of the design knowledge, the design freedom, and the cost committed vary as in Figure 5. As it is seen, the design freedom (referred to as ease of change) rapidly reduces, while the knowledge about the design slowly increases and the cost commitment is locked at early stages. This implies that the decisions made for the design variables during the early stages are crucial to be able to proceed the program. Therefore, bringing more information to the early stages is desirable in a complex system design.

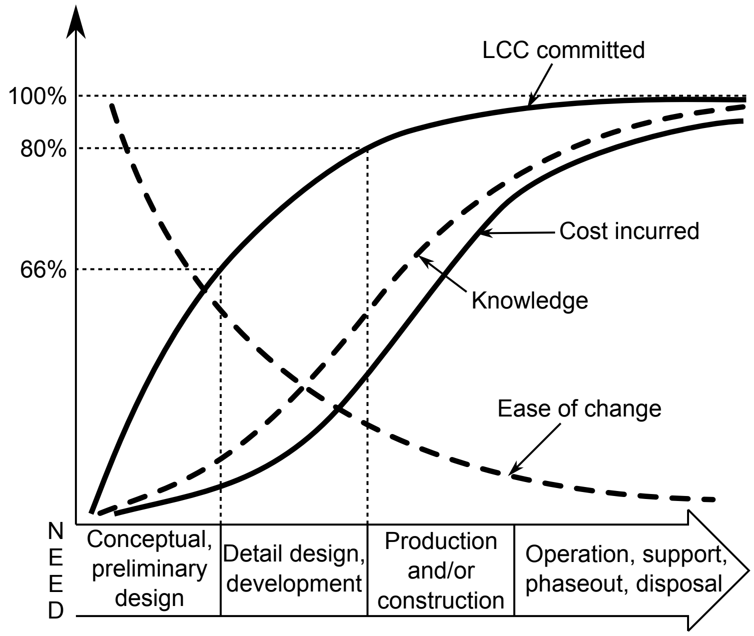


Figure 5: Life-cycle design stages (Reproduced from [51]).

The early design stages refer to the conceptual design of an aircraft. A classical approach for the conceptual design phase corresponds to a sequential process as illustrated in Figure 6 [11]. During this process, first the requirement analysis is performed. In general, expert opinion is one of the most commonly used guidelines to determine the design requirements of a system. To this end, a systems engineering approach, which employs quality engineering tools such as the Quality Function Deployment [104], enables the elicitation and rank ordering of customer needs to understand system requirements. Note that an expert opinion approach is a subjective technique to obtain the set of requirements. After the requirement analysis, an iterative low fidelity analysis is conducted to identify an initial design concept. Based on the selected concept, a deeper investigation is conducted in the following design stages through detailed disciplinary analyses. Accordingly, the results of the high fidelity analyses determine either the termination of the design project or the possibility for manufacturing the concept. Consequently, in the presence of such a sequential design process, obtaining a desired end-product strongly depends on the initial decisions made at the early design stages.

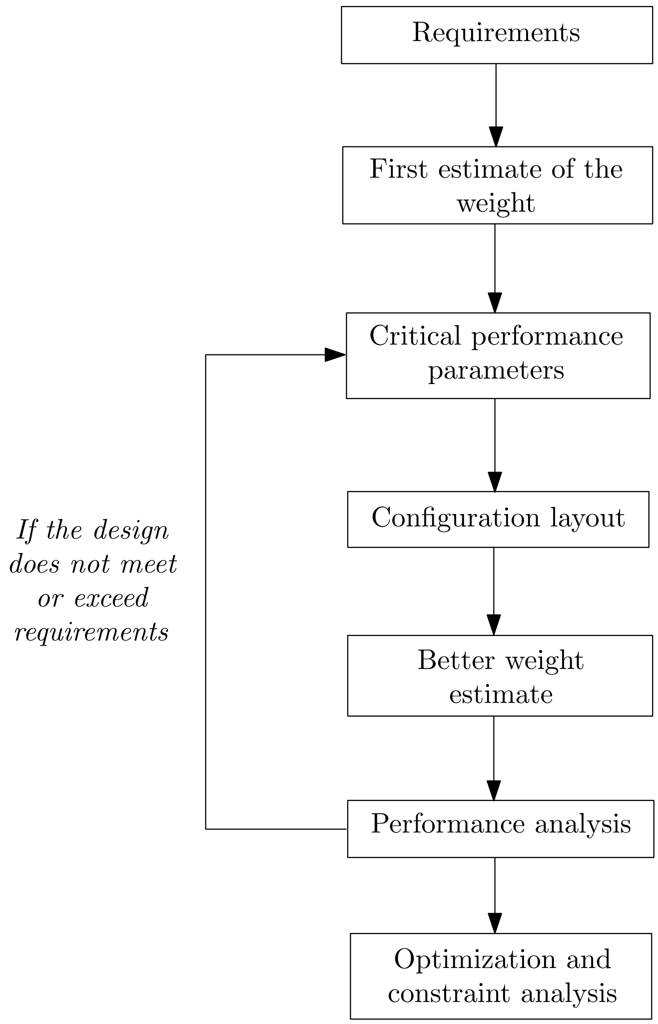


Figure 6: The seven intellectual pivot points for conceptual design (Reproduced from [11]).

The lack of design flexibility at the late design stages and the dependence of the end-product to a subjectively generated requirements have created new research areas in the design literature. In particular, modern design techniques have received a great amount of interest, which can bring more disciplinary information back to the early design phases. Accordingly, experimental design is introduced in various studies that enable design space exploration via statistical methods (e.g. [52,117,118,131]). For instance, the authors of [80] proposed a framework called Unified Trade-Off Environment that analyzes design requirements, design and economic variables, potential technologies simultaneously. Moreover, the authors of [81] introduced a methodology for technology identification, evaluation, and selection (TIES), where a design space exploration is conducted in addition to investigating the effects of technologies on the design.

2.1.2 System-of-Systems Design

As the system-of-systems (SoS) paradigm has revealed, new design methodologies are being studied to enable the design space exploration of such complex systems. Note that the combination of individually designed and optimized vehicles may not guarantee an optimal SoS due to the potential combined effect of multiple systems working together. Therefore, the traditional design techniques are seen insufficient to address the issues of networking, and new approaches are developed to identify the system requirements of SoS.

An SoS problem exhibits distinct differences when compared with a single system problem. The characteristics of an SoS problem are identified by the authors of [42] as the system type, the control of systems, and the connectivity of systems. First of all, the design of an SoS requires analysis methods that are appropriate to the type of entities constituting the system of systems. Second, it is crucial to determine the degree of control over the entities by an authority or the autonomy granted to the entities. Finally, entities of a system of systems are interrelated and communicate with other (likely a subset of) entities in the SoS. Note that the interaction among the systems of an SoS leads to an emergent behavior that may not be predictable from the systems characteristics. For instance, the emergence is defined in [41] as follows:

> "*A system exhibits emergence when there are coherent emergents at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such emergents are novel with respect to the individual parts of the system.*"

As a result, the design of an SoS should capture the effects of control and connectivity of systems as opposed to the design of a single system.

In an SoS problem, a commonly used modeling technique is *agent-based modeling* (e.g. [68, 70, 71, 75]). Accordingly, each individual system of SoS is represented as an agent interacting with its environment. A typical agent can be represented with respect to its attributes, behavioral rules, memory, resources, decision making sophistication, and rules to modify behavioral rules [75]. Accordingly, a simulation environment is created for the agents and their interactions with each other. Agent-based modeling can be used to bring operational outcomes (emergent behavior) into the design of agents. Accordingly, a particular emergent behavior can be avoided or promoted by a proper selection of the design variables based on the simulation results. Using agent-based modeling in SoS design is used in various aerospace design studies. For instance, the authors of [96] propose an agent-based approach for conceptual design exploration of a fleet of UAVs to form a private package delivery enterprise. As such, they show the trade-offs among vehicle, network, and economic parameters, and they explore design space to evaluate feasible designs through simulations. A similar kind of agent-based approach is adopted in [69] to identify the design requirements of personal air vehicle systems. In a later study, [101], the mission effectiveness of multiple UAVs in reconnaissance and surveillance missions is studied through a stochastic agent-based analysis, and the Monte Carlo simulations provide insight into the effective set of design variables for desired aircraft performance.

Alternatively, the authors of [64] introduce a different approach that creates abstractions for roles, responsibilities, services, and goals. In this manner, their proposed methodology differs from the object-oriented methodologies by focusing on the end-point that is to be reached rather than the types of behaviors that will lead to the end-point. Finally, another way to identify SoS alternatives is studied in [55] with an architecture-based approach that uses the Department of Defense Architecture Framework (DoDAF). In this manner, alternative designs are generated by modifying the baseline products, and they are checked for feasibility and consistency with the entire system architecture.

2.2 *Control Strategies for Multi-Agent Systems*

An MAS consists of a group of distributed agents, each of which operates individually. In such systems, the global or local information can be used in the action selection, which differs the type of the control. *Global information* typically refers to the states of all the agents and any data at the overall environment. On the other hand, *local information* corresponds to the states of a subset of the agents and some data at a limited part of the environment. In order to achieve a desired MAS performance, each agent uses global or local information to take necessary actions. Accordingly, the control of an MAS can be in a centralized or a decentralized fashion.

2.2.1 Centralized Control

Centralized control generally refers to the presence of a central authority that controls all agents based on the available global information. In other words, the central authority actively supervises the agents during a mission. Due to requiring the global information of an MAS, centralized control is typically common in small scale systems. There are several advantages of centralized control. Since centralized control takes into account the states of all agents as well as their possible interactions among each other, it is possible to optimally control the agents to accomplish desired tasks. Moreover, centralized control can adapt to the changes in the agents and their interactions due to globally observing the overall system. However, as the number of agents increases or the state-space gets larger, centralized control becomes intractable and inefficient from computation perspective. Furthermore, a centralized control is not robust to the possible failures. In particular, if a failure happens in the central authority, the control over the overall systems is lost. Eventually, the absence of a central authority almost always causes the loss of MAS capabilities.

2.2.2 Decentralized Control

Decentralized control refers to the distributed local controls that determines the individual agent actions based on some available local information. An overview of decentralized control is presented in [15]. In decentralized control, communication is a key concept. Since each agent is controlled separately, their emergent behavior can satisfy a desired performance if agents interact with each other. As long as the coordination is realized among the agents and efficient rules are designed to determine local agent actions, decentralized control achieves a desired performance by using only some local information. Therefore, decentralized control can be applicable to large scale systems as opposed to centralized control. One major issue with decentralized control is the challenge of developing efficient rules for local agent actions. In some cases, it may become difficult to result in optimal actions by using some local partial information.

2.3 Control Strategies for Multi-Agent Surveillance

In surveillance missions, a cost effective and safe option to gather information about an area is to send some mobile agents (i.e. unmanned vehicles) to the field and to receive information monitored by them. In an ideal case, if

- a sufficient number of agents are sent to the field to cover the overall area,
- each agent can instantly send information back to the base, and
- each agent has endurance equal to the overall mission endurance,

the base gathers the maximum amount of information about the field. In other words, situational awareness is maximized. However, the aforementioned three specifications may not be simultaneously realizable in most of the scenarios. For instance, it may not be always possible to use a sufficient number of agents to cover the overall region. As such, the agents may need to patrol the regions of interest. Alternatively, the agents may not necessarily have endurance that is equal to the mission endurance. Therefore, they may need to return to the base for refueling/recharging, which causes a degradation in situational awareness.

In persistent multi-agent surveillance problems, a major problem regarding the control of agents is on designing agent policies resulting in efficient monitoring. In the literature, this problem has been widely studied from various perspectives. For instance, one approach to achieve persistent surveillance is via randomized strategies. The authors of [123] consider a continuous coverage sweep problem, where the agents continually move over a region according to a Markov chain. Alternatively, the authors of [53] propose a strategy that uses virtual pheromones to act as potential fields. In that way, the agents are guided towards the areas that have not visited for a long time.

Another approach widely used to solve persistent surveillance problem is via a combinatorial optimization formulation. In general, the objective of the optimization problem is to minimize the time gap between the consecutive visits of a particular point. For instance, the authors of [129] and [121] solve a vehicle routing problem, [25], to achieve persistent surveillance. Alternatively, some centralized policies for multi-UAV persistent surveillance are proposed in [86] and [87], where the time between visitations to the same region are minimized. Moreover, the authors of [33] presents an optimal control formulation of the persistent surveillance problem.

In a persistent surveillance mission, it is likely to use agents that have less endurance than the overall mission endurance. In such cases, efficient persistent operations also require to consider the energy management of the agents. In the literature, there are several approaches used for determining when an agent needs to return to the base for refueling (or recharging). Some common approaches are solving the problem via a dynamic program (DP) (e.g. [26, 27, 128]) or a mixed integer linear program (MILP) (e.g. [61, 62, 116]). In general, the solution is obtained through a numerical optimization solver, which may result in an optimal or an approximate solution.

A persistent surveillance problem with some energy constraints can be modeled as a

Markov decision process (MDP), which can be solved using dynamic programming. However, as the size of the problem increases, the computation complexity dramatically grows. Therefore, there are some studies in the literature that propose approximate solutions for the MDP formulation of the persistent surveillance problem (e.g. [27, 28, 128]). Moreover, a more scalable approach utilizing a decentralized learning framework is proposed in [126], which can also capture heterogeneity among the agents.

A more recent study [61] proposes a MILP based solution to the schedules of multiple resources for persistent UAV operations. Accordingly, the authors compute the trajectories of UAVs through a minimization problem, where they define the total cost based on the costs of UAV travel, UAV purchase, and station purchase in addition to a cluster of constraints to guarantee effective UAV operations. In order to obtain an optimal solution, they first use the numerical solver CPLEX then propose a branch and bound algorithm to reduce the computational complexity. Note that the authors of [61] assume the availability of the global information about the UAVs, so they propose a centralized solution for the scheduling of UAVs performing persistent surveillance operations. Nonetheless, they do not consider any uncertainty or stochastic effect on the mission, and they do not take into account the recency of the monitored information in their objective function.

Alternatively, a decentralized approach is proposed by the authors of [43], where they formulate a distributed, energy-aware control policy to enable persistent surveillance of a region by a team of networked robots. Accordingly, they define a coverage energy function and design a controller that minimizes this function. Here, minimizing the coverage energy function implies an agents trade-off to achieve its coverage mission and to maintain energy reserves to guarantee its own safety. Moreover, the authors assume the presence of a relay agent such that whenever an agent starts to move away from the field, the remaining agents reconfigure themselves to maximize the coverage area until the relay agent arrives the surveillance area. Consequently, the agent that has returned to the base for refueling becomes the new relay agent.

2.4 Control Strategies for the Recovery of Network Connectivity

Connected communication networks play an important role in achieving various multi-agent coordination tasks (e.g. [59, 135, 136] and the literature cited within). For instance, a connected network can enable multiple spacecraft to synchronize their attitudes with each other (e.g. [38], [137]). Alternatively, a connected computer network achieves efficient peer-to-peer operations (e.g. [91, 105]). Moreover, consensus in sensor networks depends on the connectivity requirements of the network (e.g. [65]). Last but not the least, network connectivity is crucial in surveillance missions, where a group of agents such as UAVs operate around a base and stream back the surveillance data back to the base through multi-hop communications(e.g. [32, 98]). Particularly in long surveillance missions, an agent needs to return to the base for refueling or recharging. Whenever an agent leaves the surveillance area, there is no guarantee that the communication network of the remaining agents will maintain its connectivity. Therefore, there has been a growing interest in developing strategies for connectivity maintenance of networked systems.

In a networked system, a disconnection can be avoided or fixed by *proactive* or *reactive* approaches, respectively. In proactive approaches, a robust network topology is designed *a priori* such that the network can tolerate a finite number of agent or link losses (e.g. [85,122,133]). In reactive approaches, a control strategy is developed such that the network self-repairs itself in the face of agent or link removals (e.g. [5, 6]). Note that relying on proactive approaches only may be impractical in applications, where a large number of agents or links may eventually be removed from the network. Furthermore, connectivity maintenance strategies can be characterized as *centralized* or *decentralized* based on the information leveraged in the decision scheme. In a large scale system, the availability of global information to individual agents is usually not feasible. Therefore, a decentralized strategy is preferable over a centralized one due to practicality and scalability concerns.

Recently, a great amount of interest has been devoted to the analysis of multi-agent systems using graph theory. In these studies, the nodes of a graph represent the agents (such as UAVs, sensors or individuals), and edges represent the direct interactions between them. The literature on the graph theoretical connectivity control of mobile systems against edge failure is including, but not limited to, optimization based connectivity control (e.g. [135]), continuous feedback connectivity control (e.g. [107]), and control based on the estimation of the algebraic connectivity (e.g. [108]). In these studies, the authors mainly consider uncertainty in edges, and they assume a constant number of nodes. In the last few years, there has also been a significant interest in addressing agent loss problem in networked systems. In [85] and [122], the main focus is on the design of robust network topologies that can tolerate a finite number of agent removals. In [122] and [17], the authors propose self-repair strategies that create new connections among the neighbors of the failing agent.

Following a different approach than earlier studies, some researchers consider mobile agents and propose some agent movements for connectivity restoration. Some relevant works on distributed connectivity recovery of wireless sensor networks in agent failure include [6], [5], [4], [45], [83], and [134]. The authors of [6], [4], and [45] propose a set of cascaded replacements for the recovery of network connectivity. Alternatively, the authors of [5] propose block movement of agents instead of cascaded replacements, and their algorithm not only restores connectivity but also does not extend the shortest paths among agents after a failure. To this end, they use the shortest path routing table in their scheme and define a confidence level that correlates to the population of the routing table. From their simulation studies, the authors observe that agents having low-populated rows on the routing table (i.e. less than 20%) do not initiate any network recovery. In other words, a disconnection due to an agent failure may not be prevented by the remaining agents having too little information.

In order to develop efficient strategies for connectivity restoration, determining agent criticality is an important step because a recovery process is required only in the face of a critical agent removal. Unfortunately, agent criticality is a global information such that an agent may not necessarily estimate its criticality based on its local neighborhood. Therefore, even though many studies propose distributed schemes for the recovery of connectivity, they use some centralized approaches for the initiation of the process (i.e. the overall proposed schemes rely on some global information). For instance, the authors of [4] assume that each agent has perfect knowledge about its criticality, whereas the authors of [6] use the algorithm in [37] to determine the criticality of an agent based on the connected dominating set (CDS) of the whole network. Note that the efficiency of a CDS algorithm depends largely on the process of finding and maintaining a small CDS, and finding a minimum CDS is NP-complete for most graphs, even if global information is available [37]. Moreover, a topological change (e.g. agent removal/addition) in the system causes a need to update the CDS, which is stated as a complicated process by the authors of [37]. Therefore, the recovery scheme proposed in [6] requires a great amount of computation after any agent failure to refresh CDS and then to update agent assignments accordingly.

An approach to develop more localized algorithms for determining criticality is using information only in k-hop neighborhood. For instance, the authors of [83] propose a connectivity restoration framework, where each agent constructs a routing table from its k-hop neighborhood to estimate its neighbors' criticality. Accordingly, each agent is assumed to communicate with its $k+1$-hop neighbors to achieve coordinated motion in case of an agent failure. Alternatively, the authors of [134] employ a hybrid method, which combines criticality in k-hop neighborhood with CDS, to decrease overall messages in determining agent criticality.

Finally, in many studies, agent failure in sensor networks is studied based on the assumption of agents sending periodic heart beat messages (e.g. [5, 6, 83]). Accordingly, if a heart beat message is not received by the neighbors of agent i for some period of time, then agent i is assumed dead by the neighbors. Note that an agent failure is an involuntarily leave of an agent from the network, and it is analogous to a UAV crash. In this thesis, the main focus is on voluntarily leaves of agents due to reasons such as reaching a fuel threshold or changing mission strategy. These types of reasons are mostly seen in surveillance missions, where agents may require to return base for refueling (e.g. [18, 127]) or may leave the group for tracking a target (e.g. [32, 98]). Henceforth, a voluntarily leave of an agent is referred to as an *agent removal*.

CHAPTER III

METHODOLOGY DEVELOPMENT

The foundation of this research is based on multi-agent systems, whose performance is greatly influenced by the design variables of the agents and the control strategies utilized by the agents. Chapter 2 presented a survey of literature for the design and the control of MAS. The aim of this chapter is to make some observations regarding the literature and to construct the primary hypothesis of this thesis. In order to show the correctness of the hypothesis, a methodology is also proposed at the end of this chapter.

3.1 A Generic Design Methodology

In literature, there exist various system design processes applied to different types of systems. For instance, [14] presents a set of design processes and states that the generic steps can be generalized as follows: state the problem, investigate alternatives, model the system, integrate, launch the system, assess performance, and re-evaluate.

Typically, a design process starts with some *customer needs*, which determine the *mission specifications*. Then, an analysis is conducted to discover the requirements of the overall system. In general, systems engineering tools (e.g. [104]) along with some expert opinion are used in the identification of the requirements. Based on the *requirements analysis*, some *alternative designs* are generated by varying the design variables of the overall system. After that, the performance of the alternative designs are evaluated via *modeling and simulation*. Different techniques can be used in modeling the alternative designs. For instance, some well-known modeling techniques for an MAS are discrete event simulation, system dynamics, and agent-based modeling (e.g. [30, 115]). Following that the alternative designs are ranked with respect to their performances observed in the simulations. Some ranking techniques used in the literature are TOPSIS, AHP, to name a few (e.g. [7, 66, 106]).

Accordingly, if the requirements are satisfied by the top ranked candidate designs, then one of them is selected for further analysis. If the generated designs do not satisfy the requirements, then the requirement analysis is re-iterated. Consequently, the fundamental steps of a generic design methodology are illustrated in Figure 7.

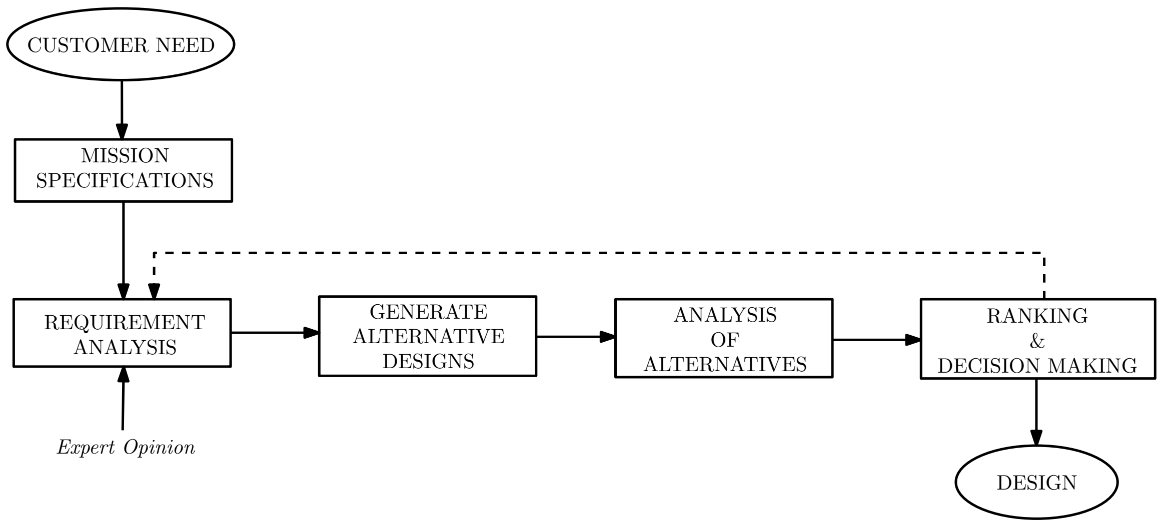


Figure 7: Fundamental steps in a generic design methodology.

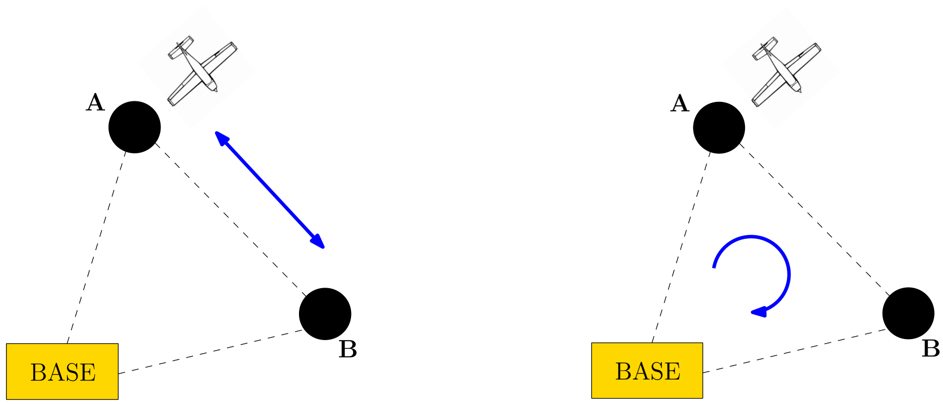
3.1.1 Observations on the Generic Design Methodology

While the design variables of the agents and the control strategies used by the agents together greatly impact the performance of an MAS, their identification and development are studied separately in the literature. For example, the control strategies are brought to the generic design methodology in the further steps. In particular, after the MAS is modeled, a fixed control strategy is assumed and the mission performance is evaluated accordingly. Such an approach brings the operational outcomes (i.e. resulting mission performance) into the selection of the design variables, which enables the design of an MAS resulting in optimal performance. However, the operational outcomes strongly depends on the selected fixed control strategy. In other words, there is no guarantee that the MAS would result in a similar optimal performance under a different control strategy. On the other hand, in the control literature of MAS, the main goal has mostly been on the design of rules (or policies) for taking actions that optimize some individual or collective objectives (e.g. decisions for the next target point to move). The details of the agent design are generally not considered in developing control strategies. For instance, agents are typically assumed as point-masses or systems with low order dynamics. Consequently, the literature has minimal precedent transparent methodology that ties together control strategies, vehicle capabilities, and operational outcomes in the design of MAS. Hence, the main research objective of this thesis is to incorporate the control strategies in the early design of MAS.

3.2 Hypothesis Formulation

A desired mission performance in an MAS may not be obtained by solely designing control strategies or solely selecting design variables. Note that the design variables determine the capabilities of the MAS. On the other hand, the control strategies guide agents to take strategic actions for achieving a task with the given capabilities. So that it is very likely to observe some interdependency between the design and the control of MAS.

Consider a canonical scenario, where there exist a base, a UAV, and two monitoring points A and B as in Figure 8. In this scenario, assume that the UAV can communicate with the base from both points. Accordingly, let the main objective of the mission be to increase situational awareness of the base that can receive instantaneous monitoring information from the UAV. Let control strategy 1 guide the UAV to travel between points A and B continuously as in Figure 8a. If the UAV has infinite fuel capacity, then control strategy 1 results in the optimal performance. However, control strategy 1 causes mission failure in real world applications because any vehicle has limited endurance. As opposed to the control strategy 1, let control strategy 2 guide the UAV to follow a cyclic route as in Figure 8b. As such, the UAV leaves the base, visits point A, and then visits point B. Finally, it return to the base. According to control strategy 2, such a trajectory is followed continuously. Note that the control strategy 2 is applicable to finite fuel capacity vehicle because it results in periodic visits to the base for possible refueling. However, control strategy 2 leads to the optimal performance only if the endurance of the UAV is exactly equal to the time required to complete the cyclic route. If the UAV has longer endurance, then when it arrives to the base it will still have some remaining fuel, which causes the inefficient use of the vehicle.



(a) Traveling between points A and B. (b) Following a cyclic route.

Figure 8: The interdependency between the design and the control of an agent.

As it is seen, a control strategy results in different performance for different vehicles. Even in this canonical scenario including only one UAV, the interdependency between the control strategies and the design variables is observed. As the number of UAVs increase and the area gets larger, the interdependency between the design variables and the control strategies likely becomes significantly complex. Therefore, the main hypothesis of this thesis is formulated as follows:

***Hypothesis* 1.** *For a desired mission performance, the selection of the design variables changes with different control strategies due to the effect of the interdependency between the design variables and the control strategies.*

3.3 Proposed Methodology

In this thesis, the main objective is to incorporate control strategies in the early design of MAS. In Chapter 1, the primary research questions are presented as

- *RQ1: Can a control strategy influence the selection of the physical design variables to result in a desired MAS performance?*

- *RQ2: How is it possible to explore the design-control couplings in an MAS at the early design stages?*

It has been previously depicted that there exists some interdependency between the design variables of the agents and the control strategies used by the agents, which greatly impacts the performance of the overall system. Exploring such interdependency can be easier in the later design stages of an MAS due to the increased knowledge about the design and the control of the agents. However, the system capabilities are often not determined in the early design stages. Accordingly, this thesis proposes a methodology that can explore the possible interdependency between the design variables and the control strategies in early design.

An overview of the proposed methodology is illustrated in Figure 9, where an additional component is introduced to the generic design methodology. As such, the proposed component, illustrated in Figure 10, translates the mission specifications to design- and control-related information to support the requirement analysis and later design stages. The input to the proposed component is the mission specifications for an MAS whereas the output are the influential design variables, the effective control strategies, and any investigated design-control pairs in an MAS. To this end, the control and the design modules are proposed, and they are detailed in the following sections. Eventually, the output of the methodology is expected to verify the correctness of Hypothesis 1.

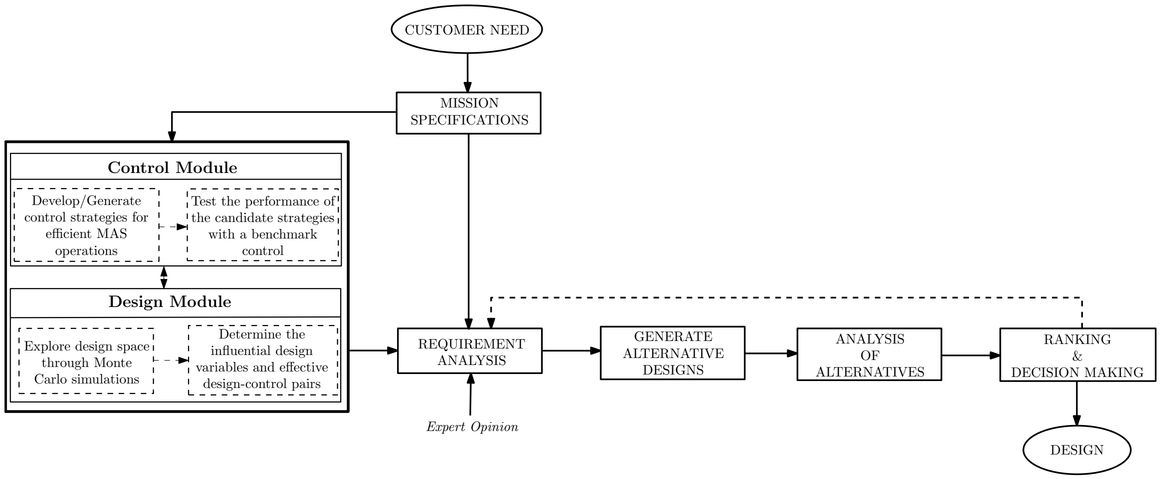


Figure 9: The overview of the proposed component introduced to the generic design methodology.

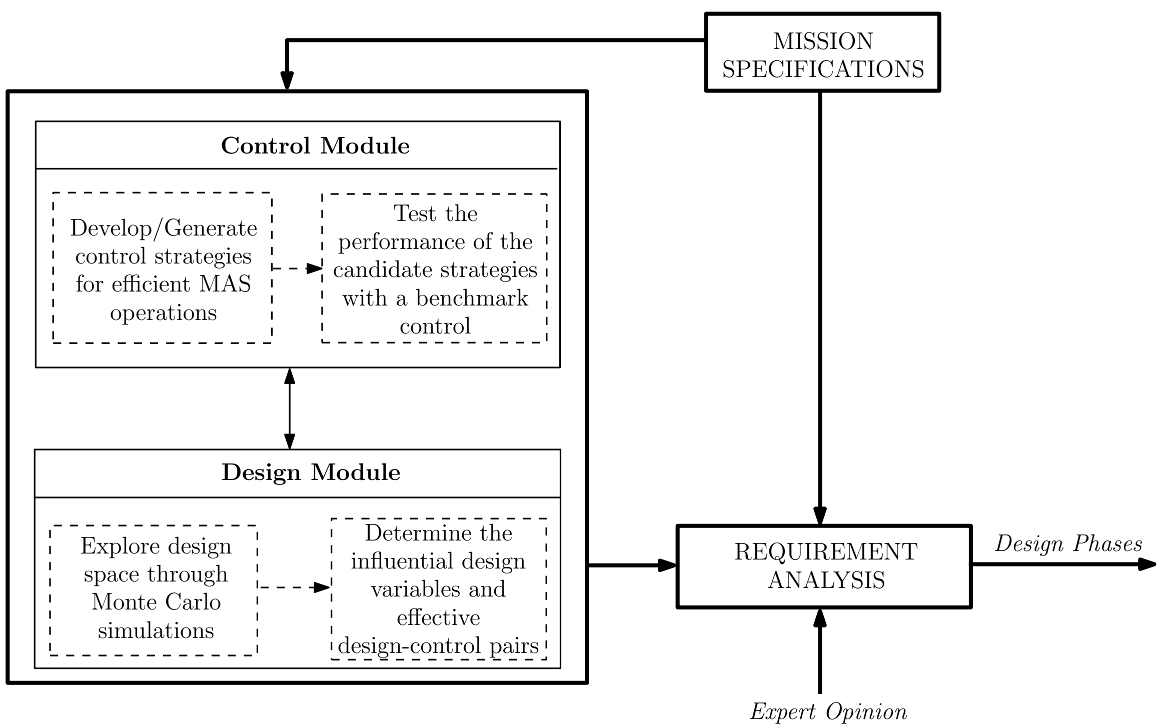


Figure 10: Zooming into the proposed component.

3.3.1 Control Module

The objective of the control module is to develop a set of candidate control strategies for an MAS, which will be used in the design module for a design space exploration. For the development of the candidate strategies, the control module attempts to answer the following questions:

3.3.1.1 What are the objectives and the constraints in a specific multi-agent mission?

In developing a control strategy, a fundamental task is to identify the objectives and the constraints in a mission. Well-formulated objectives enable the mission accomplishment through an efficient control strategy design. Furthermore, each agent in an MAS can have constraints for energy, communication, or mobility, to name a few. Therefore, a control strategy utilized by an MAS should ensure that the constraints are not violated during the mission.

3.3.1.2 How to choose the candidate control strategies that will be used in the design module?

After determining the objectives and the constraints in a mission, it is ideally desired to develop optimal control strategies. For instance, a centralized control may result in optimal performance due to employing the global information in the action selection; however, it may not be applicable to large scale MAS. On the other hand, a decentralized control is preferable in large scale systems due to relying on only some local information but it may result in a sub-optimal performance. Here, it is important to analyze the optimality gap of a decentralized strategy. Therefore, the developed strategies need to be compared with a benchmark strategy that is preferably optimal. The candidate strategies with acceptable optimality gaps can be passed to the design module to be used in the design space exploration.

3.3.2 Design Module

In an MAS, the design variables of the agents play an important role on improving the mission performance. However, not all of them have the same significance on the overall performance. Before the actual design of the agents, it is beneficial to provide insight into the significant design variables so that the high fidelity analysis is more focused on them at later design stages. To this end, the design module introduced in the proposed methodology aims to explore the design space of an MAS. The detailed steps inside the design module will be elaborated in Chapter 8. Overall, the design module attempts to answer the following questions:

3.3.2.1 What are the influential design variables of an MAS performing a specific mission?

In order to identify the influential design variables, a commonly used approach in the literature is design space exploration, which inspects the sensitivity of the mission performance to the changes in the design variables. It has been observed that the existing studies in the literature (e.g. [96, 101]) explore the design space by assuming a fixed control strategy. However, such explorations lead to results specific to the tested control strategy. In order to broaden the results of a design space exploration, the design module in the proposed methodology takes into account various control strategies that are developed in the control module. As such, the influential design variables are identified in accordance with the results of all tested control strategies.

3.3.2.2 What are the trade-offs among the design variables of an MAS performing a specific mission?

In order to obtain a desired MAS performance, understanding the design trade-offs is as important as identifying the influential design variables. In this respect, well-understood design compromises provide more insight into the overall system characteristics. Moreover, the design space exploration conducted in the design module takes into account various control strategies. Therefore, the design trade-offs also vary under different control strategies. Accordingly, the design module aims to explore the design trade-offs as well as the interdependency between the design variables and the control strategies based on the overall design space exploration. Note that the output of the design module (i.e. influential design variables and the effective design-control pairs) can greatly support the strategic decision-making in the later design stages.

3.4 Summary of the Proposed Methodology

The proposed methodology introduces an additional component to the generic design methodology, which translates the mission specifications to some influential design variables, efficient control strategies, and potential interdependency between them. To achieve this, two modules are introduced for the control and the design of agents. The control module deals with generating a set of efficient control strategies in accordance with the objectives and the constraints in a particular MAS mission. On the other hand, the design module conducts a design space exploration through Monte Carlo simulations, where various design variables and the set of generated control strategies are tested. Accordingly, the proposed additional component specifies the effective design-control pairs, as well as the influential design variables and efficient control strategies, which can potentially provide a desired performance. This information can be utilized in the later design stages to optimize the overall system through some high fidelity analyses.

CHAPTER IV

CASE STUDY: MULTI-AGENT PERSISTENT SURVEILLANCE

4.1 Mission Overview

In the previous chapters, various design and control aspects in a multi-agent surveillance mission are presented. For example, patrolling (i.e. cruise flight) and loitering are some possible agent operations in a surveillance mission. For high and low altitude operations, agents compromise between the quality of data and the energy consumption. A surveillance mission typically requires require monitoring for long periods of time, which is mostly longer than the endurance of an agent. Furthermore, there are various ways for an agent to share the monitoring information with the base (e.g. recording the monitored data, directly communicating with the base, or sending information back to the base through some agent collaboration). Accordingly, this thesis particularly focuses on missions, where agents

- loiter on some monitoring points and patrol if required,
- change operation altitude when it is necessary,
- use multi-hop communications to stream data back to the base,
- have less endurance than the overall mission time.

Since agents have less endurance than the overall mission time, they need to go back to the base for refueling/recharging. Thus, the agents travel between the base and the surveillance area multiple times, which is referred to as *persistent surveillance* in this thesis. Note that an agent removal from the surveillance area is a disturbance to the overall system because it causes performance degradation in coverage or distributed communications, to name a few. Here, the severity of performance degradation depends on both the design and the control of multi-agent system. Accordingly, Figure 11 illustrates some design and control elements that affect the overall situational awareness.

In an MAS surveillance mission, two key concepts are the energy management of the agents and the region assignment for the agents. The energy management of the agents can be impacted by the design and the control. For instance, the physical design determines the geometry and the propulsion system of the agents, which determines the maximum energy capacity and the energy consumption of an agent. On the other hand, the mission design determines the operational tasks such that the required maneuvers or the operational altitudes can vary the energy consumption of an agent.

Furthermore, the energy management can also be impacted by the control strategies used by the agents. In order to increase situational awareness, an energy-aware control strategy can lead agents to strategically move on the surveillance area in an efficient fashion. Finally, the region assignment is crucial to increase situational awareness. For instance, the situational awareness can be maximally improved by assigning agents to proper locations that are not visited for a long period of time.

