UNIVERSITY OF SOUTHAMPTON

Faculty of Physical Sciences and Engineering
School of Electronics and Computer Science

Self-organised Communication-aware Control Structures for Robot Swarms

by

Thomas Graham Kelly
MEng

A thesis for the degree of
Doctor of Philosophy

June 2025

University of Southampton

Abstract

Faculty of Physical Sciences and Engineering
School of Electronics and Computer Science

Doctor of Philosophy

Self-organised Communication-aware Control Structures for Robot Swarms

by Thomas Graham Kelly

Robotic swarms are complex systems that rely on local communication between individual members of the swarm to spread information about their state and the environment. This information informs decisions and aids in tasks such as exploration and mapping. When communications break down between members of a swarm, it can become difficult to maintain accurate and up-to-date information about the state of the swarm and the environment. This problem is pertinent when humans are involved and may act as operators or teammates of the swarm. Here it is vital that the swarm can coordinate to distil and disseminate the vast amounts of information collected to the humans and throughout the swarm effectively, to maintain situational awareness.

The collective decision-making of a swarm is one aspect that relies heavily on the ability to share information and observations to reach a swarm-wide consensus. This thesis investigates how communication constraints affect the swarm's ability to reach a consensus and implement a communication-aware coordination strategy to mitigate these effects. We propose the communication-constrained collective decision-making problem and compare the performance of several collective decision-making strategies, enhanced with our coordination algorithm. We find that using such an approach improves the speed of a swarm to reach a consensus.

Following on from this work, we examine a hybrid swarm system in a communication-limited environment. While swarms are traditionally considered decentralised systems, recent approaches have integrated decentralised and centralised control into a swarm system. We study the trade-offs in performance and communication in a hybrid system that can vary its control structure. We find that a higher level of centralisation does not guarantee higher performance and study how communication with a human operator is affected by the control structure. This work is extended to assess the feasibility of enabling a swarm system to learn the optimal control structure on the fly, according to mission requirements.

Contents

List of Figures

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as: T. G. Kelly, M. D. Soorati, K.-P. Zauner, S. D. Ramchurn, and D. Tarapore, "Collective decision making in communication-constrained environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7266–7271 T. G. Kelly, M. D. Soorati, K.-P. Zauner, S. D. Ramchurn *et al.*, "Trade-offs of dynamic control structure in human-swarm systems," *arXiv preprint arXiv:2408.02605*, 2024

Signed:.. Date:..................

Acknowledgements

Firstly, I would like to thank my University of Southampton supervisors Dr. Danesh Tarapore, Dr. Mohammad Soorati, Prof. Gopal Ramchurn and Dr. Klaus-Peter Zauner. Their input, support, encouragement and patience through all stages of the PhD has been invaluable. They gave me a great deal of insight and guidance through our meetings and I am grateful to have had the opportunity to learn from them. I would also like to thank all those in the AIC group and the Southampton Trustworthy Autonomous Systems (TAS) hub team.

My thanks also go to Prof. Tim Norman and all those involved in the MINDS CDT for the opportunity to undertake a PhD. Also to Todd Robinson and the team from DSTL for funding the project under Contract No. DSTL1000153638 and for allowing me to undertake a placement, which was invaluable to my research experience. I would also like to thank Dr. Markus Brede, who initially inspired me to pursue research, and Prof. Elio Tuci who both agreed to attend my viva.

My thanks go to my friends, including my fellow PhD students, Jack Dymond and Jack Williamson, who have been a source of motivation throughout the PhD. Also to Osian Nixon, my close friend, and Matt Venables, Jack Birch and Lucy Buck who have been a constant source of reassurance and support for my entire university life. I am indebted to my partner and best friend, Sally Lee, who has given me strength, love and motivation when I needed it most.

And finally to my family. My parents, Alison and Graham Kelly, and my sister, Emily Kelly, for their constant belief in me. Your inspiration and love have given me the fortitude to reach this stage and I certainly would not be here without it. Thank you for believing, always.

To Mum and Dad, for your inspiration and unwavering love and support.

Definitions and Abbreviations

BDI	Beliefs, Desires and Intentions Framework
DASA	Defence and Security Accelerator
DC	Direct Comparison of Option Quality
DQN	Deep Q-Networks
DMMD	Direct Modulation of Majority-based Decisions
GP	Gaussian Process
HSI	Human Swarm Interaction
IQR	Interquartile Range
MFDM	Multi-feature Decision Making
MNS	Mergeable Nervous System
PSO	Particle Swarm Optimisation
QoS	Quality of Service
SLAM	Simultaneous Localisation and Mapping
SAT	Swarm Transparency Framework
UAV	Unmanned Aerial Vehicle
WSN	Wireless Sensor Network

Chapter 1

Introduction

In a system consisting of many entities that interact to achieve a common goal, finding the best way to coordinate those entities is vital in ensuring the collective success of the system. The field of Multi-agent systems, which is a sub-field of AI, has dedicated itself to examine the mechanisms behind the behaviours and coordination of such entities for several decades. It defines an agent as an entity with goals, actions, and knowledge of its domain that is situated within an environment [3]. When these agents are robots, the system is considered a Multi-robot system. For such systems, two common control structures are used to coordinate agents and aim to provide the system with feedback either from agents' peers, known as a decentralised system, or from a central operator, known as a centralised system. A fully centralised system relies on a single computational entity that dictates the behaviour of the entire population of agents. This entity may be external to the environment that the robots are operating in or may be one of the robots that takes on the role of a leader [4, 5, 6, 7]. This entity is able to collect data from other robots via communication channels and use this global information to arrange and coordinate the system of robots. This inclusion of global data often allows such systems to perform complex tasks more quickly and accurately than their decentralised counterparts at the expense of their computationally expensive calculations, poor scalability and vulnerability to single points of failure [8]. Decentralised systems, on the other hand, do not have a single central controller and each agent/robot is left to autonomously accomplish it's tasks. Such systems typically rely on local interactions between agents to share information that is collected relating to tasks and the environment. Without a single leader or centralised control, these kinds of system are more flexible in their ability to adapt to their environment and any changes that may occur. Each agent uses the local information that they gather from the environment and from neighbours to act in the best way in order to complete the task. While decentralised coordination is possible [9, 10], it is often less optimal than centralised approaches, due to the difficulty in securing global information about the system, which often needs to be built up over time by the robots.

Ultimately, the mission requirements set out by an operator are a major factor in deciding whether a centralised or decentralised control structure is best for a given task. A centralised system is typically the optimal architecture to use when efficiency and speed are crucial [6]. However, this efficiency will come at a cost of the resilience of the system and may put the system at a higher risk of failure, should the central operator fail. Selecting a decentralised system, however, will likely sacrifice the speed and accuracy of the solution [6]. The decision of which architecture is the most optimal to use becomes even more difficult when the system is faced with unknown and dynamic environments. The state of that environment may introduce a variety of constraints on the system that is deployed, whether these are related to communications or the operability of the system. In such environments, it would be desirable to have a system that can adapt its control structure accordingly to suit the current state of the environment. To do so, the system should be able to transition back and forth between a decentralised and centralised system, to take advantage of both of these approaches' strengths while mitigating their weaknesses.

The notion of a system that can adapt its structure or organisation by itself is known as *Self-organisation*. This adaptation and reorganisation is achieved using simple local interactions without any external direction [11]. Such systems are common in nature for example the flocking behaviour of birds, where individuals change their movements to align with the positions and velocities of their neighbours [12]. Similarly, this phenomenon can also be observed in natural physical processes such as in the formation of patterns in sand dunes, like ripples or waves, as a result of interactions between wind and sand particles [13]. In the context of multi-robotic systems, this refers to the ability of robots or robotic networks to autonomously arrange their components or behaviours in a structured manner without external control or pre-defined instructions [14]. This may be through the arrangement of robots in order to form physical structures [14, 15] or through the formation of robotic networks which may extend to hierarchies present in that network [7]. These arrangements must be controlled by the swarm alone, without external intervention [14]. Doing so, allows the swarm to adapt and change the formed structures according to the state of the environment and the swarm itself. In the context of physical structures, this may involve the relocation of robots in order to allow a physical link between two points that were previously not connected. In a hierarchy, this may relate to the dynamic selection of a leader that is best suited to make decisions on behalf of a locally centralised structure of robots. Robots that follow a self-organised approach adapt their actions and structures based on interactions with their environment and with each other. This leads to the spontaneous emergence of organised behaviours, patterns and solutions.

It is possible to have a system benefit from both a centralised and decentralised approach together by combining elements of both control structures in a self-organised manner. Recently, a method inspired by the Mergeable nervous systems (MNS) [4] was proposed, which enables a swarm system to overcome the drawbacks faced by both centralised and decentralised systems. In this approach, a swarm is controlled through a self-organised hierarchical structure consisting of a brain robot which can be dynamically selected. This approach relied on physical connections between robots linking each agent to the brain robot. A system using this approach can vary the level of decentralisation or centralisation according to mission requirements and for a given task. This MNS approach was extended to use wireless communications instead of physical connections, by Zhu et al. [5]. In this approach, the behaviours of a group of robots, both ground based and unmanned aerial vehicles (UAVs) are managed by a dynamically selected UAV brain robot. The MNS approach represents the only attempt to create a hybrid control system.

1.1 Robot Swarms

One example of a decentralised Multi-robot system is a robot swarm. This is defined as a robotic system that exhibits “aggregation often combined with collective motion”. While there are many definitions of a swarm, one of the overriding themes in these definitions is that a swarm is a complex system that is made up of simple components, which act to solve a task collectively. Each member of the swarm shares the same objectives and aims to solve a complex task together which would be impossible to solve if considered by only one individual. One such example, which embodies this idea and can be observed in nature, is that of flocks of birds, with thousands of birds forming a system of complex movements using simple, local rules [16]. An important motivation for thinking about swarm systems is asking the question of how simple and local actions of individuals combine to display predictable global patterns [16]? Such a motivation is attractive from an engineering point of view. Being able to have swarms of simpler and less expensive embodied individuals, robots, deployed into an environment and have them solve a complex task collectively, as effectively as a more expensive and powerful individual robot is a tantalising prospect [17]. A key characteristic of swarm systems is their rapid adaptability to their environment. This stems from their decentralised nature which is traditionally used as a definition of a robot swarm [18]. Such a feature makes them well-suited for operation in unknown environments. It is anticipated that robot swarms will see great advances in the next decade, given that they have been shown to aid in a wide variety of problems facing humans [17]. These include offering solutions to improve the efficiency of food production, and reducing water consumption [19] and for responding to disasters [20].

A key advantage of robot swarm systems is in their ability to recover from damage, thanks to the large number of individuals which creates redundancy within the system. This, along with their ability to cover a potentially large area, makes them an excellent solution to problems in environments that may be difficult for humans to navigate in. For example, in the wake of a disaster, where the infrastructure may be damaged making it difficult or dangerous for humans to access. Here, a robot swarm could be deployed to begin the search for survivors. Such scenarios do however present their own challenges.

One such challenge considers how the swarm can communicate important information to a human who wishes to use the data collected to inform decisions. In a disaster response scenario, this information may be relayed to the operator when and where it is best to allocate resources [21]. Not only must the swarm make decisions on the information that has been collected quickly in time-sensitive situations, but must also be able to communicate these decisions to humans in a reliable manner [22]. This itself can become increasingly challenging when the intra-swarm communications and the swarm-to-human communications become unreliable. This uncertainty over the communications can be caused by a range of factors. Environmental factors may include obstacles that prevent communications between certain parts of the environment and may make the communications intermittent or even cut them out altogether [23]. Another factor to consider is the topology of the swarm which can rapidly change as the swarm performs its task. This will mean that the members who are within communication range of each other will change rapidly. Finding ways that the swarm can account for these uncertainties in the communication conditions could allow robot swarm systems to cope with rapid changes affecting the swarm and improve their decision-making under time constraints.