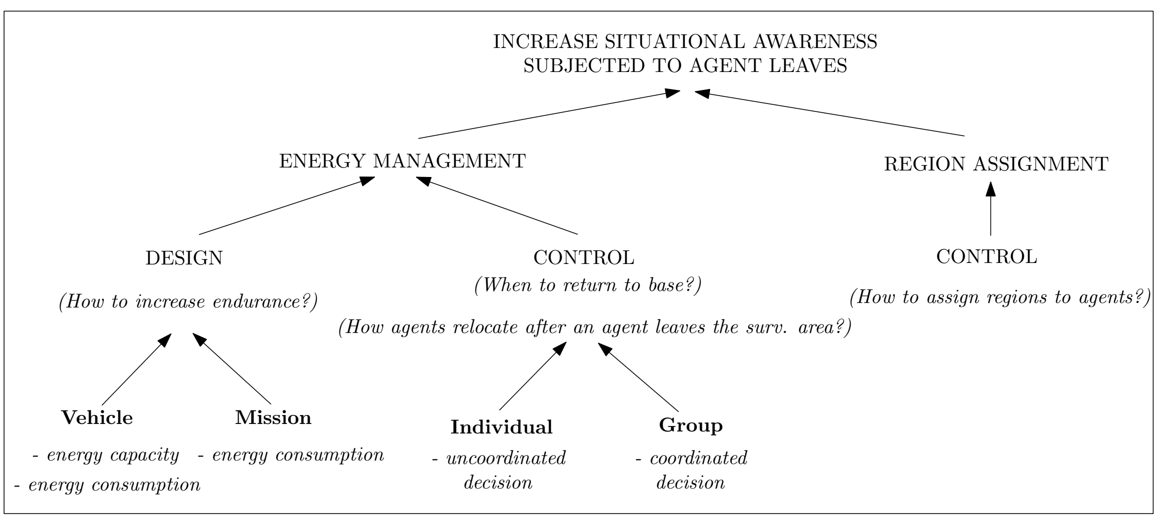


Figure 11: Influential elements to increase situational awareness in persistent surveillance missions.

4.2 Problem Formulation

This thesis addresses a discrete time multi-agent surveillance problem, where the major objective is to increase the situational awareness about a target area. Suppose that a desired region of interest is discretized in two-dimensions as in Figure 12a such that a grid cell on row i and column j has a weighting of w_{ij} and a time varying age as $\alpha_{ij}(t)$. Here, $\alpha_{ij}(t)$ is the time duration that cell ij has not been monitored, and $\alpha_{ij}(0) = 0$.

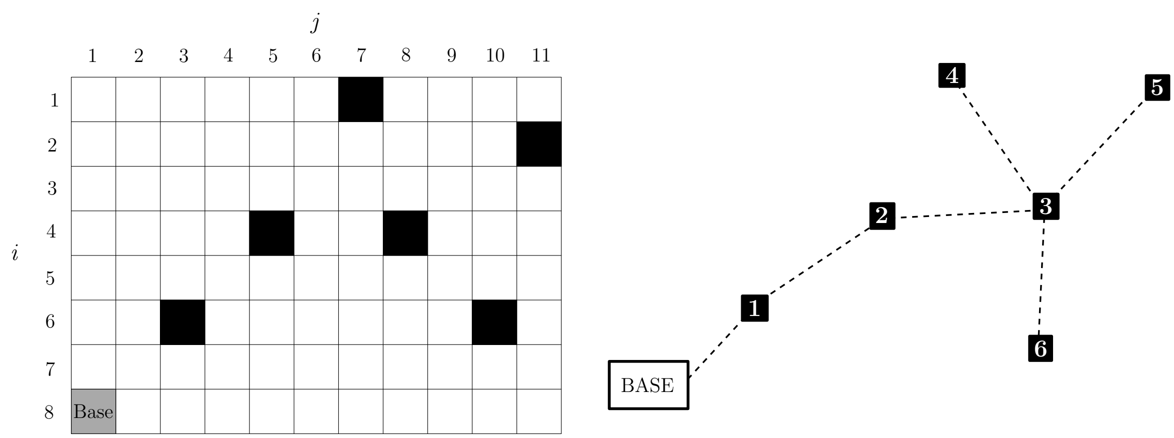
$$\alpha_{ij}(t) = \begin{cases} 0, & \text{if grid } ij \text{ is monitored,} \\ \alpha_{ij}(t-1)+1, & \text{otherwise.} \end{cases} \tag{3}$$

Accordingly, the overall degradation in situational awareness at time t, i.e. $J_{SA}(t)$, can be quantified as the summation of cell ages multiplied by the cell weights. Note that minimizing J_{SA} implies maximizing situational awareness.

$$J_{SA}(t) = \sum_{\tau=0}^{t} \sum_{i,j} w_{ij}\alpha_{ij}(\tau). \tag{4}$$

In a large surveillance area, the weightings of each grid cell may not necessarily be uniform because some cells may represent more essential coordinates (e.g. the oil rings on a sea, the high population zones in a city, or the disaster areas in an environment). In such cases, the relative weightings of the essential cells are assigned with higher values than the others.

In this thesis, the surveillance area is discretized as in Figure 12a, where the black cells denote the high priority zones with a weighting of 1 (i.e. the monitoring points), the white cells represent the rest of the points with a weighting of 0 (i.e. the points that do not require monitoring), and the gray cell denotes the base. In this setting, the base sends some agents (e.g. UAVs) to the field to gather information about the points of interest. Note that agents and the base have limited communication capability over a field because the continuous wireless communication may not be always feasible, reliable, or secure due to weather effects, jamming attacks, or service delays [110]. In the presence of limited communications, the environment can also be represented as a graph, which takes into account not only the geographical coordinates of the monitoring points and the base but also the communication ranges of agents. For instance, suppose that agents having communication range of R will be located on the grid cells in Figure 12a. Accordingly, if the distance between any pair of grid cells are less than R, then agents located on the corresponding cells can communicate with each other. Hence, the overall environment taking into account the communication range of agents can be represented as a graph shown in Figure 12b.



(a) A surveillance area represented as 2D grids.

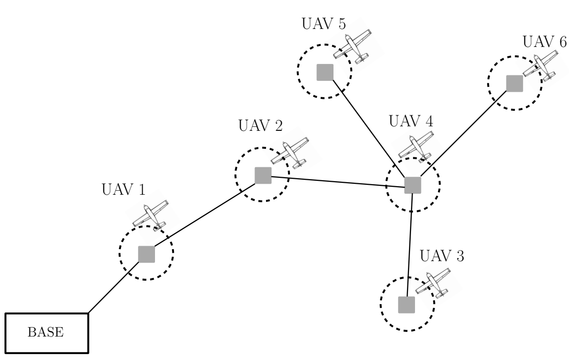
(b) A surveillance area represented as a graph.

Figure 12: Surveillance area

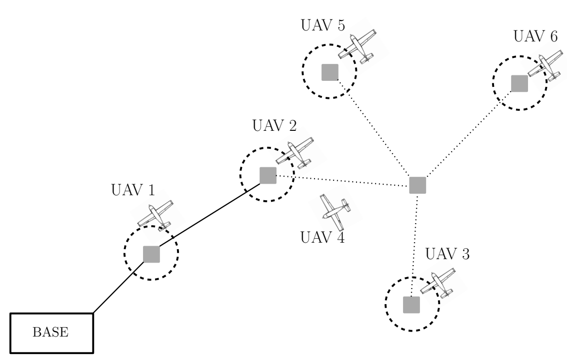
In Figure 12b, the nodes of the graph represent the monitoring points and the base, whereas the dotted edges represent the communication capability based on the agents' communication range. For instance, in Figure 12b, if there is agent i located on node 1, then agent i can directly communicate with the base (i.e. single-hop communication). If there is another agent j located on node 2, then agent j communicates with the base through agent i located on node 1 (i.e. multi-hop communication).

In an ideal case, if each node is monitored by an individual agent that has endurance equal to the mission endurance, then the base has the maximum situational awareness about the field (i.e. $J_{SA}(t) = 0$ for $t > t_{arv}$, where t_{arv} is the time an agent reaches the last unoccupied node). However, an agent has an inherent possibility of leaving the surveillance area in long-run missions due to refuel/recharge, maintenance, or a strategy change.

The removal of an agent becomes a disturbance to the overall system, and it causes a degradation in situational awareness due to preventing the regular updates of some particular nodes. For instance, Figure 13a illustrates a canonical scenario where each UAV monitors a single node and can send data back to the base due to the presence of a connected communication network. Suppose that UAV 4 needs to return to the base for refueling. As seen from Figure 13b, the removal of UAV 4 causes not only an unoccupied node but also a disconnection among UAVs 3, 5, 6, and the base. Here, even though UAVs 3, 5, and 6 monitor their assigned nodes, they cannot send information back to the base due to the disconnection.



(a) All UAVs can send data back to the base due to the presence of a connected communication network.



(b) The removal of UAV 4 causes a disconnection so UAVs 3, 5, and 6 cannot communicate with the base.

Figure 13: A canonical persistent surveillance scenario consisting of a group of UAVs.

Similar to the ages of grid cells, let $\alpha_j(t)$ be the age of node j. Here, $\alpha_j(t)$ is the duration of time that the base has not received any data from node j. Note that if a UAV loitering over node j cannot send its data back to the base due to the absence of multi-hop communication (i.e. UAV 5 in Figure 13b) or if there is no UAV loitering over node j (i.e. the unoccupied node in Figure 13b), then $\alpha_j(t)$ increases. As such, (3) for this setting can be rewritten as

$$\alpha_j(t) = \begin{cases} 0, & \text{if a UAV sends data about node } j \text{ back to the base,} \\ \alpha_j(t-1)+1, & \text{otherwise.} \end{cases} \tag{5}$$

As seen from (5), $\alpha_j(t)$ is a piecewise monotonically increasing function, and its increase implies a degradation in situational awareness for the base. Using the summation of node ages, a cost can be defined to quantify the degradation in situational awareness as

$$\mathscr{C}_{age}(t) = \sum_{j \in V} \alpha_j(t), \tag{6}$$

where V denotes the set of nodes on the surveillance area. Note that if there is no UAV on the field or there is no information flow from the UAVs to the base, $\mathscr{C}_{age}(t)$ is a monotonically increasing function. Based on the depicted scenario, the overall objective function of the surveillance problem is

$$J = \sum_{t=1}^{T} \mathscr{C}_{age}(t), \tag{7}$$

where T is the overall mission endurance.

4.3 Agent Assumptions

In this thesis, each agent represents a UAV, and a UAV monitors a node by loitering over it. A node having zero age implies that a UAV can send the image of the node back to the base. Let a UAV have a camera, which has a fixed cone angle and acquires rectangular image. Then, the area of the acquired image is directly proportional to the altitude of the UAV as (8).

$$A(I) \propto 4h^2 \tan^2\left(\frac{\theta_{cone}}{2}\right) f, \tag{8}$$

where f is the scaling factor between the length and width of the rectangular image. As the flight altitude increases, the acquired image area gets larger (e.g. Figure 14) whereas the image resolution gets lower. Assume that the camera mounted on a UAV has a fixed number of image pixels, i.e. N_{px}. Then, the resolution of the image, $Res(I)$, is inversely proportional to the size of a pixel as in (9). Accordingly, as the pixel size decreases, the image resolution increases so that more details can be captured.

$$Res(I) \propto \frac{1}{A(I)/N_{px}}. \tag{9}$$

From the operational perspective, a UAV loiters over a node to accomplish two major tasks: identifying or detecting objects. In this thesis, detecting an object is assumed to require higher resolution images than identifying an object. Therefore, a UAV loiters at high altitudes to identify objects. Whenever an object is identified, then it decreases its loiter altitude to obtain high resolution images. Note that a UAV operating at lower altitudes is subjected to more drag, which can be overcome by producing higher thrust (i.e. causing more fuel/energy consumption). Consequently, a UAV changes its loiter altitude to acquire images with required resolution, which causes a variability in the fuel/energy consumption.

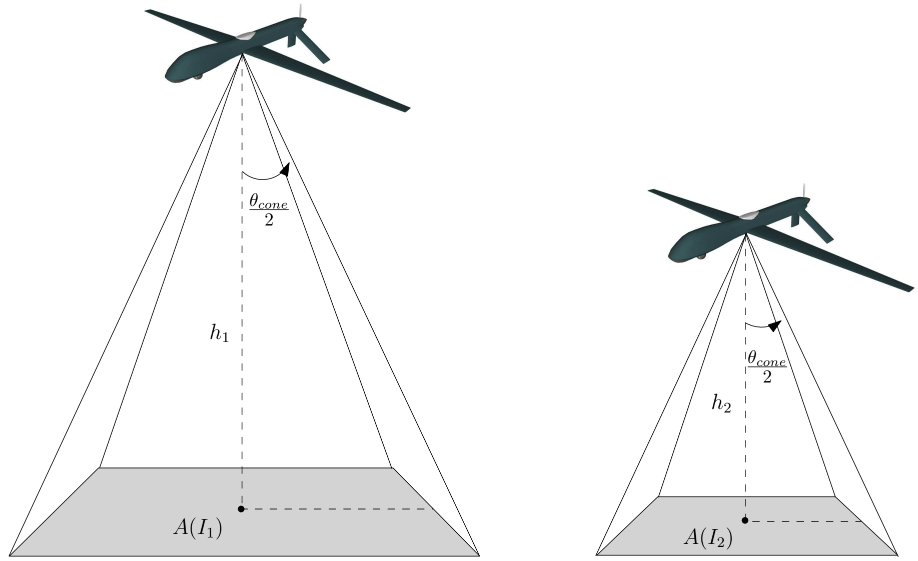


Figure 14: For a fixed camera cone angle (θ_{cone}), the monitoring areas, $A(I_1) > A(I_2)$, vary as the UAV loiters at different altitudes, $h_1 > h_2$.

4.4 Overview of the Methodology Implementation

As introduced in Chapter 3, the methodology proposed in this thesis consists of two major modules for control and design of agents. In this section, the control and the design modules are elaborated by demonstrating the methodology with the persistent multi-agent surveillance problem.

4.4.1 Control Module for Persistent Multi-Agent Surveillance

The objective of the control module is to develop (possibly decentralized) control strategies for efficient agent operations. As it is seen from Figure 15, the main input to the control module is the mission specifications.

Note that the control strategy refers to a high-level decision making that determines the actions of agents to accomplish some tasks. In the persistent multi-agent surveillance problem, the main goal is to increase situational awareness of a base that receives information from an MAS. Accordingly, the candidate control strategies developed in the control module should determine agent actions based on the following questions:

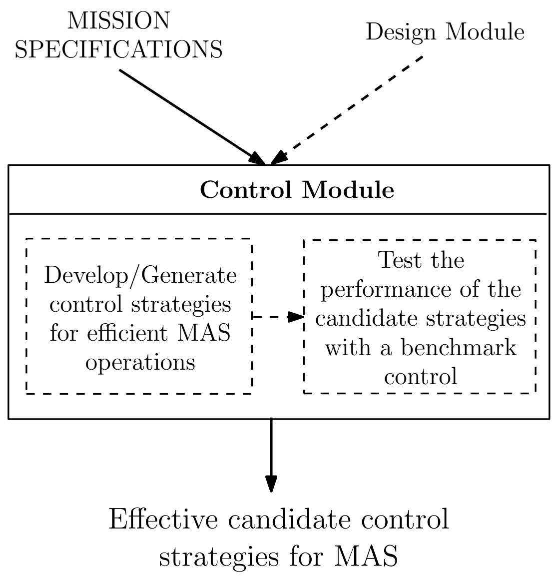


Figure 15: Zooming in the control module.

4.4.1.1 *How to choose the target location to be monitored for each agent in order to collectively optimize the situational awareness?*

A proper selection of the target coordinate is crucial to avoid degradation in situational awareness. In particular, a region of interest should not be let unoccupied for long periods of time. In the previous sections, it has been depicted that the age of a node (i.e. monitoring point) increases if it is not monitored by an agent or if no information flows to the base. Accordingly, the overall degradation in situational awareness is defined as the summation of node ages (i.e. J in (7)). In order to avoid an increase in J, one trivial solution is to assign an agent to a node that has the maximum age. However, the problem addressed in this thesis involves multiple agents, whose assignments to the same node causes inefficient agent operations. Therefore, a coordinated decision should be taken to avoid increase in J.

4.4.1.2 *When should an agent the return to base for refueling or recharging?*

As depicted in the persistent surveillance problem, each agent is assumed to have less endurance than the overall mission horizon. Here, it is crucial for an agent to avoid unsafe state such as running out of fuel (or energy). In a single agent system, it is trivial for an agent to take a return action whenever it reaches a critical fuel threshold (i.e. required fuel to return to base). However, such a decision may not be always effective in a multi-agent system. For instance, suppose that most of the agents reach their critical fuel thresholds at a similar time. Then, a severe degradation in situational awareness occurs due to the simultaneous agent returns. To avoid this, a coordinated decision should be taken to design efficient return schedules for agents.

4.4.1.3 When should an agent be relocated on the surveillance area?

When an agent leaves the surveillance area, a region of interest becomes unoccupied, and the communication network may possibly become disconnected. Note that if no agent visits the unoccupied region, then the situational awareness degrades (and even gets worse if the communication network is disconnected). On the other hand, if multiple agents simultaneously take action to visit the unoccupied region, then agents are used inefficiently (i.e. one region is monitored by multiple agents). Accordingly, some coordination is required in order to determine a set of efficient agent relocation.

As it is seen from the previous discussions, selecting a set of effective actions for an MAS requires coordination in decision-making. Note that a coordinated decision can be made by a central authority that determines the actions of all agents by tracking their states (i.e. a centralized approach). On the other hand, agents can coordinate with each other through communication so that each agent can make decision about its next action based on the information collected from the others (i.e. a decentralized approach). When determining agent actions, centralized and decentralized approaches exhibit specific advantages. For instance, the optimal set of actions can be computed via a centralized approach because an action selection is based on the states of all agents. However, such an approach becomes intractable and computationally inefficient if there exists a large number of agents. On the other hand, a decentralized approach uses only the states of some agents (i.e. the ones gathered through communication) to make an action selection. While a decentralized approach does not rely on all states of agents, i.e. a more practical solution for action selection, it may not necessarily result in optimal set of actions due to the limited information in the calculations.

Consequently, the major objective of the control module is to develop a set of control strategies (possibly decentralized), which result in effective agent operations in a persistent surveillance mission. In order to evaluate the performance of developed control strategies, it is also important to compare it with a benchmark strategy before sending it to the design module for further analysis.

4.4.2 Design Module for Persistent Multi-Agent Surveillance

The objective of the design module is to conduct a design space exploration for the identification of the effective design variables and the couplings between the design variables and the control strategies. As it is seen from Figure 16, the main input to the design module are the mission specifications and the control strategies, whereas the outputs are the influential design variables and the design-control couplings affecting the mission performance.

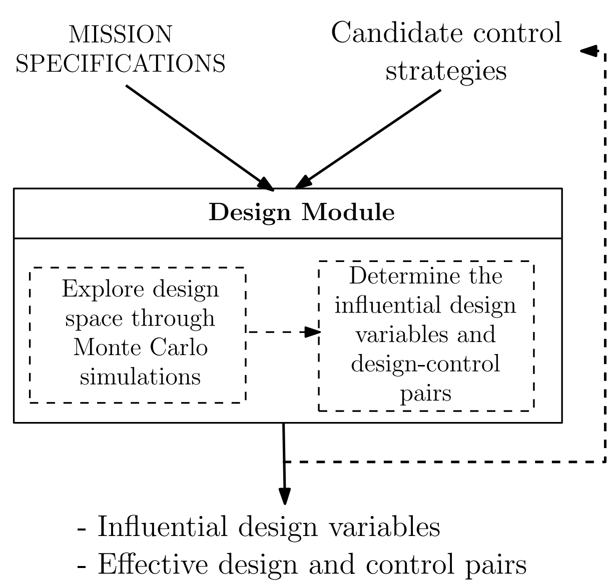


Figure 16: Zooming in the design module.

In the persistent multi-agent surveillance problem, the design variables of the agents (e.g. the maximum velocity, the maximum fuel capacity, the fuel consumption, the communication range) play an important role on the mission performance. Nonetheless, not all of the design variables impact the performance with the same significance. Accordingly, the design module implemented in the persistent surveillance problem attempts to answer the following questions:

4.4.2.1 What are the influential design variables in an MAS performing a persistent surveillance mission?

In a multi-UAV surveillance mission, there are numerous design variables related to the physical aspects of the UAVs or the networking issues among the UAVs. The physical aspects pertain to the weight, the aerodynamic characteristics, or the propulsion system performance, to name a few. Such physical variables determine the overall system capabilities. On the other hand, the networking issues relate to the communication topology in the overall system. Accordingly, it enables the interactions (e.g. coordination) among the agents. In a design space exploration, it is not feasible to take into account all design variables. Therefore, a subset of the design variables can be selected and the influential ones can be identified among the selected subset. However, an iterative process may require to choose different subsets for obtaining more generalized results.

4.4.2.2 What are the trade-offs among the design variables and the control strategies of an MAS performing a persistent surveillance mission?

In addition to the influential design variables, identifying the design trade-offs is also important to understand the overall system characteristics. In the persistent multi-agent surveillance problem, suppose that some influential design variables are determined as the maximum velocity, the maximum fuel capacity, the communication range, and the number of agents. In addition to this information, it is also critical to be able to answer the questions as follows:

- Can a large group of slow agents be compromised with a small group of fast agents?
- How is it possible to mitigate the effect of short endurance?
- Which variables or strategies do mitigate the effect of using a fewer number of agents?

Moreover, the identified design trade-offs (e.g. the answers to the previous questions) may not be similar under different control strategies. Accordingly, the design-control couplings can also be determined through the identification of the design trade-offs.

CHAPTER V

DECENTRALIZED CONNECTIVITY MAINTENANCE

Starting with this chapter, the following three chapters (Chapters 5, 6, and 7) will present an implementation of the control module shown in the proposed methodology. As proceeded within these chapters, the proposed ideas will build upon themselves to present a set of efficient control strategies for persistent multi-agent surveillance.

In a persistent surveillance mission, if the mission endurance is longer than the endurance of agents, agents need to return to the base for refueling (or recharging). When an agent leaves the surveillance area, there is no guarantee that the communication network among the rest of the agents will maintain its connectivity. In case of a network disconnection, information share cannot be accomplished among the agents and the base, which greatly hinders situational awareness.

The objective of this chapter is to introduce a decentralized strategy that maintains the network connectivity in the face of an arbitrary agent removal. This chapters starts with a brief review of graph theoretical terms that are frequently used along the following chapters. Then, the connectivity maintenance problem tackled in this chapter (i.e. replacement control problem) is presented. Following that, the proposed control scheme called Message Passing Strategy (MPS) is introduced, and its performance is improved by proposing δ-MPS. Finally, this chapter ends with some simulation results, where the proposed decentralized strategies (i.e. MPS and δ-MPS) are compared with the centralized optimal solution.

5.1 Graph Theory Preliminaries

An undirected graph, $\mathscr{G} = (V,E)$, consists of a set of nodes, V, and a set of undirected edges, E. Let X be a subset of V. Then, $\mathscr{G}_X$ refers to a *subgraph* induced by the nodes in X and their corresponding edges. In a graph, a k-length *path*, p, is a sequence of nodes $(p_0, p_1, ..., p_k)$ such that the edge between any p_i and p_{i+1} belongs to E. A path in $\mathscr{G}$ is called *simple* if it does not have any repeated nodes. An undirected graph, $\mathscr{G}$, is *connected* if there exists a path between any two nodes of the graph.

Let v_j and v_k be any two nodes in $\mathscr{G}$. Then, the *distance* between v_j and v_k is denoted as $d(v_j, v_k)$, and it is equal to the length of the shortest path between them. The *diameter* of $\mathscr{G}$, Δ, is defined as the largest distance between any two nodes of $\mathscr{G}$. The neighbor set of node v_i, $\mathscr{N}_{v_i}$, is the set including all adjacent nodes that are connected to v_i.

$$\mathscr{N}_{v_i} = \{v_j \mid (v_i, v_j) \in E\}. \tag{10}$$

The *degree* of v_i is defined as the cardinality of $\mathscr{N}_{v_i}$. Furthermore, δ-hop neighborhood of v_i, $\mathscr{N}_{v_i}^{\delta}$, is the set including nodes that are at most δ distance away from v_i.

$$\mathscr{N}_{v_i}^{\delta} = \{v_j \mid d(v_i, v_j) \leq \delta\}. \tag{11}$$

5.2 Problem Formulation

In this chapter, a discrete time problem is considered such that m agents, $V = \{v_1, ..., v_m\}$, occupy the nodes, $V^* = \{1, 2, ..., n\}$, of a connected graph $\mathscr{G}^*$, where $n \geq m$. For example, Figure 17 illustrates an example to $\mathscr{G}^*$, in which an edge implies the communication capability of two agents if they are located on the corresponding nodes.

In this setting, each agent is assumed to occupy a single node, which may correspond to monitoring a physical location such as a special access point, the coordinates of an asset, or a small region of interest. Moreover, the nodes are located far from each other, and the

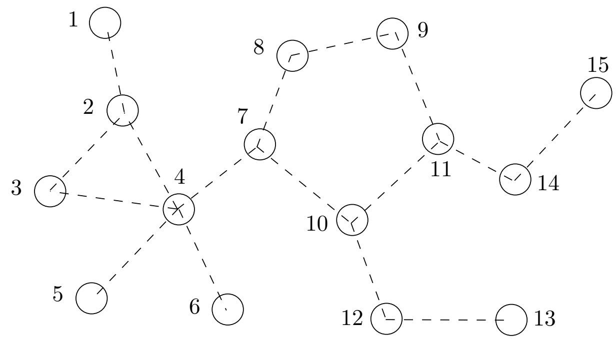
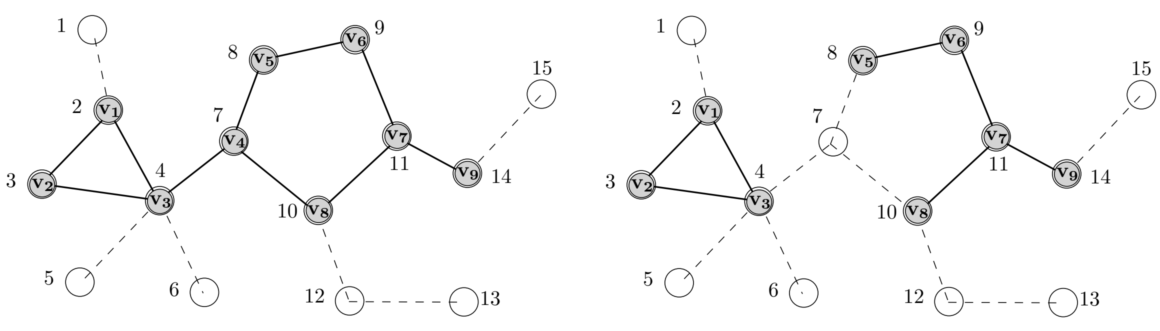


Figure 17: A connected network structure ($\mathscr{G}^*$) representing the surveillance area.

communication range of each agent is assumed larger than its monitoring range. Therefore, agents are urged to stay as close as possible to the nodes. Accordingly, the communication network of the agents becomes a subgraph of $\mathscr{G}^*$, denoted by $\mathscr{G}$. For instance, Figure 18 shows some examples for the communication networks of agents located on the nodes of a connected graph.



(a) Connected communication network

(b) Disconnected communication network

Figure 18: (a) Nine agents are located on $\mathscr{G}^*$, and $\mathscr{G}$ is a connected communication network. (b) The removal of agent 4 (v_4) causes a disconnection in $\mathscr{G}$.

Assuming that 1) agents are homogeneous with respect to network operations and 2) task co-execution can be achieved between any pair of agents, one way to recover connectivity in the face of an agent removal is to replace the removed agent by one of the remaining agents. For instance, if the removal of an agent causes a disconnection, then one of its neighbors may replace it to recover the connectivity. Similarly, if that replacement also causes a disconnection, then another replacement is required. In this manner, the replacements can be executed until a connected network is obtained. Here, any feasible solution has to recover connectivity after a finite number of replacements. Moreover, it is desired to realize the minimum number of replacements. Furthermore, it is preferable to produce a scheme that can be executed by agents making local decisions with limited information. Accordingly, the *replacement control problem* is defined as follows:

Replacement Control (RC) problem: *Given a set of agents initially forming a connected communication network, design a decentralized control scheme such that the agents maintain connectivity through as few as possible replacements in the face of an agent removal.*

5.3 Message Passing Strategy

Given a connected graph $\mathscr{G}$, where the nodes in $\mathscr{G}$ represent the agents, the RC problem finds a sequence of replacements to recover connectivity. For any solution of the RC problem, the sequence of replacements needs to end with the relocation of a noncritical node since the removal of such nodes from their previous positions does not require any further replacements.

Definition 1. *(Node Criticality) Let v_i be a node in $\mathscr{G}$ and E_i be the edges incident to v_i. v_i is critical in $\mathscr{G}$ if the graph, $\mathscr{G}' = \mathscr{G} - (v_i, E_i)$, obtained by removing v_i and E_i is disconnected; otherwise, v_i is noncritical.*

Proposition 1. *[113] Let $\mathscr{G}$ be a connected undirected graph. Suppose that each of its nodes has degree at least k. Then $\mathscr{G}$ has at least $k+1$ noncritical nodes.*

Note that a connected undirected graph always has at least 2 noncritical nodes. Thus, the goal of the RC problem is to find one of them. The following remark presents a trivial sufficient condition for a node to be noncritical in a graph.

***Remark* 1.** *Given a connected graph $\mathscr{G} = (V, E)$, let $v_i \in V$ be a leaf node such that $|\mathscr{N}_{v_i}| = 1$. Then, v_i is noncritical in $\mathscr{G}$ because any simple path involving v_i either starts or ends with v_i. Hence, its removal does not cause a disconnection between any two nodes.*

In the RC problem, a replacement is assumed to occur between a node and one of its neighbors. Therefore, the sequence of replacements can be represented as a path from the removed node to a noncritical node. Note that for a connected undirected graph $\mathscr{G}$, there always exists a path from any node in $\mathscr{G}$ to a noncritical node.

A centralized controller can provide the optimal solution to the RC problem by finding a shortest path between the removed node and a noncritical node. However, the optimal solution is obtained by assuming the availability of the overall graph structure. One of the goals of this thesis is to find a decentralized scheme that can perform "close to optimal".

Definition 2. *(Maximal simple path) Let $\mathscr{G} = (V,E)$ be a connected undirected graph, and let $\mathscr{N}_{v_i}$ denote the neighbors of $v_i \in V$. Suppose that $p = (p_0, p_1, ..., p_k)$ is a simple path of length k. Then p is a maximal simple path if $\mathscr{N}_{p_k} \subseteq \{p_0, p_1, ..., p_k\}$.*

Theorem 1. *For any connected undirected graph $\mathscr{G}$, a maximal simple path always ends with a noncritical node.*

Proof. From Definition 2, a maximal k-length simple path, $p_k = (v_0, v_1, ..., v_k)$, ends when $\mathscr{N}_k \subseteq \{v_0, v_1, ..., v_k\}$. Let v_i and v_j be any nodes in $\mathscr{G}$. In a connected graph, by definition, there exist at least one path connecting v_i and v_j. In general, there are two possibilities: (1) Assume that there exist a simple path, $\tilde{p}$, between v_i and v_j that doesn't contain v_k. Then the removal of v_k does not cause disconnection between v_i and v_j due to the existence of $\tilde{p}$. (2) Suppose all paths between v_i and v_j contain v_k. Note that any such path can be written as $p_{ij} = (v_i, ..., v_{k1}, v_k, v_{k2}, ...v_j)$ and $\{v_{k1}, v_{k2}\} \in \mathscr{N}_k$. Since $\mathscr{N}_k \subseteq \{v_0, v_1, ..., v_k\}$, there exist a subpath p_k^{sub} in p_k such that it connects v_{k1} and v_{k2}. Hence, the removal of v_k will not cause disconnection because the new path from v_i to v_j becomes $p_{ij}^{new} = (v_i, ..., v_{k1}, p_k^{sub}, v_{k2}, ...v_j)$. Note that p_{ij}^{new} does not guarantee to be a simple path due to the possibility of common nodes in p_k^{sub} and p_{ij}. However, the existence of p_{ij}^{new} show that a disconnection will not happen between v_i to v_j due to the removal of v_k. Consequently, v_k is a noncritical node. □

Corollary 1. *For any connected $\mathscr{G}$ and arbitrarily removed p_0, a sequence of replacements along a maximal simple path, $(p_0, p_1, ..., p_k)$, such that any $p_{i+1} \in \mathscr{N}_{p_i} \setminus \{p_0, p_1, ..., p_i\}$ for $i < k$, recovers the graph connectivity.*

Proof. The maximal simple path $(p_0, p_1, ..., p_k)$ is the replacement path where p_0 is any arbitrarily removed node and any p_i is replaced by p_{i+1} for $0 \leq i \leq k-1$. After the replacements are realized, the graph will have a new structure as if p_k was removed from the system. From Theorem 1, p_k is noncritical so its removal from its previous location does not cause any loss of connection in $\mathscr{G}$. □

In light of the preceding facts, a decentralized connectivity maintenance scheme called the *message passing strategy* (MPS) is introduced. Let p_0 be any arbitrary node that is removed from $\mathscr{G}$. The objective of MPS is to find a sequence of replacements, which is initiated by p_0 and ended with a noncritical node, by using only local information. In this manner, the replacements result in a reconfiguration such that the graph becomes identical to the initial graph minus a noncritical node, which is the final node in the replacement sequence.

The outline of MPS is as follows: Before the removal of p_0, first p_0 creates a message including its own node ID as $\{p_0\}$ and checks whether it is a leaf node. If it is a leaf node, then it is noncritical (from Remark 1) and its removal does not cause a disconnection. Otherwise, it selects a node, p_1, from $\mathscr{N}_{p_0} \setminus \{p_0\}$. Then, p_0 sends the message to p_1, which will replace p_0. In this respect, whenever a node, p_i, receives a message, $\{p_0, ..., p_{i-1}\}$, it appends its individual node ID to the message as $\{p_0, ..., p_{i-1}, \mathrm{p_i}\}$. Then, it sends the message to one of its neighbors from the set $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$ before replacing p_{i-1}. Eventually, the message passing process, whose pseudo-code is displayed in Algorithm 1, stops when $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\} = \emptyset$ or p_0 is a leaf node.

Algorithm 1: Message Passing Strategy (MPS)

$Input$: An arbitrary node, p_0, from $\mathscr{G}$

$Output$: Connectivity maintenance in the removal of p_0

$Assumption$: Each node shares its unique node ID with its neighbors.

initialization: $p_i \leftarrow p_0$; $\mathscr{N}_{p_i} \leftarrow \mathscr{N}_{p_0}$; $message \leftarrow (p_0)$;
if $|\mathscr{N}_{p_0}| = 1$
    no replacements required;
else
    while $\mathscr{N}_{p_i} \setminus message \neq \emptyset$
        $p_{i+1} \leftarrow v \quad s.t. \quad v \in \mathscr{N}_{p_i} \setminus message$;
        p_i sends $message$ to p_{i+1};
        if $i = 0$
            p_i is removed;
        else
            p_i replaces p_{i-1};
        end if
        $p_i \leftarrow p_{i+1}$; $\mathscr{N}_{p_i} \leftarrow \mathscr{N}_{p_{i+1}}$;
        $message \leftarrow (message, p_i)$;
    end while
    p_{i+1} replaces p_i;
end if

Proposition 2. *MPS always stops, and it stops at a noncritical node.*

Proof. Let $p_0 \in V$ be any arbitrary node that will be removed from $\mathscr{G}$. If p_0 is a leaf node, MPS stops at p_0. Otherwise, p_0 generates a message as $\{p_0\}$, and the message is modified as $\{p_0, ..., p_i\}$ whenever it is received by $p_i \in V$. Let $N+1$ be the total number of nodes in $\mathscr{G}$. In this respect, as $i \to N$, $\{p_0, ..., p_i\} \to \{p_0, ..., p_N\} = V$. Eventually, there exist an instant $k = i \leq N$, at which $\mathscr{N}_{p_k} \subseteq \{p_0, ..., p_k\}$. From Theorem 1, p_k is a noncritical node because it satisfies $\mathscr{N}_{p_k} \setminus \{p_0, ..., p_k\} = \emptyset$. Consequently, MPS always stops at a noncritical node. □

Corollary 2. *The message obtained when MPS stops is a set of ordered nodes, which corresponds to either a single leaf node or a maximal simple path. Hence, MPS guarantees connectivity maintenance in the removal of any arbitrary node from* $\mathscr{G} = (V, E)$.

Proof. The message obtained from MPS is either $\{p_0\}$ or $\{p_0, ..., p_i, ..., p_k\}$. If it is $\{p_0\}$, then $|\mathscr{N}_{p_0}| = 1$ implying that p_0 is a leaf node, and the connectivity maintenance is an immediate result. If the message is $\{p_0, ..., p_i, ..., p_k\}$, it involves consecutive pairs of nodes, $(p_i, p_{i+1}) \in E$, thus the message is always a path in $\mathscr{G}$. Additionally, the message never involves repeated nodes because each p_i selects p_{i+1} from $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$. Thus, the path is always simple. Finally, MPS stops whenever $\mathscr{N}_{p_k} \setminus \{p_0, ..., p_i, ..., p_k\} = \emptyset$. From Definition 2, the ordered nodes in the message is a maximal simple path. From Corollary 1, the replacements based on a maximal simple path always guarantee connectivity recovery because the final graph is reconfigured as the initial graph without p_k. □

An illustration for MPS is displayed in Figure 19, where there is initially a connected graph with 7 nodes. As it is seen in Figure 19(b), the removal of v_0 will create a disconnection in the graph. If each node runs MPS, then a possible replacement path is $\{v_0, v_2, v_4\}$ such that v_2 replaces v_0, and v_4 replaces v_2. Note that $\{v_0, v_2, v_4\}$ is not the only replacement path, e.g. $\{v_0, v_1, v_5\}$. Consequently, the system reconfigures itself to maintain connectivity, and the resulting graph becomes the initial graph without a noncritical node (e.g. v_4).

5.3.1 Performance of MPS

Given a networked system, reactive schemes for connectivity maintenance result in some changes in the graph topology. While maintaining the graph connectivity, an important aspect is to avoid causing significant changes in the graph properties, such as the total number of edges or the maximum node degree. Note that the total number of edges and the maximum node degree can be directly related to the overall and individual communication

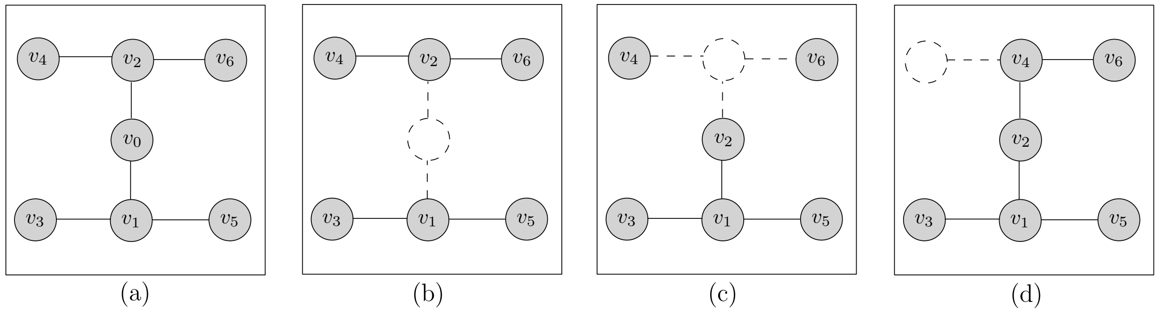


Figure 19: An illustration for MPS. (a) Initially connected graph. (b) v_0 will leave the system. Since it is not a leaf node, it creates a *message* as $\{v_0\}$ and selects a neighbor from $\mathscr{N}_{v_0} \setminus \{v_0\} = \{v_1, v_2\}$ for its replacement. (c) Let v_2 receive the *message* and modify it as $\{v_0, v_2\}$. Then, it selects a neighbor from $\mathscr{N}_{v_2} \setminus \{v_0, v_2\} = \{v_4, v_6\}$ for its replacement. (d) Let v_4 receive the *message* and modify it as $\{v_0, v_2, v_4\}$. It attempts to select a neighbor from $\mathscr{N}_{v_4} \setminus \{v_0, v_2, v_4\} = \emptyset$ for its replacement. Since $\mathscr{N}_{v_4} \setminus \{v_0, v_2, v_4\} = \emptyset$, v_4 cannot send the message to any node and the algorithm stops.

loads, whose increase is not desirable for a network system containing agents with limited energy capacity.

Proposition 3. *A sequence of replacements along a maximal simple path,* $(p_0, p_1, ..., p_k)$, *on* $\mathscr{G}$*, such that any* $p_{i+1} \in \mathscr{N}_{p_i} \setminus \{p_0, p_1, ..., p_i\}$ *for* $i < k$*, guarantees no increase in the total number of edges and the maximum node degree in the presence of any arbitrary node removal.*

Proof. Let $p = (p_0, p_1, ..., p_k)$ be the replacement path, and let $\mathscr{G}'$ be the new graph structure after the replacements. Then, this corollary is proven in two parts: (1) In the removal of an arbitrary node, p_0, p results in $\mathscr{G}'$, which corresponds to the removal of p_k and its adjacent edges from $\mathscr{G}$. As a result, the total number of edges decreases as the agents are removed. (2) Let p_0 in p be the agent that has the maximum degree d_{max} in $\mathscr{G}$. If p_0 is removed, then p_1 replaces p_0. Now, if $k = 1$, then p_1 is the noncritical node that will not be replaced. As a consequence, the degree of p_1 becomes $d_{max} - 1$ after the replacement. If $k \neq 1$, then p_1 will be replaced by p_2. Hence, the degree of p_1 becomes d_{max} after the replacements. In both cases, p_1 becomes the node with the maximum degree in $\mathscr{G}'$ after replacing p_0. Finally, in the removal of an arbitrary node, which does not correspond to the maximum degree node $\tilde{v}$, either no replacements occur in the neighborhood of $\tilde{v}$, or the replacements in the neighborhood of $\tilde{v}$ may cause at most one reduction in d_{max}. As a result, the maximum node degree in $\mathscr{G}'$ becomes either d_{max} or $d_{max}-1$. □

The optimal solution satisfying the minimum number of replacements for the RC problem can be obtained by a centralized controller by finding the shortest path between the removed node and a noncritical node on the graph. Note that such a centralized controller requires the complete information about the graph. The objective of MPS is to solve the RC problem only by using some local and partial information. Due to utilizing limited information, MPS may not necessarily guarantee the optimal solution for any graphs.

Proposition 4. *In any undirected connected graph, $\mathscr{G}=(V,E)$, the maximum number of replacements that can occur via MPS is $|V|-1$.*

Proof. From Corollary 2, MPS results in a message that is the sequence of replacements represented as a maximal simple path, p. Let $|p|>|V|$, then at least one node appears multiple times in p, thus p is not simple. This is a contradiction, hence $|p|\leq|V|$, which implies an upper bound for the number of replacements as $|V|-1$. □

A tighter bound for Proposition 4 can be obtained in a tree graph, in which any two nodes are connected by exactly one simple path.

Proposition 5. *In a tree graph, $\mathscr{G}=(V,E)$, the maximum number of replacements that can occur via MPS is $\Delta-1$, where Δ is the diameter of $\mathscr{G}$.*

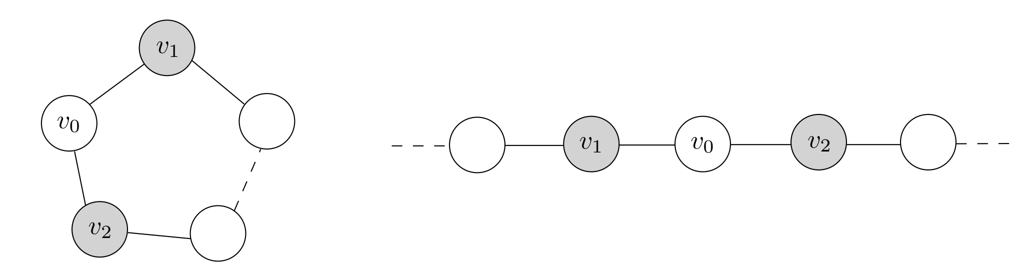
Proof. In a tree graph, a noncritical node is always a leaf node, and a critical node always has a degree of 2. Thus, the diameter of a tree graph corresponds to the length of the longest maximal simple path. Let $\{p_0,p_1,....,p_{\Delta-1},p_\Delta\}$ denote to the longest maximal simple path, where p_0 and p_Δ are leaf nodes and the nodes in between are critical. If p_0 is the removed node, then MPS does not initiate replacements. If p_1 is the removed node, then the maximum number of replacements based on MPS may occur along the sequence $\{p_1, ..., p_{\Delta-1}, p_\Delta\}$ resulting in $\Delta - 1$ replacements. □

A biconnected graph is a connected graph that does not have any critical nodes, i.e. the removal of any single node cannot disconnect a biconnected graph.

***Remark* 2.** *In biconnected graphs, MPS cannot achieve optimal solution for connectivity maintenance. Using MPS, the removal of an arbitrary node initiates some replacements even though none is necessary.*

Note that MPS may not always result in the minimum number of replacements in agent removal. For instance, if the removed agent is not a leaf node, but noncritical, MPS still initiates a sequence of replacements as depicted in Remark 2. On the contrary, a centralized perspective certainly identifies the criticality of a node due to the availability of the overall graph structure.

***Remark* 3.** *The node criticality cannot be always determined with local information. As shown in Figure 20, let $\mathcal{G}_1$ and $\mathcal{G}_2$ be an infinite cycle and infinite path graphs, respectively. Suppose that any node in $\mathcal{G}$ knows its δ-hop neighborhood. Let v_0 be the removed node. In Figure 20, v_0 is noncritical in $\mathcal{G}_1$, but critical in $\mathcal{G}_2$. Note that for any finite δ, v_0 has the same neighborhood in $\mathcal{G}_1$ and $\mathcal{G}_2$, hence it can not differentiate its criticality by just looking at its δ-neighborhood.*



(a) Infinite cycle graph ($\mathcal{G}_1$) (b) Infinite path graph ($\mathcal{G}_2$)

Figure 20: Examples to graphs with infinite nodes.

Since the criticality cannot be always determined locally, MPS may sometimes initiate a sequence of replacements when a noncritical node is removed. This occurs due to the limitation of local information in the computations. However, it is important to emphasize that, for any undirected connected graph, connectivity maintenance in the presence of any node removal is guaranteed by MPS by using only some local information. In this respect, for all $i \geq 0$, p_i selects the node, p_{i+1}, which will replace itself, from $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$. Here, a question arises as which node from $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$ should be selected to increase the efficiency of MPS. For instance, a random selection scheme requires very little information to be shared among nodes, or a node selection based on the minimum degree may capture the leaf node neighbors. Consequently, as the information possessed by a node and shared in the neighborhood increases, the solution is expected to approach the optimal solution.

5.4 Message Passing Strategy with Minimum Degree

This section introduces a heuristic algorithm for the node selection in MPS. Note that any node using MPS randomly selects a node from its neighborhood to forward the message. Instead of a random selection, d_{min}-MPS suggests to select a neighbor that has the minimum degree in the neighborhood. Let p_i be the node that will replace the message to one of its neighbors in $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$, and let $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$ contain m nodes. Suppose that v_j is the node with the minimum degree in $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$. If p_i executes d_{min}-MPS, then it selects p_{i+1} as v_j with probability 1. On the other hand, p_i executing MPS selects v_j with probability $1/m$. Accordingly, in cases where there is an immediate leaf node neighbor, random selection via MPS causes a longer replacement sequence with probability $(m-1)/m$.

5.5 *Message Passing Strategy with δ-Criticality*

This section introduces a variant of MPS, which uses δ-criticality information for each node. Here, the δ-criticality is defined as Definition 3.

Definition 3. *(δ-criticality) A node, v_i, is called δ-critical if the subgraph, induced by the δ-neighborhood of v_i, is disconnected by the removal of v_i; otherwise, v_i is δ-noncritical.*

Let $\mathscr{G} = (V, E)$ be a connected graph, and let $v_i \in V$ be δ-noncritical. Suppose that a simple path, p^{nm}, in $\mathscr{G}$ connects any arbitrary two nodes $v_n, v_m \in V$ and includes v_i as an intermediate node. In p^{nm}, v_i appears between two of its neighbors. In the removal of v_i, there exist another path, p^i, consisting of some nodes within δ hops of v_i since v_i is δ-noncritical by definition. Hence, the removal of v_i does not cause a disconnection between v_n and v_m because v_i can be replaced by p^i.

***Remark* 4.** *A δ-noncritical node is always noncritical in $\mathscr{G}$.*

In light of Remark 4, δ-criticality is used in MPS as in Algorithm 2. In this respect, each node knows whether itself and immediate neighbors are δ-noncritical. Note that a node can obtain its own δ-criticality by sending a query to each node in its delta-neighborhood to understand who is linked to whom within the delta-neighborhood. In δ-MPS, whenever a node, p_i, receives a message, it appends its own individual ID to the message likewise MPS. Then, it selects a neighbor from the candidate set, $\mathscr{N}_{p_i} \setminus \{p_0, ..., p_i\}$, based on δ-criticality. In the case, where the candidate set does not contain a δ-noncritical node, p_i selects a random node from the candidate set.

It has been shown in Remark 4 that a δ-noncritical node is globally noncritical in $\mathscr{G}$. Now, a question arises as when a δ-critical node assures global criticality. In this respect, Proposition 6 presents a sufficient condition that guarantees global node criticality by relating δ to a graph structure.

Definition 4. *A chordless cycle in $\mathscr{G}$ is a cycle such that no two nodes of the cycle are connected by an edge that does not itself belong to the cycle.*

Proposition 6. *Let c_{max} be the length of the longest chordless cycle in a graph. Then a δ-critical node is globally critical in $\mathscr{G}$, if $\delta \geq \frac{c_{max}}{2}$.*

Proof. Assume that there is at least one chordless cycle in $\mathscr{G}$. Let v be a noncritical node in $\mathscr{G}$, and let $\mathscr{N}^{\delta}$ be the δ-neighborhood of v for some $\delta \geq \frac{c_{max}}{2}$, where c_{max} is the length of the longest chordless cycle in $\mathscr{G}$. Suppose that v is a δ-critical node, then the graph, $\mathscr{G}'$, induced by the nodes in $\mathscr{N}^{\delta}$ is disconnected. Now, since v is noncritical, there must exist a shortest path between a pair of nodes $(v_n, v_m) \in \mathscr{N}^{\delta}$, which is not connected in $\mathscr{G}'$ but connected in $\mathscr{G} - \{v\}$. Thus, there exist a shortest path, (v_n, p^*, v_m), where no elements on p^* is connected to v (i.e. no elements on p^* is in $\mathscr{N}^{\delta}$). Note that (v_n, p^*, v_m, v, v_n) is a chordless cycle and its length, c, cannot be larger than c_{max}. However, v does not know the existence of such a path, so $c > 2\delta$, which is a contradiction because $2\delta \geq c_{max} \geq c$.

Now, assume that there is no chordless cycle in $\mathscr{G}$, so $c_{max} = 0$. Then, $\mathscr{G}$ is a tree graph, where each noncritical node is a leaf node. As such, a node can determine its criticality by checking its degree, d, where $d > 1$ implies a critical node. Consequently, for any $\delta \geq 0$, a δ-critical node is globally critical in $\mathscr{G}$. □

Corollary 3. *If δ is selected based on Proposition 6, then the replacement sequence generated via δ-MPS involves only one noncritical node, which is the last node on the replacement sequence.*

Proof. Based on Algorithm 2, a message travels from a δ-critical node to a neighboring δ-critical node until finding a δ-noncritical node. In the case of $\delta \geq \frac{c_{max}}{2}$, Proposition 6 shows that a δ-critical node is globally critical. Hence, the replacement sequence generated via δ-MPS contains only one noncritical node, which is the last node on the sequence. □

Algorithm 2: δ-MPS

Input : An arbitrary node, p_0, from $\mathscr{G}$

Output : Connectivity maintenance in the removal of p_0

Assumption : Each node shares both its ID and δ-criticality with its neighbors.

initialization: $p_i \leftarrow p_0$; $\mathscr{N}_{p_i} \leftarrow \mathscr{N}_{p_0}$; $message \leftarrow (p_0)$;
if p_0 is δ-noncritical
    no replacements required;
else
    while $\mathscr{N}_{p_i} \setminus message \neq \emptyset$
        if any $v \in \mathscr{N}_{p_i} \setminus message$ is δ-noncritical;
            $p_{i+1} \leftarrow v$ s.t. v is one of the δ-noncritical nodes;
            p_i sends *message* to p_{i+1};
            p_i replaces p_{i-1}; p_{i+1} replaces p_i;
            break;
        else
            $p_{i+1} \leftarrow v$ s.t. v is randomly selected from $\mathscr{N}_{p_i} \setminus message$;
            p_i sends *message* to p_{i+1};
            if $i = 0$
                p_i is removed;
            else
                p_i replaces p_{i-1};
            end if
            $p_i \leftarrow p_{i+1}$; $\mathscr{N}_{p_i} \leftarrow \mathscr{N}_{p_{i+1}}$;
            $message \leftarrow (message, p_i)$;
        end if
    end while
end if

Note that δ-MPS for $\delta = 0$ is equivalent to MPS introduced in the previous section. In this respect, a node uses its zero-neighborhood, which is only itself, to identify is criticality. Thus, a leaf node in a graph is 0-noncritical whereas any other nodes are 0-critical. If the removed node is not a leaf node (i.e. 0-critical), a sequence of replacements is initiated. Consequently, the same steps are followed in MPS and 0-MPS.

***Remark* 5.** *If δ is selected in accordance with Proposition 6, a δ-critical node is always a critical node. Hence, when a δ-critical node is removed, some replacements are necessary for connectivity maintenance. Furthermore, Corrollary 3 indicates that unnecessary message forwarding does not occur since the message either visits critical nodes or stops at a noncritical node.*

In executing δ-MPS, if δ is selected such that any δ-critical node is globally critical, then only globally critical nodes generate or forward replacement request messages. Nonetheless, this is not sufficient to ensure the minimum number of replacements to recover connectivity. For instance, if there is no δ-noncritical node in the immediate neighborhood of a node v, then v selects a neighbor randomly for its replacement. As such, δ-MPS does not always guarantee a shortest path to a noncritical node due to such randomizations.

In Figure 21, two cases are illustrated to discuss the optimality gap of δ-MPS. For the tree graph shown in Figure 21(a), Proposition 6 suggests that if $\delta \geq 0$, then δ-criticality implies global criticality. Assume that v_0 is removed from the graph, and let δ be 1. From the perspective of v_0, selecting v_1 or v_2 is indifferent because v_0 can only see $\mathscr{G}_{\mathscr{N}^1}$, which contains the highlighted nodes in Figure 21(a). Hence, 1-MPS can result in the shortest maximal simple path as (v_0, v_1, v_3) or the longest maximal simple path as $(v_0, v_2, ..., v_{n-1}, v_n)$. In Figure 21(b), assume that the longest chordless cycle is a triangle and $\delta = 2$ in accordance with Proposition 6. Like the previous case, v_0 is removed from the graph and selecting v_1 or v_4 is indifferent for v_0 (since v_1 and v_4 are both 2-critical.) Note that selecting v_1 leads to a replacement sequence as (v_0, v_1, v_2) whereas selecting v_4 causes a much longer route as $(v_0, v_4, ..., v_{n-1}, v_n)$.

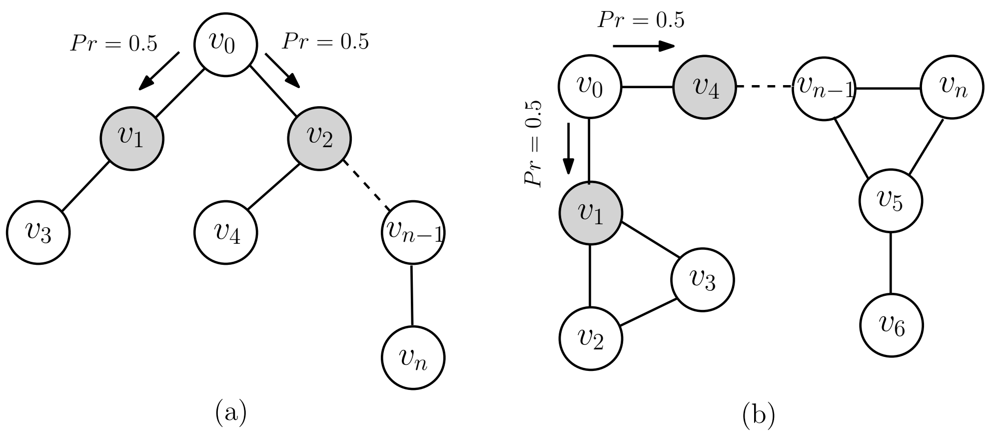


Figure 21: In the case of v_0 is removed, (a) 1-MPS can generate the shortest sequence as (v_0, v_1, v_3) or the longest sequence as $(v_0, v_2, ..., v_{n-1}, v_n)$, (b) 2-MPS can generate the shortest sequence as (v_0, v_1, v_2) or the longest sequence as $(v_0, v_4, ..., v_{n-1}, v_n)$.

Consequently, the optimality of δ-MPS depends on the lengths of the maximal simple paths between critical and noncritical nodes. On any graph $\mathscr{G}$, the optimality of δ-MPS is quantified as its maximum possible deviation (in terms of the number of replacements) from the optimal centralized solution in any node removal.

***Remark* 6.** *For any connected $\mathscr{G} = (V, E)$, let $V_c \subset V$ be the set of critical nodes. For each $v_i \in V_c$, let $l_{max}(v_i)$ and $l_{min}(v_i)$ be the lengths of the shortest and the longest maximal simple paths from v_i to any noncritical node, respectively. If δ is selected based on Proposition 6, then δ-MPS deviates form the optimal centralized solution by at most* $\max_{v_i \in V_c}(l_{max}(v_i) - l_{min}(v_i))$.

5.6 Results

The RC problem introduced in the preceding section seeks a decentralized strategy that results in as few as possible replacements to recover connectivity in the face of a node removal. Note that minimum number of replacements can be obtained by an optimal centralized solution. To this end, the centralized optimal scheme identifies the closest noncritical node to the removed node, and then it executes the replacements along the shortest path between them. However, a centralized solution requires the availability of overall graph topology to each agent, which becomes inefficient in large scale systems. This sections presents some experiments to compare the optimality of δ-MPS, as well as MPS and $d_{min} - MPS$, with respect to the optimal centralized solution. Note that the optimal centralized solution is selected as a benchmark due to resulting in minimum number of replacements.

All simulations involve connected random geometric graphs consisting of 50 nodes. The nodes are randomly and uniformly distributed on an area having a size of 75×75 unit^2. Different graph properties are obtained by varying the radius of connection, R, which is assumed to be the same for each agent. For any given R, 200 connected graphs are generated. In each simulation, a randomly selected node is removed from the graph, and the RC problem is solved via a centralized controller, then via d_{min}-MPS, then via MPS, and then via δ-MPS for $\delta = 1, 2$. The strategies are compared based on the resulting costs that is selected as the total number of node replacements. As such, Table 2 illustrates the mean costs of the strategies to solve the RC problem for various graph structures. For $R = 11, 15, 19$, the distributions of average degree and graph diameter, as well as the costs, are presented in Figures 22, 23, and 24.

Table 2: Based on 200 simulations, mean cost* of various strategies to recover connectivity in graphs with 50 nodes

Graph Properties			Mean Cost*				
R (unit)	Mean $\bar{d}$**	Mean Δ	Centralized	MPS	d_{min}-MPS	1-MPS	2-MPS
11	2.65	20.97	0.725	4.570	6.330	0.895	0.835
12	3.02	18.83	0.500	6.775	8.050	0.725	0.610
13	4.11	14.94	0.225	8.720	9.585	0.410	0.305
14	4.56	13.29	0.170	9.770	11.880	0.330	0.240
15	5.14	11.64	0.120	12.085	14.930	0.240	0.205
16	5.79	10.13	0.075	14.720	16.350	0.205	0.135
17	6.47	9.02	0.040	16.935	19.545	0.165	0.075
18	7.10	8.17	0.030	19.165	23.985	0.120	0.065
19	7.80	7.49	0.025	20.340	24.125	0.105	0.050
20	8.50	6.85	0.010	23.790	29.690	0.050	0.030

* number of agent replacements

** average node degree

For graphs consisting of equal number of nodes, as the radius of connection increases, it is more likely to have nodes with more connections and graphs with smaller diameters. Such graphs are in general more robust to agent removal, and the average number of replacements needed to recover connectivity is reduced. This is illustrated in Figures 22, 23 and 24, where the cost distribution of the centralized solution is more skewed towards 0 as the diameter decreases.

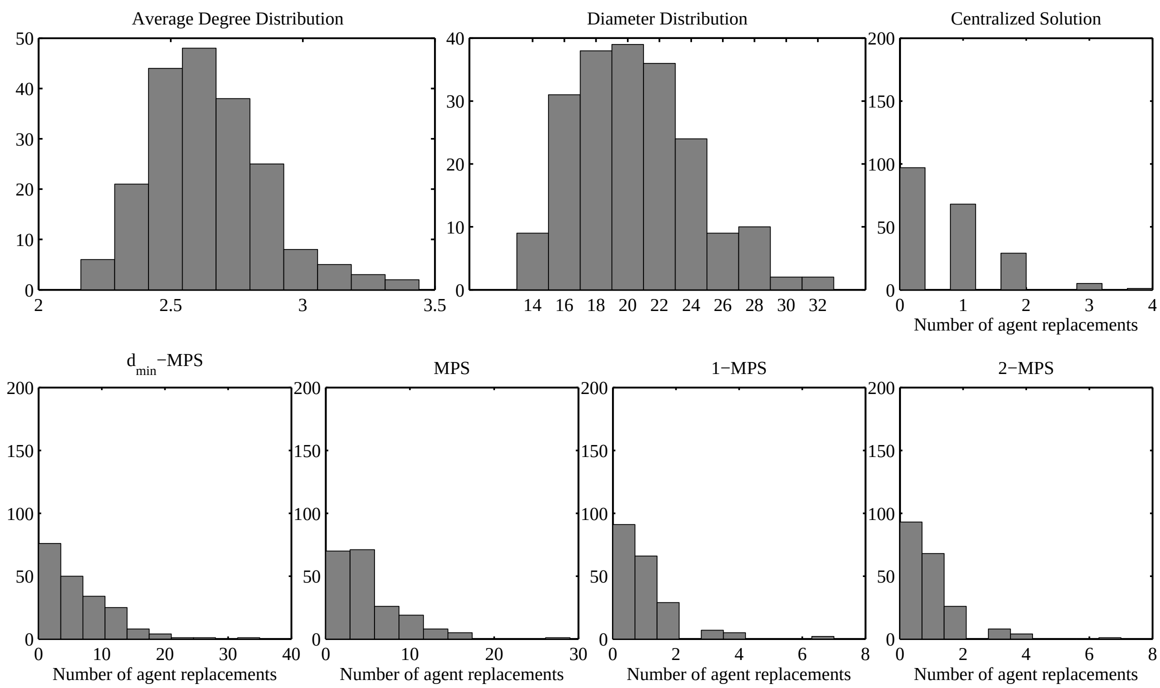


Figure 22: For $R = 11$, the distributions of graph properties and mean costs for various strategies to solve the RC problem.

Simulations indicate an interesting result when d_{min}-MPS and MPS are compared with each other. Even though d_{min}-MPS is using more information than MPS (i.e. each agent running d_{min}-MPS shares its degree and ID in its neighborhood whereas each agent using MPS shares only its ID), it performs slightly worse than MPS. This result is justifies as follows: suppose that the removed agent is p_0 and the closest noncritical agent to p_0 is p_k, whose distance to p_0 is k. Let $\mathbf{Pr}_{\mathrm{MPS}}$ and $\mathbf{Pr}_{\mathrm{d_{min}-MPS}}$ denote the probabilities of finding a k-length replacement path as $(p_0, p_1, ..., p_k)$ via MPS and d_{min}-MPS, respectively. As such,

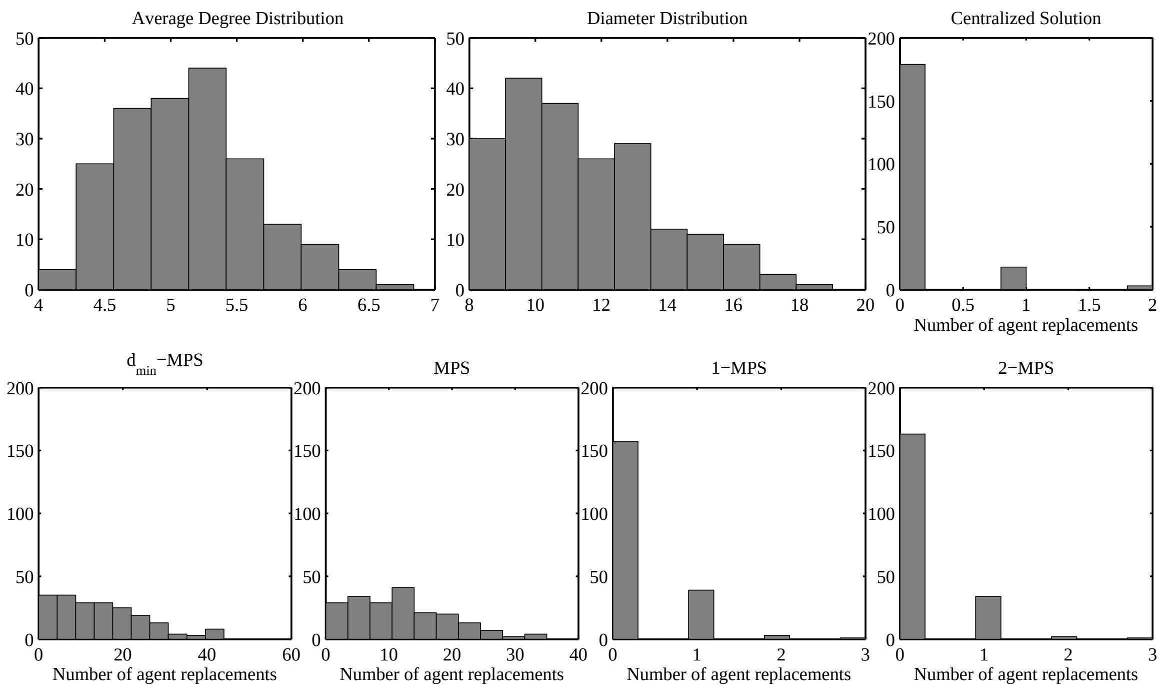


Figure 23: For $R = 15$, the distributions of graph properties and mean costs for various strategies to solve the RC problem.

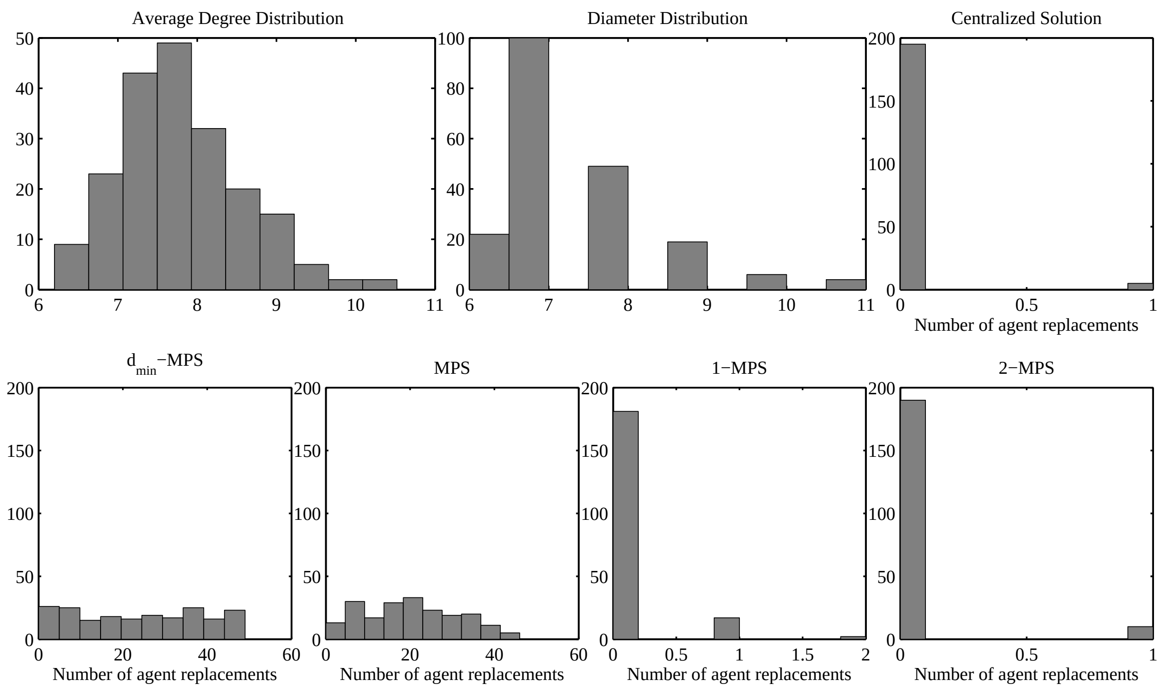


Figure 24: For $R = 19$, the distributions of graph properties and mean costs for various strategies to solve the RC problem.

$$0 < \prod_{0<i<k-1} \frac{1}{|\mathcal{N}_{p_i}|} \leq \mathbf{Pr}_{\text{MPS}} \leq \frac{1}{|\mathcal{N}_{\text{p}_0}|}, \tag{12}$$

$$0 \leq \mathbf{Pr}_{\text{d}_{\text{min}}-\text{MPS}} \leq 1, \tag{13}$$

where $\mathcal{N}_{p_i}$ is the neighbor set of agent p_i. For example, if p_k is a leaf node (i.e. $k = 1$), the randomized nature of MPS causes to select p_k with a probability of $1/|\mathcal{N}_{p_0}|$ whereas d_{min}-MPS certainly selects p_k. Nonetheless, there can be cases where $\mathbf{Pr}_{\text{d}_{\text{min}}-\text{MPS}}$ becomes zero due to the strict selection of minimum degree neighbor, while $\mathbf{Pr}_{\text{MPS}}$ is always bounded away from zero. For instance, let p_0 in Figure 25 be the removed node. Then, the gray nodes are the closest noncritical nodes to p_0. Accordingly, p_0 running d_{min}-MPS never creates a message that ends with one of the gray nodes since p_0 tends to send the replacement message to the node with a degree of 2. On the contrary, p_0 utilizing MPS has a probability of 0.5 to create a message ending with one of the gray nodes.

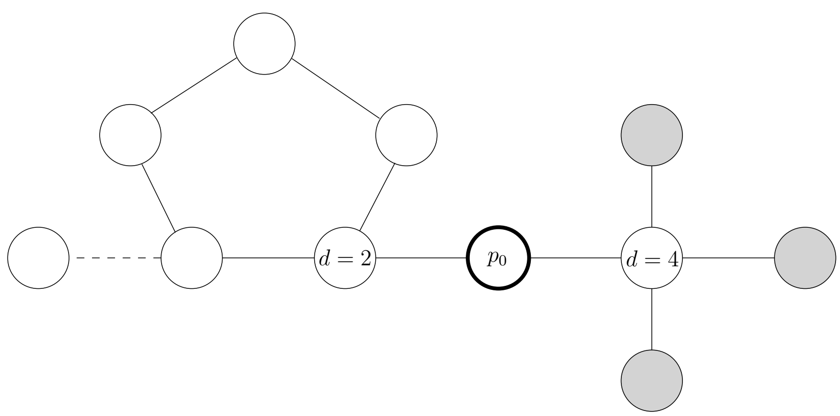


Figure 25: In the face of p_0 removal, d_{min}-MPS never stops at one of the gray nodes (i.e. closest noncritical agents) whereas MPS has a finite probability to stop at one of them.

In MPS and d_{min}-MPS, a replacement sequence initiates regardless of criticality unless the removed node is a leaf node. Such schemes become inefficient in well-connected graphs because they cause a large number of unnecessary replacements. As it is seen from Table 2, the mean costs of MPS and d_{min}-MPS dramatically increase as the radius of connection increases. On the other hand, δ-MPS incorporates the δ-criticality of nodes in replacement decisions. The results show that utilizing δ-criticality, even for δ=1, significantly improves the performance of MPS. Moreover, the solutions driven by δ-MPS are observed close to optimal for sparser graphs as indicated in Table 2. Finally, the results show that while utilizing 1-MPS over MPS results in a significant performance improvement; using 2-MPS over 1-MPS does not improve the optimality significantly. Hence, the simulation results suggest that 1-MPS is a very efficient and local approach to solve the RC problem.

CHAPTER VI

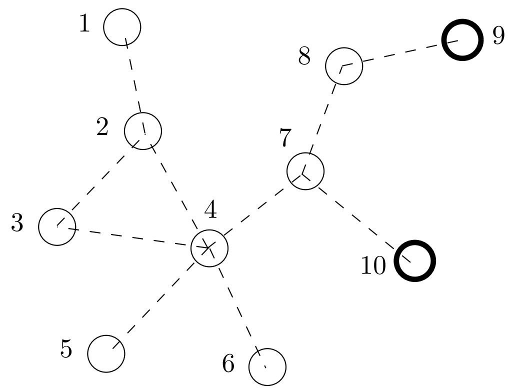
DECENTRALIZED CONNECTIVITY MAINTENANCE FOR MULTI-AGENT SURVEILLANCE

6.1 Problem Formulation

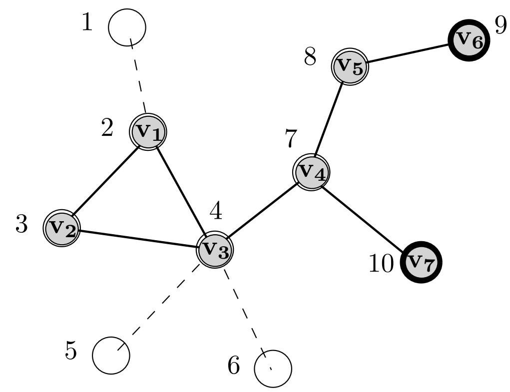
Previously, the surveillance environment has been represented as a connected graph $\mathscr{G}^*$, whose nodes correspond to some areas of interest. In this representation, any two nodes are linked by an edge if the distance between them is smaller than the communication range of the agents. For a given $\mathscr{G}^*$, let some nodes have more priority than the others. In this setting, an MAS having a connected communication network as well as occupying the high priority nodes becomes more preferable than an MAS having a connected communication network but not occupying the high priority nodes.

Suppose that some agents are located on the nodes of $\mathscr{G}^*$ as in Figure 26b, where nodes 9 and 10 have high priority. In this setting, the agents' communication network, $\mathscr{G}$, is a connected subgraph of $\mathscr{G}^*$ shown in Figure 26a. Note that $\mathscr{G}$ becomes disconnected in the face of a critical agent removal such as the removal of v_4 as shown in Figure 26c. Here, if each agent utilizes δ-MPS, the connectivity of $\mathscr{G}$ can be recovered by a single replacement as in Figure 26d, where v_7 replaces v_4. However, v_7 is located on a high priority node, and it should not accept any replacement request.

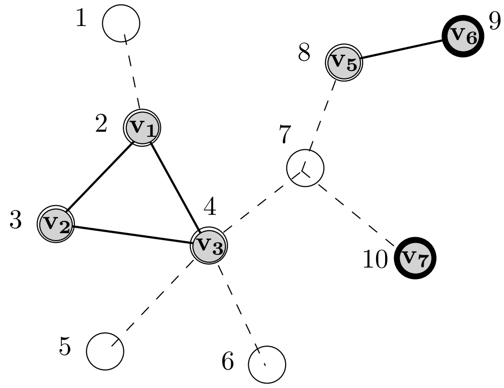
In the presence of high priority nodes in $\mathscr{G}^*$, the connectivity of $\mathscr{G}$ should be recovered by not only pursuing as few as possible replacements but also ensuring the occupancy of high priority nodes. For example, one feasible sequence of replacements is illustrated in Figure 26e, where v_4 is replaced by v_3, v_3 is replaced by v_2, and the high priority nodes 9 and 10 are preserved by v_7 and v_6.



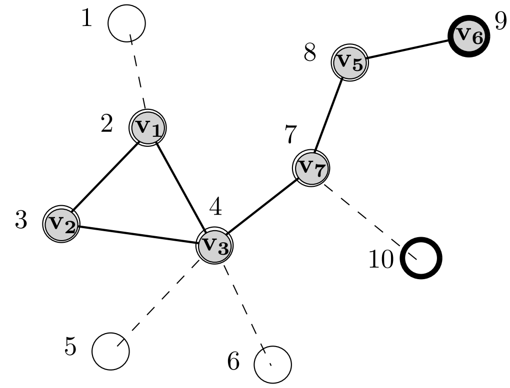
(a) Nodes 9 and 10 have priority on $\mathscr{G}^*$.



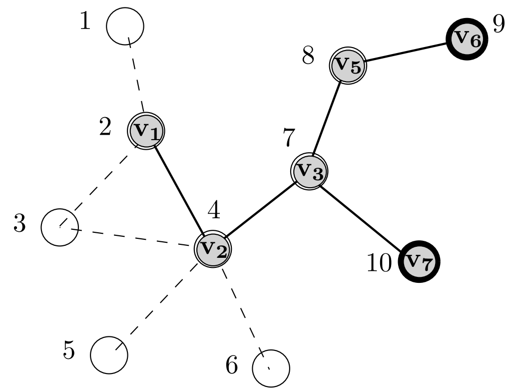
(b) 7 agents are located on the nodes of $\mathscr{G}^*$.



(c) Removal of v_4 causes a disconnected $\mathscr{G}$.



(d) An infeasible $\mathscr{G}$ by utilizing δ-MPS.



(e) A feasible $\mathscr{G}$ after the removal of v_4.

Figure 26: Motivation for the constrained MPS.

Based on the presented motivation, the goal of this chapter is to answer the following research question.