As such, the study of human swarm interaction (HSI) has become increasingly important in recent years [16]. This field combines the studies of traditional human-robot interaction with swarm systems. In these systems, feedback with the operator is particularly important due to the seemingly chaotic interactions and behaviours that individual agents can appear to exhibit. Swarm transparency, defined as "the quality of an interface to support a human operator's comprehension of an intelligent agent's intent, performance, future plans and reasoning process" [24], is an important concept when considering how a swarm may present information to a human. The situational awareness-based agent transparency (SAT) framework was outlined in [24] and incorporates Endsley's situational awareness framework [25], Rao and Georgeff's beliefs, desires and intentions (BDI) framework [26] and Lee's 3Ps model (purpose, process and performance) for human operator trust calibration [27]. It highlights important factors that improve the situational awareness of human-robot teams through providing greater transparency. In [22], Roundtree at al. examined criteria that designers of human-robot systems can consider to achieve greater transparency. Their analysis was based around the SAT framework, presented in [24]. These studies found that when communication conditions are limited, breakdown or change, the swarm's ability to relay information to a human operator or disseminate the information throughout the swarm can be hindered. This breakdown can also affect the human operator's situational awareness of the environment [28, 22, 29, 30], by hindering the swarm's ability to communicate up-to-date and accurate information regarding the swarm's current states, future states and intentions. These changes to the communication conditions can be caused by several factors. For example; environmental factors such as in a forest environment, changes in the topology of the swarm network or in contested environments where the swarm must reduce communication to avoid detection. It is important that when these factors are present, the swarm can continue to inform human operators and perform its task.

Swarm systems embody many of the key principles and features of a self-organised system. They are resilient to single points of failure, scalable to different tasks and environments and autonomous in their decision-making. Importantly, they are also able to coordinate and collaborate to accomplish tasks through local interactions, without centralised control, resulting in effective collective behaviours. This capability enhances their flexibility and enables them to adapt their behaviours and actions according to their environment, which may be dynamic or unpredictable. They do, however, face significant challenges when effectively communicating up-to-date and accurate information to human operators. Maintaining the situational awareness of an operator and improving the swarm transparency are key issues that must be considered in the development of swarm systems. While many aspects of HSI have been studied, the impact on a human operator and on the performance of a swarm itself has not been considered for a swarm using a dynamic self-organising system, that may be able to vary its level of centralised and decentralised control.

1.2 Self-organised Hybrid Control Structures

A swarm system presents a unique opportunity to exploit a huge number of simple agents and the redundancy that comes with it to complete tasks that would be impossible to solve by a single agent. Decentralised approaches are proposed as scalable and resilient methods for organising swarms but with low efficiency. In contrast, centralised coordination can be efficient (e.g. in area coverage [31]) but is often not scalable and they rely on central units to control the system which can be points of system failure (i.e., low resilience). A combination of centralised and decentralised control could provide a swarm control method that makes use of the benefits of both approaches. Traditionally, swarms are defined as decentralised systems with shared sets behaviours embedded into each member of the swarm and, as such, no hierarchy, no centralisation and with no predetermined leader [18].

However, recent efforts have endeavoured to incorporate elements of hierarchy and centralised control into robots swarms in order to increase the adoption of robot swarms into real-world applications [7]. The inclusion of centralised control contradicts the definition that a swarm system should be decentralised in order to benefit from the scalability, flexibility and fault tolerance that decentralised systems exhibit. However, if hierarchy and centralised control are applied in a way such that the centralised control is formed by the swarm itself and is not imposed from outside the swarm then it is possible to design a system that incorporates swarm controlled centralisation, while allowing the swarm to dynamically adapt this hierarchy in a self-organised way [7]. This could be achieved by designing a system where agents switch between acting as a centralised or decentralised system. The overall time spent in each of these states by the system could dictate the level of centralisation or decentralisation exhibited. This method is an important aspect of the future of robot swarms [32]. Such a method would further enhance the ability of a swarm to adapt quickly and effectively according to its environment, when it has been deployed in an *a priori* unknown environment. The relative advantages of using such an approach have been examined for varying groups of robot sizes in both homogeneous [4, 5] and heterogeneous [33, 6] systems.

Work considering theoretical foundations of self-organised hierarchical control frameworks has also been considered by Zhang et al. [34]. Hierarchical swarm control, despite being a relatively new field within swarm robotics, represents a promising area of research with important implications for the future of swarms [32]. The self-organised nervous system for swarms was proposed by Zhu et al. [7] as a hierarchical control method for swarms, examining the scalability and stability of such a system. Work in this area, so far has focused on the application of a mergeable nervous system to distribute and coordinate a swarm based on a set of network topologies. While the relative performance of a bio-inspired hierarchical control structure has been studied, the performance of a more generic hybrid approach and its effect on human operators have not yet been explored.

It has been found in previous works on a variety of platforms, that the level of human input needed to achieve the highest performance is task-dependent [35, 36]. Higher levels of automation reduce the time spent by a human operator performing actions but can increase the cognitive load due to the vast amount of data that is being collected and presented by the swarm [37, 28]. Cummings suggested that this can affect the situational awareness of the human operator in the task [28]. This is because large quantities of data make comprehending the current and future states of the swarm very difficult. As such it is important to include the human at some level of decision-making [28], however, too much involvement may reduce the overall system performance [35, 36]. When operating a swarm, the volume and quality of information exchanged with an operator and within the swarm is a key factor in maintaining situational awareness. Managing multiple tasks or interruptions at once increases the effort required by an operator to restore their situational awareness [38]. To aid with the operation and the situational awareness, the focus has been placed on developing novel interfaces for the visualisation and control of swarms [39, 40, 41]. Abioye et al. [42] studied the effect of cognitive workload on the overall human-swarm performance and showed that a higher volume or quality of information sent to the operator does not necessarily increase the number of accurately completed tasks and instead slows down the operation.

While these issues have been studied extensively in the context of HSI for traditional decentralised swarms, they have not been considered in the context of self-organised hybrid control systems. It's unclear how these aspects of HSI, such as the cognitive load and situational awareness vary with different levels of decentralised or centralised control that may be present in self-organised control structures. A swarm system presents an often unpredictable system with emergent behaviours that can appear chaotic to a human operator as a result of the local interactions that are present. In a system that can change the control structure quickly and potentially without warning, it is possible that the system could present additional mental overhead for a human operator. If this is the case, it is important to understand the mechanisms that cause the system to change its control structure. Comprehending how these mechanisms affect communication with an operator could allow the prediction of future changes in the swarm structure and reduce the mental load placed on the operator.

1.3 Problem Definition

Communication limitations in unknown environments pose significant challenges for human-swarm systems [37, 22, 30, 29, 23] and limit their deployment into real-world applications [43]. A variety of factors may affect this, from environmental obstacles to rapidly changing and dynamic swarm topologies. As a result, it is important that a swarm can adapt and optimise its own parameters to these conditions and that it is aware of its communication environment. The swarm must be able to maintain up-to-date information, along with providing accurate information to any operators, improving their situational awareness. Many of the approaches presented in literature consider planning to reconnect in communication constrained environments [44, 10, 45], maintaining connectivity throughout the swarm [46, 47] and collective decision making [48, 49]. Each of these are important considerations when implementing a system that can provide a human operator with situational awareness. Maintaining connectivity can help mitigate some of the negative effects of latency in the system by ensuring commands and feedback are exchanged between the human operators and the swarm promptly. Re-connection protocols can reduce the risk of the swarm splitting into fragments unexpectedly, which may occur as a result of the human operator issuing several commands in a short time [37]. Many of the approaches, however, rely on *a priori* knowledge of either the possibility of two individuals being able to communicate at any point in time or the environment that the swarm is operating in [50, 23, 45, 44]. These approaches also aim to find an optimal plan, by simulating and planning many timesteps into the future. Such plans may require replanning in a changing environment, resulting in a waste of initial computation. The resulting assumptions are difficult to fulfil in swarm systems where the topology of the swarm and the communication landscape may change rapidly due to human input or changing environmental conditions. Methods that avoid the need for this information and can consider the uncertainties present in a large swarm setting could mitigate this issue. Particularly where imperfect information may be available to each agent relating to the locations of neighbours and where plans may not be possible due to the reactive nature of the swarm. Alongside this, many of the proposed methods for coordination and coverage aim to be optimal and plan accordingly. However, in a dynamic environment, planning too far in advance may result in wasted time, where plans become obsolete a few time steps in the future. Constraining the planning process to consider changes in the swarm topology and the environment may lead to more energy-efficient plans.

We address these issues in Chapter 3, by considering a collective decision-making scenario with communication constraints that are an intrinsic feature of the environment. These constraints prevent agents from being able to share information completely. We examine how communication constraints affect the ability of a swarm to reach a consensus and implement a communication-aware coordination approach to mitigate and overcome these communication constraints. This approach considers the uncertainty over the environment and the swarm to give individual agents the best chance at being able to communicate with other members of the swarm.

We also consider work on hybrid swarm control systems. This represents an important avenue of research for the future of swarm systems [32]. A system with a hybrid control structure would enable a swarm to adapt to the conditions of an unknown environment appropriately. This area of research is relatively young, with a majority of the work focusing on implementing a hybrid control structure into different robotic platforms and tasks, including a swarm of Kilobots [4], a collective perception problem [33], a coverage control problem [6] and formation control [5]. There are however several gaps present in the literature relating to this kind of control system. While a robust hybrid control approach has been proposed [4], and applied to several problems [33, 5, 6], the level of centralisation and decentralisation present within these systems remains the same for the duration of the experiment. No problems have been explored where it may be beneficial to vary the level of centralisation and decentralisation throughout the experiment. An example scenario could require the swarm to cover a dynamic area, with hot spots of concentrated activity, where the swarm size is too small to do so optimally. As such, having a swarm system that can fluidly transition between centralised and decentralised control could reveal in more depth the relative trade-offs of each type of control system, as has been studied in [6]. Such an approach may prove important in unknown and dynamic environments where the swarm must be able to adapt to changing states of the task, swarm, environment or communication capabilities. As such, there are several trade-offs to be considered when deciding what type of systems should be deployed for a given task. These ultimately hinge on what the priorities of the mission are. Does the task need to be completed quickly and accurately or is it more important that the operator is kept up to date and the task is completed safely? Where is the optimal point where both of these conditions can be met satisfactorily for the task and the operator? Enabling a swarm with this ability to dynamically change its level of centralisation (or decentralisation) on the fly may have important impacts on a human operator, receiving feedback on the environment and the swarm. It's unclear how aspects of HSI, such as the cognitive load and situational awareness are affected by systems combining decentralised and centralised control. A system that can change its control structure or topology quickly as in some of the works mentioned [4, 5], may present additional challenges for a human operator. As such, it is important to understand the mechanisms that cause the system to change control and how this affects the communication with an operator. Doing so could open up new directions for HSI research relating to hybrid and self-organising systems.

These issues are explored in Chapters 4 and 5, where we examine the trade-offs of hybrid swarm control structures against hierarchical and decentralised methods. To do this, a simple, generic hybrid control approach is proposed which allows for the level of centralisation (or decentralisation) to be varied. In Chapter 4, this level is varied across simulation runs, while in Chapter 5, it is varied on the fly during the simulations. We examine the relationship between the performance and communications of a swarm using this approach for different levels of centralisation. We also examine the feasibility of enabling a swarm to learn the optimal level of centralisation for a given task.