***Research Question* 3.** *In the presence of some nodes that require agent occupancy, how is it possible to recover connectivity as few as possible replacements?*

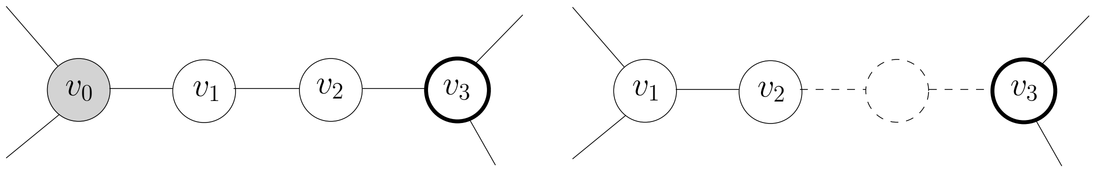
6.2 Message Passing Strategy with Constraints

Like the previous sections, the nodes of $\mathscr{G}$ represent the agents. Different than the previous sections, the nodes of $\mathscr{G}$ may have constraints based on their locations at $\mathscr{G}^*$. In this thesis, a node constraint is defined as follows:

Definition 5. *(Node Constraint) A node in $\mathscr{G}$ is constrained if it can not move from its current position unless it is removed, otherwise it is unconstrained.*

In order to ensure a sequence of replacements among the unconstrained nodes, a node should not move immediately after it sends a replacement request message as in δ-MPS. Let v_0 be the removed node, and let v_3 be the constrained node in Figure 27a. Suppose that v_0 initiates the replacements by sending a replacement request message to v_1 and it leaves the system immediately. Then, v_1 sends the replacement request message to v_2 according to the δ-MPS and replaces v_0. Similarly, v_2 sends the replacement request message to v_3, which is its only neighbor that has not received the message earlier. Accordingly, v_2 replaces v_1 by assuming that v_3 will eventually replace itself. However, v_3 is a constrained node that will not move from its current position. As a result, the replacements result in a reconfiguration as in Figure 27b, which illustrates a graph disconnection. In order to avoid such cases, the feasibility of a replacement sequence should be verified before the replacements are actually executed.

In constrained δ-MPS for $\delta \geq 0$, each node knows its own constraint in addition to its individual ID and δ-criticality. Moreover, it can share these information with its immediate neighbors. Accordingly, each node attempts to forward the replacement request message to



(a) Before replacements. (b) After replacements.

Figure 27: An example where δ-MPS does not guarantee connectivity maintenance in the presence of a constrained node.

one of its unconstrained δ-noncritical neighbors from $\mathcal{N}_{p_i} \setminus \{p_0, ..., p_i\}$. If $\mathcal{N}_{p_i} \setminus \{p_0, ..., p_i\}$ contains only constrained nodes, the sequence of selected replacements becomes infeasible. Thus, the removed node should be informed about the infeasibility of the sequence, and it should initiate a new replacement sequence.

Like the replacement request message, let replacement approval and replacement refusal messages be passed among the nodes. In constrained δ-MPS, whenever a node, p_0, is removed, it creates a replacement request message by including its own individual ID, picks an unconstrained neighbor p_1 from $\mathcal{N}_{p_0}$ in order to send the message, and records the ID of p_1. Similar to δ-MPS, if an unconstrained δ-critical node, p_i, receives a replacement request message, it appends its own individual ID to the message and sends it to one of its unconstrained neighbors from the candidate set, $\mathcal{N}_{p_i} \setminus \{p_0, ..., p_i\}$. Accordingly, if the message passing stops at p_n, which is a δ-noncritical unconstrained node, p_n creates a replacement approval message by copying the replacement request message it has received. Then, it sends the approval message to the last node on the sequence, i.e. p_{n-1}. Note that the replacement request message is an ordered sequence of nodes. Whenever a node p_j receives an approval message from p_{j+1}, it sends the message to p_{j-1} and replaces p_{j-1}. As a result, a replacement proceeds only if a node receives a replacement request and a replacement approval messages. Moreover, p_0 leaves the graph whenever the approval message reaches it.

If a replacement request message reaches a constrained node p_{i+1} (i.e. $\mathcal{N}_{p_i} \setminus \{p_0, ..., p_i\}$ contains only constrained nodes), p_{i+1} creates a replacement refusal message by replicating the replacement request message received from p_i and sends it to p_i. Whenever a node p_j receives a refusal message from p_{j+1}, it sends the message to p_{j-1}. Consequently, when the refusal message reaches p_0, it creates a new replacement request message by selecting another neighbor from $\mathcal{N}_{p_0} \setminus p_1$, where p_1 was the neighbor of p_0 it selected in the previous replacement sequence. Note that a major assumption in the execution of constrained MPS is that the message forwarding is much faster than the movement of agents.

Definition 6. *(Connected component) A connected component is a maximal connected subgraph of $\mathcal{G}$.*

***Remark* 7.** *Each node belongs to exactly one connected component of $\mathcal{G}$.*

Proposition 7. *Let v' be any critical node in a connected graph $\mathcal{G}$. If the removal of v' partitions $\mathcal{G}$ into n connected components, then $\mathcal{G}$ has at least n noncritical nodes.*

Proof. Let $\mathcal{G}_1$,...,$\mathcal{G}_n$ denote the connected components of $\mathcal{G}$ after the removal of v_i. Note that for each $\mathcal{G}_i$, at least one node in $\mathcal{G}_i$ is an immediate neighbor of v'. Let $(p_0, p_1, ..., p_m)$ be a maximal simple path where $p_0 = v'$ and any p_{i+1} is selected from $\mathcal{N}_{p_i} \setminus \{p_0, p_1, ..., p_i\}$ as in MPS. Note that if MPS is initiated by v' and p_1 is a node in $\mathcal{G}_i$, then the rest of the nodes on the sequence are also in $\mathcal{G}_i$. Moreover, the last node on the sequence is noncritical because MPS always stops at a noncritical node as shown in Proposition 2. This implies that any $\mathcal{G}_i$ for $1 \leq i \leq n$ contains at least one noncritical node. Thus, if the removal of v_i causes n number of connected components, then $\mathcal{G}$ has at least n noncritical nodes. □

Algorithm 3: Constrained δ-MPS

$Input$: An arbitrary node, p_0, from $\mathscr{G}$

$Output$: Connectivity maintenance in the removal of p_0

$Assumption$: Each node shares its unique node ID, δ-criticality, and constraint with its neighbors.

initialization: $p_i \leftarrow p_0$; $\mathscr{N}_{p_i} \leftarrow \mathscr{N}_{p_0}$; $RequestMessage \leftarrow (p_0)$; $ApprovalMessage \leftarrow \emptyset$; $RefusalMessage \leftarrow \emptyset$;
if p_0 is a δ-noncritical and an unconstrained node
no replacements required;
else
while $\mathscr{N}_{p_i} \setminus RequestMessage \neq \emptyset$
if $\mathscr{N}_{p_i} \setminus RequestMessage$ contains at least one unconstrained node
if An unconstrained node in $\mathscr{N}_{p_i} \setminus RequestMessage$ is δ-noncritical
$p_{i+1} \leftarrow v$ s.t. v is one of the δ-noncritical unconstrained nodes;
p_i sends $RequestMessage$ to p_{i+1};
$p_i \leftarrow p_{i+1}$; $RequestMessage \leftarrow (RequestMessage, p_i)$;
break;
else
$p_{i+1} \leftarrow v$ s.t. v is one of the unconstrained δ-critical nodes in $\mathscr{N}_{p_i} \setminus RequestMessage$;
if $RequestMessage = (p_0)$
$saveID = p_{i+1}$;
end if
p_i sends $RequestMessage$ to p_{i+1};
$p_i \leftarrow p_{i+1}$; $\mathscr{N}_{p_i} \leftarrow \mathscr{N}_{p_{i+1}}$; $RequestMessage \leftarrow (RequestMessage, p_i)$;
end if
else
$RefusalMessage \leftarrow RequestMessage$;
for $j = length(RequestMessage)$ **to** 2
p_j sends $RefusalMessage$ to p_{j-1};
end for
$\mathscr{N}_{p_i} \leftarrow (\mathscr{N}_{p_0} \setminus saveID)$; $RequestMessage \leftarrow (p_0)$; $ApprovalMessage \leftarrow \emptyset$; $RefusalMessage \leftarrow \emptyset$;
end if
end while
$ApprovalMessage \leftarrow RequestMessage$;
for $j = length(RequestMessage)$ **to** 1
p_j sends $ApprovalMessage$ to p_{j-1}; p_j replaces p_{j-1};
end for
end if
p_0 is removed;

Proposition 8. *Let v' be any node in a connected graph $\mathscr{G} = (V,E)$. There always exist a simple path, p, from any critical node in $V \setminus \{v'\}$ to a noncritical node such that v' is not in p.*

Proof. Let v_c be any critical node in $\mathscr{G}$, and let p be any simple path from v_c to a noncritical node, $v_{nc,1}$. This proposition is proven in two parts: 1) Assume that v' is a noncritical node and p' contains v' as $p' = (v_c, ..., v_i, v', v_j ..., v_{nc,1})$. If v' is removed from $\mathscr{G}$, v_i does not disconnect from v_j because there exist another path $p^{i,j}$ that connects v_i to v_j after the removal of v'. Hence, there always exist a path from v_c to v_{nc1} as $p'' = (v_c, ..., v_i, p^{i,j}, v_{i+1} ..., v_{nc,1})$ that does not contain v'. Note that any path (e.g. p'') contains a simple path. Hence, there is a simple path p in p'' such that v' is not in p. 2) Assume that v' is a critical node, and its removal from $\mathscr{G}$ causes n number of connected components as $\mathscr{G}_1$,...,$\mathscr{G}_n$. Let v_c be a node in $\mathscr{G}_j$. If the simple path p from v_c to $v_{nc,1}$ contains v', then $v_{nc,1}$ should be a node in another $\mathscr{G}_i$, where $i \neq j$. Note that each $\mathscr{G}_i$ has at least one noncritical node as shown in Proposition 7. Hence, there always exist a path from v_c to a noncritical node without containing v'. □

Proposition 9. *Let $v' \in V$ be the only constrained node in a connected graph $\mathscr{G} = (V,E)$. Assume that δ is selected according to Proposition 6. After an arbitrary node removal, the constrained δ-MPS ensures a connected $\mathscr{G}$ without violating the node constraint.*

Proof. Let p_0 be the removed node in $\mathscr{G}$. 1) If p_0 is a δ-noncritical unconstrained node, the constrained δ-MPS stops with a request message $\{p_0\}$, which is not forwarded to any other nodes. Thus, p_0 is removed from the graph. 2) If p_0 is the constrained node v' (either δ-critical or δ-noncritical), the constrained δ-MPS initiates the message forwarding among the unconstrained nodes of $\mathscr{G}$. Consequently, the constrained δ-MPS algorithm never executes the lines between 20 and 26. Whenever the request message reaches a δ-noncritical node, the approval message is forwarded accordingly. Finally, the connectivity of the communication graph is maintained, and the node constraint is not violated because

v' is replaced by another node and the replacing node becomes the new constrained node. 3) If p_0 is a δ-critical unconstrained node, then it initiates the message forwarding until an approval message reaches itself. Note that based on Proposition 6 and Proposition 8, there always exist a simple path from the removed node (i.e. critical node) to a noncritical node without including v'. Hence, the constrained δ-MPS results in a sequence of replacements that is a simple path from the removed node to a noncritical node without including v'. Therefore, the connectivity is maintained without violating the node constraint after the removal of p_0. □

Corollary 4. *For a connected tree graph $\mathcal{G}$, let v' be the only constrained node in $\mathcal{G}$. Assume that an arbitrary node other than v' is removed from $\mathcal{G}$, and utilizing the constrained δ-MPS results in a connected graph $\mathcal{G}^+$ after the node removal. Let d' and d'_+ denote the maximum distances from v' to any node in $\mathcal{G}$ and $\mathcal{G}^+$, respectively. Then $d'_+ \leq d'$.*

Proof. Assume that v_1 is the furthest point to v' in $\mathcal{G}$. Then $p' = (v', v_n, ..., v_2, v_1)$ has a length of d'. Note that v_1 is a noncritical node. Let v_i be the removed node in $\mathcal{G}$. 1) If $v_i = v_1$, then no replacements are required. After the removal of v_1, the furthest node to v' becomes v_2 and d'_+ becomes $d' - 1$. 2) If v_i is a node on p' other than v_1, then a sequence of replacements will be proceeded because v_i is a critical node. For $i > 1$, if all neighbors of v_i are on p', i.e. $\mathcal{N}_{v_i} = \{v_{i-1}, v_{i+1}\}$, then the path $(v', v_n, ..., v_{i+1}, v_i, v_{i-1}, ...v_2, v_1)$ becomes $(v', v_n, ..., v_{i+1}, v_{i-1}, ...v_2, v_1)$ after the replacements. Thus, the distance from v' to v_1 becomes $d' - 1$. 3) For $i > 1$, if v_i has a neighbor, v_k, other than v_{i-1} and v_{i+1}, then the path from v' to v_1, i.e. $(v', v_n, ..., v_{i+1}, v_i, v_{i-1}, ...v_2, v_1)$, can alternatively become $(v', v_n, ..., v_{i+1}, v_k, v_{i-1}, ...v_2, v_1)$ after the replacements. Hence, the distance from v' to v_1 stays d'. □

***Remark* 8.** *Let v' be the only constrained node in a connected tree graph $\mathcal{G} = (V, E)$. Suppose that $|V|$-1 nodes are sequentially removed from $\mathcal{G}$. Then, the final graph after the* *removal of $|V|$-1 nodes becomes a single node v'.*

Corollary 5. *Let V' be the set of all constrained nodes in $\mathscr{G} = (V,E)$, and let n be the length of the shortest path that contains all nodes in V'. Assume that δ is selected according to Proposition 6. For any arbitrary node removal, if the constrained δ-MPS maintains connectivity without violating the constraints, then $|V| - 1 \geq n$.*

Proof. Assume that $\mathscr{G} = (V,E)$ has initially n number of nodes. Then $\mathscr{G}$ corresponds to p that is the shortest path containing all nodes in V'. Note that p starts and ends with a node from V', so any node removal in p initiates the replacements. However, the replacements do not stop at a noncritical node, because all noncritical nodes in $\mathscr{G}$ are constrained, and they cannot accept any replacement request. Consequently, if one of the constrained nodes at the beginning and end of the path is removed, the resulting graph becomes connected but violates the node constraints. Otherwise, the resulting graph becomes disconnected due to not proceeding a necessary sequence of replacements. Hence, the constrained δ-MPS cannot maintain connectivity without violating the constraints, if $|V| < n+1$. □

6.3 Results

In this section, some experiments are conducted to compare the outcomes of no control strategy, the δ-MPS, and the constrained δ-MPS. In accordance with the persistent surveillance problem, let a surveillance area be represented as a connected graph that contains 7 unconstrained and 1 constrained nodes (i.e. points of interest) as in Figure 28. Here, the constrained node denotes the position, where an agent can directly communicate with the base.

In this scenario, suppose that there exist 8 agents, each of which is located on the nodes of the graph as in Figure 29. Note that information flows from any agent to the base as long as an agent is located on the constrained node (e.g. agent 5 is located on the highlighted constrained node). Based on the initial setting as in Figure 29, agent 5 should not move

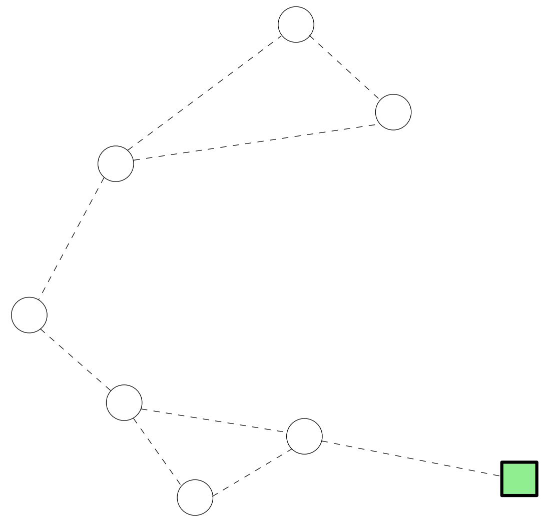


Figure 28: A surveillance area involving 1 base, 7 unconstrained nodes, and 1 constrained node.

from its position except its removal. Moreover, if it is removed from the system, a sequence of replacements should be pursued to occupy the constrained node.

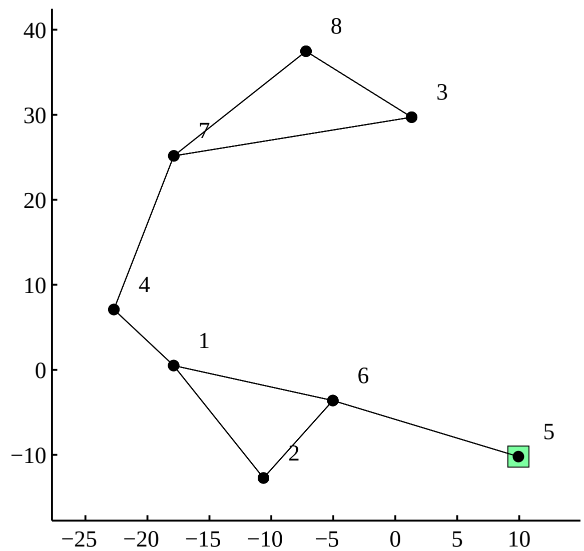


Figure 29: Initial MAS containing 8 agents, one of which is located on the constrained node.

An experiment is performed in order to compare the performance of no control startegy, 1-MPS, and the constrained 1-MPS. Accordingly, a randomly selected agent is removed from the surveillance area at each time step, and the graph evolution (i.e. the communication network of the remaining agents) is inspected as illustrated in Figures 30 and 31.

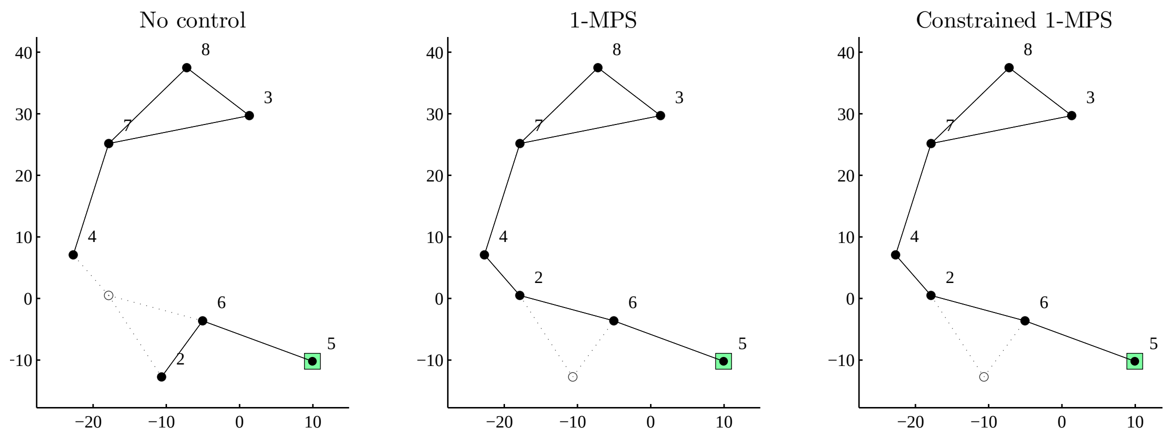
At $t = 1$, agent 1 leaves the surveillance area. As seen from Figure 30a, utilizing no control strategy causes a graph disconnection whereas 1-MPS and the constrained 1-MPS recovers the connectivity with one replacement (i.e. agent 2 replaces agent 1). At $t = 2$, agent 5 leaves the surveillance area. From the connectivity perspective, it is a noncritical agent so its removal does not actually cause a graph disconnection. However, its removal causes the constrained node become unoccupied, thus information does not flow from agents to the base. As seen from Figure 30b, 1-MPS does not result in any replacements, and the remaining agents on the surveillance area maintain a connected communication network among each other. However, they become disconnected from the base due to the unoccupied constrained node. On the contrary, the constrained 1-MPS leads a sequence of five replacements as follows: agent 5 is replaced by agent 6, agent 6 is replaced by agent 2, agent 2 is replaced by agent 4, agent 4 is replaced by agent 7, and agent 7 is replaced by agent 8. In this respect, agents relocate to occupy the constrained node and to maintain network connectivity.

At $t = 3$, agent 3 leaves the surveillance area. Since agent 3 is noncritical, both 1-MPS and the constrained 1-MPS do not drive agents to replace each other. However, due to the previous graph configuration at $t = 2$, the MAS using 1-MPS maintains connectivity within itself but disconnects from the base. On the other hand, the MAS using the constrained 1-MPS maintains connectivity within itself and with the base. Moreover, the MAS using no control strategy has two connected components, and it is disconnected from the base.

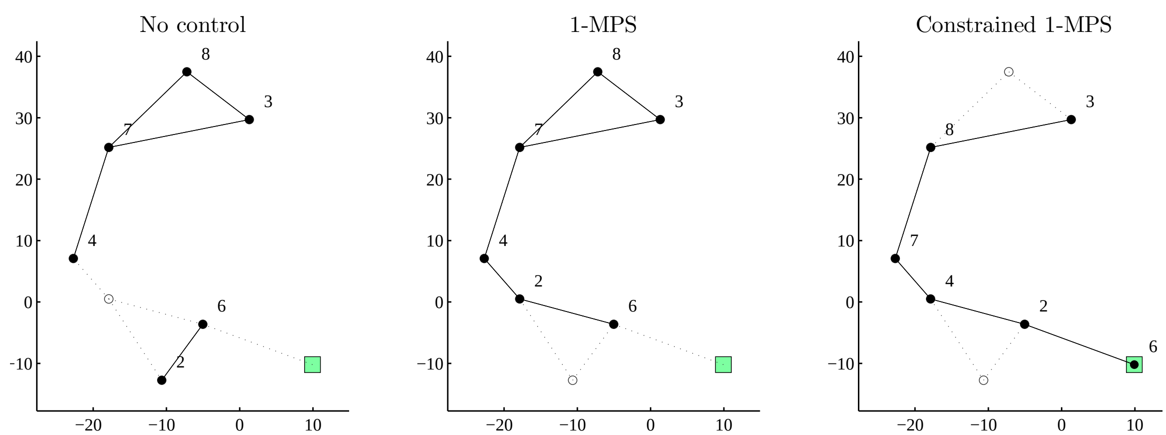
At $t = 4$, agent 2 leaves the surveillance area. Since it is a critical agent, both 1-MPS and the constrained 1-MPS drive agents to relocate on the surveillance area. Using 1-MPS results in agent 6 replacing agent 2. On the other hand, utilizing the constrained 1-MPS leads agents to perform the following replacements: agent 4 replaces agent 2, agent 7 replaces agent 4, and agent 8 replaces agent 7. As a result, two connected components, which do not connect with the base, exist by using no control strategy. One connected component, but disconnected from the base, occurs via 1-MPS. Finally, one connected component, as well as connected with the base, occurs with the constrained 1-MPS.

At $t = 5$, agent 8 leaves the surveillance area. Since it is a noncritical agent in all graph configurations, no replacements are required. However, due to being disconnected at $t = 4$, the graphs continue to be disconnected from the base via no control strategy and 1-MPS at $t = 5$. On the other hand, the constrained 1-MPS ensures the connectivity.

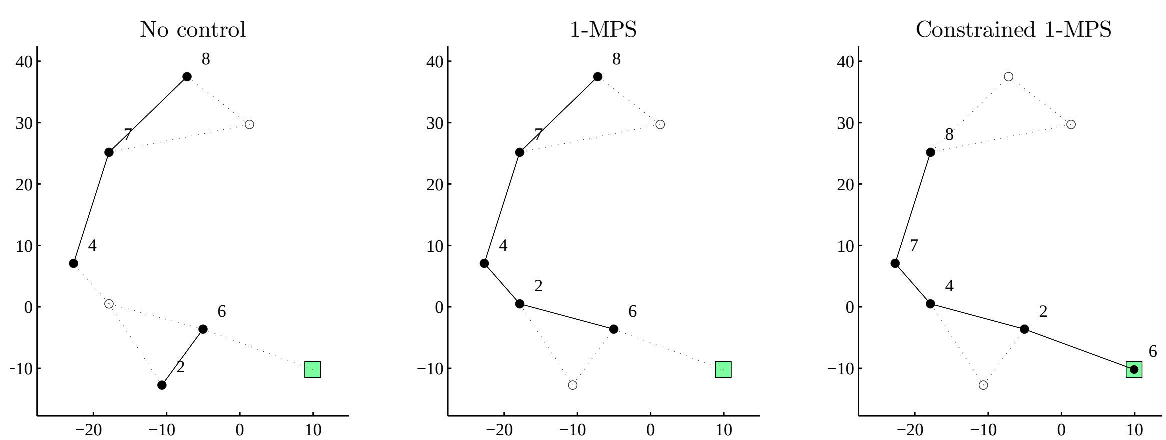
Finally, at $t = 6$, agent 7 leaves the surveillance area. Eventually, agents 4 and 6 exist on the surveillance area. As seen from Figure 31c, the last two agents are connected and do not violate the node constraint via the constrained 1-MPS. On the other hand, 1-MPS leads to a connected configuration, where no agent occupies the constrained node. Moreover, if agent 4 leaves the surveillance area at the next time step, the constrained 1-MPS results in a graph consisting only 1 agent locating on the constrained node. On the other hand, if agent 6 leaves the area at $t = 7$, then agent 4 replaces it and occupies the constrained node. As it is seen from the simulations, the constrained node behaves as if it is an attractor for the agents. Consequently, the simulations demonstrated that the constrained δ-MPS is a locally applicable decentralized strategy that ensures connectivity and occupancy of the constrained node at all time.



(a) Resulting graphs after the removal of *agent 1* at $t = 1$

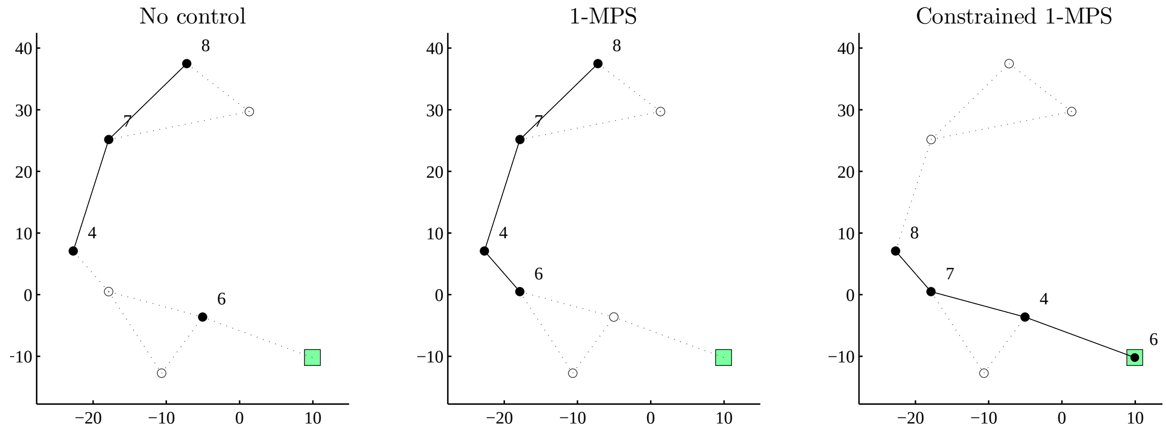


(b) Resulting graphs after the removal of *agent 5* at $t = 2$

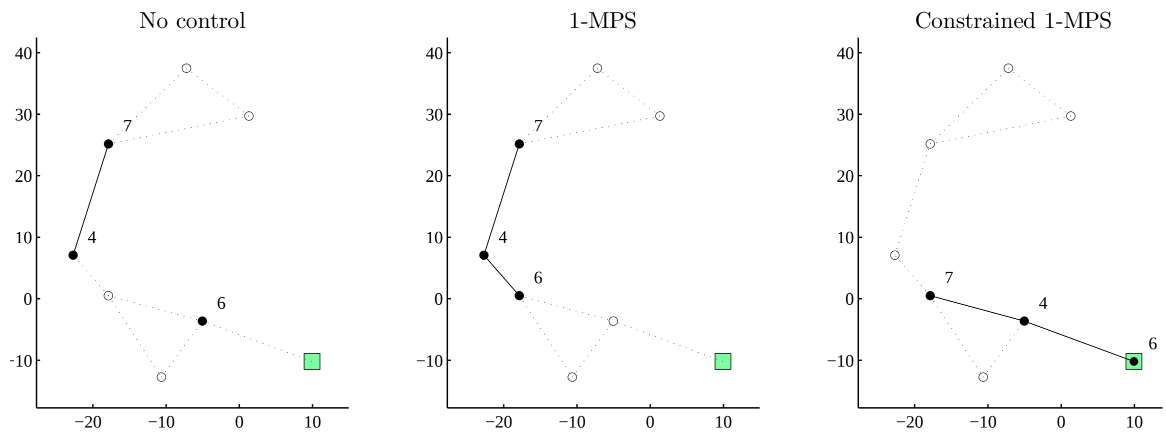


(c) Resulting graphs after the removal of *agent 3* at $t = 3$

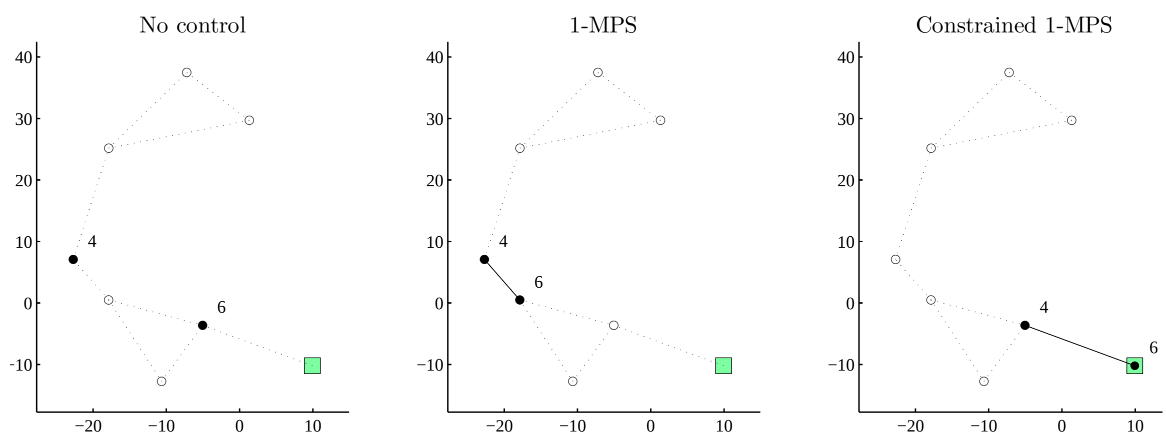
Figure 30: Graph evolution via various strategies at $t = 1, 2, 3$.



(a) Resulting graphs after the removal of *agent 2* at $t = 4$



(b) Resulting graphs after the removal of *agent 8* at $t = 5$



(c) Resulting graphs after the removal of *agent 7* at $t = 6$

Figure 31: Graph evolution via various strategies at $t = 4, 5, 6$.

CHAPTER VII

ENERGY-AWARE PERSISTENT SURVEILLANCE WITH MULTI-AGENT SYSTEMS

In a multi-agent system, each agent typically senses its environment, processes the gathered information, and makes a decision about its next action. In physical systems, agent actions typically require some energy consumption. Hence, it is crucial for an agent to be aware of its remaining energy when planning its actions. For example, in a multi-UAV surveillance mission, each UAV is a valuable asset that has a limited fuel capacity. Hence, each UAV should be aware of its remaining fuel, and it should maintain a sufficient amount of fuel in order to safely return to base at any instant. Hence, designing energy-aware strategies for multi-agent surveillance systems is essential for desired mission outcomes. In this respect, the motivation of this chapter is based on the following research question:

***Research Question* 4.** *How is it possible to design an energy-aware (preferably decentralized) strategy that results in efficient persistent surveillance operations?*

This chapter presents two approaches for the design of energy-aware strategies that can be applicable to networked multi-agent systems. First, a centralized strategy is obtained via a Markov decision process (MDP) formulation and its approximate dynamic programming solution. Second, some decentralized strategies are proposed as alternatives to the centralized strategy. Finally, the last section compares the performance of the presented centralized and decentralized strategies through simulations.

7.1 Markov Decision Process

Markov decision process is a mathematical framework that models decision making in a stochastic system (e.g. [92, 100]). A discrete time MDP is defined by a tuple $\{X, U, \mathscr{P}, g\}$ as follows:

- X is the finite state space.
- U is the finite set of actions available at each time step.
- $\mathscr{P}$ are the transition probabilities such that $p(x^+|x,u) \in \mathscr{P}$ is the probability of moving from state x to state y by taking action u.
- $g : (x,u) \to \mathbb{R}$ is the cost incurred due to taking action $u \in U$ when in state $x \in X$.

Let the history at time t be the sequence of states and actions as well as the current state as $h(t) = \{x(0), u(0), x(1), u(1), ..., x(t-1), u(t-1), x(t)\}$. Let μ denote a policy. Then, a policy is a mapping from the state space to the action space as $\mu : X \to U$, which can choose an action based on the current state of the system at time t. Accordingly, the main objective in an MDP formulation is to find an optimal policy that minimizes the overall expected cost incurred along the time horizon $[0, T]$.

$$J^{\mu}(x) = \mathbb{E}\Big[\sum_{t=0}^{T} \gamma^t g\Big(x(t), \mu\big(x(t)\big)\Big)\Big|x(0) = x\Big], \tag{14}$$

where $\gamma \in [0,1]$ is a discount factor, and T can be finite or infinite.

An MDP problem can be solved by various optimization techniques such as linear programming or dynamic programming (e.g. [73, 111]). For example, linear programming deals with problems that have linear objectives and constraints, while dynamic programming depicts a recursive solution technique that can be applicable to a wider range of problems having linear or nonlinear equations (e.g. [49, 74]).

7.1.1 Dynamic Programming Preliminaries

Dynamic programming is an optimization method that solves complex problems by breaking them down into simpler sub-problems (e.g. [24]). As such, a larger solution is synthesized from smaller sub-solutions, and a sequential procedure is utilized to optimize a desired objective function. A dynamic programming algorithm tries many possibilities and choices before it arrives at the optimal set of choices. The foundation of dynamic programming is based on Bellman's principle of optimality.

Theorem 2. *(Bellman's principle of optimality) From any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem over the remaining number of stages or time interval initiated at that point.*

Given a finite MDP formulation, for any policy μ and a cost function as in (14), a cost-to-go function can be computed by solving the linear equations

$$J^{\mu}(x) = g\big(x, \mu(x)\big) + \gamma \sum_{x^+ \in X} p\big(x^+ | x, \mu(x)\big) J^{\mu}(x^+). \tag{15}$$

Note that for the expected total discounted cost, there exists a policy, μ^*, that minimizes the cost-to-go for all initial states. Accordingly, the minimum cost-to-go J^* is unique and satisfies the Bellman equation.

$$J^* = \min_{u \in U} \Big(g(x,u) + \gamma \sum_{x^+ \in X} p(x^+ | x, u) J^*(x^+) \Big). \tag{16}$$

7.2 *Centralized Approach*

In this section, the energy-aware persistent surveillance problem is formulated as a Markov Decision Process (MDP) and solved via approximate dynamic programming. Let a finite coordinate space, C, denote all possible coordinates an agent can be on a surveillance area, and let a finite energy space, F, denote all discretized energy levels an agent can have during a mission. Accordingly $c_i(t) \in C$ is the coordinate of agent i at time t, and $f_i(t) \in F$ is the remaining energy agent i has at time t. Let the overall state space X be $C \times F$. Then, the state of agent i at time t is denoted by $x_i(t) = (c_i(t), f_i(t))$, where $x_i(t) \in X$.

In this thesis, agents are assigned to some points to monitor a local region around the point. Accordingly, each agent's action refers to the coordinate of its target point. Since a surveillance area is assumed to have a finite number of monitoring points, each agent has a finite number of actions at each time step. Let U denote the admissible action space that contains the coordinates of the nodes (i.e. monitoring points) and the base. Then, the action of agent i at time t, $u_i(t)$, is its target coordinate. Note that $u_i(t)$ drives the state of agent i to a new coordinate and a new energy level. Accordingly, if an agent takes action $u_i(t) \in U$, its state at the next time step, i.e. $x_i(t+1) = \big(c_i(t+1), f_i(t+1)\big)$, evolves according to the following dynamics.

$$c_i(t+1) = \begin{cases} c_i(t) + V_i \frac{u_i(t)-c_i(t)}{\|u_i(t)-c_i(t)\|}, & \text{if } \|u_i(t)-c_i(t)\| \geq V_i \\ u_i(t), & \text{if } otherwise, \end{cases} \tag{17}$$

$$f_i(t+1) = \begin{cases} f_i^{max}, & \text{if } c_i(t) = c_{base} \\ f_i(t) - \Delta f_i\big(c_i(t), u_i(t)\big), & \text{if } otherwise, \end{cases} \tag{18}$$

where V_i is the maximum velocity of agent i, c_{base} is the coordinate of the base, and the energy consumption of agent i is denoted by $\Delta f_i\big(c_i(t), u_i(t)\big)$ as in (19).

$$\Delta f_i\big(c_i(t), u_i(t)\big) = \begin{cases} \Delta f_{cruise}, & \text{if } c_i(t) \neq u_i(t) \\ \Delta f_{loiter}, & \text{if } c_i(t) = u_i(t). \end{cases} \tag{19}$$

Let $\mathscr{X}(t) = \{x_1(t), ..., x_n(t)\}$ and $\mathscr{U}(t) = \{u_1(t), ..., u_n(t)\}$ denote the set of states and the set of actions of n agents at time t. The overall objective function in the persistent surveillance mission is defined as the summation of the node ages as in (7). Moreover, the age of a node depicts the duration of time the base has not received any information from it at time t. The recursive dynamics of node age is given in (5). Note that the age of a node is a function of the history of the agent states, i.e. $\alpha_j(t) = \alpha_j(\mathscr{X}(0), ..., \mathscr{X}(t))$. Accordingly, the objective function in (15) can be rewritten as

$$J^{\mu}(\mathscr{X}) = \mathbb{E}\Big[\sum_{t=1}^{T} g\big(\mathscr{X}(0), ..., \mathscr{X}(t), \mu(\mathscr{X}_t)\big)\Big|\mathscr{X}(0) = \mathscr{X}\Big], \tag{20}$$

where V is the set of nodes, T is the overall mission horizon, and

$$g\big(\mathscr{X}(0), ..., \mathscr{X}(t), \mu(\mathscr{X}_t)\big) = \sum_{j \in V} \alpha_j(t+1). \tag{21}$$

The persistent surveillance problem considered in this thesis is a long-run mission, and the agents may need to return to the base for replenishing their energies. In order to avoid any crash due to running out of energy, each agent needs to maintain a sufficient amount of energy for going back to base at any instant. Hence, each agent must satisfy an energy constraint such as

$$f_i(t) > f_i^{Cr}\big(c_i(t)\big) \quad \forall i, \tag{22}$$

where $f_i^{Cr}\big(c_i(t)\big)$ is the required energy for agent i to return to the base from the coordinate $c_i(t)$. Consequently, the MDP problem becomes to find a policy that solves

$$\mu^* = \underset{\mu}{\operatorname{argmin}} \, \mathbb{E}\Big[\sum_{t=1}^{T} g\big(\mathscr{X}(0), ..., \mathscr{X}(t), \mu(\mathscr{X}_t)\big)\Big|\mathscr{X}(0) = \mathscr{X}\Big] \tag{23}$$

$$\text{s.t. } f_i(t) > f_i^{Cr}\big(c_i(t)\big) \quad \forall i \text{ and } \forall t \in [0, T].$$

The agents states have nonlinear dynamics as given in (17) and (18). The MDP problem posed in (23), containing nonlinear objective and constraint, can be solved via dynamic programming. A dynamic programming solution requires the computation of all possible states and their corresponding costs-to-go over the time horizon $[0,T]$. Given a set of initial states for n agents, i.e. $\mathscr{X}_0 = \mathscr{X}$ where $\mathscr{X} \in \mathbb{R}^{n\times 2}$, the total number of combined actions at any t is $(|V|+1)^n$, so $\mathscr{U} \in \mathbb{R}^{(|V|+1)^n}$, where $|V|+1$ denotes the total number of feasible target states (including all nodes and the base). Accordingly, the costs-to-go over the time horizon $[0,T]$ can be computed by considering $(|V|+1)^{nT}$ possibilities. Here, as T or the number of nodes or agents increase, the required number of computations increases exponentially.

As depicted before, this thesis focuses on long-run persistent surveillance missions. Even though n or $|V|$ is assumed small, computing the optimal solution severely suffers from the curse of dimensionality if a large T is considered. One way to reduce the problem complexity is to use approximations in the solution. Recently, a great amount of interest has been devoted to approximate dynamic programming, where the researchers relax the problem by reducing the state-space, using approximate functions for the objective function, or optimizing over a shorter time horizon, to name a few (e.g. [8], [23], [99], [130], [67]).

In this section, the MDP problem depicted in (23) is solved via approximate dynamic programming, where two relaxation methods are adopted. First, the objective function is optimized over a shorter time horizon. Accordingly, a sequence of optimal actions $(\mathscr{U}(t),...,\mathscr{U}(t+\tau))$ are computed at any t by solving (23) from t to $t+\tau$, where $\tau << T$. Moreover, a receding horizon control is used to enable action correction at any time step. Note that receding horizon control is a well-studied control approach in the literature (e.g. [79], [19], [47]). Accordingly, an optimal course of action is computed at each time step by solving an optimization problem over a short time interval. This implies that only $\mathscr{U}(t)$ is selected from $(\mathscr{U}(t),...,\mathscr{U}(t+\tau))$. Hence, $\mathscr{U}(t+1)$ is computed by finding the optimal sequence of actions to minimize (23) over $[t+1,t+1+\tau]$. Here, the goal of using the receding horizon control is to mitigate the deviation of the suboptimal solution from the optimal one.

Second, some conditions are derived to reduce the size of the action space.

Definition 7. *(Crash-free action) An action $u_i(t)$ is called crash-free, if the corresponding agent satisfies $f_i(t) - \Delta f_i\big(c_i(t), u_i(t)\big) \geq f_i^{Cr}\big(c_i(t+1)\big)$.*

***Remark* 9.** *Let $c_i(t) \neq c_{base}$ and $f_i(t) > f_i^{Cr}\big(c_i(t)\big)$. For all $u_i(t) \neq c_{base}$, suppose that $f_i(t) - \Delta f_i\big(c_i(t), u_i(t)\big) < f_i^{Cr}(c_i(t+1))$. Then the only crash-free action is $u_i(t) = c_{base}$.*

According to Remark 9, $c_i(t) \neq c_{base}$ and $f_i(t) > f_i^{Cr}\big(c_i(t)\big)$, which implies that agent i is operating on the surveillance area (i.e. either in cruise or loiter mode) with enough energy to return to the base at t. Since $f_i(t) - \Delta f_i\big(c_i(t), u_i(t)\big) < f_i^{Cr}(c_i(t+1))$ for all possible $u_i(t) \neq c_{base}$, then $u_i(t) \neq c_{base}$ cannot be a crash-free action due to Definition 7. Hence, the only crash-free action becomes $u_i(t) = c_{base}$

***Remark* 10.** *Let $\sum_{j \in V} \alpha_j(t) = 0$ and let $u_i(t) = c_i(t)$ be crash-free for all i. Then $u_i(t) = c_i(t)$ minimizes $\sum_{j \in V} \alpha_j(t+1)$.*

According to Remark 10, $\sum_{j \in V} \alpha_j(t) = 0$, which implies that the base collects the maximum amount of information. Then, $u_i(t) = c_i(t)$ leads agents not to move from their coordinates. Thus, $\sum_{j \in V} \alpha_j(t+1) = 0$.

7.3 Decentralized Approach

In the previous section, an approximate dynamic programming solution is discussed for the solution of (23). Accordingly, the states of all agents are assumed to be available to the base, and the base computes a sequence of optimal actions based on the proposed approximate solution. Even though the approximate solution is relaxed by reducing the action space and optimizing over a shorter time interval, it is not a scalable approach since it requires the states of all the agents. Note that as the number of actions increases, the tractability of the agent states become harder from the base perspective. Therefore, this thesis proposes an efficient locally applicable strategy that achieves energy-aware persistent surveillance in a decentralized fashion.

In a typical persistent surveillance mission, there exist three major tasks, whose efficient implementation greatly improves the surveillance performance. The first task is how to assign nodes to agents. An efficient node assignment strategy minimizes the summation of node ages. The second task is to design efficient return schedules for agents to replenish energy. An effective return strategy not only prevents agents to run out of energy but also enables agents to stay on the surveillance area as long as possible. Finally, the third task is to relocate agents after an agent leaves the surveillance area. An efficient relocation strategy mitigates the increase in the summation of node ages due to an agent leave. In this thesis, an energy-aware decentralized strategy is proposed to reduce the amount of computations made by the base. Accordingly, the base only tracks the ages of the nodes based on its communication with the agents. Moreover, it assigns nodes to agents only if they come back to the base for refueling/recharging. Apart from that, each agent is assumed to operate autonomously in making decisions about return and relocation based on its individual state and the information it gathers from its vicinity.

7.3.1 Node Assignment Policy

In the persistent surveillance problem, the major tasks of the base are tracking the ages of the nodes and assigning a node to an agent whenever it is ready to return to the mission area. At the beginning of each mission, the base randomly assigns a single node to each agent. As the agents operate at the surveillance area, the base tracks the ages of the nodes by considering the age dynamics introduced in (5). Accordingly, the ages of m nodes $\left(\alpha_1(t), \ldots, \alpha_m(t)\right)$ are always available to the base at any time t. During the mission, let agent i return to base and replenish its energy. Then, the base needs to assign a node to agent i. For the computation of the assignments, suppose that the base uses a greedy strategy for the node assignment. As such, agent i is assigned to the node j, which is the node with the maximum age, i.e. $\alpha_j(t) = \max\big(\alpha_1(t), ..., \alpha_m(t)\big)$.

7.3.2 Return Policies

In this thesis, each agent makes its own decision about when to return to the base. As such, it is crucial to design a locally applicable return strategy that avoids an agent to have unsafe states (e.g. running out of fuel/energy without reaching the base). Note that $f_i^{Cr}(c_i(t))$ denote the energy required for agent i to go from its current position, $c_i(t)$, to the base, c_{base}. Suppose that each agent can estimate $f_i^{Cr}(c_i(t))$. An effective return policy must ensure that $f_i(t)$ will not be less than $f_i^{Cr}(c_i(t))$ at any instant during the mission. Note that this is also the constraint of the *MDP* problem depicted in (23). This thesis presents two types of return policies as follows.

7.3.2.1 Deterministic Return Policy

The deterministic return policy is the most trivial one that changes an agent's action whenever its energy state approaches to the critical energy level. Let $\beta > 0$ denote some amount of buffer energy. Then, an agent utilizing the deterministic policy compares its energy state, $f_i(t)$ with a critical energy threshold, $f_i^{Cr}(c_i(t)) + \beta$, at any instant. Whenever $f_i(t)$ becomes less than the critical energy threshold, then the agent returns to the base, i.e. $u_i(t) = c_{base}$. The flow diagram of the deterministic policy is illustrated in Figure 32.

7.3.2.2 Randomized Return Policy

The agents using the deterministic return policy do not coordinate with each other when taking a return action. In case of the majority of agents reach their critical energy threshold, the deterministic policy leads agents to return to base simultaneously, which causes a huge degradation in situational awareness for some time instances. One way to avoid this situation is that an agent can communicate with the neighbors in the vicinity and make a

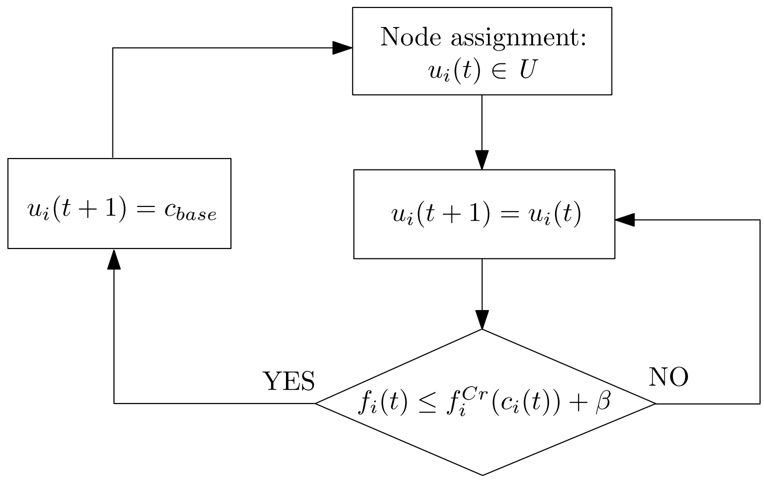


Figure 32: The flow diagram of the deterministic return policy.

return decision based on its individual as well as its neighbors' energy states. However, this increases the communication load of an agent. Alternatively, an agent may return to the base with a small probability even though its remaining energy has not approached the critical energy threshold. In this manner, random returns to the base increase the control of the base on the surveillance area by performing more frequent node assignments. The flow diagram of the randomized return policy is illustrated in Figure 33.

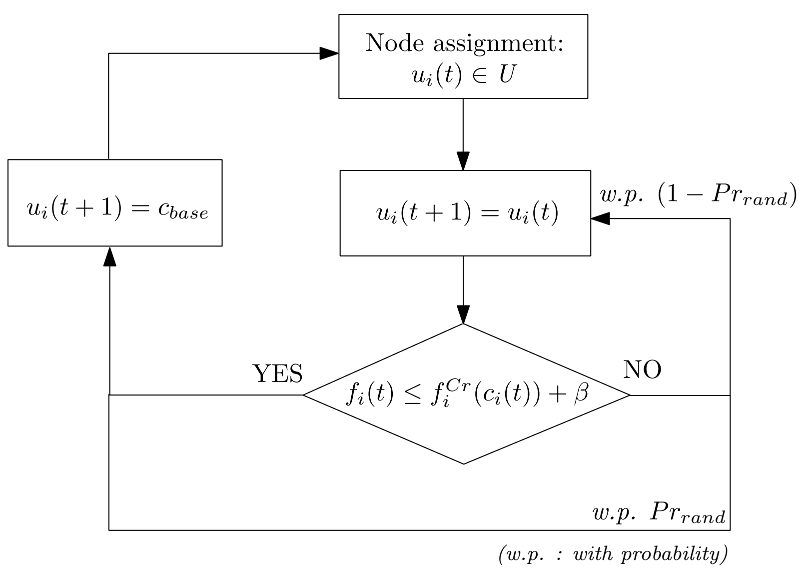


Figure 33: The flow diagram of the randomized return policy.

Note that if the probability to return to base (Pr_{rand}) is too large, then the agents frequently return to the base with plenty of energy remaining, which causes an ineffective use of the overall energy capacity. On the other hand, if $Pr_{rand} = 0$, then it is equivalent to the deterministic return policy. Therefore, the probability to return to base behaves like a tuning parameter to vary the performance of the MAS.

7.3.3 Relocation Policies

Whenever an agent leaves the surveillance area, the rest of the agents on the surveillance area are assumed to make their own decisions for their relocation. As discussed in the previous sections of this thesis, an agent leave (i.e. removal) may cause a disconnection in the communication network. In case of a disconnection, some of the agents cannot send any monitoring information back to the base. For example, if agent i monitors node j and there is a disconnection between agent i and the base, then the age of node j increases even though it is monitored by agent i. Thus, an efficient relocation policy is expected to maintain the network connectivity in the face of any agent removal.

7.3.3.1 Replacement Policy

In Chapter 5, the constrained δ-MPS has been proposed as an effective connectivity recovery strategy for persistent surveillance missions. Accordingly, the constrained δ-MPS recovers connectivity through as few as possible replacements while agents never lose communication with the base. Hence, if each agent utilizes the constrained δ-MPS during a mission, then a connected communication network including the base exists all the time.

7.3.3.2 NoReplacement Policy

In order to observe the effect of the replacement policy, the noReplacement policy is proposed as a counter strategy. As such, agents do not relocate themselves in any condition. In other words, the action of each agent changes only if the agent completes refueling or recharging after its return to the base.

Note that a relocation policy does not only imply the maintenance of network connectivity. Even though a connected network is present, there can be cases where some nodes can be unoccupied. In that case, agents may need to patrol the surveillance area to minimize the node ages. However, one critical issue is to ensure connectivity with the base as frequently as possible to be able to send the new information. For instance, the authors of [29] propose a decentralized connectivity strategy for robotic swarms that patrol the surveillance area in a coordinated fashion and by considering the communication constraints. Alternatively, this thesis suggests a decentralized strategy that comprises the replacement and the randomized policies mentioned above. Accordingly, the replacement policy ensures connectivity with the base in the face of an agent leave. Instead of designing a strategy for agents to provide coordinated movement on the surveillance area, the randomized policy is utilized by each agent to result in random returns to the base, which increases the control of the base over the surveillance area.

7.4 Results

The preceding sections have presented some possible centralized and decentralized solutions for energy-aware persistent surveillance. As depicted, a centralized solution becomes computationally inefficient as the number of states increases in a problem. Overcoming this issue motivates the development of decentralized solutions. In this section, a set of experiments are performed to investigate the outcomes of persistent multi-agent surveillance under different control strategies. In order to understand the effectiveness of the decentralized energy-aware strategies, their results are compared with the centralized strategy that is considered as a benchmark. Note that the centralized strategy uses approximate dynamic programming with a receding horizon optimization for the decision of actions, whereas the decentralized strategies suggest locally applicable rules for the agents.

For all practical purposes, a simple representative scenario is considered as follows: suppose that the surveillance area contains a base and three nodes (i.e. monitoring points) as illustrated in Figure 34. Note that node 1 in Figure 34 is constrained because an agent on nodes 2 or 3 can only send information back to the base as long as there exist an agent located on node 1.

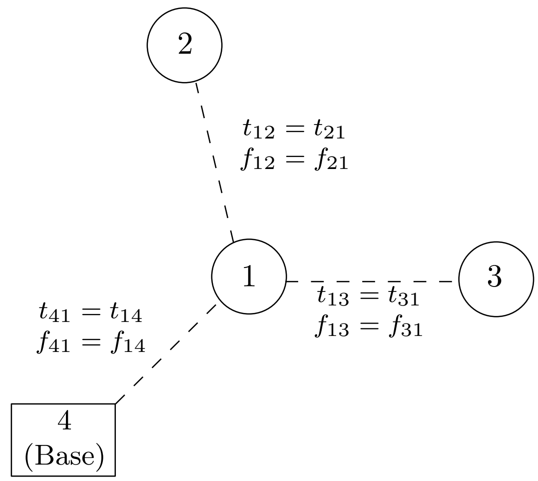


Figure 34: A canonical scenario used in the experiments to investigate energy-aware strategies.

Let t_{ij} denote the duration of time to go from node i to node j, and suppose that t_{ij} is equal to t_{ji}. In the representative scenario, let each t_{ij} be 1. This implies that an agent travels from the base to node 1 in one time step. Similarly, an agent travels from node 1 to nodes 2 or 3 in one time step, too. Moreover, let f_{ij} denote the required energy for an agent to go from node i to node j, and assume that any f_{ij} is equal to f_{ji}. Note that an agent may not necessarily visit node 1 to travel from the base to nodes 2 or 3, or vice versa.

$$f_{42} \leq f_{41} + f_{12} \quad , \quad t_{42} \leq t_{41} + t_{12} \tag{24}$$

$$f_{43} \leq f_{41} + f_{13} \quad , \quad t_{43} \leq t_{41} + t_{13} \tag{25}$$

$$f_{23} \leq f_{21} + f_{13} \quad , \quad t_{23} \leq t_{21} + t_{13} \tag{26}$$

In the experiments, three derivatives of the centralized strategy are created by assuming different prediction horizons, i.e. τ. On the other hand, various decentralized strategies are designed by selecting a combination of a node assignment, a return, and a relocation policies. The experimented strategies are displayed in Table 3.

Table 3: Experimented energy-aware strategies.

	Centralized			**Decentralized**				
	Prediction Horizon			Node Assignment Policy	Return Policy		Relocation Policy	
	$\tau = 1$	$\tau = 2$	$\tau = 3$	Greedy	Deterministic	Randomized	NoReplacement	Replacement
Strategy 1	×							
Strategy 2		×						
Strategy 3			×					
Strategy 4				×	×		×	
Strategy 5				×		×	×	
Strategy 6				×	×			×
Strategy 7				×		×		×

In the following parts, the performance (i.e. J in (7) and the mean of J) of each strategy is presented individually. Note that J is the summation of node ages along the mission horizon, which quantifies the overall degradation in situational awareness. On the other hand, $\bar{J}$ is the time average of J, which denotes the average age of a node during a mission. The smaller values are desirable both for J and $\bar{J}$. In simulations, each strategy in Table 3 is executed at missions having various time horizons, i.e. $T = \{250, 500, 750, 1000, 1250, 1500, 1750, 2000\}$. Moreover, for each T, the strategies involving randomness are repeated 100 times in order to inspect the variability in their results.

The results of the experimented strategies are illustrated in Figures 35, 36, 37, 38, 39, 40, 41, 42, 43, and 44. These figures show J and $\bar{J}$ of each strategy with respect to various mission horizons. The results of the strategies including randomness are displayed by a box plot, which shows the variability of J and $\bar{J}$ for a particular mission horizon.

Based on the canonical problem, a central authority using Strategy 1 makes a decision for the actions of three agents by considering their four feasible locations in the next 1 time step. Accordingly, it evaluates the outcome of 64 possible agent configurations (i.e. $4^{3\tau}$ where $\tau = 1$). Only looking 1 step ahead causes some variability in the results of Strategy 1, which are illustrated in Figures 35 and 36. As seen from Figure 35, the variability of J increases as the mission horizon increases. Moreover, Figure 36 illustrates that the variability in $\bar{J}$ is more significant in short missions than the longer missions. Finally, the median of $\bar{J}$ mostly stays in the range of $(0.4, 0.5)$ that is displayed as the horizontal lines in the box plots.

In the canonical problem shown in Figure 34, an agent can go from any node to an adjacent node in 1 time step. Moreover, an agent can go from any node to another node in 2 time steps because the graph has a diameter of 2. Accordingly, any strategy having a prediction horizon more than or equal to 2 can observe the consequences of all possible actions that drive agents from their current state to any state in the state-space. The results of Strategies 2 and 3 are illustrated in Figures 37 and 38, where no variability is observed in the performance. As illustrated in Figure 38, the medians of $\bar{J}$ for both strategies stay in the range of $(0.34, 0.38)$ as the mission horizon varies. Moreover, the values of J in Figure 37 are often smaller than the values in Figure 35, which indicates that the performances of Strategies 2 and 3 are potentially better than the performance of Strategy 1. Note that in return for performance improvement, action selection through Strategies 2 and 3 require the evaluation of 4096 (i.e. $4^{3\tau}$ where $\tau = 2$) and 262144 (i.e. $4^{3\tau}$ where $\tau = 3$) possible agent configurations, respectively.

Strategy 4 differs from the previous strategies due to being decentralized, and its results are illustrated in Figures 37 and 38. Note that the results do not involve any variability because it is a deterministic strategy. As seen from the figures, Strategy 4 performs poorer than Strategies 2 and 3 (e.g. the lines in Figure 37 are diverging from each other as the mission horizon increases). Moreover, Figure 38 illustrates that the median of $\bar{J}$ for Strategy 4 mostly stays in the range of $(0.44, 0.46)$ for various mission horizons, which is worse than the median $\bar{J}$ of Strategies 2 and 3.

The results of Strategy 5 are illustrated in Figures 39 and 40. Note that this strategy is decentralized, and it employs the randomized return policy. As seen from the results, random agent returns create a variability in $\bar{J}$, whose median stays in the range of $(0.45, 0.48)$ as displayed in Figure 40. These results are worse than the ones observed with Strategies 2,3, and 4.

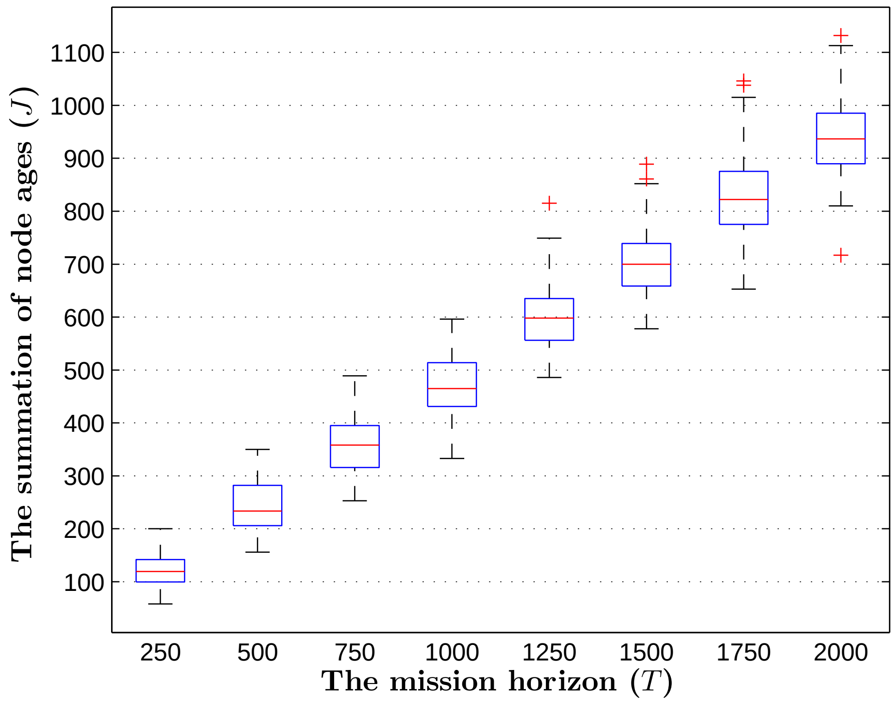


Figure 35: Box plot for the summation of node ages vs. the mission horizon via Strategy 1

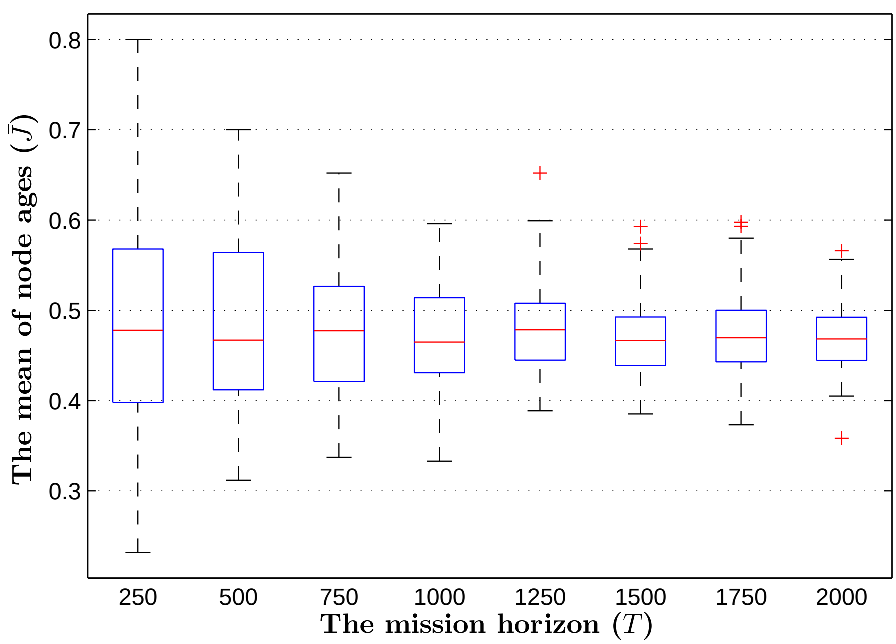


Figure 36: Box plot for the mean of node ages vs. the mission horizon via Strategy 1

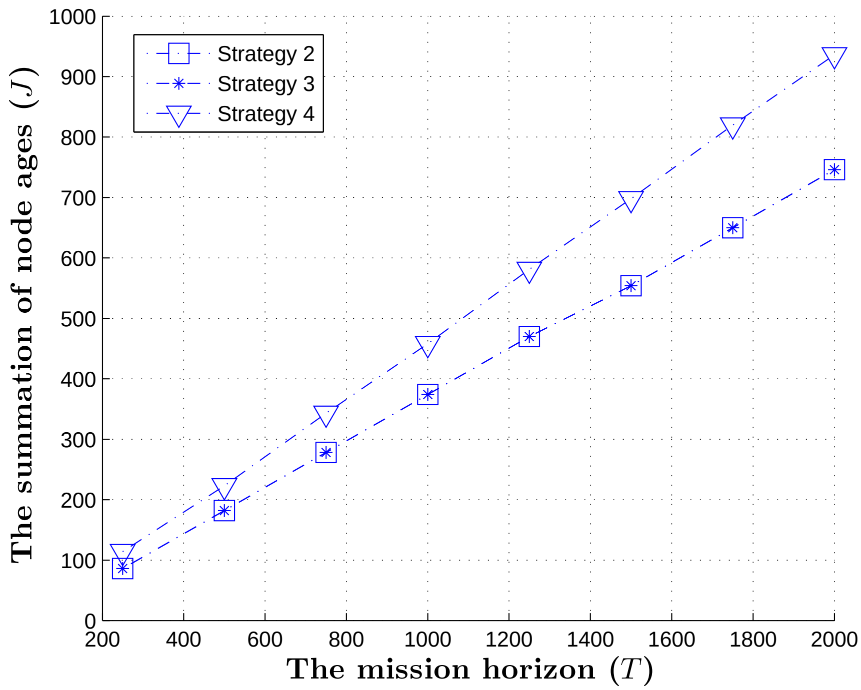


Figure 37: The summation of node ages vs. the mission horizon via Strategies 2,3,4

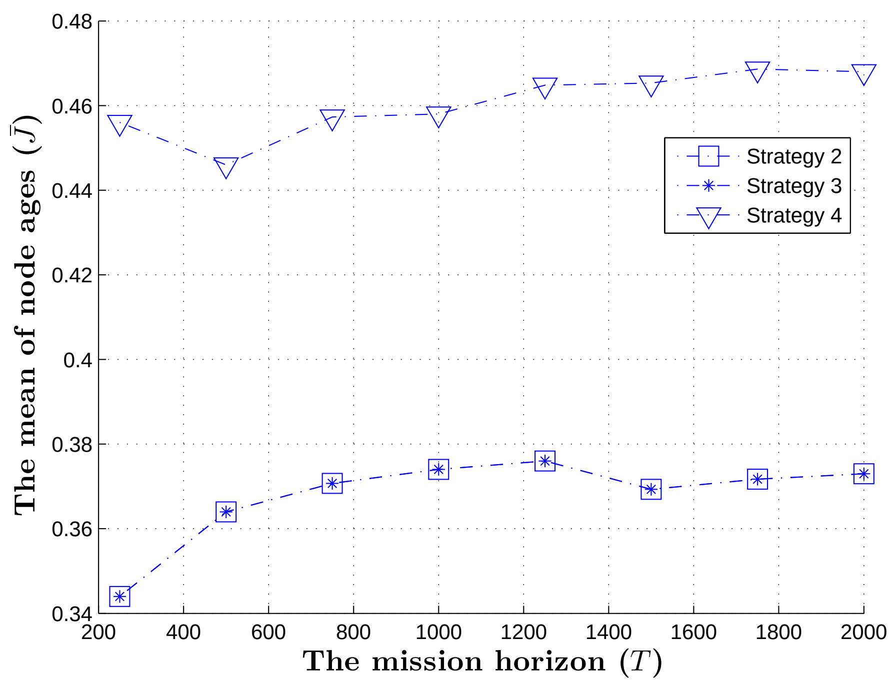


Figure 38: The mean of node ages vs. the mission horizon via Strategies 2,3,4

Strategies 6 and 7 differ from Strategies 4 and 5 by employing the replacement policy for the relocation of agents after an agent removal. The results of both strategies are observed better than the results of Strategies 4 and 5. This shows that utilizing the replacement policy can greatly improve the mission performance.

In addition to the replacement policy, Strategy 6 employs a deterministic return policy, whose results are illustrated in Figures 41 and 42. Note that the performance variability in this strategy is due to the randomly selected agents for the replacements. As seen from Figure 42, the median of $\bar{J}$ stays in the range of $(0.37, 0.38)$. On the other hand, the results of Strategy 7 are illustrated in Figures 43 and 44, which exhibit slightly worse performance than Strategy 6. This is due to the random returns of agents.

Finally, the summary of the results are illustrated in Table 4. The best performance is obtained via Strategies 2 and 3, which are centralized strategies employing approximate dynamic programming. As a decentralized solution, Strategies 6 and 7 perform close to Strategies 2 and 3. Here, it is important to emphasize that the computational complexity of centralized solutions are significantly higher than the decentralized ones. For example, using Strategy 3 in the canonical problem for a mission horizon of 2000 took approximately 33 hours on a 3.40GHz Intel Core i7 PC (for the same scenario, using Strategy 6 or 7 took less than 1 minute on the same PC). Consequently, the simulation results suggest that Strategies 6 and 7 are efficient and local strategies that can achieve energy-aware persistent surveillance.

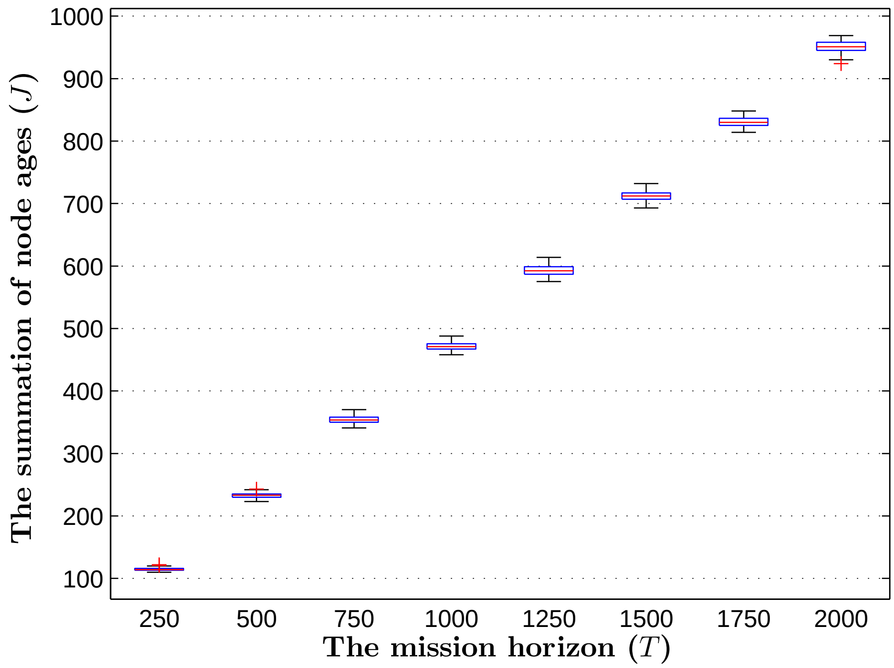


Figure 39: Box plot for the summation of node ages vs. the mission horizon via Strategy 5

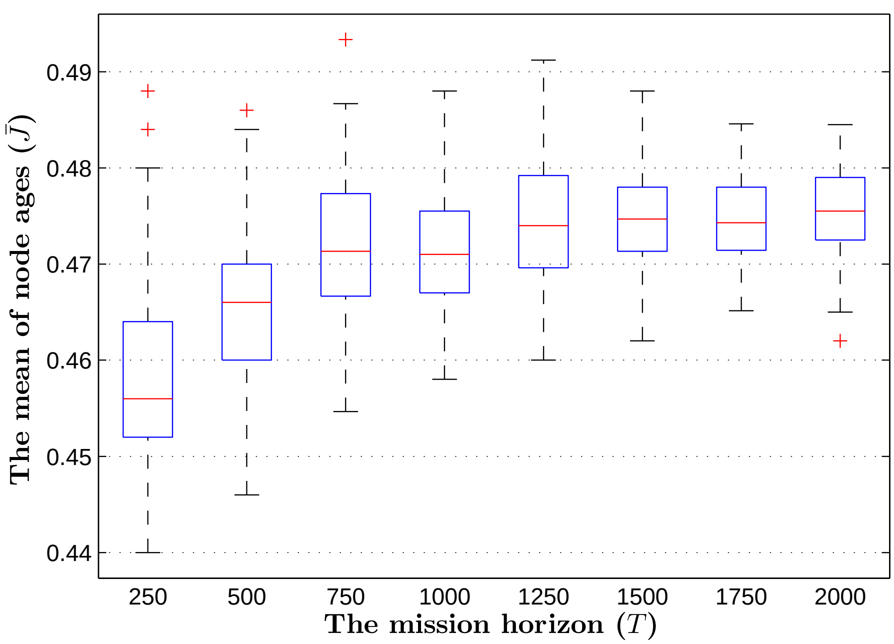


Figure 40: Box plot for the mean of node ages vs. the mission horizon via Strategy 5

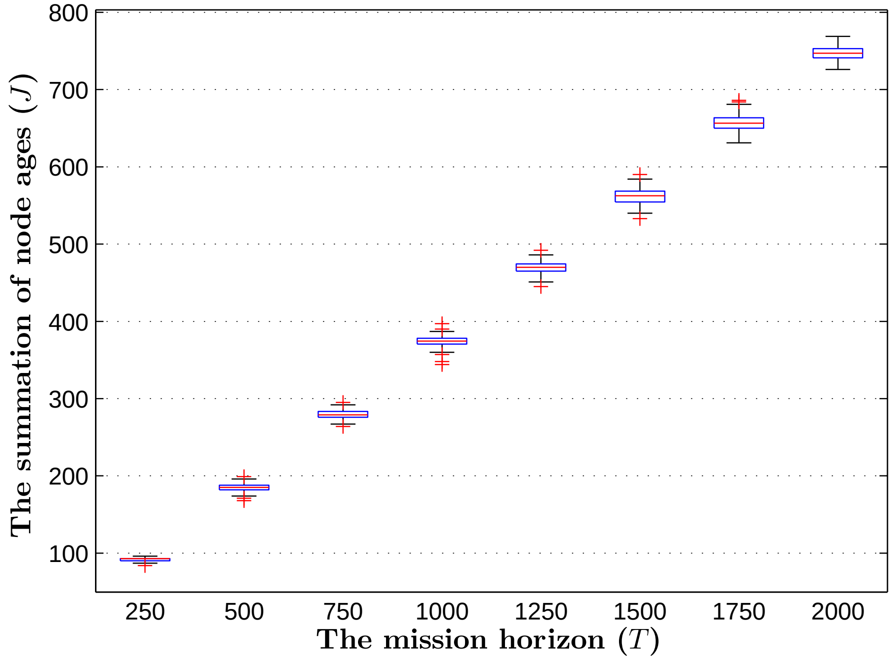


Figure 41: Box plot for the summation of node ages vs. the mission horizon via Strategy 6

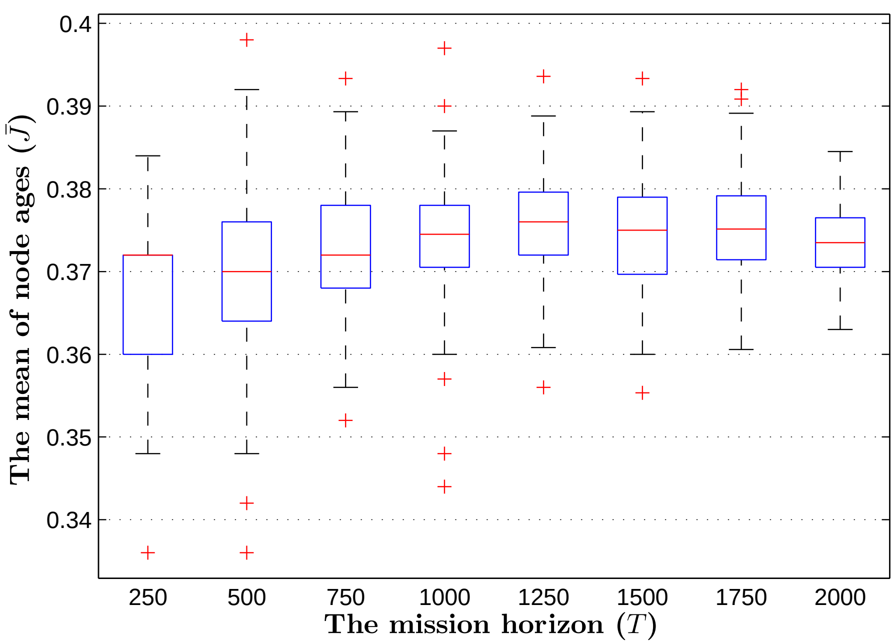


Figure 42: Box plot for the mean of node ages vs. the mission horizon via Strategy 6

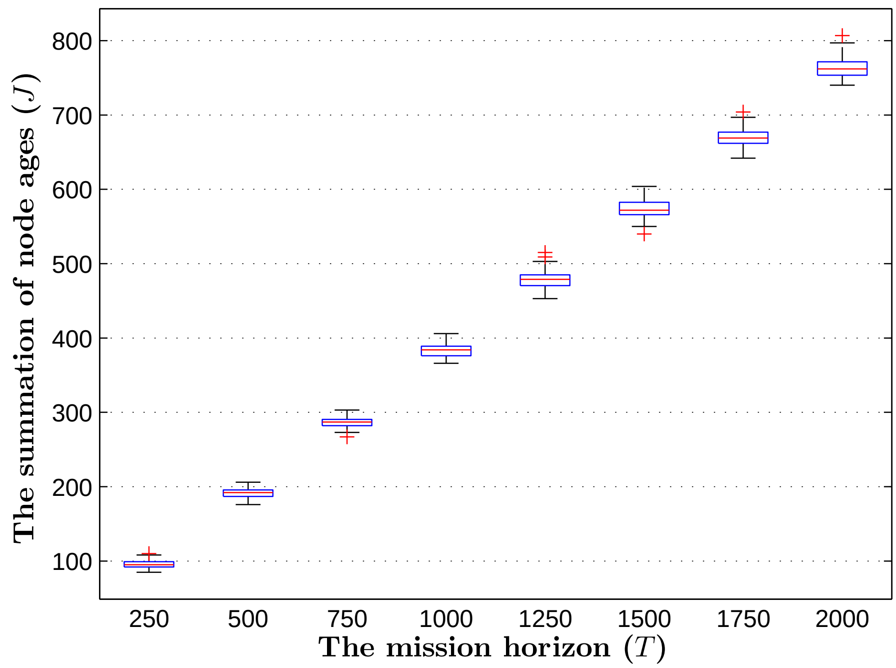


Figure 43: Box plot for the summation of node ages vs. the mission horizon via Strategy 7

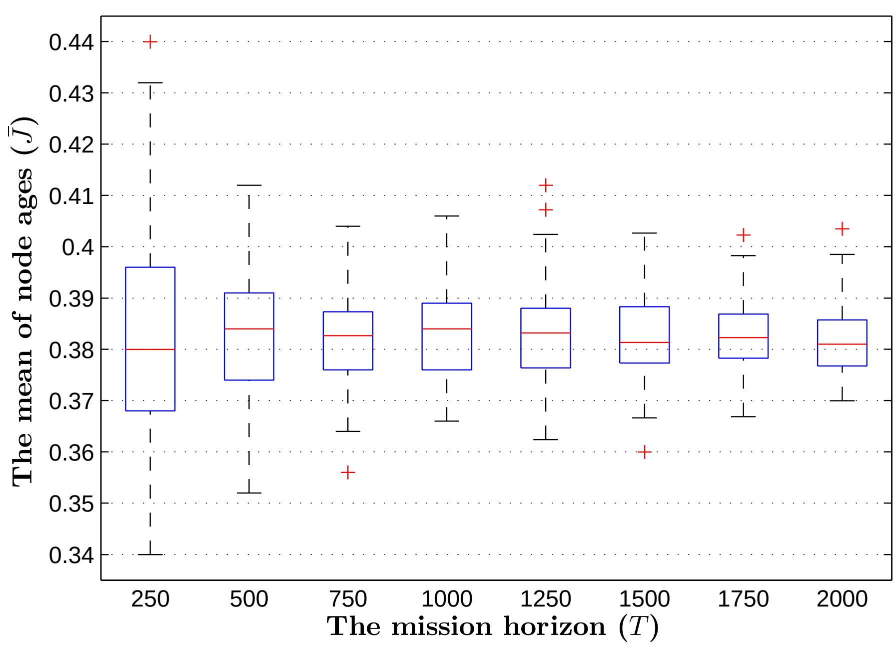


Figure 44: Box plot for the mean of node ages vs. the mission horizon via Strategy 7

Table 4: Summary of the experimented energy-aware strategies.