In this thesis, we aim to address the following research questions:

1. **What are the effects of spatial communication constraints on a swarm's ability to make decisions?**

   As has been highlighted in the literature, a significant challenge for robot swarms is in their ability to coordinate and effectively collaborate when communications are limited. In a collective decision-making task, the swarm relies heavily on the ability to share observations and beliefs to decide on the best quality option present in the environment. Previous works have considered communication limitations [48] and communication noise [51] as constraints in a collective decision-making task. However, currently, there is no literature examining how spatial communication constraints affect a swarm's ability to reach this consensus.

2. **How does making a swarm communication-aware affect its decision-making capabilities?**

   Previous work has highlighted the need for collective decision-making strategies to limit the spread of beliefs to avoid reaching a consensus based on incorrect information [52, 53, 54]. Introducing a communication-aware approach based on improving the ability of the swarm to connect with neighbours could introduce issues that improve speed, but hinder the accuracy of the swarm. We examine how introducing a communication-aware protocol to the swarm impacts its ability to make decisions correctly.

3. **How are the communications of a swarm affected by a hybrid swarm system with a variable control structure?**

   Work by Jamshidpey et al. [6] has considered the relative advantages of a decentralised, centralised and hybrid swarm system for a wide range of metrics. However, the trade-offs of a hybrid system that can vary its control structure according to the state of the environment or user-defined mission requirements have not been considered. This is of particular importance in a communication-constrained environment, where such a system may be well suited to adapt to harsh communication limitations.

4. **What mechanisms are required to smoothly transition between centralised and decentralised control structures?**

   This builds on the previous question, considering a hybrid swarm control structure that can vary on the fly. Such a system has not been considered and as such it is unclear what underlying mechanisms are required to facilitate the transition between different control structures.

1.4 Contributions of the Thesis

In this section, we present the main contributions of the thesis relating to multi-robot and swarm systems.

1.4.1 Communication constrained collective-decision making

Here we study the performance of a swarm system in a collective perception problem under communication constraints. We consider the time taken for the swarm to reach a consensus on the problem when a proportion of the environment does not allow the swarm to exchange information. We propose a new benchmark scenario to study the collective perception of a robot swarm in a communication-constrained environment. In our scenario inter-robot communication may be completely denied in *a priori* unknown areas for the duration of the experiment.

We also develop algorithms for the swarm to adapt to these communication restrictions, achieving high-speed consensus on decisions of features of interest in the environment. Using these algorithms, our results show that the performance is higher than the state-of-the-art collective decision-making algorithms in communication-constrained setups.

The details on this work can be found in Chapter 3.

1.4.2 Trade-offs of Hybrid Swarm Control

Here we examine the assumption that a centralised approach always achieves the best performance in a task over a decentralised or hybrid solution. We propose a simple generic hybrid approach which combines elements of decentralised and centralised control systems. It is shown that the proposed hybrid system may achieve higher performance in some tasks (e.g., in our environmental monitoring task) compared to a centralised and decentralised approach. While this performance comes with trade-offs in communications, both between a human operator and within the swarm, we show that these trade-offs can be balanced with the performance of the overall system according to end-user requirements while using our proposed hybrid approach.

The details on this work can be found in Chapter 4.

1.4.3 Adaptive Hybrid Control Structures

In this chapter we examine the possibility of being able to learn the optimal values, dictating the level of (de)centralisation for the best performance of the swarm in a given task. The swarm must learn and vary its parameters depending on whether performance or communication to the operator is the preferred focus of the mission. In doing so, we examine the mechanisms required to smoothly transition a swarm system back and forth between a decentralised and centralised system. An extension is proposed to the hybrid approach, that is introduced in Chapter 5. We find that by using this approach, it is possible to optimise the parameters for a swarm to maximise the performance and communication requirements in an environmental monitoring problem.

This work is presented in Chapter 5.

1.5 Organisation of the Thesis

The remaining structure of the thesis is as follows:

- In Chapter 3 an approach is proposed to improve a swarm's ability to adapt to communication constraints within the environment. This is applied in a collective perception task. The performance of the approach is assessed by evaluating the time taken for the swarm to reach a consensus. We examine the performance under different communication constraints and varying decision-making strategies. This work was accepted at *IEEE / RSJ International Conference on Intelligent Robots and Systems* (IROS 2022) [1]. T. G. Kelly, M. D. Soorati, K.-P. Zauner, S. D. Ramchurn, and D. Tarapore, "Collective decision making in communication-constrained environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7266–7271
- In Chapter 4 we examine the trade-offs in hybrid swarm control between communications to an operator and the performance of the system. An environmental monitoring task is utilised to compare a simple hybrid approach against a hierarchical and a decentralised swarm control structure. We study how performance and communications to an operator can be varied to adhere to end-user requirements. This work has been accepted to *The International Symposium on Distributed Autonomous Robotic Systems* (DARS 2024) [2]. T. G. Kelly, M. D. Soorati, K.-P. Zauner, S. D. Ramchurn *et al.*, "Trade-offs of dynamic control structure in human-swarm systems," *arXiv preprint arXiv:2408.02605*, 2024
- In Chapter 5, the hybrid system introduced in Chapter 4 is extended to examine the mechanisms required by a swarm system to be able to transition from a centralised system to a decentralised system and back again. An environmental monitoring task is also utilised here with agents being required to vary and optimise parameters related to the level of (de)centralisation according to the state of the environment and preferences given by a human operator. This chapter contains work that has been completed in preparation for submission to *Swarm Intelligence*.
- In Chapter 5 we conclude this thesis with a summary of our results and provide some directions for potential future research.

1.6 Impact

This research project is aimed at addressing some of the issues faced by large-scale swarms in gaining and maintaining situational awareness. A focus will be placed on considering scenarios that see the swarms deployed into hostile, adversarial and rapidly changing environments. While much of the project, particularly initially, will likely focus on using simulations to test and develop algorithms, there is scope within the project for the developed solutions to be deployed and tested on real robots. Regardless of whether the project applies any algorithms to real robots, it is important to consider the possible implications of the outcomes of the project on a wide variety of stakeholders. This is particularly pertinent, given the defence and security applications of the project.

In 2019, the Defence and Security Accelerator (DASA) injected £2.5m into developing drone swarm technology. This came following £31m being allocated into new mini-drones and a total of £160m being pledged for developing swarm squadrons [55]. Such projects are aimed at developing technologies that will give troops and pilots an edge in hostile situations, as part of the Tempest future combat air system [56, 57]. This may come from providing greater awareness, or additional tactics involving the swarms. Overall, it is hoped that military forces will have a greater ability to operate in more dangerous and complex environments in future engagements [55].

Enabling the use of unmanned swarms of vehicles will provide multiple benefits in such scenarios. Firstly, it will likely allow for lower operation costs, with swarms accompanying pilots as 'wingmen'. This could result in having one fighter jet with multiple drones, being able to perform the same task as several jets, but with a much lower operation cost. This would also ensure that fewer people are put into dangerous and complex situations, protecting pilots and troops.

It is clear to see that social desirability is at the forefront of a project such as this, in improving the ability for defence and security forces to respond to threats in many different environments. Given this, there are a vast number of possible defence and security applications, including [58]:

- Improving the ability to gain and maintain situational awareness.
- The ability to provide medical assistance.
- The resupply of logistics.
- Identification, and disposal of explosive ordnance.
- The confusion, deception and neutralisation of threats.

As such, there is a wide range of possible applications for the research that aim to provide benefits to not only personnel responsible for maintaining the security of assets but also the people who will be protected by systems that make use of the research. However, there is also the possibility that the research could be exploited to counter such systems. With this in mind, it may be pertinent to consider withholding some information from publications of the research, relating to specific applications of such research. Instead, the focus should be on the general framework of the approaches explored and proposed. Such research should also be extended or adapted with time, to ensure that any exploits that may be developed to counter the techniques proposed, can be responded to. It is hoped that such strategies will help to ensure socially desirable outcomes.

With a project such as this, all possible stakeholders must be considered. This will help to ensure trust and confidence in the systems that make use of the research carried out in the project. This is particularly important considering that such systems could be deployed alongside military forces in hostile engagements. These stakeholders include the people who will be making decisions in such scenarios, which may be high-ranking military officers or operators of the swarms. These kinds of stakeholders will need to be trained to be able to anticipate the responses and capabilities of the systems that use the research. This will enable them to understand the limits of such systems, such as the time it takes for an input to be reflected in the actions of the swarm. It is also vital that the people who such systems may be working alongside the swarms in scenarios are involved, such as ground troops or pilots. By considering both sets of stakeholders, it is hoped that the resulting systems that use research and techniques developed in this project will be inherently trustworthy. This will be aided by ensuring that focus within the project is placed on ensuring any methods that are explored account for explainability.

Not only is the explainability of the systems important to consider from a trustworthy point of view, but also from an accountability standpoint. As has been highlighted, many applications place human lives in danger and under the protection of systems that may use the outcomes of this project. As a result, it is important that any decisions that are made by such systems, can be traced and examined to ensure that the final decision that is made can be made with confidence that all the available intelligence and data has been used effectively. This is something that must be addressed given that some uses of the research could be controversial and placed in the public eye, should it not perform to a high standard.

To keep the numerous stakeholders involved in the research and applications of the project, most trials of the research will likely be conducted in test environments in simulation, however, if the research were to be used beyond this, and applied to real scenarios, multiple tests would be needed before deployment to build the level of trust and confidence in operating such a system that has previously been outlined for each stakeholder.

Chapter 2

Literature Review

One challenge that robot swarms face is in their ability to communicate mission critical information not only between members of the swarm, but also with humans. These humans may be acting as supervisors, operators, team-mates or a mixture of these roles within the human-swarm system [37]. It is vital that the swarm is able to distil the vast amounts of information that it is collecting to the humans in an effective way. This information may include the current state of the swarm, environmental estimations from sensors within the swarm, the predicted future state of the swarm or advice on the next course of action that a human or the swarm should take. With swarm systems typically consisting of a large number of members, sometimes up to thousands of individual robots, presenting information from each, would easily overwhelm the ability for a human to effectively monitor and control the swarm [59]. When operating in an unknown environment, a swarm must be able to coordinate, even in the face of potentially limited and changing communication conditions. This is important, not only to maintain and disseminate up to date information throughout the swarm but also to relay appropriate information to human team-mates, whether this is the current state of the swarm and the environment, the projected future state of the swarm and the environment or the intent of the swarm, relating to what the swarm anticipates will be its future actions. By doing this, while maximising the coverage of the swarm in an area of interest, the swarm can improve the situational awareness of humans that are using the swarm to accomplish tasks in a dynamic environment. This literature review will focus on exploring the existing literature relating to a swarm's ability to maintain situational awareness, coordination and mapping in an unknown environment and in maximising the coverage of a swarm.

2.1 Swarm Coordination

The issue of co-ordination in communication constrained environments has been extensively studied in the context of multi-robot teams. Multi-robot teams, while similar to robot swarms, in that they consist of a number of individuals that carry out a task, differ in that individuals are often capable of carrying out a complex task, such as the exploration of an area regardless of how many individuals make up the system. They may coordinate and communicate the same as a swarm but may still be able to go about performing a task in their own complex way. Swarms on the other hand, rely on emergent behaviours that come about as a result of simple rules and interactions between members of the swarm, in order to achieve a task, that would be impossible for a single member of the swarm to achieve.