		Mission horizon (T)							
		250	500	750	1000	1250	1500	1750	2000
J	*Strategy 1*	[58,200]	[156,350]	[253,489]	[333,596]	[486,815]	[578,889]	[653,1046]	[717,1132]
	Strategy 2	86	182	278	374	470	554	650	746
	Strategy 3	86	182	278	374	470	554	650	746
	Strategy 4	114	223	343	458	581	698	820	936
	Strategy 5	[110,122]	[223,243]	[341,370]	[458,488]	[575,614]	[693,732]	[814,848]	[924,969]
	Strategy 6	[84,96]	[168,199]	[264,295]	[344,397]	[445,492]	[533,590]	[631,686]	[726,769]
	Strategy 7	[85,110]	[176,206]	[267,303]	[366,406]	[453,515]	[540,604]	[642,704]	[740,807]
$\bar{\mathbf{J}}$	*Strategy 1*	[0.23,0.80]	[0.31,0.70]	[0.34,0.65]	[0.33,0.59]	[0.39,0.65]	[0.39,0.59]	[0.37,0.59]	[0.36,0.57]
	Strategy 2	0.34	0.36	0.37	0.37	0.38	0.37	0.37	0.37
	Strategy 3	0.34	0.36	0.37	0.37	0.38	0.37	0.37	0.37
	Strategy 4	0.46	0.45	0.46	0.46	0.46		0.47	0.47
	Strategy 5	[0.44,0.49]	[0.44,0.49]	[0.45,0.49]	[0.46,0.49]	[0.46,0.49]	[0.46,0.48]	[0.46,0.48]	[0.46,0.48]
	Strategy 6	[0.34,0.38]	[0.34,0.40]	[0.35,0.39]	[0.34,0.40]	[0.36,0.39]	[0.36,0.39]	[0.36,0.39]	[0.36,0.38]
	Strategy 7	[0.34,0.44]	[0.35,0.41]	[0.36,0.40]	[0.37,0.41]	[0.36,0.41]	[0.36,0.40]	[0.37,0.40]	[0.37,0.40]

CHAPTER VIII

DESIGN SPACE EXPLORATION FOR MULTI-AGENT SURVEILLANCE SYSTEMS

As depicted in the beginning of this thesis, the overall performance of an MAS depends on both the control strategies used by the agents and the design variables of the agents. Until now, the control module of the proposed methodology is presented by introducing effective decentralized control strategies for persistent MAS surveillance. This chapter elaborates the design module of the proposed methodology, whose goal is to conduct a design space exploration for multi-UAV surveillance systems. In the design module, the foundation of the proposed steps are based on the design of experiments, thus the next section presents some preliminaries about this topic. Then, the following sections describe the steps of the design module and show some results for various case studies.

8.1 Design of Experiments Preliminaries

In design, experiments are performed to investigate the characteristics of a system (e.g. [16, 112]). The characteristics, also known as responses, are the results observed from the experiments, and they depend on the controllable and uncontrollable variables. The controllable variables also refer to the design variables of the system, whereas the uncontrollable variables are anything other than the design variables. The experiments are pursued to appropriately analyze the responses according to the objectives of the experiments. For instance, some objectives for the experiments can be as follows (e.g. [84], [93]):

- Determining which design variables are most influential on the response,
- Determining where to set the influential design variables so that the response is almost always near the desired nominal value,
- Determining where to set the influential design variables so that the variability in the response is small,
- Determining where to set the influential design variables so that the effects of the uncontrollable variables are minimized.

Well-designed experiments are important because the results and conclusions drawn from the experiments mostly depend on the manner in which the data was collected. Hence, a framework is introduced for experimental planning as Design of Experiments (DOE) (e.g. [12,50]). Some major steps of DOE are

1. Determining the experimental objectives such as screening, optimization, or robustness test,
2. Defining the controllable (i.e. design) or uncontrollable variables that have a potential effect on the system,
3. Specifying the responses that are the outputs of the system, which will be observed from the experiments,
4. Generating the experimental design (e.g. full factorial design for screening purpose, fractional factorial design for screening or robustness test purposes, or composite designs for optimization purpose).

8.2 *Proposed Steps of the Design Module*

In this section, the steps of the design module shown in Figure 45 are elaborated. Note that the inputs to the design module are the mission specifications and the agent strategies coming from the control module, while the output of the design module is the influential design variables and effective design-control pairs influencing the mission performance. Here, the design variables of an agent refer to the vehicle capabilities such as velocity, endurance, communication range, or several others. Whereas a control strategy is a set of rules that guide agents to achieve efficient persistent surveillance. In order to provide insight into the effects of the design variables and the control strategies on the mission performance, one way is to perform DOE with Monte Carlo simulations. Accordingly, the steps of the design module are as follows.

8.2.1 Step 1: Select a set of design variables and responses

In order to perform a design space exploration, first a set of potentially influential design variables and responses in interest need to be selected. Here, a response corresponds to an operational outcome with respect to a particular set of design variables. In general, there is not a systematic approach for the selection of these variables. Thus, it is a subjective step such that the selected variables are based on the decision-maker's preferences.

8.2.2 Step 2: Create the DOE cases

After the selection of the design variables, the next step is identifying the size of the design space. Note that the dimension of the design space is equal to the number of selected design variables, for which a finite range is selected for to conduct design space exploration. In general, the ranges are determined with respect to a baseline configuration. Accordingly, the design variables of the baseline are varied by $+x\%$ and $-y\%$ to create a design space around them. Note that high values of x and y result in a large design space to explore, which may potentially cause the experimented designs become completely different than the baseline design. In such cases, if particular assumptions for the baseline are made in the simulations, then these assumptions may not be valid for the experimented cases and the simulation results may not capture the actual performance.

After determining the ranges for the design variables, the next step is to generate the experimental design cases withing the selected ranges. There are various options for the generation of the experimental design cases such as full factorial design for screening purpose, fractional factorial design for screening or robustness test purposes, or composite designs for optimization purpose.

8.2.3 Step 3: Replicate the DOE cases for candidate control strategies

The design space exploration proposed in this thesis is different than the existing studies by taking into account multiple control strategies instead of a single strategy. Thus, experimental design cases are replicated for each control strategy developed in the control module.

8.2.4 Step 4: Conduct Monte Carlo simulations

After determining the experiment cases, the next step is performing simulations to evaluate the performance of each case. Since a control strategy may potentially contain some randomness in selecting agent actions and there might exist some uncertainty in the design variables of the agents, the response, i.e. output, of each experiment case is a random variable Thus, the response of each experiment is considered as a sample from the actual response distribution. As a result, repeated simulations are performed to demonstrate the expected performance of each experiment case.

8.2.5 Step 5: Visualize the results

The simulation results are evaluated through various visualization techniques to depict conclusions about influential design variables or effective design-control pairs. At the end of Step 5, if insufficient conclusions are depicted, then a modification might require for the design variables and the control strategies (i.e. the dashed arrows in Figure 45 feed back the control module and the first step of the design module). Otherwise, the conclusions support the requirement analysis as well as the later design stages of a generic design process.

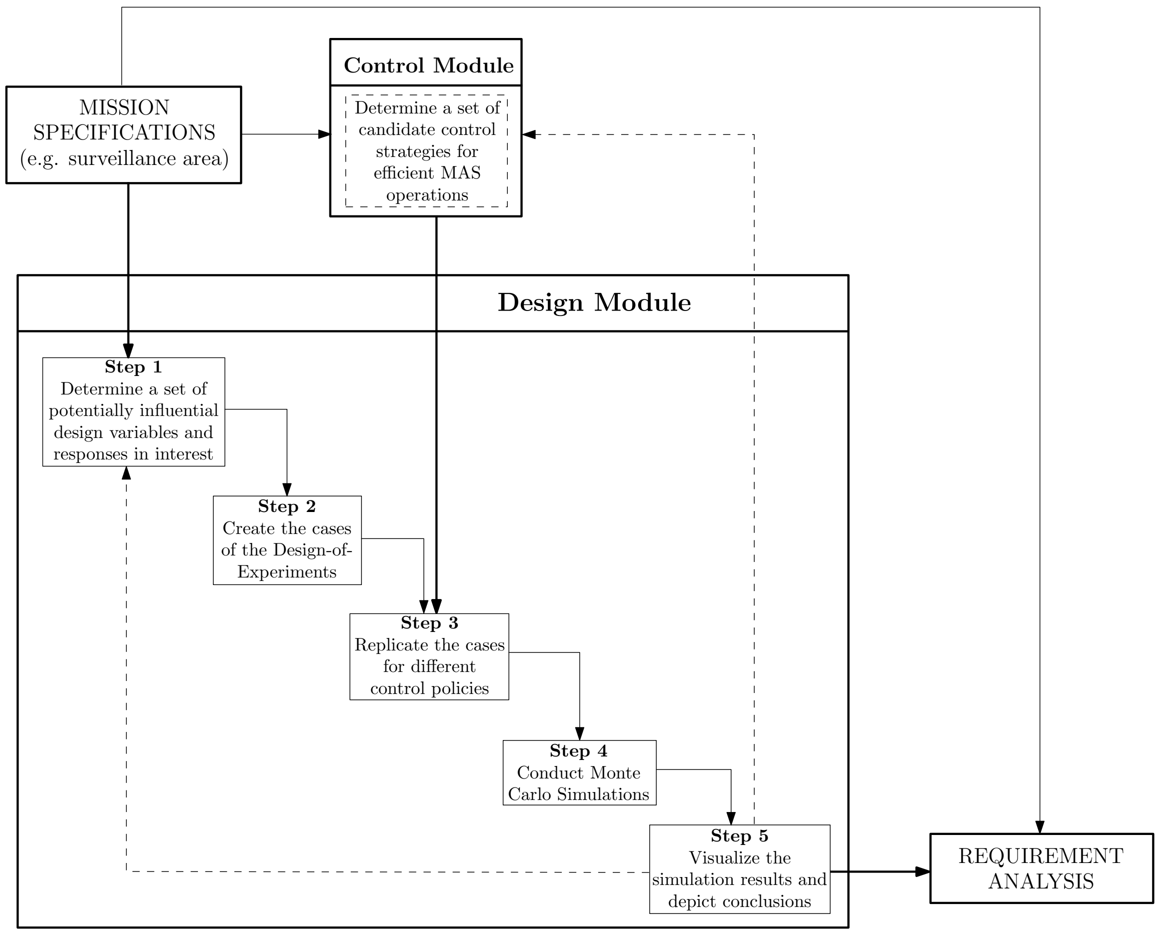


Figure 45: Zooming in the design module of the proposed methodology

8.3 *Implementation of the Design Module*

In the preceding sections, the goals and the details of the design and the control modules have been presented. In this section, the overall methodology is applied to the multi-UAV persistent surveillance problem introduced in Chapter 4. Throughout this section, the scenario description, the modeling assumptions, and the implementation of the design module will be detailed.

8.3.1 Scenario Description

Recently, there is a huge interest in using UAVs for maritime surveillance (e.g. [90]), natural hazard search (e.g. [109]), biological or chemical detection (e.g. [10]), to name a few. In this thesis, a surveillance scenario is used to demonstrate the steps of the design module. The surveillance area is assumed as a square region with a size of $20\,km \times 20\,km$. This area contains a base and some monitoring regions as illustrated in Figure 46. In this scenario, a monitoring region may represent an urban zone with high population or the coordinate of a valuable immobile asset. On the other hand, the base is the ground station that aims to collect information from the monitoring regions via remote sensing. Accordingly, it increases its situational awareness.

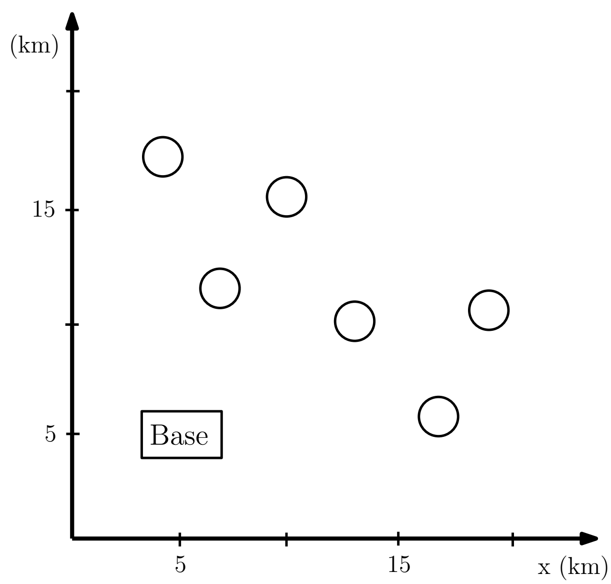


Figure 46: A canonical surveillance scenario including a base and some monitoring regions.

In this scenario, there are various options to increase situational awareness. For instance, a few high technology UAVs (e.g. Global Hawk) can operate on the surveillance area. Such large and complex UAVs have long endurance, high velocity, and good sensor capabilities. However, they are expensive so that it is not economically viable to use a bunch of them. Alternatively, another option is using low technology UAVs (e.g. Raven RQ-11), which may have short endurance, low velocity, and less sensor capabilities. Nonetheless, it becomes feasible to use a large group of them due to being cheap. The scenario considered in this thesis contains a group of small UAVs such as Raven RQ-11 to perform persistent surveillance on the desire region.

8.3.1.1 Specifications of AeroVironment Raven RQ-11

The Raven RQ-11 is a light-weight small UAV that is designed by AeroVironment for rapid deployment and high mobility for both military and commercial applications requiring low-altitude reconnaissance, surveillance, and target acquisition [2]. The Raven does not require a runway to take-off. It is a hand-launched system. It can be used manually or programmed for autonomous operations. A major feature of the Raven is to be capable of delivering real time situational awareness. A picture of the Raven RQ-11B is displayed in Figure 47, and its specifications are illustrated in Table 5.

Figure 47: Raven RQ-11B [1]

Table 5: The specifications of a Raven RQ-11B [2, 89]

Speed	32 – 81 km/h
Endurance	60 – 90 minutes
Maximum communication range	10 km
Operating altitude	30 – 152 m
Wing span	1.4 m
Length	0.9 m
Weight	1.9 kg
Cost	$34000 (in FY2012)

8.3.2 Modeling a Multi-UAV System

In this study, each agent and the base represent an individual entity. Thus a proper technique to model the overall system by considering the capabilities and the behaviors is the agent-based modeling. As such, the base corresponds to an immobile agent that is responsible to track the ages of the monitoring points (note that *age* is the time duration the base has not received any information from a particular point). Each UAV corresponds to a mobile agent that visits the monitoring points to send information back to the base. Throughout the section, *agent* is referred to a UAV, and the base is denoted by its individual name.

In the case study, the surveillance mission contains one base, n number of monitoring zones, and m number of identical UAVs. Let each UAV have a discrete dynamics with two degrees of freedom. Moreover, each UAV has two flight phases such as *cruise* to travel from one point to another and *loiter* to monitor a particular zone. Accordingly, the modeling assumptions of the overall system are listed as follows:

- Each agent can send information back to the base directly or indirectly. *Direct communication* implies that the distance between the agent and the base is less than the agent's communication range. *Indirect communication* implies that an agent sends information back to the base as long as it is connected to the base through some other agents. Accordingly, agents forward messages (containing monitoring information) among them.

- The base has a dashboard that counts the ages of the monitoring zones at each time step. Accordingly, whenever an agent returns to the base for energy replenish, the base assigns a target zone to the UAV according to the greedy assignment strategy presented in Chapter 7.

- During the mission, each agent sends information back to the base to minimize the age of the monitoring zone. Moreover, it tracks its individual states (e.g. remaining energy) to take a return action for energy replenish. If an agent leaves the surveillance area, it can also initiate a sequence of replacements to maintain the connectivity of the communication network according to the constrained δ-MPS presented in Chapter 6. Consequently, any agent takes an individual action to stay at/leave/relocate on the surveillance area. The base does not guide agents for their actions.

- Each agent represents a notional Raven RQ-11B, which is a battery-powered aircraft. When an agent returns to the base for energy replenish, it *swaps batteries* (instead of recharging) to quickly return to the surveillance area. Accordingly, it is assumed that the base has enough resources to perform battery swaps whenever an agent returns to the base.

- Each agent cruises at a high velocity and loiters at a lower velocity. Therefore, the power required varies in cruise and loiter modes, which creates a variation in the energy consumption.

- In addition to the variability in energy consumption for cruise and loiter, some stochastic effects are also taken into account. Here, a stochastic effect can be due to the possible altitude variations, which may possibly influence the power required due to the air density change. Moreover, some uncertainty can also exist in the propulsion system, which may cause some slight changes in the energy consumption. Hence, the energy consumption is modeled as

$$f(t+1) = f(t) - g(\Delta f_{cruise}, \Delta f_{loiter}) + w, \tag{27}$$

where $f(t)$ is the remaining energy at time t, $g(\Delta f_{cruise}, \Delta f_{loiter})$ denotes the energy consumption in cruise or loiter, and w is a random variable uniformly picked from the range of $(-0.1\Delta f_{cruise}, 0.1\Delta f_{cruise})$

8.3.3 Demonstrating the Steps of the Design Module

Based on the depicted scenario and the modeled multi-UAV system, the proposed steps of the design module are implemented as follows:

Step 1: The persistent surveillance problem addressed in this thesis is a complex problem involving a large number of variables affecting the overall performance. In the case study, five major design variables are initially selected. The first design variable is selected as the maximum *velocity* of a UAV, which directly affects the time required to travel from one point to another on a surveillance area. The second design variable is chosen as the *endurance* of a UAV because it determines the overall flight time without any energy replenish. The third variable is selected as the *radius of communication*, which greatly impacts the topology of the overall communication network. Accordingly, an increase in the radius of communication improves the connectivity of the network topology so the return of a UAV to the base likely causes a disconnection.

This thesis addresses the persistent multi-UAV surveillance problem, where the objective is to maximize the situational awareness. In Chapter 1, it has been presented that maximizing situational awareness is equivalent to minimize the objective function in (7), which is the summation of the node ages along the mission horizon. Note that J is an increasing function with respect to time. In order to provide am more generalized conclusion, the response of the experimented cases is selected as the time normalized of J, which particularly represents the *average degradation in situational awareness* during the mission. Consequently, the case study contains only one response, which is $\bar{J}$.

Step 2: The specifications of the Raven RQ-11 are illustrated in Table 5. Based on these values, the ranges of the design variables in the DOE study are selected as in Table 6. The velocity and endurance ranges are chosen according to the capabilities of the current Raven. Due to the potential improvements in the communication technologies, the maximum value of the communication range is assumed a bit higher than the current capability.

Table 6: The ranges of the design variables of a notional Raven RQ-11B for the DOE study

V_{max}	Velocity	$0.83-1.33 \quad (50-80)$	km/min (km/h)
E	Endurance	$60-90$	minutes
R_{comm}	Maximum communication range	8-12	km

After determining the ranges, the next step is to choose the experiment cases. In this study, it is desired to explore the overall space within the determined ranges, so a latin-hyper cube design is used to generate the DOE cases.

Step 3: In this thesis, the control module has developed four decentralized strategies for persistent multi-agent surveillance as the Strategies 4, 5, 6, and 7 introduced and tested in Chapter 7. Note that Strategy 4 includes the greedy node assignment strategy, a deterministic return policy, and no relocation policy. Strategy 5 differs from Strategy 4 by using a randomized return policy instead of the deterministic one. Strategy 6 comprises the greedy node assignment strategy, a deterministic return policy, and the constrained 1-MPS as a relocation policy. Finally, Strategy 7 differs from Strategy 6 by using a randomized return policy instead of the deterministic one. In order to be descriptive in the plots of the experimental results, the strategies considered in this chapter will be denoted as follows:

- *DetnoReplace*: Strategy 4
- *RandnoReplace*: Strategy 5
- *DetReplace*: Strategy 6
- *RandReplace*: Strategy 7

Accordingly, the DOE cases generated in the step 3 are replicated four times for the developed control strategies.

Steps 4 & 5: A MATLAB script is written to perform the Monte Carlo simulations. Accordingly, the design space exploration contains four set of DOE study based on the preceding assumptions. The simulation results are displayed by some scatter and contour plots, which will be elaborated on the next section.

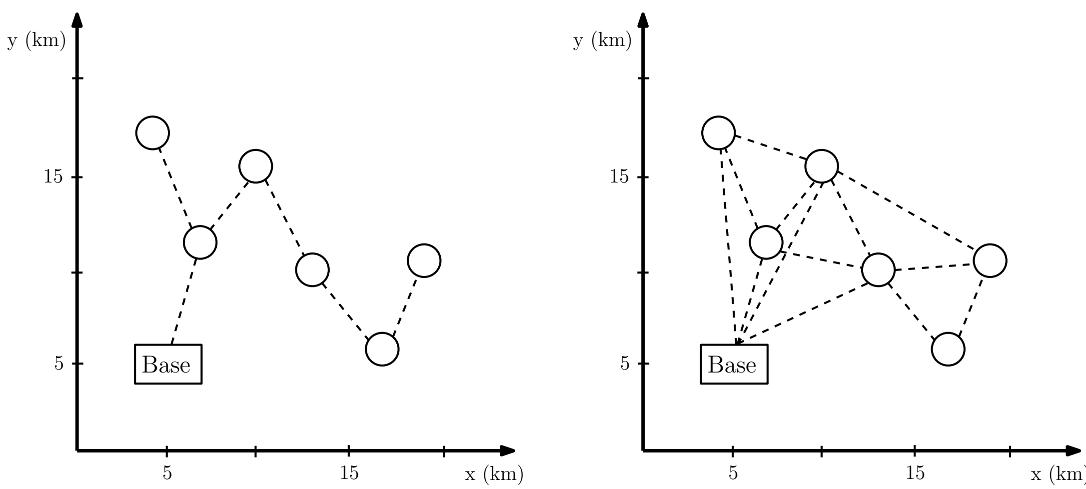
8.4 Results

This section presents the results of two experiments. The first experiment considers a scenario that contains n UAVs and n monitoring zones. The second experiment considers a more generalized scenario that contains n UAVs and m monitoring zones for $n < m$. The objectives of the experiments are to demonstrate the proposed steps of the design module, to investigate the influential design variables and the control strategies on the mission performance, and to explore the interdependency between the design variables and the control strategies.

8.4.1 Experiment 1: Persistent Surveillance with n UAVs monitoring n Zones

In the first experiment, the scenario contains 1 base, 6 UAVs, and 6 monitoring zones. Based on Table 6, the changes in the communication ranges generate different communication topologies. For instance, if all UAVs are on the surveillance area and each one has the minimum or the maximum communication ranges (i.e. 8 or 12 *km*), then the communication networks are shown in Figures 48a or 48b, respectively.

As mentioned before, each UAV is represented by three design variables, i.e. velocity, endurance, communication range. In the first experiment, the DOE size is selected 300 (implying that 300 cases are generated through a latin-hyper cube design). Each case in the DOE is repeated 20 times, and each repetition is executed for 3000 time steps (i.e. mission horizon). In this experiment, one time step in the simulations corresponds to two minutes of a UAV operation. Thus, the simulations represent a persistent mission performed for 4 days. As depicted before, all cases in the DOE are executed under 4 different control strategies (i.e. *DetnoReplace*, *RandnoReplace*, *DetReplace*,*RandReplace*). To this end, a MATLAB script was written to conduct the Monte Carlo simulations. For an individual control strategy, each experiment case, which runs for 3000 time steps and repeated 20 times, took approximately 2.5 minutes on a 3.40GHz Intel Core i7 PC.



(a) Communication network with min R_{comm} (b) Communication network with max R_{comm}

Figure 48: Changes in the communication network for various R_{comm} in the Experiment 1.

8.4.1.1 Influential Design Variables

Based on the presented DOE study, this section discusses the influential design variables and control strategies on the system response (i.e. $\bar{J}$ refers to the average degradation in situational awareness). After performing the Monte Carlo simulations, the results are presented by some scatter plots as in Figures 50, 51, 52, and 53. A scatter plot matrix consists of multiple scatter plots, each of which contains all the experimented DOE cases. Note that a DOE case corresponds to some particular values of the selected design variables as illustrated by the highlighted point in Figure 49.

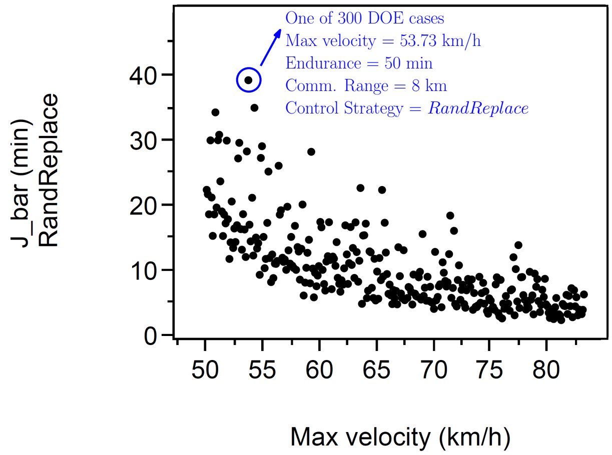


Figure 49: Zooming in a box in a scatter plot matrix.

In Figure 50, the y-axis of each scatter plot displays the objective function, which is the average degradation in situational awareness ($\bar{J}$). In other words, it refers to the average age observed in a monitoring zone during the mission, and its unit is in minutes. Accordingly, minimizing the objective function is desired because small values of $\bar{J}$ imply that the monitoring zones do not become unoccupied for long periods of time. The distribution of the points in each plot illustrates the influence of the corresponding design variable and the control strategy on $\bar{J}$. For instance, the scatter plots corresponding to V_{max} in Figure 50 show that the maximum velocity has a significant effect on the objective function because its increase greatly reduces $\bar{J}$ for all the control strategies. On the other hand, the increased values of endurance and communication range seem favorable to decrease the objective function.

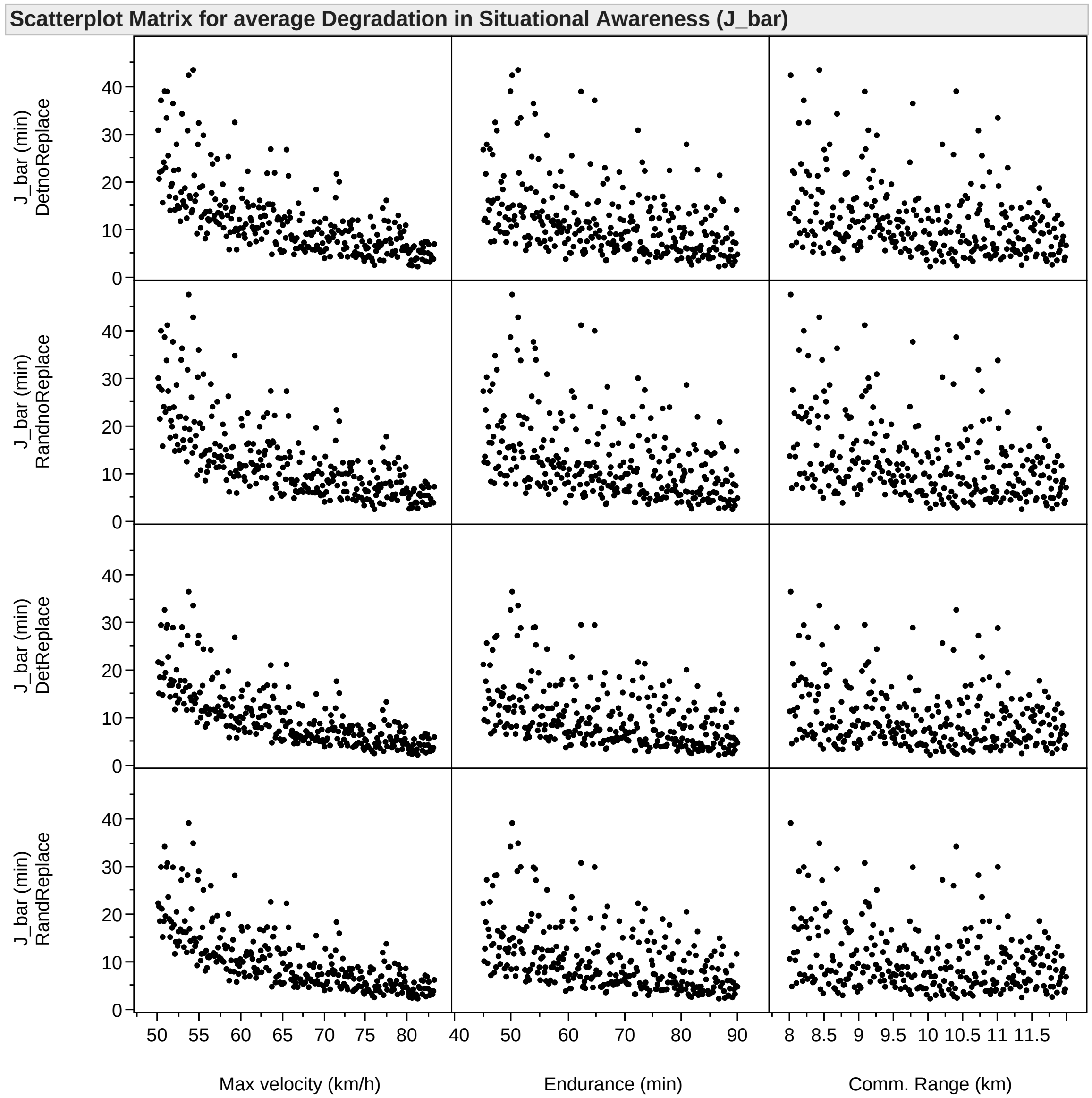


Figure 50: The scatter plot matrix displaying the objective function ($\bar{J}$) with respect to three design variables and four control strategies used in Experiment 1.

Suppose that there are some constraints for the maximum values of the communication range and the endurance as 10 *km* and 70 minutes, respectively. These constraints are displayed in Figure 51 as vertical lines to the corresponding boxes. Let a desired value for $\bar{J}$ not exceed 15 minutes, which is shown as the horizontal lines in Figure 51. Based on these constraints and the objective function's upper limit, the design points are filtered. The results indicate that high values of velocity become crucial to satisfy low values of $\bar{J}$ with the assumed constraints in endurance and communication range. This result can be seen by the highlighted points in Figure 51. Accordingly, the results in Figure 51 support that the velocity of a UAV is a significant design variable on the mission performance, and its higher values are favorable to minimize the objective function.

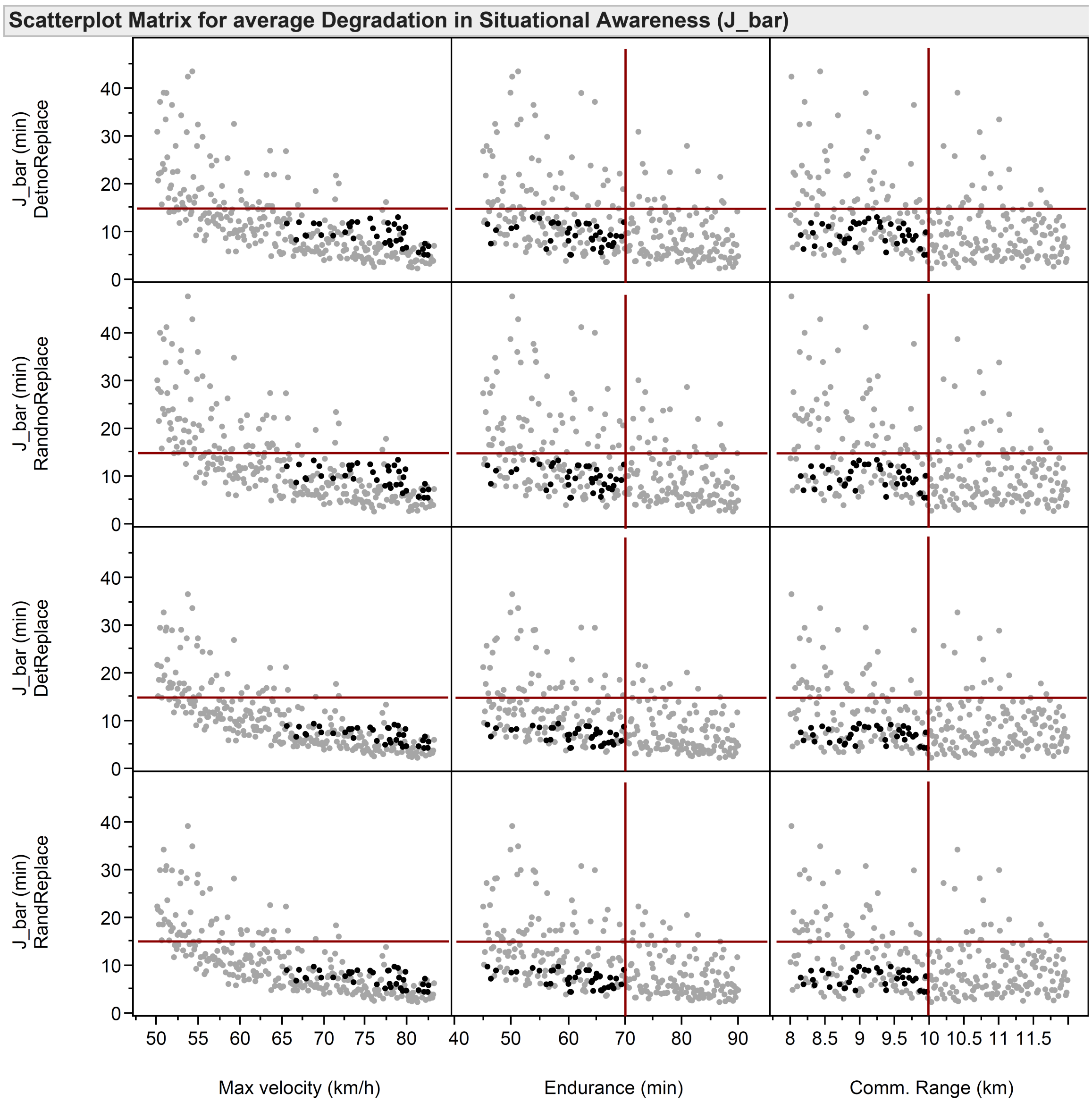


Figure 51: Highlighted points represent short endurance and short comm. range UAVs, which can obtain small values for the objective function ($\bar{J}$) by increasing velocity.

Assume that an upper bound for the objective function is chosen as 15 minutes, and Figure 52 illustrates the filtered points satisfying $\bar{J} \leq 15$ according to *DetReplace* and *RandReplace* (control strategies using the replacement policy). The results show that some particular design cases fail the mission by using *DetnoReplace* and *RandnoReplace* (illustrated by the highlighted points above the constraint line). On the contrary, the same particular design cases accomplish the mission by executing *DetReplace* or *RandReplace*.

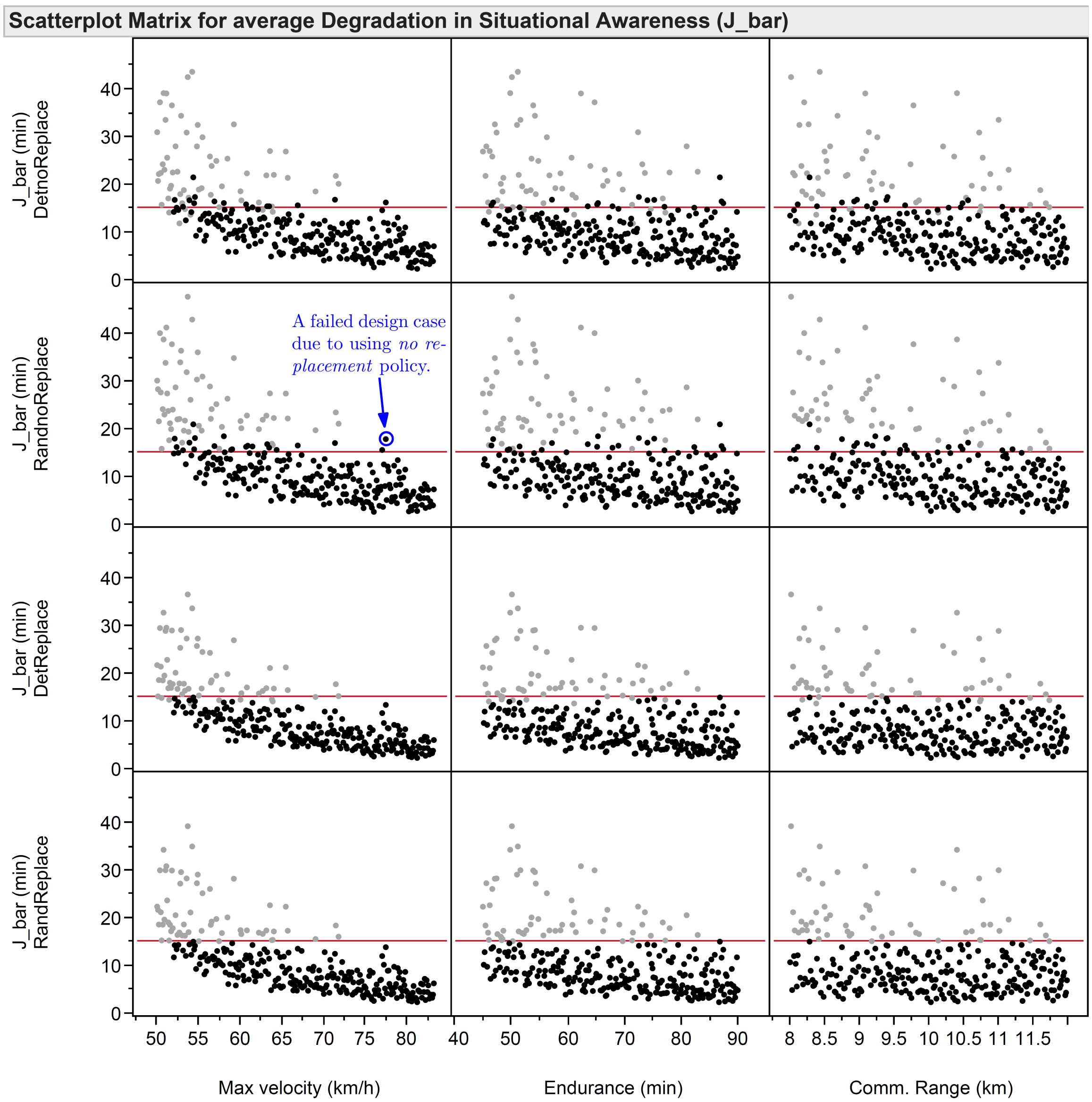


Figure 52: The highlighted points accomplish the mission by using the control strategies including the replacement policy.

The final scatter plot for the Experiment 1 is illustrated in Figure 53, where a desired value for $\bar{J}$ is set to 15 minutes. The design cases corresponding to $V_{max} \leq 65\ km/h$ with a close value of $\bar{J}$, i.e. the highlighted points, accomplish the mission by using *DetReplace* or *RandReplace*. On the contrary, the same design cases likely fail if *DetnoReplace* or *RandnoReplace* are utilized. Hence, the results support that selecting design variables based on a specific control strategy may result in poor performance under a different control strategy.

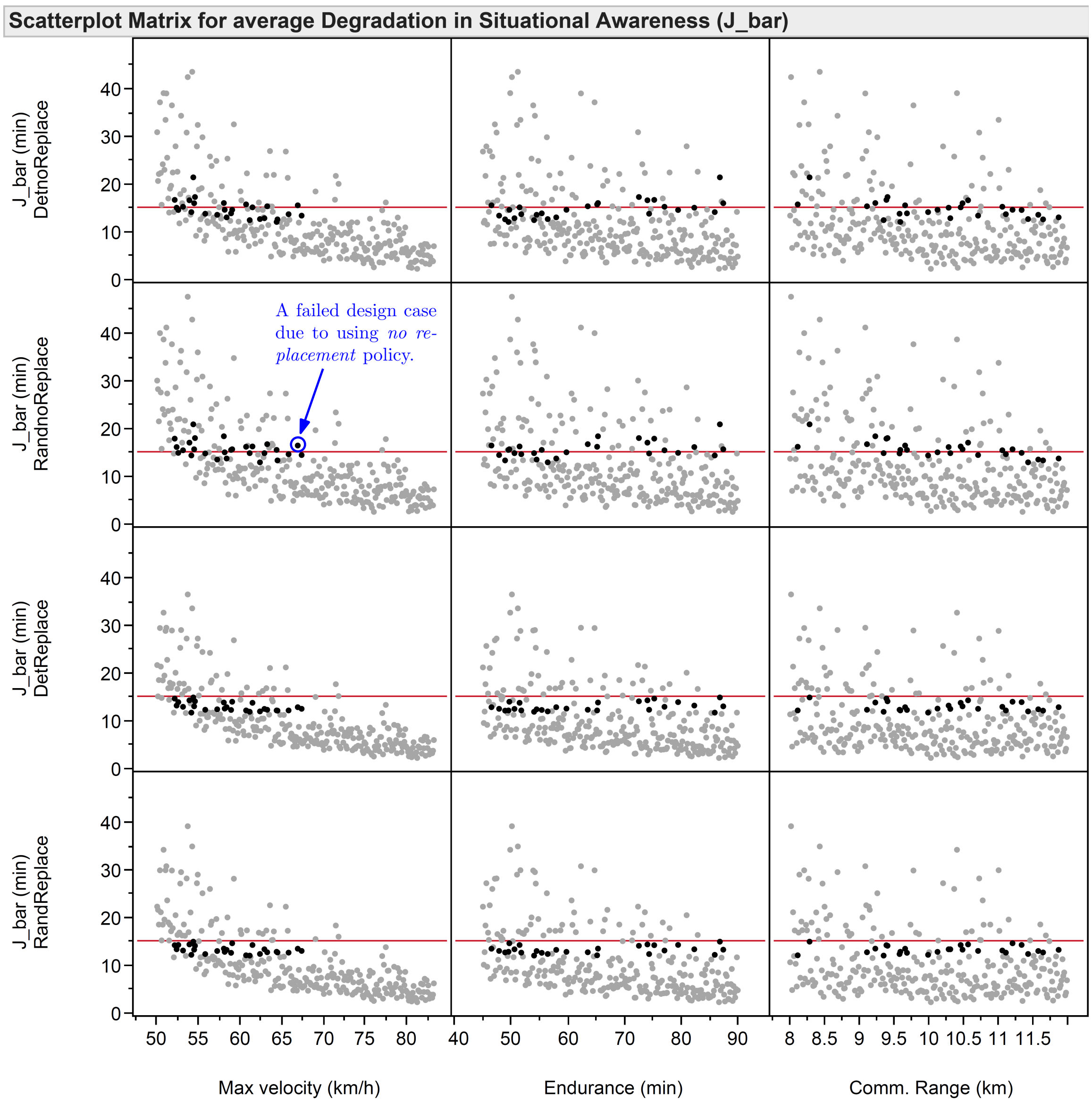


Figure 53: For the highlighted designs, using *DetnoReplace* and *RandnoReplace* likely violates the constraints whereas *DetReplace* and *RandReplace* do not cause a violation.

8.4.1.2 *Influence of the Control Strategies on the Mission Performance*

In this section, the influence of the control strategies on the mission perofrmance are discussed via some contour plots in Figures 54, 55, 56, 57, 58, and 59. In these figures, the objective function $\bar{J}$ is displayed by various filled colored contours with respect to a pair of design variables on the x- and y-axises. In all of the figures, the black regions represent the high values of $\bar{J}$ whereas the white regions indicate the low values. For instance, the color patterns in Figure 54 indicate the performance variation with respect to the maximum velocity and the endurance for different control strategies.

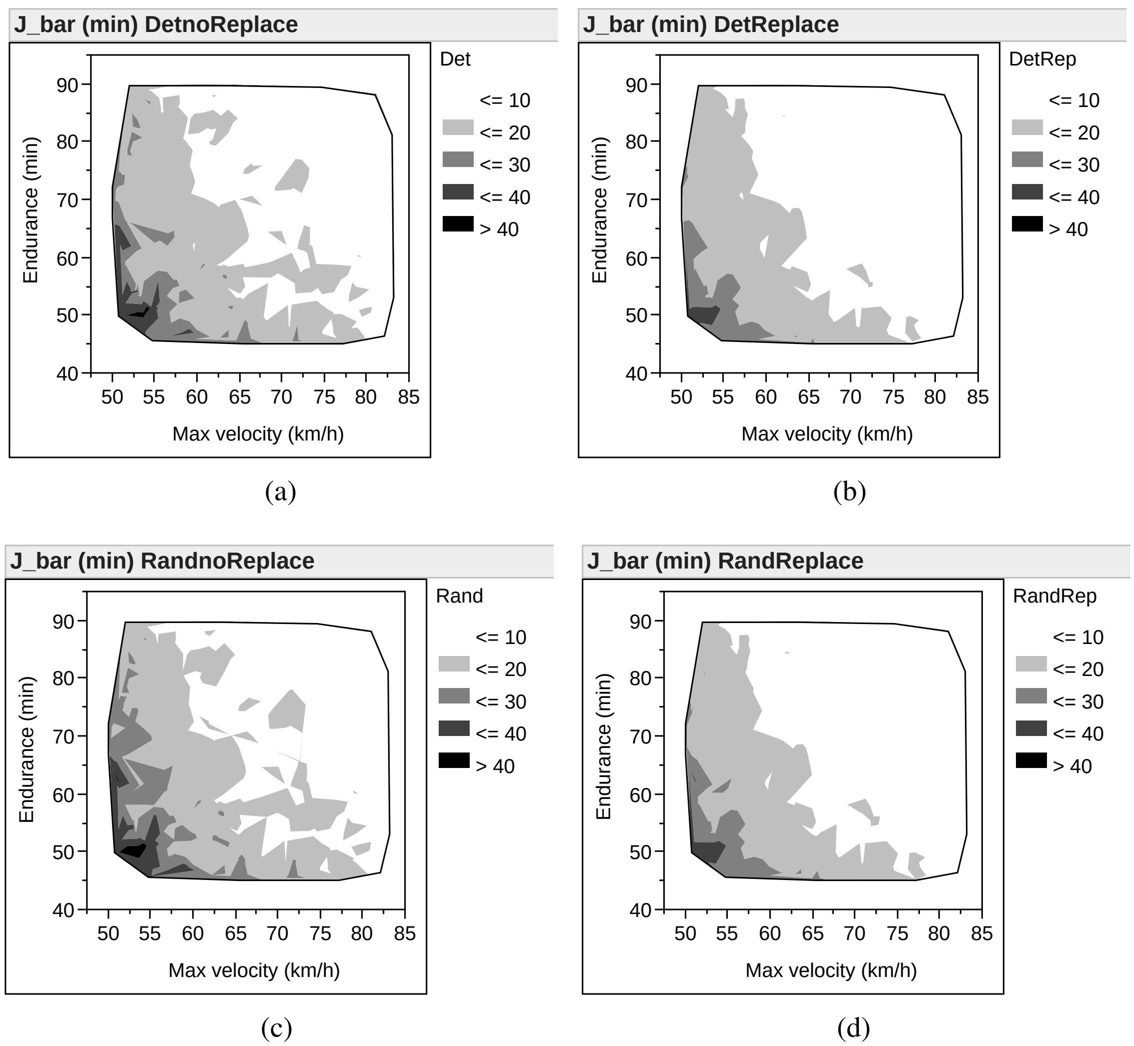


Figure 54: Contour plots for the objective function $\bar{J}$ with respect to the maximum velocity and endurance in Experiment 1.

The results in Figure 55 show that the design cases having lower values of velocity and shorter endurance most likely perform poor (indicated by the dark regions in the highlighted circle). The same figure also shows that utilizing replacement policy (as in *DetReplace* and *RandReplace*) significantly reduces the dark patterns, which result in significant performance improvement. Moreover, when the highlighted circles in Figures 55a and 55b are compared with each other, it is observed that the randomized return policy causes a slight performance degradation (increase of dark regions in Figure 55b when compared to Figure 55a).

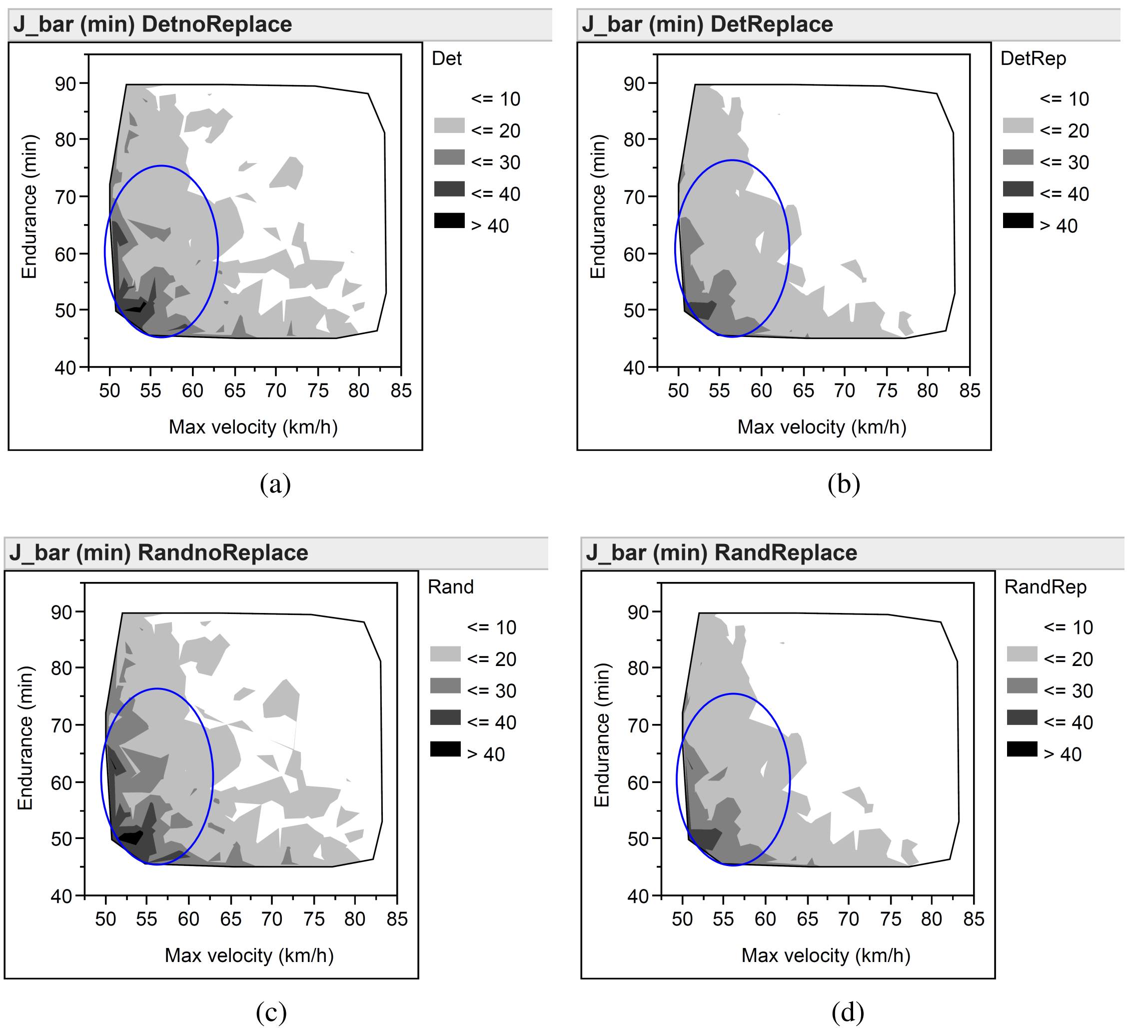


Figure 55: Low velocity and short endurance cause poor performance (high values of $\bar{J}$) in Experiment 1.

Finally, the results in Figure 56 show that high values of velocity and long endurance are desirable to obtain low values of $\bar{J}$ in almost all control strategies. Like the previous case, utilizing the replacement policy (in *DetReplace* and *RandReplace*) improves the performance as seen by the disappearance of gray regions in the highlighted circles.

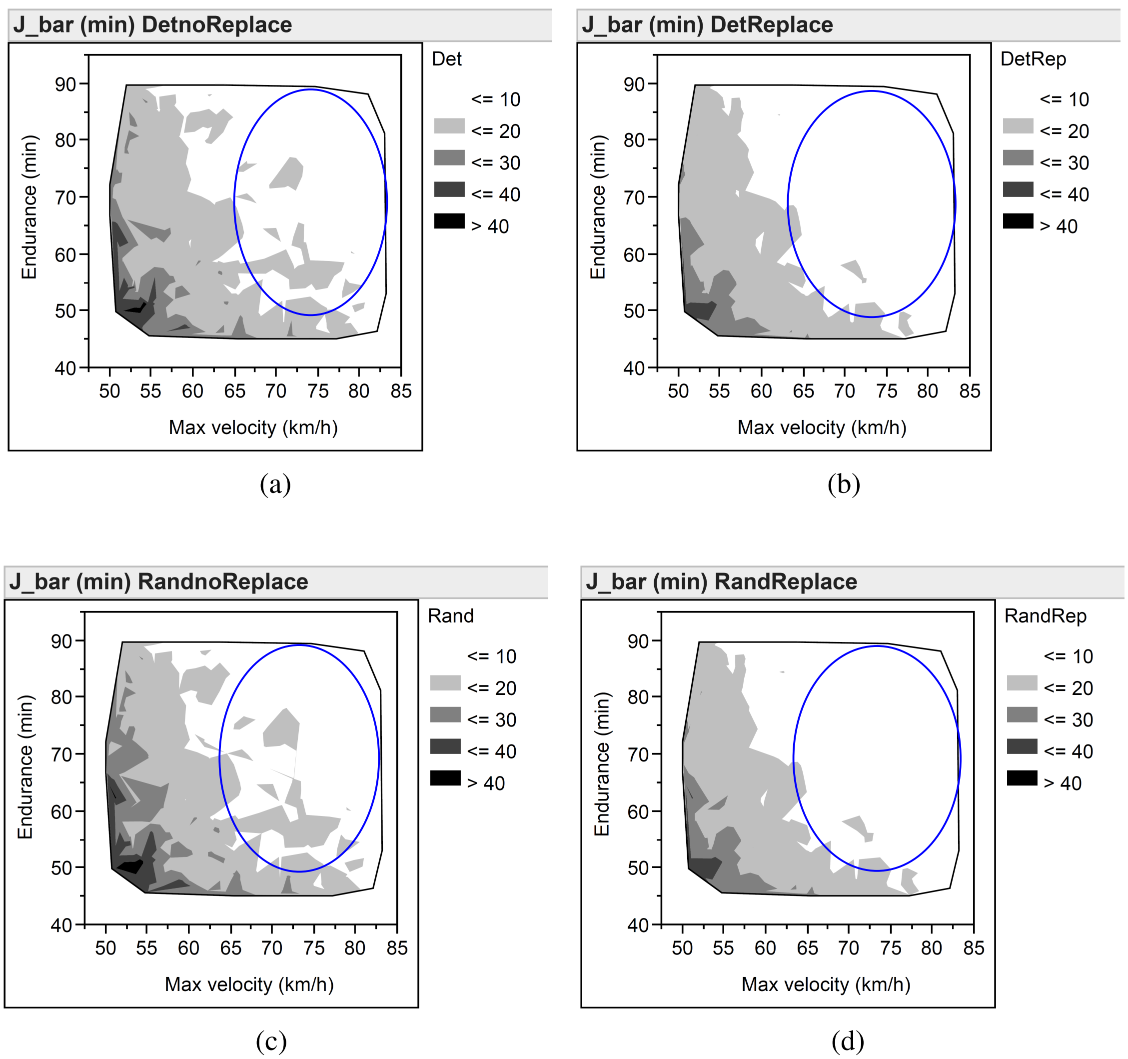


Figure 56: The performance improvement via utilizing the replacement policy as seen by the gray region disappearance in the highlighted circles.

The next set of contour plots in Figures 57, 58, and 59 demonstrate on the performance variation with respect to the maximum velocity and the communication range. Like the previous plots, the dark regions indicate high values of objective function whereas the white regions represent the low values.

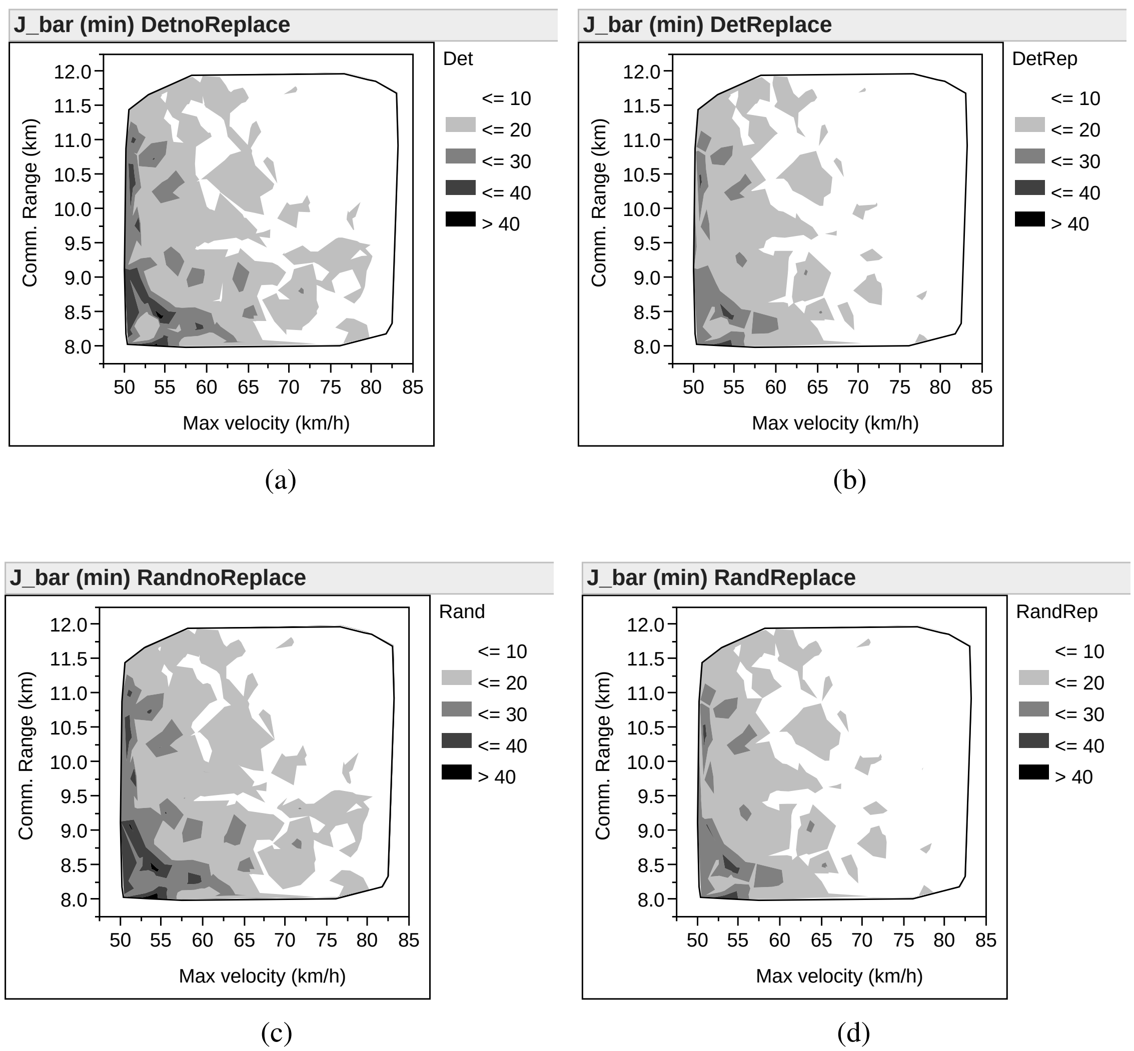


Figure 57: Contour plots for the objective function $\bar{J}$ with respect to the maximum velocity and the communication range in Experiment 1.

The results in Figure 58 show that the design cases having low velocity and short communication range perform poor as seen by the majority of the dark regions in the highlighted area. An interesting result is the significant impact of low velocity on the mission performance. Accordingly, if the design cases have very low velocity (around 50 km/h), the objective function almost always has a high value regardless of the communication range. Furthermore, the design cases utilizing the replacement policy result in improved performance as seen from the reduction of dark patterns in Figures 58c and 58d.

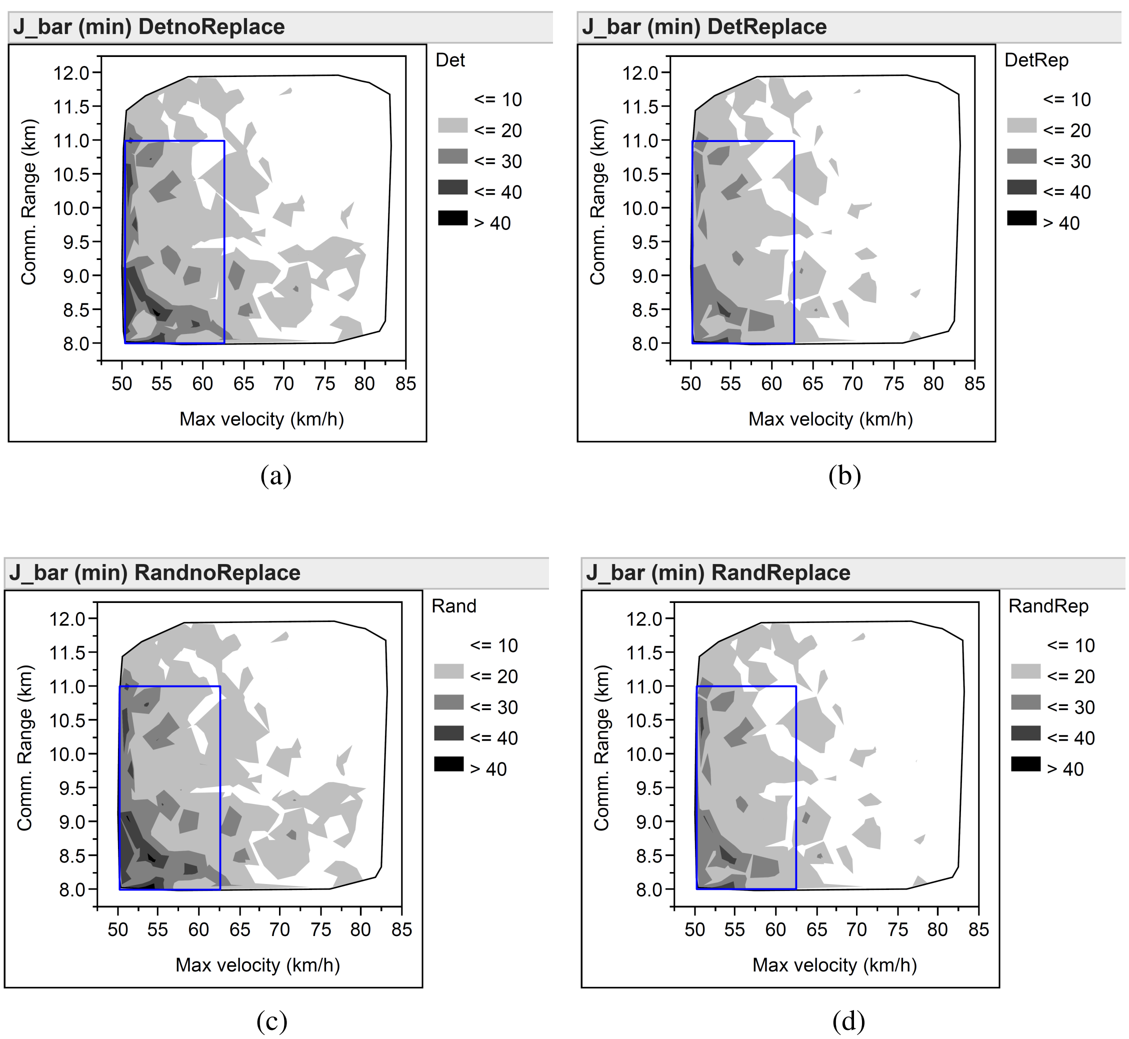


Figure 58: Low velocity causes poor performance independent to the changes in the communication range in Experiment 1.

Finally, the results in Figure 59 indicate that utilizing the replacement policy reduces the sensitivity of the performance to the changes in the communication range. For instance, consider a design case having velocity around $65-70$ km/h. Even though the communication range is very low (close to $8-8.5$ km), the design cases using the replacement policy likely result in a desired performance ($\bar{J} \leq 15min$). On the other hand, for the same design cases, it is likely to obtain a poorer performance as seen from the presence of gray regions in the highlighted area in Figures 59a and 59b.

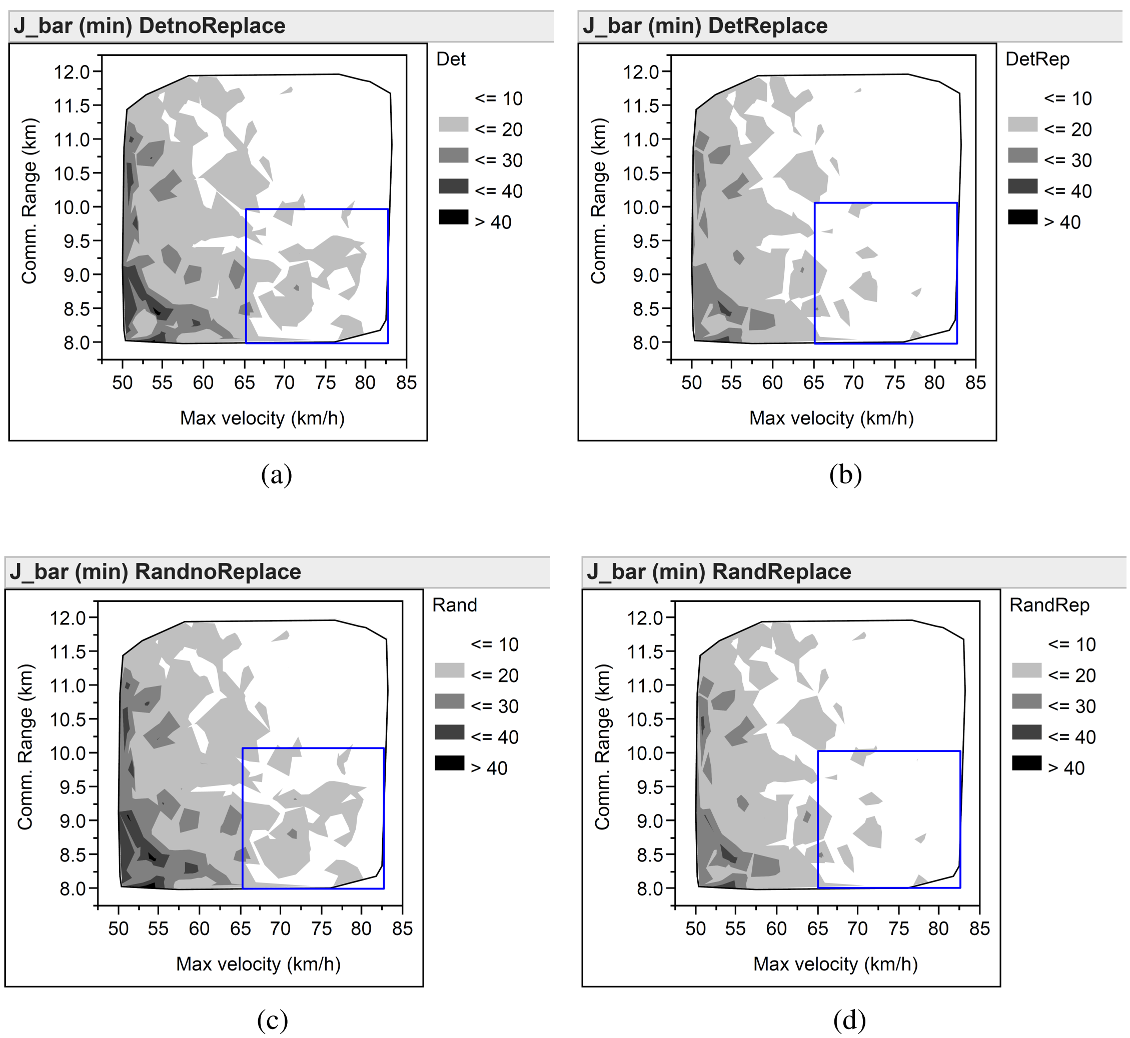


Figure 59: Reducing the sensitivity of the objective function ($\bar{J}$) to the changes in the communication range by using the replacement policy.

Results of the Experiment 1: For a surveillance scenario including n number of UAVs and n number of monitoring zones, the following observations are obtained:

- The most influential design variables are the maximum velocity and the endurance.
- High velocity and long endurance vehicles are observed efficient in the considered scenario.
- The radius of communication has an effect on the overall performance such that its higher values improve the mission performance.
- The strategies using the replacement policy greatly impact the resulting performance of the multi-UAV group, whereas the strategies using the randomized return policy do not exhibit a significant performance change.

8.4.2 Experiment 2: Persistent Surveillance with n UAVs monitoring m Zones

The second experiment differs from the first experiment by assuming a different surveillance scenario, where the number of UAVs is less than the number of monitoring zones. In particular, the case study contains 3 UAVs and 4 monitoring zones. As such, there exist 64 possible configurations the UAVs can monitor the regions. However, some of these possible configurations may lead to disconnection from the base. Hence, some feasible configurations with connected communication network are illustrated in Figure 60.

The results of the previous experiment show that the replacement strategy has a significant effect on improving the mission performance. Moreover, the same results indicate that the randomized return policy has not presented a positive effect on the mission performance. Actually, it has caused a slight performance loss as seen from 55b, which has more dark regions than Figure 55a. Note that the scenario considered in the first experiment was containing equal number of UAVs and the monitoring zones.

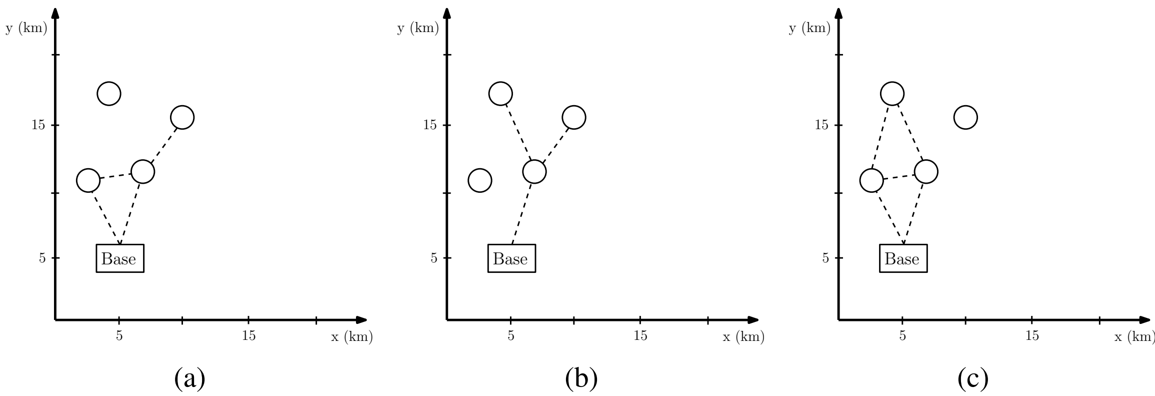
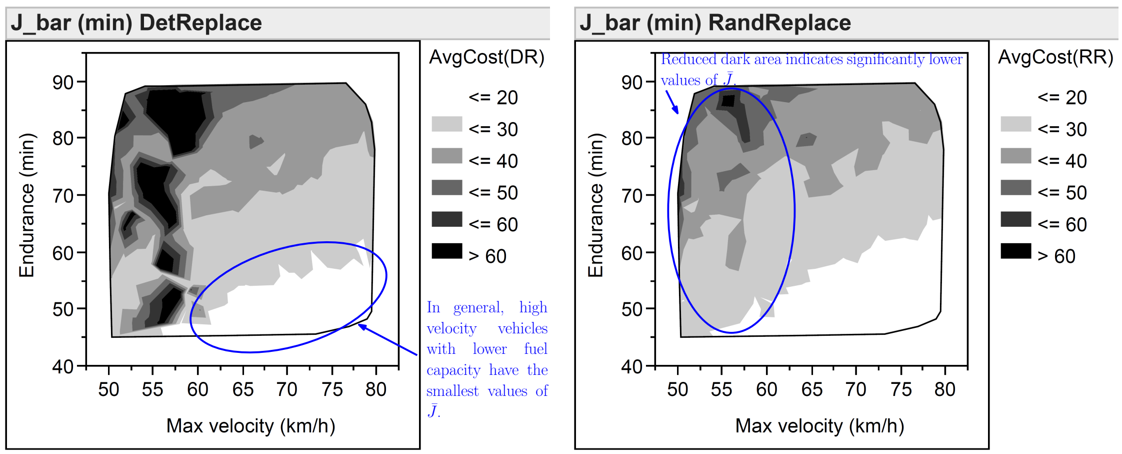


Figure 60: Some feasible connected communication networks while 3 UAVs are monitoring 4 monitoring zones.

Based on the conclusions from the Experiment 1, two control strategies (*DetReplace* and *RandReplace*) are tested in the second experiment. The results indicate a positive effect of the randomized return policy on the performance as opposed to the first experiment. For instance, the contour plot in Figure 61 shows that utilizing the randomized policy significantly reduces the value of the objective function ($\bar{J}$). This can be seen by comparing the dark regions in Figure 61a and the removal of dark regions from the left half of Figure 61b. Furthermore, high velocity and low fuel capacity vehicles perform better to reduce $\bar{J}$. This can be seen from the dominant white regions at the bottom right quadrant of Figures 61a and 61b.



(a) (b)

Figure 61: Contour plots of $\bar{J}$ with respect to the maximum velocity and endurance for Experiment 2.

The scatter plots of the Experiment 2 are displayed in Figures 62 and 63. As seen from Figure 62, the design cases using *DetReplace* distribute to the overall region whereas the same cases using *RandReplace* are more cluttered around the low values of $\bar{J}$. In particular, low velocity vehicles perform very poorly if they use *DetReplace*.

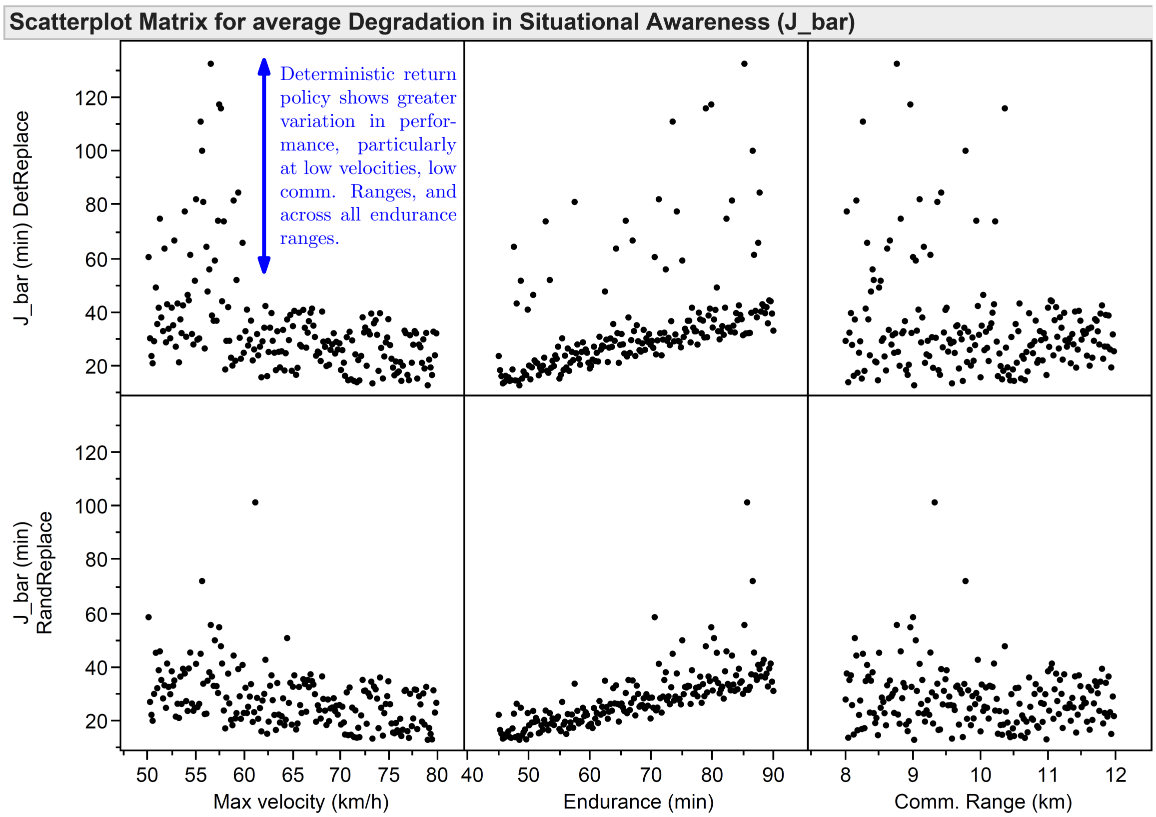


Figure 62: The scatter plot matrix displaying the objective function ($\bar{J}$) with respect to three design variables and 2 control strategies used in Experiment 2.