In [44], Hollinger et al. presented a decentralised algorithm where agents planned paths based on the time each individual would be disconnected from each other, in a mapping task. Each agent planned to reconnect at some future time to share information gathered about the environment and to replan. One of the main drawbacks to this approach is that each agent requires prior knowledge of where reconnections will take place and does not consider scenarios where individuals fail to reconnect in the future. The idea of planning when, where and with whom to communicate was also examined in [10]. Here, Best et al. developed an algorithm to have agents plan a communication schedule, deciding who to communicate with at future time steps. They incorporated assumptions about imperfect communication conditions and made use of the Decentralised Monte Carlo Tree Search (Dec-MCTS) algorithm for the path planning in an information foraging task. This approach improved on the method presented in [44] in that imperfect communications were incorporated into the design of the system, so that even if an individual could not communicate with its target, it would plan to reconnect in the future. This new approach however, aims to plan optimally and plans many steps ahead. In an environment that is dynamic, this may result in time that is wasted planning for an eventuality that is obsolete a few time steps down the line. It is also worth mentioning that optimal planning can be computationally expensive and may not scale well to relatively simple robotic platforms.

Vandermeulen et al. addressed the problem of having a set of robots reconnect communication capabilities with other members of the swarm without making explicit plans [45], extending and improving on the work done in [44]. Each member of the swarm maintained beliefs over the locations of other individuals in the form of a graph. This was used to find the optimal paths for reconnection. When team-mates connected, they exchanged information on their beliefs and acted based on their objective function which was defined as the probability that they would find their targets for reconnection. While this approach did consider dynamic environments and swarm topologies, it again required each agent to plan several time steps ahead, despite the fact that any plans made may become useless soon after.

In [46], Teacy et al. proposed an algorithm that used a radio propagation model to influence swarm behaviours dependant on how an action would affect the quality of service of the individual's communication. Using this algorithm, connectivity was maintained within the swarm and uncertainty about the model of the communication environment was incorporated. This approach incorporated uncertainty information related to the radio signals between members of the swarm, which makes this approach potentially applicable to environments with dynamic communication landscapes. However, once again, it does not consider scenarios where members of the swarm become unable to exchange information with others and contingency plans for when communication is not possible. Majcherczyk et al. also addressed the issue of preserving connectivity within a swarm in [47]. Here, the authors made use of two algorithms that were used to construct a long-range communication backbone between two points by treating the swarm as a logical tree. Constraints were placed on the tree to ensure a certain flow of data between the two points can be maintained. This method provides an effective way of enforcing connectivity constraints on a swarm network while being resilient to faults that may occur in the network due to changing communication landscapes.

Ruetten et al. [60] proposed an algorithm to find optimal configurations of a network of UAVs. This method made use of virtual spring forces between members of the network, similar to early physics based swarm control methods [61, 62]. While other approaches define these forces based on proximity to other members of the swarm [61] or obstacles [62], here the forces were based on the radio signal strength between two individuals. Members of the swarm exchange positioning information with certain neighbours that they 'rely on', and is determined at the beginning of the task, however, this means that some members of the swarm do not exchange information with individuals which may move to be in close proximity of one another during the task. As a result, this method is not suitable for rapidly changing swarm topologies and dynamic communications environments.

In [63], Wang et al. developed a decentralised algorithm that was tolerant to faults [63] and was used to enable a swarm of self-organising robots to cover a pre defined area. The swarm would generate gradients, according to their distance from a leader. Each agent periodically queried the gradients of neighbours and moved to cover the area. The approach itself required very little computational power making such an approach ideal for low complexity systems. While this approach worked well in a simple coverage problem with a number of grouped swarms, the deployment of individuals is relatively slow and may not be suited to dynamic environments where the area of interest that the swarm needs to monitor may be changing very quickly.

In these approaches, the environment that the swarms are observing are static, with the relative proportions of features remaining constant throughout the experiments and do not consider scenarios where these proportions may be changing. It is also apparent in these approaches that there is very little done by the swarm in the way of actively improving their chances of converging towards a decision. In each of the approaches given above, members of the swarm perform random walks throughout the course of the task. A key benefit of this is that the swarm will not have any biases towards certain areas of the environment where the the features may not be a good representation for the environment as a whole. However, the overall performance of these algorithms could be improved by making the swarm more proactive in communicating information around the rest of the swarm. Taking inspiration from other works explored in this section, relating to the idea of making agents in the swarm aware of their communication environments, such as in [10, 46, 60], could improve the collective decision making and allow for more reliable convergence towards a decision in such scenarios.

The problem of coordinating a set agents to maximise coverage is one that has been extensively addressed in the context of wireless sensor networks (WSNs), but has recently also been applied to swarm systems. Many approaches consider maintaining connectivity while maximising the coverage of the network by finding an optimal deployment of sensors to cover the environment, given the size of the environment and the number of sensors available [64]. Early work by Zou et al. in this area made use of the virtual forces algorithm [65] originally proposed in [66]. This algorithm aimed to maximise the coverage of the sensors, while minimising the collisions in the network, by reducing the number of adjacent sensors. This method has been extended by several works [67, 68] and is also a common approach for the control of robot swarms [37, 16]. Other approaches to maximising swarm coverage include using computational geometry via the construction of Voronoi diagrams based on sensor locations [69] and Delaunay triangulation [70]. Each of these approaches aim to minimise the number of sensors used to cover the area.

Work by Ruetten et al. [60] focused on the problem of optimising a network of UAVs to maximise area coverage in a disaster scenario. Here the agents organised according to virtual spring forces that were defined based on the radio signal strength between two individuals. The UAVs maintained readings of radio signal strength for other individuals that they ”rely on”, however, this is determined at the beginning of the deployment and does not change during the task which results in sub optimal configurations for larger numbers of individuals over time [60] resulting in a relatively poor coverage of the environment. Wang et al. also considered the problem of optimising the deployment a swarm of robots to maximise coverage and they developed a decentralised algorithm that was tolerant to faults [63]. This algorithm utilised a simple gradient based approach whereby agents generated a gradient within their group, according to their distance from a leader. While this provided a low complexity algorithm, that was resilient to faults in swarm individuals, the organisation of the swarm was relatively slow and may not hold up to environments that change rapidly, or where the area of interest is not initially known.

In [71], Stranders et al. developed a greedy algorithm to determine the optimal subset of sensors that should be activated in order to maximise coverage, while minimising message conflicts that may occur over the network. Unlike many previous works, this approach also considered a reconnection phase, that would be carried out when sensors were lost due to their power being depleted. This contingency plan in the event that sensors are lost is something that is vital in robot swarms whose topology may change very rapidly, requiring constant refinements to the distribution of the swarm to maximise coverage. This idea is not commonly addressed in the work relating to WSNs. It is also proposed by Vandermeulen et al. in [45] that their reconnection algorithm could be applied to a coverage task, but is not formally evaluated.

Current approaches to optimising swarm coverage have addressed issues such as maintaining quality of service [60, 72] and fault tolerance [63]. One area that has not been addressed is that of the issue where contact is lost with members of groups of the swarm, which may occur regularly in real robot swarms due to changing topology and environmental factors such as moving into a forest environment. While Vandermeulen et al. [45] present an algorithm that may be used in a coverage problem, this approach requires optimal planning to re-establish connections. In high risk situations, particularly those that involve human operators, it is vital that the swarm attempts to regain connection with lost members quickly to reduce the loss of situational awareness of an operator [28] and before any belief models become highly uncertain. Approaches that use more reactive techniques such as that presented by Ruetten et al. [60] may provide better options for quickly regaining a connection. The short comings of such methods are that predicting future changes in the swarm is difficult due to the reactive nature of the swarm.

2.2 Collective Decision Making

The collective decision making process itself is also a vital element of a swarm system being able to function autonomously [16]. When the swarm is located in an environment which imposes limitations on the swarm's ability to communicate with one another, which may be in the form of obstructions in the environment (e.g. foliage, electric cables), this can hinder the decision making power of the swarm and may lead to scenarios where a decision takes much longer to be made. In scenarios that are time critical, it is important that decisions are made quickly and accurately by the swarm in order to improve the system's situational awareness [22]. For example, consider a swarm that has been tasked with assessing the best route of a finite number of options, to take through a densely packed forest, following a disaster. It is unknown how the existing routes have been affected by the disaster and may pose a risk to humans entering the area. The speed of a swarm to reach a consensus may also vary depending on the control structure of the swarm. The two main mechanisms being centralised and decentralised control.

Coping with failure is a major challenge for designing adaptive robotic systems. Swarm robotics aims to solve the issue of individual robot failures by using redundancy in the system. A large group of simple robots locally interact with each other and the environment yielding an emerging behaviour. While being characterised as adaptable, scalable and robust, robot swarms face challenges in the design of decentralised collective decision making processes. A collective decision-making strategy has to define a process that solves the *best-of-n* problem where the swarm is tasked to identify the best choice from a set of given options [73].

The collective perception problem was introduced by Valentini et al. [52] where robots are tasked with feature detection. In the problem they posed, the agents are able to locally observe features in their environment, these being black and white tiles, and must estimate whether there are more white or black tiles present in the environment. Agents maintain an estimate and a belief over which colour is most dominant and are able to share this information with neighbouring agents. Over time, the agents collate their estimates to form a consentaneous belief about which feature is more dominant in the environment. The trade-off between speed and accuracy is examined for different collective decision making strategies, each of which had been previously proposed [54, 53]. These approaches focused on making use of positive feedback mechanisms to improve the speed at which the swarm converges to a consensus. This positive feedback mechanism was achieved by modulating the time agents spent attempting to share their beliefs and estimates.

In [48], Ebert et al. propose a multi-feature collective decision making algorithm for a swarm of robots, which is applied to a scenario where the swarm must collectively estimate the proportion of a set of options within an environment. Members of the swarm switched between two states, observation and dissemination. During the observation phase, individuals would only observe options in the environment, and would communicate this with other members, for an amount of time that is proportional to the confidence of their prediction for the feature. The authors show that their algorithm converges to the correct overall distribution in finite time. This approach is similar to the one used by Bartashevich et al. in [49] in which the authors also made use of an observation and dissemination phase, which was switched based on the quality of each individual's estimate, derived using evidence theory.

A key element of the approaches that have been proposed in the literature is the mechanisms that under-pin the collective decision-making of agents. These mechanisms dictate how agents update their beliefs and estimates over the observed options in the environment. An agent does this by incorporating observations of option quality along with beliefs that are shared by neighbours through local communication [74]. This social dissemination of information on option qualities drives individual agents to make decisions over the options in the environment and the system succeeds when a majority of agents choose the best-quality option. While many mechanisms have been proposed, the most well-known ones are the voter model and majority rule [16]. Agents using the voter model update their belief to the belief of a random neighbour [75]. When using the majority rule, agents update their belief based on the majority belief of their all their neighbours [52, 75]. Valentini et al. [53] compared these two types of models to examine the trade-off of speed and accuracy when using the models. They found that on average the majority rule mechanism allowed for faster convergence to a consensus, at the cost of accuracy compared to the voter model which was slower to reach consensus but did so more reliably. Some notable mechanisms that have been proposed to improve the speed without sacrificing accuracy are using distributed Bayesian hypothesis testing [76] and the use of Social Impact Theory in the form of the Ising model [49]. Ebert at al. [48] made use of the voter model and extended it to give agents a long term memory of previously adopted beliefs to allow for smoother convergence to a consensus in a multi-feature scenario. More recently, evolutionary robotics approaches have been proposed to evolve decision-making mechanisms [77, 78, 79, 74, 8]. These approaches typically focus on evolving artificial neural networks (ANNs) which dictate the decision making behaviours and define the opinion selection of robots. The aim of this kind of implementation targets improving the adaptability of the decision making behaviours for dynamic environments and scenarios with a larger number of options present in the environment [74].