In Figure 63, an interesting result is observed as the robustness of short endurance low velocity vehicles if they use the randomized return policy. Let a desired limit for $\bar{J}$ be set to 30 minutes, which is shown as the horizontal line in in Figure 63. In this figure, the highlighted points represent the design cases having low velocity and short endurance. As it is seen, if the UAVs corresponding to these highlighted points use the randomized return policy, $\bar{J}$ does not deviate from the limit line greatly (covered by small dashed rectangles). On the other hand, if the same UAVs use the deterministic return policy, then a great deviation is observed such that a portion of the highlighted points locate above the constraint line (covered by large dashed rectangles).

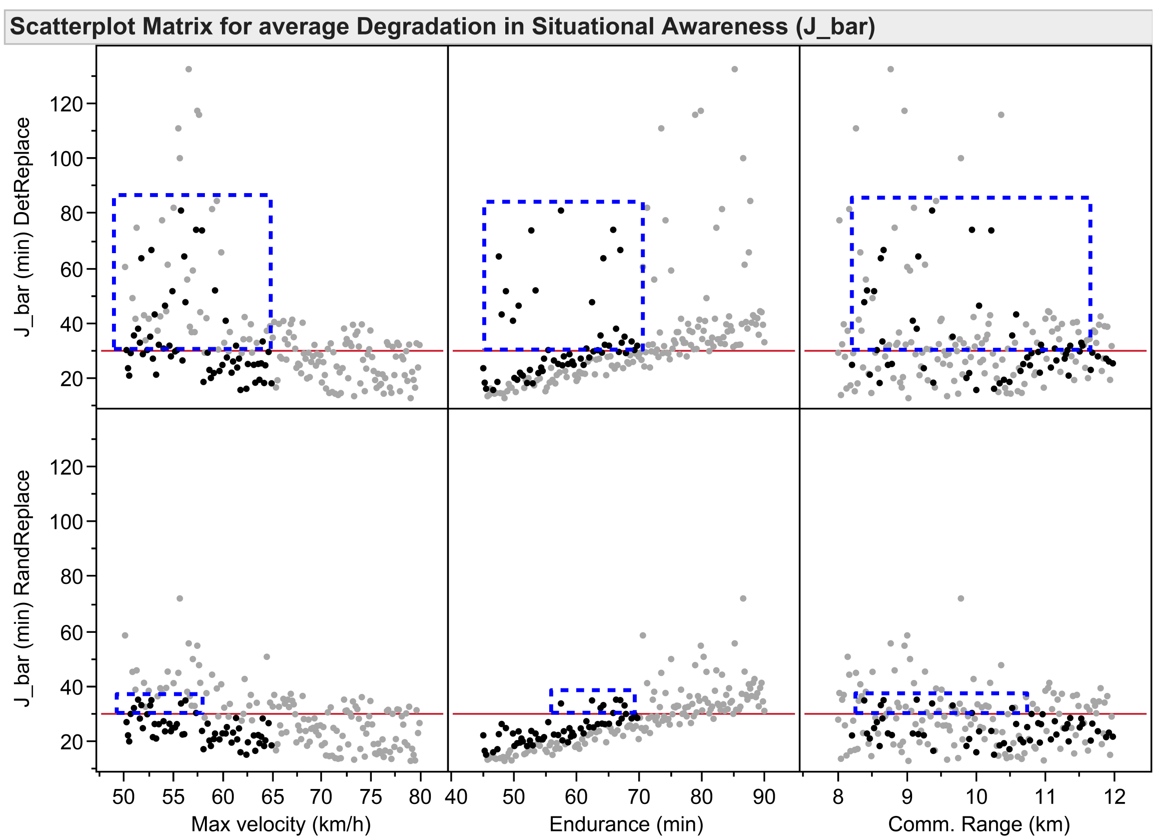


Figure 63: Using the randomized policy in Experiment 2 mitigates the effect of low velocity on the mission performance.

Results of the Experiment 2: For a surveillance scenario including n number of UAVs and m number of monitoring zones, where $n < m$, the following observations are obtained:

- The most influential design variables are observed as the maximum velocity and the endurance.
- High velocity short endurance vehicles are favorable to reduce the objective function $\bar{J}$ in the scenarios containing UAVs less than the number of monitoring zones.
- The randomized return policy greatly impacts the overall performance as oppose to the results of the Experiment 1, where a slight performance loss has been observed with the randomized return policy.

8.5 *Revisiting the Hypothesis 1*

In this section, the primary hypothesis of this thesis is revisited, and it is supported by the experimental results. As a reminder, Hypothesis 1 was formulated in Chapter 3 as:

For a desired mission performance, the selection of the design variables changes with different control strategies due to the effect of the interdependency between the design variables and the control strategies.

The results driven from the performed experiments indicate that the control strategies have a significant impact on the trends between the design variables and the mission performance. Based on the experiment results, one distinct interdependency is observed between the randomized return policy and the number of vehicles. Accordingly, if the number of vehicles is less than the number of monitoring zones, then the randomized policy significantly improves the overall performance. Otherwise, it degrades the situational awareness. This result is mainly caused by the dramatic increase in the ages of some particular monitoring zones due to being unoccupied for a long time if the vehicle has long endurance.

Accordingly, if a scenario includes more monitoring zones then the number of UAVs, then frequent returns to the base (due to the randomized policy) results in new assignments to the UAVs to monitor the high age zones.

Another conclusion driven from the experiments is that *high velocity short endurance* vehicles using the randomized return policy are favorable in missions containing more monitoring zones than the number of vehicles. As such, these vehicles quickly travel between the base and the surveillance area, and they are subjected to more zone assignments due to the frequent returns to the base. On the other hand, *high velocity long endurance* vehicles using the deterministic return policy become more effective in missions containing equal number of monitoring zones and vehicles. Finally, the results indicate that utilizing the replacement policy reduces the need to use vehicles with high radius of communication. In other words, each UAV does not require direct communication with the base if the connectivity maintenance is ensured during a mission. Consequently, the results of the experiments support that the selection of the design variables varies if the interdependency between the design variables and the control strategies are considered.

CHAPTER IX

CONCLUSION AND FUTURE WORK

This thesis addresses the consideration of the control strategies in the early design stages of multi-agent surveillance systems. It was presented that the performance of a multi-agent system (MAS) is greatly influenced by the physical design of the agents and the control strategies utilized by the agents. For any given system, the physical design determines the capabilities of each agent. On the other hand, the control strategies are developed to achieve a task with the given capabilities. In the early design stages, the MAS capabilities are often not certain. The first research question motivating this thesis was stated as follows:

Research Question 1: *Can a control strategy influence the selection of the physical design variables to result in a desired MAS performance?*

In response to the first research question, the first hypothesis was formulated as follows:

Hypothesis 1: *For a desired mission performance, the selection of the design variables changes with different control strategies due to the effect of the interdependency between the design variables and the control strategies.*

The verification of the hypothesis was conducted by generating a set of control strategies for efficient multi-agent surveillance and investigating their effects on the mission performance through Monte Carlo simulations. Due to the potential interdependency between the design variables and the control strategies of agents, the second research question dealt with the exploration of such interdependency.

Research Question 2: *How is it possible to explore the interdependency between the control strategies and the design variables of an MAS at early design stages?*

The second research question was addressed by the development of a methodology that takes into account a set of control strategies in the design space exploration. The experiment results showed that there exist some interdependency between the design variables and the control strategies, which can be successfully explored by the proposed methodology and potentially influence the design decisions. The proposed methodology and other contributions of the thesis are summarized in the next section. Then, this chapter ends with presenting the possible directions for future research.

9.1 Contributions of the Thesis

This thesis proposes a methodology that aids the early design of a multi-agent system by considering the effect of interdependency between the design variables and the control strategies on the mission performance. The proposed methodology comprises two major modules pertaining to the control and the design of agents. In the control module, a set of candidate control strategies (preferably decentralized) is generated. In the design module, the influential design variables are identified for each candidate strategy through a design space exploration. Accordingly, the methodology specifies the effective design-control pairs that can potentially provide a desired performance. This information can be utilized in requirement analysis as well as in the later design stages to optimize the overall system through some high fidelity analyses.

In this thesis, the proposed methodology is applied to a persistent multi-UAV surveillance problem. In this problem, there exists a base and a group of UAVs, each of which has limited energy and limited communication capability. The objective of the problem is to increase the situational awareness of the base through the flow of instantaneous monitoring information from the group of autonomous UAVs. To achieve this objective, this thesis studies both the physical capabilities and the decision mechanism of the UAVs for efficient operations in the surveillance area.

In long-run missions, each agent (e.g. a UAV) needs to return to the base for refueling at a frequency depending on its endurance. When an agent leaves the surveillance area, the information flow from the remaining agents to the base can be degraded. Note that the network connectivity is essential for the base to receive instantaneous information from the agents having limited communication ranges. In order to avoid a disconnection, this thesis proposes a decentralized connectivity maintenance strategy, the *Message Passing Strategy* (MPS), which can be applicable to any initially connected network. The proposed strategy is based on a sequence of replacements initiated by the agent leaving the surveillance area. Moreover, some variants of the MPS are introduced to increase the overall performance and the applicability to surveillance missions. In particular, the constrained δ-MPS is proposed as an efficient decentralized strategy that can be applicable to multi-agent surveillance missions. It is theoretically shown that the proposed strategy, i.e. the constrained δ-MPS, ensures the connectivity maintenance with the base in the face of an arbitrary agent removal. Moreover, the theoretical results are demonstrated via some simulations.

In a persistent surveillance mission, refueling is necessary to avoid unsafe states (e.g. running out of fuel). However, the absence of a refueling agent causes a degradation in the surveillance performance. As such, energy-aware strategies are desirable for efficient surveillance. To achieve this, one important issue is the decision for when an agent needs to return to base. In this thesis, deterministic and randomized return policies are studied. In the deterministic policy, each agent returns to the base whenever its remaining fuel (or energy) reaches a critical threshold (e.g. the minimum amount of fuel required to return to the base). In the randomized policy, each agent may return to the base with a small probability even though its remaining fuel (or energy) has not reached a critical threshold. Furthermore, whenever an agent leaves the surveillance area, the remaining agents may need relocation to mitigate the impact of its absence. In this thesis, the proposed connectivity maintenance strategies are used for such relocation. Accordingly, a set of decentralized energy-aware surveillance strategies are proposed by combining a relocation and a return policy. In order to test the performance of the proposed strategies, a benchmark solution is obtained by using approximate dynamic programming. Some simulations are performed to demonstrate the performance of the proposed decentralized strategies by comparing them with the benchmark solution. The results indicate that a combination of the constrained δ-MPS with one of the proposed return policies exhibit a performance close to the benchmark solution.

Finally, this thesis presents a design space exploration for a group of multi-UAVs by taking into account both the physical variables and the control strategies. To this end, a design-of-experiments study with Monte Carlo simulations are conducted to investigate the interdependency between the design variables and the control strategies. For instance, the results of the case study indicate the following design-control pairs: the maximum velocity and the maximum fuel capacity have a significant effect on the multi-UAV surveillance performance. If the number of UAVs is equal to the number of the monitoring points, high velocity and long endurance vehicles are desirable. Utilizing the constrained δ-MPS reduces the need for high radius of communication. Moreover, using the randomized return policy results in a slight performance loss. On the other hand, if the number of UAVs is less than the number of monitoring points, then high velocity short endurance vehicles become more favorable. Moreover, utilizing a randomized return policy greatly improves the surveillance performance. Consequently, the overall results driven from the design space exploration show that the control strategies affect the influence of the design variables (i.e. physical characteristics) on the mission performance. Hence, investigating the possible interdependency between the design variables and the control strategies enables a more efficient multi-agent system design.

9.2 Directions for Future Research

The persistent multi-agent surveillance problem studied in this thesis provides various avenues for future research. From design perspective, exploring a larger design space will help to design UAVs with better surveillance performance. Note that this thesis has investigated only three design variables, namely the maximum velocity, the endurance, and the communication range. As a future extension, the design variables pertaining to aerodynamic characteristics and propulsion performance can be incorporated in the design space exploration. As such, a high dimensional design space can be explored to depict more generalized results. Nonetheless, in the presence of a high dimensional design space, an additional step needs to be developed in the design module for the prioritization and ranking of the influential design variables.

From control perspective, developing close to optimal control strategies will result in more efficient agents. This thesis has developed a connectivity maintenance strategy that can deal with a single agent removal from the system, i.e. equivalent to assume a single UAV leave for refueling. However, it is possible to observe multiple UAV returns to the base in a surveillance mission. As such, one possible research path is to extend the proposed strategy for dealing with simultaneous agent removals. Furthermore, the combination of replacement and randomized return policies has been proposed as an effective decentralized strategy for long-run surveillance missions. Note that the randomized return policy is a perturbation of the deterministic return policy with a constant probability to return. As a future extension, the probability to return can be designed such that some characterization of the limiting behavior of an MAS can be obtained analytically. Finally, in the proposed energy-aware surveillance strategies, an agent is assigned to a new location only when it returns to the base for refueling. Such an approach reduces the responsibility of a central authority with a compromise of degraded performance. In order to obtain better surveillance performance, the proposed strategy may be improved by allowing for the reassignment of the agents even when they are in the surveillance area.

REFERENCES

[1] “DoD photo by Lance Cpl. James Purschwitz, U.S. Marine Corps..” http://www.defense.gov/dodcmsshare/newsphoto/2009-03/hires_090315-M-2581P-377.jpg.

[2] “The specification of Raven RQ-11B.” http://www.avinc.com/uas/small_uas/raven/. Accessed: 2014-08-07.

[3] ABATE, F. R., *The Oxford Dictionary and Thesaurus: The Ultimate Language Reference for American Readers*. Oxford Univ. Press Oxford, 1996.

[4] ABBASI, A. A., YOUNIS, M., and AKKAYA, K., “Movement-assisted connectivity restoration in wireless sensor and actor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 9, pp. 1366–1379, 2009.

[5] ABBASI, A. A., YOUNIS, M., and BAROUDI, U., “Recovering from a node failure in wireless sensor-actor networks with minimal topology changes,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 256–271, 2013.

[6] AKKAYA, K., THIMMAPURAM, A., SENEL, F., and ULUDAG, S., “Distributed recovery of actor failures in wireless sensor and actor networks,” in *IEEE Wireless Communications and Networking Conference, WCNC 2008.*, pp. 2480–2485, IEEE, 2008.

[7] AKSARAY, D., LI, Y., and MAVRIS, D., “Development and implementation of a hybrid multi-criteria decision making technique,” in *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference*, 2011.

[8] ALIGHANBARI, M. and HOW, J. P., "Cooperative task assignment of unmanned aerial vehicles in adversarial environments," in *American Control Conference*, pp. 4661–4666, IEEE, 2005.

[9] ALMEIDA, A., RAMALHO, G., SANTANA, H., TEDESCO, P., MENEZES, T., CORRUBLE, V., and CHEVALEYRE, Y., "Recent advances on multi-agent patrolling," in *Advances in Artificial Intelligence–SBIA 2004*, pp. 474–483, Springer, 2004.

[10] ALTENBAUGH, R., BARTON, J., CHIU, C., FIDLER, K., HIATT, D., HAWTHORNE, C., MARSHALL, S., MOHOS, J., MCHUGH, V., and NICOLOFF, B., "Application of the raven uav for chemical and biological detection," in *SPIE Defense, Security, and Sensing*, pp. 766505–766505, International Society for Optics and Photonics, 2010.

[11] ANDERSON, J. D., *Aircraft performance and design*, vol. 1. McGraw-Hill New York, 1999.

[12] ANTONY, J., *Design of experiments for engineers and scientists*. Elsevier, 2014.

[13] AUSTIN, R., *Unmanned aircraft systems: UAVS design, development and deployment*, vol. 54. Wiley, 2011.

[14] BAHILL, A. T. and GISSING, B., "Re-evaluating systems engineering concepts using systems thinking," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 4, pp. 516–527, 1998.

[15] BAKULE, L., "Decentralized control: An overview," *Annual Reviews in Control*, vol. 32, no. 1, pp. 87 – 98, 2008.

[16] BATES, R., BUCK, R., RICCOMAGNO, E., and WYNN, H., "Experimental design and observation for large systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 77–94, 1996.

[17] BAUSO, D., GIARRE, L., and PESENTI, R., "Quantized dissensus in networks of agents subject to death and duplication," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 783–788, 2012.

[18] BEARD, R. W., MCLAIN, T. W., NELSON, D. B., KINGSTON, D., and JOHANSON, D., "Decentralized cooperative aerial surveillance using fixed wing miniature UAVs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[19] BELLINGHAM, J., RICHARDS, A., and HOW, J. P., "Receding horizon control of autonomous aerial vehicles," in *American Control Conference*, vol. 5, pp. 3741–3746, IEEE, 2002.

[20] BENDOR, T., SCHEFFRAN, J., and HANNON, B., "Ecological and economic sustainability in fishery management: A multi-agent model for understanding competition and cooperation," *Ecological economics*, vol. 68, no. 4, pp. 1061–1073, 2009.

[21] BERT, C. W., "Prediction of range and endurance of jet aircraft at constant altitude," *Journal of Aircraft*, vol. 18, no. 10, pp. 890–892, 1981.

[22] BERT, C. W., "Range and endurance of turboprop, turbofan, or piston–propeller aircraft having wings with or without camber," *Aircraft Design*, vol. 2, no. 4, pp. 183–190, 1999.

[23] BERTSEKAS, D. P., "Approximate dynamic programming," 2011.

[24] BERTSEKAS, D. P., BERTSEKAS, D. P., BERTSEKAS, D. P., and BERTSEKAS, D. P., *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.

[25] BERTSIMAS, D. J. and VAN RYZIN, G., "A stochastic and dynamic vehicle routing problem in the euclidean plane," *Operations Research*, vol. 39, no. 4, pp. 601–615, 1991.

[26] BETHKE, B., HOW, J. P., and VIAN, J., "Group health management of UAV teams with applications to persistent surveillance," in *American Control Conference, 2008*, pp. 3145–3150, IEEE, 2008.

[27] BETHKE, B., HOW, J. P., and VIAN, J., "Multi-UAV persistent surveillance with communication constraints and health management," in *Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, IL*, 2009.

[28] BETHKE, B., REDDING, J., HOW, J. P., VAVRINA, M. A., and VIAN, J., "Agent capability in persistent mission planning using approximate dynamic programming," in *American Control Conference (ACC)*, pp. 1623–1628, IEEE, 2010.

[29] BEZZO, N., YUAN, Y., FIERRO, R., and MOSTOFI, Y., "A decentralized connectivity strategy for mobile router swarms," in *Proc. of the 18th World Congress of the International Federation of Automatic Control (IFAC), Milan, Italy*, 2011.

[30] BORSHCHEV, A. and FILIPPOV, A., "From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools," in *Proceedings of the 22nd international conference of the system dynamics society*, no. 22, 2004.

[31] BOUSQUET, F. and LE PAGE, C., "Multi-agent simulations and ecosystem management: a review," *Ecological modelling*, vol. 176, no. 3, pp. 313–332, 2004.

[32] CASBEER, D. W., SWINDLEHURST, A. L., and BEARD, R., "Connectivity in a UAV multi-static radar network," in *AIAA Guidance, Navigation, and Control Conference*, (Keystone, CO), 2006.

[33] CASSANDRAS, C. G., DING, X. C., and LIN, X., "An optimal control approach for the persistent monitoring problem," in *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 2907–2912, IEEE, 2011.

[34] CHASPARIS, G. C. and SHAMMA, J. S., “Distributed dynamic reinforcement of efficient outcomes in multiagent coordination and network formation,” *Dynamic Games and Applications*, vol. 2, no. 1, pp. 18–50, 2012.

[35] CHEVALEYRE, Y., “Theoretical analysis of the multi-agent patrolling problem,” in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology, (IAT 2004)*, pp. 302–308, IEEE, 2004.

[36] CORTES, J., MARTINEZ, S., KARATAS, T., and BULLO, F., “Coverage control for mobile sensing networks,” in *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, vol. 2, pp. 1327–1332, IEEE, 2002.

[37] DAI, F. and WU, J., “An extended localized algorithm for connected dominating set formation in ad hoc wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 10, pp. 908–920, 2004.

[38] DAI, R., MAXIMO, J., and MESBAHI, M., “Formation of connected networks for fractionated spacecraft,” in *AIAA Guidance, Navigation, and Control Conference*, (Minneapolis, MN), 2012.

[39] DAVE, S., SOORIYABANDARA, M., and YEARWORTH, M., “System behaviour modelling for demand response provision in a smart grid,” *Energy Policy*, vol. 61, no. 0, pp. 172 – 181, 2013.

[40] DE SOUSA, J. B. and PEREIRA, F. L., “Specification and design of coordinated motions for autonomous vehicles,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 1, pp. 101–106, IEEE, 2002.

[41] DE WOLF, T. and HOLVOET, T., “Emergence versus self-organisation: Different concepts but promising when combined,” in *Engineering Self-Organising Systems*

(BRUECKNER, S., DI MARZO SERUGENDO, G., KARAGEORGOS, A., and NAGPAL, R., eds.), vol. 3464 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin Heidelberg, 2005.

[42] DELAURENTIS, D., "A taxonomy-based perspective for systems of systems design methods," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 1, pp. 86–91 Vol. 1, Oct 2005.

[43] DERENICK, J., MICHAEL, N., and KUMAR, V., "Energy-aware coverage control with docking for robot teams," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3667–3672, IEEE, 2011.

[44] DIMEAS, A. and HATZIARGYRIOU, N., "A multi-agent system for microgrids," in *Methods and applications of artificial intelligence*, pp. 447–455, Springer, 2004.

[45] DU, J., XIE, L., SUN, X., and ZHENG, R., "Application-oriented fault detection and recovery algorithm for wireless sensor and actor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[46] DUDEK, G., JENKIN, M. R., MILIOS, E., and WILKES, D., "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.

[47] DUNBAR, W. B. and MURRAY, R. M., "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[48] ELMALIACH, Y., AGMON, N., and KAMINKA, G. A., "Multi-robot area patrol under frequency constraints," *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3-4, pp. 293–320, 2009.

[49] ENNS, R. and SI, J., "Helicopter trimming and tracking control using direct neural dynamic programming," *Neural Networks, IEEE Transactions on*, vol. 14, no. 4, pp. 929–939, 2003.

[50] ERIKSSON, L., *Design of experiments: principles and applications*. MKS Umetrics AB, 2008.

[51] FABRYCKY, W. J. and BLANCHARD, B. S., *Life-cycle cost and economic analysis*, ch. 1: System life-cycle concepts, pp. 1–14. International series in industrial and systems engineering, Englewood Cliffs, N.J: Prentice Hall, 1991.

[52] FORRESTER, A., SOBESTER, A., and KEANE, A., *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[53] FU, J. G. M. and ANG, M., "Probabilistic ants (pants) in multi-agent patrolling," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1371–1376, IEEE, 2009.

[54] GIRARD, A. R., HOWELL, A. S., and HEDRICK, J. K., "Border patrol and surveillance missions using multiple unmanned air vehicles," in *43rd IEEE Conference on Decision and Control, CDC*, vol. 1, pp. 620–625, IEEE, 2004.

[55] GRIENDLING, K. and MAVRIS, D., "An architecture-based approach to identifying system-of-systems alternatives," in *5th International Conference on System of Systems Engineering (SoSE)*, pp. 1–6, IEEE, 2010.

[56] HAMILTON, S., *UAVs: Unmanned Aerial Vehicles*. ABDO Publishing Company, 2012.

[57] HAZON, N. and KAMINKA, G. A., "Redundancy, efficiency and robustness in multi-robot coverage," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005*, pp. 735–741, IEEE, 2005.

[58] JAMESON, T., “A fuel consumption algorithm for unmanned aircraft systems,” tech. rep., DTIC Document, 2009.

[59] JI, M. and EGERSTEDT, M., “Distributed coordination control of multiagent systems while preserving connectedness,” *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[60] JILLA, C. D. and MILLER, D. W., “Multi-objective, multidisciplinary design optimization methodology for distributed satellite systems,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 39–50, 2004.

[61] KIM, J. and MORRISON, J. R., “On the concerted design and scheduling of multiple resources for persistent uav operations,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 479–498, 2014.

[62] KIM, J., SONG, B. D., and MORRISON, J. R., “On the scheduling of systems of UAVs and fuel service stations for long-term mission fulfillment,” *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 347–359, 2013.

[63] KINGSTON, D., BEARD, R. W., and HOLT, R. S., “Decentralized perimeter surveillance using a team of UAVs,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394–1404, 2008.

[64] KINNY, D. and GEORGEFF, M., “Modelling and design of multi-agent systems,” *Intelligent Agents III Agent Theories, Architectures, and Languages*, pp. 1–20, 1997.

[65] KNORN, F., STANOJEVIC, R., CORLESS, M., and SHORTEN, R., “A framework for decentralised feedback connectivity control with application to sensor networks,” *International Journal of Control*, vol. 82, no. 11, pp. 2095–2114, 2009.

[66] LAI, Y.-J., LIU, T.-Y., and HWANG, C.-L., “Topsis for modm,” *European Journal of Operational Research*, vol. 76, no. 3, pp. 486–500, 1994.

[67] Le Ny, J., Dahleh, M., and Feron, E., "Multi-agent task assignment in the bandit framework," in *45th IEEE Conference on Decision and Control*, pp. 5281–5286, Dec 2006.

[68] Lee, S. M. and Pritchett, A., "Predicting interactions between agents in agent-based modeling and simulation of sociotechnical systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, pp. 1210–1220, Nov 2008.

[69] Lewe, J.-H., Ahn, B.-H., DeLaurentis, D. A., Mavris, D. N., and Schrage, D. P., "An integrated decision-making method to identify design requirements through agent-based simulation for personal air vehicle system," 2002.

[70] Lewe, J.-H., Upton, E. G., Mavris, D. N., and Schrage, D. P., "An agent-based framework for evaluating future transportation architectures," *AIAA paper*, vol. 6769, 2003.

[71] Liao, C. and Barooah, P., "A novel stochastic agent-based model of building occupancy," in *American Control Conference (ACC), 2011*, pp. 2095–2100, IEEE, 2011.

[72] Lima, L. and Barros, J., "Random walks on sensor networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, pp. 1–5, April 2007.

[73] Lincoln, B. and Rantzer, A., "Relaxing dynamic programming," *Automatic Control, IEEE Transactions on*, vol. 51, no. 8, pp. 1249–1260, 2006.

[74] Luus, R., "Application of dynamic programming to high-dimensional non-linear optimal control problems," *International Journal of Control*, vol. 52, no. 1, pp. 239–250, 1990.

[75] MACAL, C. M. and NORTH, M. J., "Agent-based modeling and simulation," in *Winter Simulation Conference*, WSC '09, pp. 86–98, Winter Simulation Conference, 2009.

[76] MACHADO, A., RAMALHO, G., ZUCKER, J.-D., and DROGOUL, A., "Multi-agent patrolling: An empirical analysis of alternative architectures," in *Multi-Agent-Based Simulation II*, pp. 155–170, Springer, 2003.

[77] MARINO, A., PARKER, L., ANTONELLI, G., and CACCAVALE, F., "Behavioral control for multi-robot perimeter patrol: A finite state automata approach," in *IEEE International Conference on Robotics and Automation, 2009. ICRA'09.*, pp. 831–836, IEEE, 2009.

[78] MATARIĆ, M. J., "Designing and understanding adaptive group behavior," *Adaptive Behavior*, vol. 4, no. 1, pp. 51–80, 1995.

[79] MATTINGLEY, J., WANG, Y., and BOYD, S., "Receding horizon control," *Control Systems, IEEE*, vol. 31, no. 3, pp. 52–65, 2011.

[80] MAVRIS, D. N. and DELAURENTIS, D. A., "Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design," in *22nd Congress of the International Council on the Aeronautical Sciences (ICAS)*, 2000.

[81] MAVRIS, D. N. and KIRBY, M. R., "Technology identification, evaluation, and selection for commercial transport aircraft," 1999.

[82] MESBAHI, M. and HADAEGH, F. Y., "Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 369–377, 2001.

[83] MI, Z., YANG, Y., and LIU, G., "HERO: A hybrid connectivity restoration framework for mobile multi-agent networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1702–1707, IEEE, 2011.

[84] MONTGOMERY, D. C., *Design and analysis of experiments*. John Wiley & Sons, 2008.

[85] MOTEVALLIAN, S. A., YU, C., and ANDERSON, B. D., "Robustness to the loss of multiple nodes in the localizability of sensor networks," *in IFAC WC*, 2011.

[86] NIGAM, N., BIENIAWSKI, S., KROO, I., and VIAN, J., "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2012.

[87] NIGAM, N. and KROO, I., "Persistent surveillance using multiple unmanned air vehicles," in *IEEE Aerospace Conference*, pp. 1–14, IEEE, 2008.

[88] NORMILE, D., "Lost at sea," *Science*, vol. 344, no. 6187, pp. 963–965, 2014.

[89] OESTERGAARD, J. K., "About the RQ-11 Raven." `http://www.bga-aeroweb.com/Defense/RQ-11-Raven.html`, 2014.

[90] O'YOUNG, S. and HUBBARD, P., "Raven: A maritime surveillance project using small uav," in *IEEE Conference on Emerging Technologies and Factory Automation*, pp. 904–907, IEEE, 2007.

[91] PANDURANGAN, G., RAGHAVAN, P., and UPFAL, E., "Building low-diameter peer-to-peer networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 995–1002, 2003.

[92] PAPADIMITRIOU, C. H. and TSITSIKLIS, J. N., "The complexity of markov decision processes," *Mathematics of operations research*, vol. 12, no. 3, pp. 441–450, 1987.

[93] PARK, G.-J., "Design of experiments," *Analytic Methods for Design Practice*, pp. 309–391, 2007.

[94] PASQUALETTI, F., FRANCHI, A., and BULLO, F., "On cooperative patrolling: optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 592–606, 2012.

[95] PESCHEL, J. M. and MURPHY, R. R., "On the human–machine interaction of unmanned aerial system mission specialists," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 1, pp. 53–62, 2013.

[96] PFAENDER, H., DELAURENTIS, D., and MAVRIS, D., "An object-oriented approach for conceptual design exploration of UAV-based system-of-systems," in *Proceedings of 2nd AIAA "Unmanned Unlimited" Conference, ser. AIAA*, vol. 2003, p. 6521, 2003.

[97] PIPATTANASOMPORN, M., FEROZE, H., and RAHMAN, S., "Multi-agent systems in a distributed smart grid: Design and implementation," in *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pp. 1–8, IEEE, 2009.

[98] PONDA, S. S., JOHNSON, L. B., CHOI, H.-L., and HOW, J. P., "Ensuring network connectivity for decentralized planning in dynamic environments," in *Proceedings of the AIAA Infotech@ Aerospace Conference, St. Louis, MO*, 2011.

[99] POWELL, W. B., *Approximate Dynamic Programming: Solving the curses of dimensionality*, vol. 703. John Wiley & Sons, 2007.

[100] PUTERMAN, M. L., *Markov decision processes: discrete stochastic dynamic programming*, vol. 414. John Wiley & Sons, 2009.

[101] RANQUE, P., FREEMAN, D., KERNSTINE, K., LIM, D., GARCIA, E., and MAVRIS, D., "Stochastic agent-based analysis of UAV mission effectiveness," in *11th AIAA Aircraft Technology, Integration, and Operations (ATIO) Conference*, 2011.

[102] RAYMER, D. P., "Aircraft design: A conceptual approach, american institute of aeronautics and astronautics," *Inc., Reston, VA*, 1999.

[103] RAYMER, D. P., "Approximate method of deriving loiter time from range," *Journal of aircraft*, vol. 41, no. 4, pp. 938–940, 2004.

[104] REVELLE, J. B., MORAN, J. W., and COX, C. A., *The QFD handbook*. Wiley, 1998.

[105] RIPEANU, M., FOSTER, I., and IAMNITCHI, A., "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *arXiv preprint cs/0209028*, 2002.

[106] SAATY, T. L., *What is the analytic hierarchy process?* Springer, 1988.

[107] SABATTINI, L., CHOPRA, N., and SECCHI, C., "On decentralized connectivity maintenance for mobile robotic systems," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 988–993, IEEE, 2011.

[108] SABATTINI, L., SECCHI, C., and CHOPRA, N., "Decentralized connectivity maintenance for networked lagrangian dynamical systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2433–2438, IEEE, 2012.

[109] SAGGIANI, G., PERSIANI, F., CERUTI, A., TORTORA, P., TROIANI, E., GIULETTI, F., AMICI, S., BUONGIORNO, M., DISTEFANO, G., BENTINI, G., and OTHERS, "A uav system for observing volcanoes and natural hazards," in *AGU Fall Meeting Abstracts*, vol. 1, p. 05, 2007.

[110] SAMPIGETHAYA, K., POOVENDRAN, R., and BUSHNELL, L., "Security of future eenabled aircraft ad hoc networks," *AIAA Aviation Technology, Integration and Operations (ATIO)*, 2008.

[111] SANNER, S. and BOUTILIER, C., "Practical solution techniques for first-order mdps," *Artificial Intelligence*, vol. 173, no. 5, pp. 748–788, 2009.

[112] SANTNER, T. J., WILLIAMS, B. J., and NOTZ, W., *The design and analysis of computer experiments*. Springer, 2003.

[113] SAVCHENKO, S. V., "On the number of noncritical vertices in strongly connected digraphs," *Mathematical Notes*, vol. 79, no. 5, pp. 687–696, 2006.

[114] SCHETTER, T., CAMPBELL, M., and SURKA, D., "Multiple agent-based autonomy for satellite constellations," *Artificial Intelligence*, vol. 145, no. 1, pp. 147–180, 2003.

[115] SCHOLL, H. J., "Agent-based and system dynamics modeling: a call for cross study and joint research," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pp. 8–pp, IEEE, 2001.

[116] SCHUMACHER, C., CHANDLER, P., PACHTER, M., and PACHTER, L., "Constrained optimization for uav task assignment," in *Proceedings of the AIAA guidance, navigation, and control conference*, pp. 1–14, American Institute of Aeronautics and Astronautics Inc, 2004.

[117] SHAN, S. and WANG, G., "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, 2010.

[118] SIMPSON, T., POPLINSKI, J., KOCH, P. N., and ALLEN, J., "Metamodels for computer-based engineering design: Survey and recommendations," *Engineering with Computers*, vol. 17, no. 2, pp. 129–150, 2001.

[119] STIPANOVIĆ, D. M., INALHAN, G., TEO, R., and TOMLIN, C. J., "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[120] STONE, P. and VELOSO, M., "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[121] STUMP, E. and MICHAEL, N., "Multi-robot persistent surveillance planning as a vehicle routing problem," in *IEEE Conference on Automation Science and Engineering (CASE)*, pp. 569–575, IEEE, 2011.

[122] SUMMERS, T. H., YU, C., and ANDERSON, B., "Addressing agent loss in vehicle formations and sensor networks," *International Journal of Robust and Nonlinear Control*, vol. 19, no. 15, pp. 1673–1696, 2009.

[123] TIWARI, A., JUN, M., JEFFCOAT, D. E., and MURRAY, R. M., "Analysis of dynamic sensor coverage problem using kalman filters for estimation," in *Proceedings of the 16th IFAC world congress*, 2005.

[124] TRAUB, L. W., "Range and endurance estimates for battery-powered aircraft," *Journal of Aircraft*, vol. 48, no. 2, pp. 703–707, 2011.

[125] TUMER, K. and AGOGINO, A., "Improving air traffic management with a learning multiagent system," *IEEE Intelligent Systems*, vol. 24, no. 1, pp. 18–21, 2009.

[126] URE, N. K., CHOWDHARY, G., CHEN, Y. F., HOW, J. P., and VIAN, J., "Distributed learning for planning under uncertainty problems with heterogeneous

teams," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 529–544, 2014.

[127] VALENTI, M., BETHKE, B., HOW, J. P., DE FARIAS, D. P., and VIAN, J., "Embedding health management into mission tasking for UAV teams," in *American Control Conference, (ACC)*, pp. 5777–5783, IEEE, 2007.

[128] VALENTI, M., DALE, D., and HOW, J., "Mission health management for 24/7 persistent surveillance operations," in *AIAA Guidance, Navigation, and Control Conference*, 2007.

[129] WEINSTEIN, A. L. and SCHUMACHER, C., "Uav scheduling via the vehicle routing problem with time windows," in *Proc. AIAA Infotech@ Aerospace 2007 Conference and Exhibit. Rohnert Park, California*, 2007.

[130] WILLIAMS, J., FISHER, J., and WILLSKY, A., "Approximate dynamic programming for communication-constrained sensor network management," *Signal Processing, IEEE Transactions on*, vol. 55, pp. 4300–4311, Aug 2007.

[131] WILSON, B., CAPPELLERI, D., SIMPSON, T., and FRECKER, M., "Efficient pareto frontier exploration using surrogate approximations," *Optimization and Engineering*, vol. 2, no. 1, pp. 31–50, 2001.

[132] WOLLKIND, S., VALASEK, J., and IOERGER, T. R., "Automated conflict resolution for air traffic management using cooperative multiagent negotiation," in *AIAA guidance, navigation, and control conference*, pp. 16–19, 2004.

[133] YAZICIOĞLU, A. Y., EGERSTEDT, M., and SHAMMA, J. S., "Decentralized degree regularization for multi-agent networks," in *52nd IEEE Conference on Decision and Control*, IEEE, 2013.

[134] ZAMANIFAR, A., KASHEFI, O., and SHARIFI, M., "AOM: An efficient approach to restore actor-actor connectivity in wireless sensor and actor networks," *International Journal of Computer Networks & Communications*, vol. 1, no. 1, pp. 61–72, 2009.

[135] ZAVLANOS, M. M., EGERSTEDT, M. B., and PAPPAS, G. J., "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[136] ZAVLANOS, M. M., JADBABAIE, A., and PAPPAS, G. J., "Flocking while preserving network connectivity," in *46th IEEE Conference on Decision and Control*, pp. 2919–2924, IEEE, 2007.

[137] ZHOU, J., HU, Q., and FRISWELL, M. I., "Decentralized finite time attitude synchronization control of satellite formation flying," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 185–195, 2013.