Prasetyo et al. also proposed an extension of the best-of-n problem to dynamic environments [80]. In this problem, option qualities can abruptly change over time. This requires an adaptive system to allow individual members of a swarm to modify their opinions over the best option even after a consensus has been reached. This is something that a traditional voter model cannot achieve [81]. They proposed two mechanisms which modify the voter model, enabling it to be adaptable. The first mechanism relied on a small proportion of stubborn agents which do not change their initial opinion and remain committed to this option throughout the experiment. The second mechanism relied on spontaneous opinion switching where each agent may probabilistically change their opinion, independently of the mechanism used to decide on the best option.

Similarly, Soorati et al. considered a collective perception problem with a dynamic environment where option qualities change over time. Here, the collective perception problem introduced by Valentini et al. [52] was adapted such that the environment was split into two sides. The swarm was required to decide which side had the highest proportion of white tiles. This proportion associated with each side was varied over time, requiring the swarm to exhibit adaptability. Individuals in the swarm relied on local, multi-hop communication to propagate information about their beliefs throughout the swarm in order to create a global reference frame. The proposed solution relied on preventing the swarm from being able to lock in its decision. This was achieved by implementing a confidence score over the available options which was incremented based on received opinions and periodically reduced in order to avoid individuals accumulating large confidences which would stop them from reacting to changes in the environment.

Against this background, our approach considers a scenario where agents' communication is constrained by the environment. As such, agents must adapt their behaviour based on their current communication conditions. Collective perception is mainly considered in environments with homogeneous communication quality distribution [54, 53, 52]. It has been shown that collective behaviour is possible even without any communication but the emerging behaviour is limited to simple spatial configurations (e.g., aggregation) [82, 83]. Limited range and spatially targeted communications have also been proposed that are design decisions to improve the collective behaviour rather than an externally-imposed constraint [84, 85, 86]. The resilience of different control approaches to communication noise and temporal constraints are considered [51, 87] but, to the best of our knowledge, literature lacks a collective decision-making strategy that adapts to environments with inhomogeneous communication quality distribution.

2.3 Hierarchical Swarm Control

A swarm system presents a unique opportunity to exploit a huge number of agents and the redundancy that comes with it to complete tasks that would be impossible to solve by a single agent. Often, a decentralised approach is proposed as a scalable and resilient method for organising swarms. However, centralised approaches can provide more efficient solutions in certain applications [6]. Finding an optimal level of centralised control and decentralised control that can be adapted to changing mission requirements could provide a swarm control method that could make use of the benefits of both approaches. Such a method is an important aspect for the future of robot swarms [32].

Zhu et al. proposed a method for having a swarm establish distributed control via a self organised communication network [5]. To do this, the network was given a target topology which the swarm must form. Using the mergable nervous system (MNS) approach, proposed by Mathews et al. [4], a set of robots would form a self organised directed rooted tree that would be used as a communication network. The root of the tree would act as the brain of the MNS. The swarm would also be given a target topology for the network based on the scenario. This topology was made up of a graph, G, and a set of attributes, A, which relate to links within G. These attributes included information such as the orientation of a child target robot relative to its parent robot and what type of robot the child should be (relating to a heterogeneous swarm). If the robot is a brain then it has access to the full graph, G and set of attributes, A, as its reference, otherwise a robot will receive a subgraph of G with its associated subset of A. The authors examined the swarm's ability to recreate the target topologies and also examined the fault tolerance of their approach. They found that using a MNS approach proved to be resilient to robot failures and also to displacement of the swarm.

Jamshidpey et al. extended this work by using the idea of a MNS approach for controlling a heterogeneous swarm of robots [33]. In this work, the authors considered a swarm comprised of UAVs and ground robots in a coverage task. The aim was to ensure an even coverage of the environment with robots spending an equal amount of time exploring small areas of the environment. In their approach, UAV robots acted as the brains in the MNS and would coordinate to control the ground robots to ensure even coverage. Again, the swarm relied on a set of target topologies to do this. This MNS approach was compared to a fully decentralised approach and a fully centralised approach in which one UAV controlled all the ground robots. The uniformity of the coverage was used to compare the performance of each approach. The authors found that their MNS approach significantly outperformed the decentralised approach but was slightly outperformed by the centralised approach. However, MNS approach did prove to be robust to failures.

The relative advantages and disadvantages of centralised control and decentralised control were studied by Jamshipdey et al. [6]. They compared a predetermined approach, centralised approach, decentralised approach and a hybrid approach which were modelled in the ARGoS simulator [88]. In the decentralised approach, agents explore the environment, moving randomly without any supervisor or coordinating agent. The centralised approach saw agents make use of a single supervisor agent that was able to give motion commands to the other members of the swarm, which are calculated dynamically based on the swarm's exploration of the environment. Similarly for the predetermined approach, the swarm made use of a centralised member, however in this case, paths were calculated offline by the supervisor, based on *a priori* knowledge, and then passed onto the members of the swarm. The hybrid approach made use of the remote mergeable nervous system which was proposed by Zhu et al. [5]. A coverage task was used to compare the various approaches. The authors found that, unsurprisingly, the predetermined approach performed the best, achieving the most complete coverage in the quickest time. However, the centralised and hybrid approaches performed similarly and resulted in a high coverage completeness, while exhibiting a higher level of fault tolerance than the predetermined approach. The hybrid approach was also resilient to a single point of failure and a bottleneck in the communications, a property that the centralised and predetermined approaches did not show.

Work considering theoretical foundations of self-organised hierarchical control frameworks has also been considered by Zhang et al. [34]. Although this is a relatively young area of swarm robotics, hierarchical swarm control is a promising area of research that has important implications for the future of swarms [32]. Work in this area, so far has focused on the application of a mergeable nervous system to distribute and coordinate a swarm based on a set of network topologies. One interesting avenue of research here may consider how a swarm might self organise to form its own topologies, given a set of application and scenario requirements. Whether these requirements are specified by a human operator or are implied by the environment that the swarm is operating in, particularly when the swarm's communication is constrained. Another interesting area that could be investigated here is in the way that leaders within the swarm (or brains in the case of the MNS) coordinate to allocate their available child robots to accomplish a set of tasks in the environment.

Against this background, this thesis aims at understanding how a simple hybrid control approach compares to either of the two (central or decentralised) in the overall system performance, the volume of messages sent to the operator (associated to cognitive workload) and the number of messages passed within the swarm. We study the trade-offs in using various levels of centralised coordination in an environment monitoring application where tasks appear randomly and the agents must move to the task area as soon as possible. We consider the human-swarm interaction implications intrinsically within the swarm behaviours. Previous literature has examined the use of communication within a hybrid control approach but, to our knowledge, never from the perspective of a human swarm system and how such communications could potentially affect a human in the loop. We extend previous work done using photomorphogenesis for robot self-assembly [89] by allowing the formations to reconfigure themselves depending on the environment via the swarm behaviour. A recent study confirms that the swarm control strategy can be modified to reduce the cognitive workload [90]. We, therefore, apply a dynamic hierarchical control approach and study the communication and the performance trade-offs that exist in using such a hybrid centralised and decentralised approach.

2.4 Mapping in Unknown Environments

Simultaneous localisation and mapping (SLAM) has been explored in the context of multi-robot systems, with early work from Thrun et al. providing a framework for mapping 3D environments with multiple robots [91], expanding on the literature from Elfes and Elfes et al. which provided initial methods for single robot mapping and navigation using sonar [92] and occupancy maps [93]. Other methods have included the use of Extended Kalman filters [94] and particle filters [95]. It has only been recently however, that swarm SLAM has been considered with many of the existing methods for SLAM relying on centralised architecture [96]. In [97], Ramachandran et al. presented a distributed algorithm for constructing occupancy maps using a swarm of robots. Their method made use of Lévy walks, which are a variant of random walks. Each member of the swarm, took laser range sensor readings and broadcast these to their neighbours. The authors showed that the resulting map generated by the swarm converged to the actual map. Kegeleirs et al. also performed a swarm SLAM experiment in which they use the Multirobot map merge algorithm, presented in [98] then compared the quality of the maps generated by five different variants of random walks [99].

While the presented approaches focus on mapping unknown environments, they do so in environments that are static and do not consider environments that may be changing very rapidly. We address this problem by utilising a probabilistic model of the swarm's communication conditions in the form of a GP. Osborne et al. presented an algorithm which was used to decide when and which sensor to acquire readings from in a WSN [100]. It made use of a multi-output GP to predict missing sensor data, which may have been caused by network outages. This estimation was made with an uncertainty that was maintained to be below a specified threshold. O'Callaghan et al. also made use of a GP to map dependencies between points in an environment to estimate the occupancy for that area [101]. This makes use of the occupancy grid mapping framework developed by Elfes et al. [93]. The model was able to account for sparse and noisy readings taken in the environment. Such a method however can be computationally expensive and may not scale well to simple robotic platforms, particularly when updating the GP, however, some of the methods employed by Osborne et al. may aid this for example, in their use of a windowing technique to reduce the number of data samples.

In [102], Luo et al. utilised Gaussian processes (GPs) to estimate the distribution of environmental phenomena in a multi robot sensor coverage task. The robots were tasked with finding the optimal stationary configuration of robots in order to maximise the sensor coverage. Each robot maintained a GP over it's belief of the distribution of the phenomena and used a mixture of GPs and active sampling to estimate the global distribution in the environment. While GPs provide a powerful method for estimating information in a range of applications, they must be used with care with robot swarms, given a swarm's low individual computational complexity and the potentially high computational demands of updating and maintaining a GP. Such methods could however, be used to estimate projections of future states of a swarm, along with uncertainties. As has been discussed in section 2, this kind of information can help to reduce the impact of limited communications in terms of latency [29] and may help to enhance the situational awareness of human operators [24]. Such mapping techniques could be used to estimate the communication conditions within an environment, accounting for uncertainty of the generated map and push members of the swarm towards areas in their environment where they are most likely to disseminate relevant information in an efficient way and to the most members of the swarm. This could aid in maintaining more accurate and up to date information pertaining to the state of the environment and facilitate quicker convergence towards decisions, that the swarm needs to make.

Chapter 3

Communication Constrained Collective Decision-Making

3.1 Introduction

A key issue that is highlighted from the literature is a swarms ability to cope with uncertain and rapidly changing communication conditions. Environmental factors can impact a swarms ability to effectively communicate with one another. For example, the presence of trees in a forest can cause intermittent communications, by breaking the line of sight between robots. The rapidly changing topology of the swarm can also affect the optimal location in the environment to communicate with other members. To address this issue, we propose a method for mapping the communication conditions of the environment in which the swarm is operating. Previous approaches to improving the communications within a multi-robot team have proposed planning methods to determine when and with whom to communicate with [44, 10, 45]. As has been discussed, these approaches aim to find an optimal solution, by simulating and planning many timesteps into the future. Such plans could become obsolete quickly and may require replanning, resulting in a waste of computation. These approaches cannot also quickly formulate contingency plans, when communication is not possible.

In this chapter, the work advances the state of the art in the following ways: First, a new benchmark scenario is proposed to study the collective perception of a robot swarm in a communication-constrained environment. In this scenario, inter-robot communication may be completely denied in *a priori* unknown areas for the duration of the experiment. Second, a set of algorithms is developed for the swarm to adapt to these communication restrictions. These algorithms help the swarm achieve consensus at a high speed, on decisions over features of interest in the environment.

Finally, the results show that our algorithms outperform the state-of-the-art collective decision-making algorithms in communication-constrained setups.

3.2 Simulation

The experiments performed in this chapter were run on a simple simulator which was designed to focus on the interaction between agents in the environment. The agent class allowed for easy integration of different behaviours within the experiment. Within this class, the agents' communication, movement and decision-making strategies were defined. The focus on the agent interaction and the simple nature of the simulator also allowed for a large number of agents to be simulated before the time taken to run each timestep made it infeasible, in the simulator's current version, to increase the number of agents present.

3.3 Environment

Agents move at a maximum speed of 1 m/s in a continuous bounded environment, $Q \subseteq \mathbb{R}$ with their position given by $(x, y) \in Q$. The size of the environment is 64m $\times$ '64m. Agents use one of two movement programmes during the experiments these being the communication-aware path planning algorithm, described in 2.4.4, and a random walk routine which is as follows. An agent moves in a straight line for a random distance, d, with a mean distance of 5m that is sampled from an exponential distribution, given by:

$$d = \lambda e^{-\lambda x} \tag{3.1}$$

where λ is the rate parameter, and for our experiments, $\lambda = 0.2$. After travelling this distance, agents perform a turn. This turn has equal probability of being performed in a clockwise or counter-clockwise direction. The agents turn for a period of time that is uniformly distributed between 0 and 2 seconds. Then repeats these steps for the duration of the experiment. We also implemented a collision avoidance algorithm which is as follows. When agents detect one or more nearby obstacles, defined as neighbouring agents within 2m or the bounds of the environment, agents reduce their speed to 0.2 m/s and calculate the distance and bearing of each obstacle. The agent then defines a new direction that is opposite to the direction of the obstacles and turns clockwise or counter-clockwise to align its direction of motion to match the computed direction. The agent then continues with its walk routine.

3.3.1 Collective Perception Problem

We define a collective perception scenario with 2 features ($|f| = 2$). Agents are initially placed randomly within the environment with their initial (x, y) values drawn uniformly. Every point in the environment is assigned a binary value, representing the value of the feature at that location, given by equation 3.2.

$$f_i : Q \longrightarrow \{0,1\}, \quad i \in \{1,2\} \tag{3.2}$$

Where Q is the environment and i is the feature, corresponding to a value of either 0 or 1. The feature is locally defined at every point (x, y) in the environment. It is assumed that agents have the ability to obtain and observe the features in their environment at their current location. In our experiments we do not consider noise present in the sensing ability of agents. The task of the agents is to decide which feature is dominant in the environment. For our experiments, the feature of 1 is always the dominant feature and, as such, the swarm's task is to reach a consensus for 1. The ratio of points which correspond to a feature of 1 in the environment is given by r_f in equation 3.3.

$$r_f = \frac{1}{|Q|} \iint_Q f(x,y) dA \tag{3.3}$$

An example of the feature environment can be seen in figure 3.1 which shows an environment with a fill ratio of $r_f = 0.65$. In the figure, the white area corresponds to points in the environment with a feature value of 1, while the black areas correspond to points with a feature value of 0. The feature distribution is characterised by a grid consisting of 4m $\times$ 4m cells. Algorithm 1 was used to generate the feature distribution. The ratio of features after generation, r_g, is checked and the distribution is accepted if the ratio is within 1% of the expected ratio, until $|r_f - r_g| \leq 0.01$. Otherwise, cells are inverted to either increase of decrease the ratio towards the accepted threshold.

3.3.2 Communication-Constrained Environment

The environment is also characterised by a communication quality, q_c, defined for each point, $q_c \in [0, 1]$. This quality relates to the probability that a message broadcast by an agent is successfully received by neighbouring agents in communication range. Communication quality varies across the environment. Moreover, the agents are denied communication when in some regions of the environment ($q_c = 0$). The ratio of communication denied cells in the environment is given by r_c. Agents are able to perceive the value of the feature at their current location in the environment but must estimate the communication link quality themselves. When an agent is at a location

Algorithm 1 Feature Generation

for each Cell $\in Q$ **do**
    if $A \sim \mathcal{U}(0, 1) < r_f$ **then**
        $Cell \leftarrow 1$
    else
        $Cell \leftarrow 0$
    end if
end for
$r_g \leftarrow \sum_{i=0} Cell_i$
while $|r_f - r_g| \geq 0.01$ **do**
    $i \sim \mathcal{U}(0, 1)$
    if $r_f - r_g > 0.01$ **then**
        $Cell_i \leftarrow 0$
    else
        $Cell_i \leftarrow 1$
    end if
end while

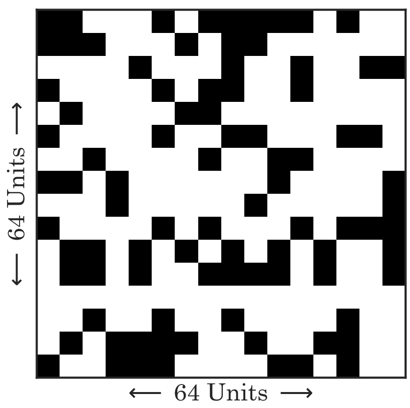
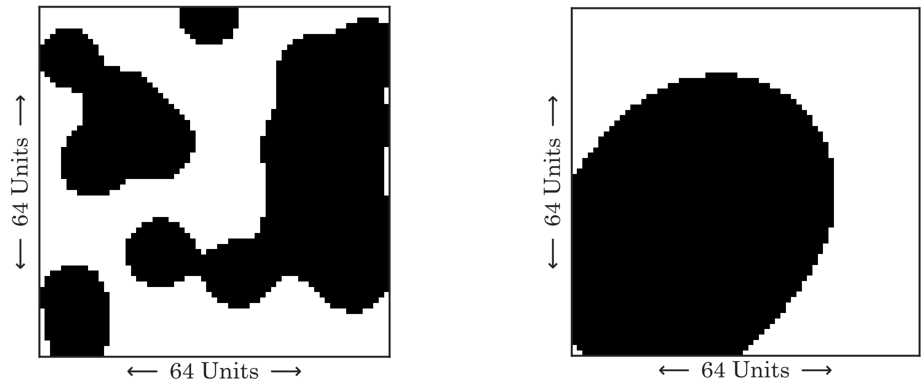


FIGURE 3.1: Example of an environment with binary features, indicated by black (feature = 0) and white (feature = 1) squares, and with $r_f = 0.65$. This means $65 \pm 1\%$ is white with a feature value of 1.

where $q_c = 0$, that agent is unable to transmit messages to neighbours, but they are able to receive messages from neighbours within communication range.

To generate the communication denied areas, we use a method that takes inspiration from the method proposed in [103] for generating surfaces of objects in computer graphics. First, several centroids are uniformly generated in the environment, each of which has an assigned radius which is also uniformly generated. This allows us to define a force-field function, given by equation 3.4.

$$F(p) = \sum_{n=1}^{k} \frac{c_n}{|m_n - p|^g} > \beta \tag{3.4}$$



(A) (B)

FIGURE 3.2: Examples of a distributed communication denied (B), and a continuous communication denied (B) environment. In both environments, agents are denied communication in half the environment ($r_c = 0.5$), indicated in black. Areas shaded lighter allow communication with a higher probability.

Where the force, F, acting on a point in the environment, $p = (x, y)$ is given by the summation over all k generated centroids, c_n. m_n is the location of centroid n while g defines a parameter which dictates how the overall shapes are generated. In our experiments, we consider two types of communication constrained environments; distributed and continuous communication constrained environments. Examples of these can be found in figure 3.2. The distributed communication denied environment is characterised by a number of small regions in the environment where agents are unable to communicate with one another, while the continuous communication denied environment has one large area where agents cannot communicate.

These two different environments are generated by varying the centroids used to generate the field, characterised by equation 3.4. For the distributed communication denied environment 18 centroids were drawn uniformly, which resulted in an environment shown in figure 3.2(B) with more merged comms denied areas. For the continuous communication denied environment, 2 centroids were drawn uniformly resulting in an environment shown in figure 3.2(A) with more separation between the communication denied areas. β defines the threshold that decides whether point p is communication denied. Each point in the environment is evaluated against this function, with $F(p) \in [0, 1]$. The ratio of communication denied points in the environment is compared to the desired r_c value for the experiment. The threshold, β is then iteratively incremented or decremented until $|\beta - r_c| \leq 0.01$.

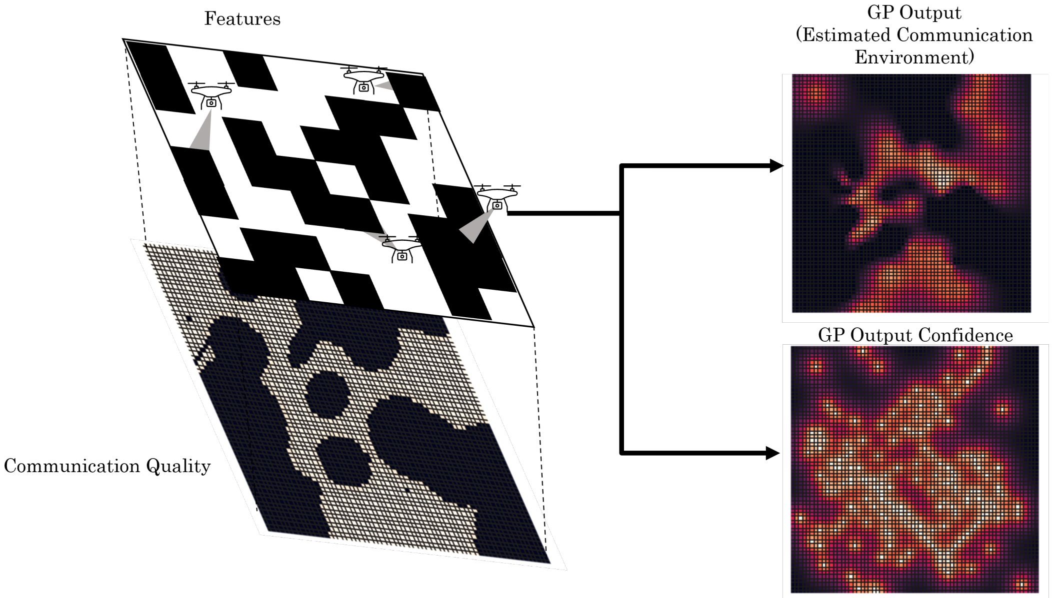


FIGURE 3.3: Diagram showing the proposed communication constrained collective decision-making problem. Agents operate in an environment with features (black and white tiles) and an underlying communication quality. Agents make estimates over the communication quality at points in the environment and use this data to construct a GP. The outputs from the GP are the estimated communication quality over the environment and the confidence over that estimation.

3.4 Method

Our communication-aware approach consists of three elements. These are the collective decision-making strategies on the dominant feature in the environment, augmented with communication-aware path planning, and coordination algorithms for the swarm. This is outlined in figure 3.3. To evaluate our communication-aware algorithms, we consider two types of communication environments: 1) Distributed: With many, small communication denied areas which are spread throughout the environment, for example, see Figure 3.2a. 2) Continuous: With one large communication denied area, for example, see Figure 3.2b. For our experiments, we consider an environment of size $64m \times 64$m with 36 agents in the swarm. Agents have a maximum communication range, CR, of 5m.

3.4.1 Agent Behaviour

Algorithm 2 outlines the main loop executed by agents at each timestep of our simulation up to t_{max} timesteps. Agents spend their time in one of two states, these being the *Dissemination* state (D), and the *Exploration* state (E).

Agents can communicate with neighbours that are within communication range. In our experiments, this range is set to a distance of 5 body lengths of agents. This

Algorithm 2 Agent Behaviour

while $t < t_{max}$ **do**
  BroadcastHeartbeat()
  ReceiveHeartbeats()
  BroadcastAcknowledgements()
  $obs_c \leftarrow$ ObserveCommunicationQuality(x, y)
  ObservationsUpdate(obs_c)
  if State $==$ E **then**
    $obs_f \leftarrow$ ObserveFeatures(x, y)
    EstimateUpdate(obs_f)
    if $t\%60 == 0$ **then**
      State $= D$, $t = 0$, $g =$ UpdateModulation()
    end if
  else if State $==$ D **then**
    UpdateStoredBeliefs()
    if $t\%120g == 0$ **then**
      State $= D$, $t = 0$
    end if
  end if
  Planning(State)
  Movement(State)
  $t = t + 1$
end while

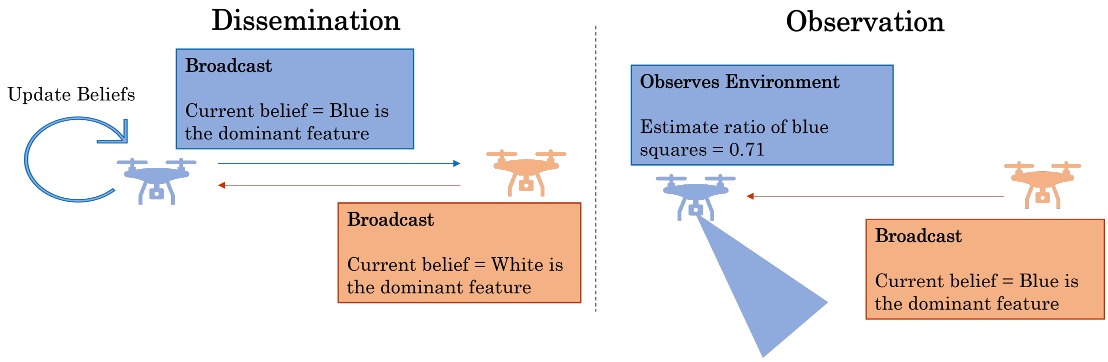


FIGURE 3.4: Diagram showing the operation of agents during the dissemination and observation states. Here the blue agent is the one acting in the dissemination and observation state respectively. In both states, it receives broadcasts from the red agent. In the dissemination state, the blue agent can also broadcast and update its own beliefs.

constraint on direct communication of agents is similar to the communication model used in [48]. As a result, agents can only rely on local interactions, instead of sharing information globally, to all agents, as is done in [52]. This constraint further complicates the problem faced by the swarm in this communication-constrained collective decision-making task. Not only does the swarm need to ensure they are attempting to communicate in an area that is not constrained by the environment, but also that the area has other agents to receive that information. As such, the swarm must be aware of the communication environment to the extent that they can select suitable locations to disseminate information effectively. This approach to modelling communications assumes that agents are reliably able to communicate within a short distance of each other. This could be practically achieved by using low-power, short-range wireless communication technologies such as Bluetooth or Zigbee.

In the D state, agents broadcast their beliefs over the dominant feature in the environment to neighbouring agents within the communication range. In this state, agents cannot observe the features in the environment. The time spent in the D state is determined via a parameter, g, which is defined for each of the decision-making strategies and may last for a maximum of 120 timesteps. Agents in the E state, observe the features in the environment, but cannot disseminate their beliefs to neighbouring agents. The time spent in the E state is fixed at 60 timesteps. When the agent has spent enough time in its current state, it switches to the other state (from D to E or vice versa) and resets its timestep counter, t. An example of this is shown in figure 3.4.

At each timestep, agents estimate the communication link quality at their current location and use this to update their communication quality map, which is modelled using a Gaussian Process (GP). To estimate this value, agents make use of heartbeat messages that are broadcast at every timestep. These messages include the transmitting agent's beliefs over the features in the environment and their current state. While agents always transmit their beliefs, it will only be considered by a receiving agent if the sender's state is set as D. When an agent receives a heartbeat message, they immediately send an acknowledgement message. Agents keep a record of the IDs of neighbours (assigned randomly at the start of the experiment), which they have received acknowledgements and heartbeats from for the current timestep. This process is shown in figure 3.5. Such an exchange allows the agent to estimate the probability that a message is successfully received from the agent's current location by using equation 3.5.

$$obs_c = \frac{|Acknowledgements|}{|Acknowledgements \cup Heartbeats|} \tag{3.5}$$

This can be seen in lines 4 and 5 of Algorithm 2. Agents also perform path planning, based on their communication quality map, in both D and E states and use the result

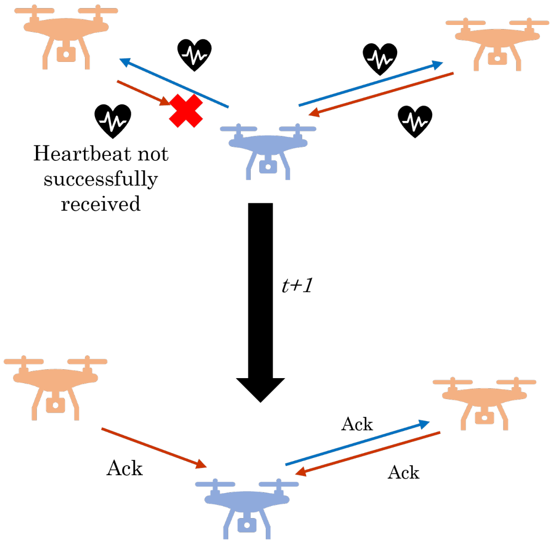


FIGURE 3.5: Diagram showing the exchange of heartbeat messages between three neighbouring agents at two timesteps. In the first timestep, at time t, the blue agent exchanges heartbeat messages with two neighbours. These neighbours successfully receive these messages, but the blue agent only receives one heartbeat message in return. At $t+1$, the agents acknowledge the receipt of the heartbeat messages. Using these acknowledgements and heartbeats, agents are able to estimate the communication quality at their location.

of this planning to move to a favourable location, depending on their state, seen in lines 18 and 19 of Algorithm 2.

3.4.2 Decision Making Strategies

We apply three collective decision-making strategies to test our communication-aware algorithm, which are taken from previous literature, detailed in Section 1.3.2. These are *Direct Comparison of Option Quality* (DC) [52], *Direct Modulation of Majority-based Decisions* (DMMD) [52, 75] and the multi-feature decision-making algorithm proposed by Ebert et al., which will henceforth be labeled as MFDM [48]. These mechanisms were chosen because they each have unique properties related to how an individual agent's beliefs are updated and how the population is able to reach a consensus. DC allows agents to directly modify their beliefs based on quality estimates of their neighbours, while DMMD and MFDM both consider an aggregation of received beliefs from neighbours. Another property which DMMD and MFDM share is in their modulation of the time spent by agents in each phase of the agent's behaviour, while DC sets fixed time periods for each phase. MFDM differs from DMMD in that the agents maintain a concentration value that is used as a long-term memory of the agents' perceived beliefs. This concentration is also used to lock in agents' decision on the dominant feature, something that the DC and DMMD approaches do not do. As such, the effect of communication limitations on these properties will be examined. The full strategy algorithms are as follows.

In DC, agents share their beliefs, $b_i \in \{0, 1\}$, about which feature in the environment is dominant and also their current estimate of the ratio for the dominant feature, ρ_i, during the D state. This information is broadcast to other agents that are within communication range of agent i. When i receives a broadcast from another agent, j, containing b_j and ρ_j, agent i stores this information in a vector, B, where it is held until agent i enters the E state. In DC, the parameter g shown in line 15 of Algorithm 2 is equal to 1, meaning the D state lasts for a fixed period of 120 timesteps in our simulation. When leaving the D state, agents compare their current estimate of the dominant feature with an estimate, ρ_j, selected at random from B, shared by a neighbouring agent of the swarm with belief $b_j \in 0, 1$. If the estimate, ρ_j, is greater than the agent's own estimate ρ_i, then the agent alters its belief to b_j. This belief-update process is outlined in Algorithm 3. During the E state, which lasts for a fixed period of 60 timesteps in our simulation, agents cannot broadcast their own beliefs or estimates of which feature is dominant in the environment, but can receive them from other agents and are limited to moving and sensing in the environment, while building their current estimate, ready for the next D period.

Algorithm 3 DC Belief Update

Require: ρ_i Agent's current estimate, B Beliefs and estimates of dominant feature received from neighbouring agents

```
$\rho_j, b_j \leftarrow$ Random $\rho, b \in B$
if $\rho_j > \rho_i$ then
    $b_i \leftarrow b_j$
end if
```

In DMMD, agents are limited to sharing their beliefs during the D state. Agents maintain an estimate of the ratio for their current belief which is used to modulate the time spent in the D state. This is done by setting $g = \rho_i$. Thus, a higher estimate results in a longer D state and a greater chance of influencing the belief of other members of the swarm. At the end of the D state, agents sum the number of each belief, b_j, received during that D state, including their own, and alter their belief to reflect the most frequent belief. Agents then enter the E state for a fixed period of 60 timesteps. This belief update is detailed in Algorithm 4.

MFDM also modulates the time spent in the D period. In this strategy, g is assigned as the maximum of an agent's estimates for both possible beliefs, $g = \max\{\rho_1, \rho_0\}$ where ρ_1 and ρ_0 correspond to the estimates of the ratio of feature 1 and feature 0 respectively. This is labelled as the confidence of the agent. Agents with estimates close to 0 or 1 will have a high confidence that their belief is correct, while agents with estimates close to 0.5 will have a lower confidence and as such will spend less time disseminating. During the D state, agents share their estimates with neighbouring

Algorithm 4 DMMD Belief Update

Require: b_i Agent's current belief, B Beliefs of all agents encountered including b_i
$n_1 \leftarrow \text{SUM}(b_j)$ if $b_j = 1\ \forall b_j \in B$
$n_0 \leftarrow \text{SUM}(b_j)$ if $b_j = 0\ \forall b_j \in B$
if $n_0 > n_1$ **then**
    $b_i \leftarrow 0$
else
    $b_i \leftarrow 1$
end if

agents, and at the end of this state, agents take the most frequent estimate that they have received in that period as their belief, similar to DMMD, detailed in Algorithm 4. Agents also maintain a concentration value that is updated when an agent receives a belief from another agent that it has not been in contact with for 180 timesteps. This concentration term is defined as $C^* = 0.9C + 0.1b$ where C is the current concentration and b is the received belief. This acts as a long term memory of the agent's perceived belief of the swarm over time and makes this approach more robust to changes in the environment. When the concentration remains at 0.1 or 0.9 for 30 timesteps, in our simulation, the agent locks in its decision which is equal to the agent's current belief.

3.4.3 Communication Mapping Algorithm

We adopt a generic framework for considering the quality of service (QoS) of the communication links between agents. Since QoS metrics are application-dependent and may be affected by environmental factors or the type of data being shared, we consider an approach that is independent of any specific application requirements. Our approach also allows for the model to take into account changes that may be observed, differing from the theoretical values.

In our problem, agents have no *a priori* information relating to the location or severity of the communication limitations in the environment and they need to build up a decentralised representation of the communication environment. To do this, we implement a Gaussian Process (GP) model [104].

A GP is specified by its mean function, $\mu(x)$, and covariance function, $k(x, x')$ of a real process, $f(x)$, given by

$$\mu(x) = \mathbb{E}[f(x)] \tag{3.6}$$

$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))] \tag{3.7}$$

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')). \tag{3.8}$$

A GP aims to estimate the function, $f(x)$. Formally, let $P_i = [p_1^i, ..., p_N^i]$ be the vector of N_i points in the environment with corresponding communication quality observations, $Q_i = [q^i_{c1}, ..., q^i_{cN}]$, made by agent i. Then given that $q = f_i(p)$, for a set of test locations, $P_i^* \subseteq P$, we can model the joint distribution of the training outputs, f_i and the test outputs, f_i^* as the following which assumes a mean of 0 and noise-free observations:

$$\begin{bmatrix} \boldsymbol{f_i} \\ \boldsymbol{f_i^*} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(P_i, P_i) & K(P_i, P_i^*) \\ K(P_i^*, P_i) & K(P_i^*, P_i^*) \end{bmatrix}\right) \tag{3.9}$$

The covariance function specifies the joint variability between pairs of cells in the environment p and p', and for our approach we use the Matérn kernel as the covariance function, defined as the following where ν is a smoothness parameter and ℓ is the length scale:

$$k(p, p') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{||p - p'||}{\ell} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{||p - p'||}{\ell} \right) \tag{3.10}$$

Each agent, i, maintains a GP model which is learned using estimations of the communication link quality, which are gathered from the environment, $\{P_i, Q_i, T_i\}$ where T_i is the time when the estimation was made. As has been highlighted in the literature, GPs present a powerful method for estimating spatial distributions with uncertainty [102, 101, 100]. However, they come with significant costs in constructing the GP from observations. Exact inference on a data set of size n has complexity $O(n^3)$ and storage demands of $O(n^2)$. With alternative low rank approximation methods a complexity of $O(nm^2)$ and storage demands of $O(nm)$ are possible, where m is user selected parameter defining the number of inducing variables [105]. Considering a GP that stores communication data for each point in a discretized environment, as is done in our approach, would result in an implementation that would scale poorly for larger environments. To mitigate the computational costs of constructing a GP on what may be a low-complexity device, we use a sampling method to reduce the number of data points stored. In our implementation, 100 estimations are kept. This results in data that represents 2.44% of the environment in our experiments.

The sampling method is implemented as follows. When an estimate of the communication link quality is made by agent i, Algorithm 5 is performed to determine whether or not the estimate should be stored in P_s, to be used for the generation of the communication quality map. Agent i calculates the minimum distances from all stored estimations, $p_s \in P_s$ to any other one of its stored estimations, $p_{s'} \in P_s$ s.t. $p_{s'} \neq p_s$. These distances are stored in D^i_{min}. Agent i then calculates the minimum distance between the current estimation, p_e and each of its stored estimations, $p_s \in P_s$. This process is shown in lines 2-6 of Algorithm 5. If the minimum distance for the current estimation, p_e is greater than the smallest minimum distances in D^i_{min}, then one of the observations with this minimum distance is randomly selected to be replaced by p_s. This can be seen in lines 7-10 of Algorithm 5.

Algorithm 5 Observation sampling

Require: P_e Communication estimate, P_s Stored estimations, D^i_{min} Minimum distances for agent i
1: $min\_distance \leftarrow \infty$
2: **for** $p_s \in P_s$ **do**
3:     **if** $||p_e - p_s|| < min\_distance$ s.t. $p_s \neq p_e$ **then**
4:         $min\_distance \leftarrow ||p_s - p_e||$
5:     **end if**
6: **end for**
7: **if** $min\_distance > \min(D^i_{min})$ **then**
8:     $P_s[\text{index\_of}(\min(D^i_{min}))] \leftarrow p_e$
9:     Update D^i_{min}
10: **end if**

Agents also share communication link quality estimations in their heartbeat messages, alongside their feature beliefs. This was another important consideration in the implementation. While members of the swarm could exchange the communication quality maps that are generated by the GP, the communication costs of such an approach would place great strain on the communication capabilities of the swarm. A large volume of data would be need to be transmitted by each agent whenever they broadcast their heartbeat messages. This approach would also exhibit poor scalability for larger environments and swarm sizes, due to the greater number of data points contained within the maps. Instead, in our implementation, a constant number of communication quality estimations are sampled uniformly from an agent's set of stored observations. In our experiments, this value was set to 10. These are then broadcast to neighbouring agents that are in communication range.

Following this algorithm, agents aim to maximise the coverage of the estimations that are stored in their GP model and as such build up a representation of the communication environment, given a small number of data points. If the agent's stored estimations set is not at capacity (100 estimates), then the agent adds its most recent estimation of the communication environment. This allows for the wide area map to be initiated at the beginning of the task.

3.4.4 Path Planning Algorithm

Using the GP model, each agent, i is able to assign an estimated communication quality, $\hat{q}^i_{x,y}$, and uncertainty over the modelled communication quality environment, $\hat{\sigma}^i_{x,y}$ for each point, (x, y) in the environment.

During the D state, agents consider a square portion of their communication quality map, A, with the agent positioned at the centre. A has sides of length l, in our simulation this is set to $l = 10$. Agents then split this region into n equally sized areas, $a_n^i \in A$, each with lengths of size $\frac{l}{\sqrt{n}}$, in our simulation $n = 9$. The potential rewards for moving to each area in the D state, $R_D(a_n^i)$, is given by the sum of the estimated communication quality at each point in that area.

$$R_D(a_n^i) = \sum_{x,y \in a_n^i} \hat{q}_{x,y}^i \tag{3.11}$$

A similar approach is used in the E state, however instead of using a portion of the estimated communication quality map, agents use a portion of the uncertainty map. Hence, the potential rewards for moving to each area in the E state, $R_O(a_n^i)$ is given by the following equation:

$$R_O(a_n^i) = \sum_{x,y \in a_n^i} 1 - \hat{\sigma}_{x,y}^i \tag{3.12}$$

Once an agent has determined which area holds the highest potential rewards, it moves in a straight line towards the centroid of that area. When the agent reaches the centroid the path planning algorithm is performed again and the agent moves from area to area based on the expected rewards. This is the planning algorithm without any coordination between agents, abbreviated to CA-G in our study.

3.4.5 Agent Coordination

We now introduce our swarm coordination algorithm which is implemented to reduce the possibility of agents moving to the same area, possible when agents only employ the CA-G algorithm. The intuition behind our coordination algorithm, CA-Co, is for agents to maximise their possibility of successfully communicating during the D state and to maximise coverage to reduce uncertainty over the environment in the E state. Consider a single agent, i, in the D state. When entering the planning state, agent i uses the path planning algorithm to determine the potential rewards, $R_D(a_n^i)$ for moving to each of the n areas, $a_n^i \in A$ within their considered portion of the communication quality map, A. Instead of moving to the centroid of the best area, as in the path planning algorithm, agent i assigns a score to each of the areas, s_n^i, which reflects the contribution of each area, to the total reward of all considered areas, for agent i. This is given by $s_n^i = \frac{R_D(a_n^i)}{\sum_{a_m^i \in A} R_D(a_m^i)}$. Agents then choose a random area to move to using the calculated scores as weights. As such, areas with higher scores, and thus expected rewards, are more likely to be visited.

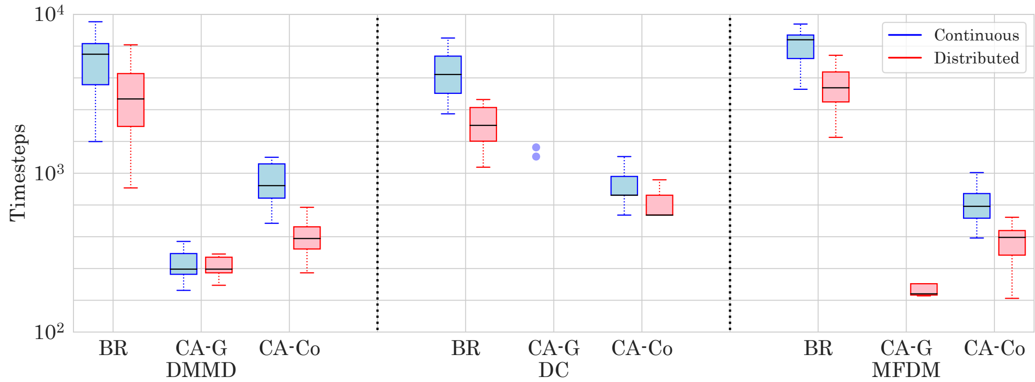


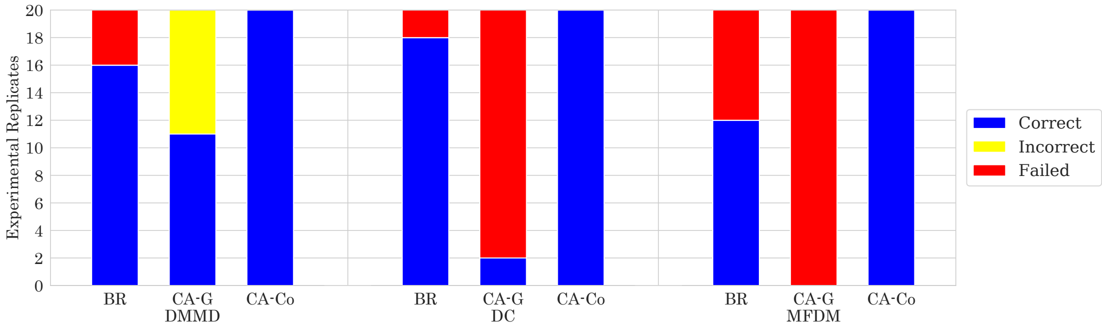
FIGURE 3.6: Time to convergence for each decision making strategy (DMMD, DC and MFDM), with $r_c = 0.5$ and $r_f = 0.65$ in continuous communication denied region (blue) and distributed communication denied region (red).

3.5 Results

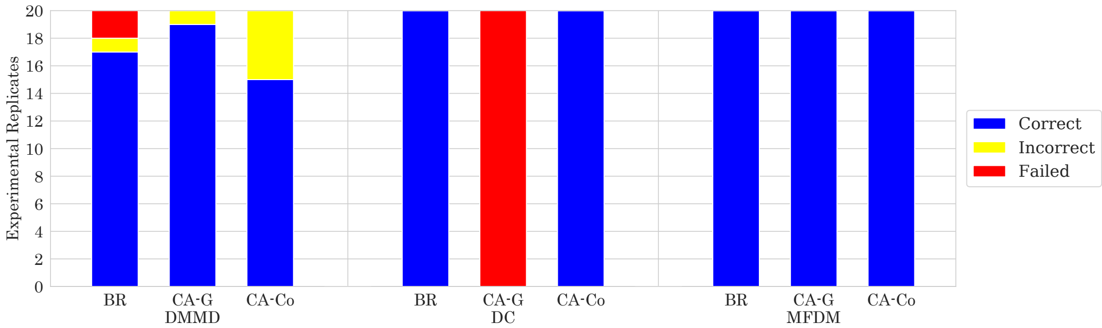
We compare our proposed communication aware algorithms, CA-Co and CA-G, with the DC, DMMD and MFDM decision making strategies. Results are compared with Random Baseline (RB), the original implementations of these decision making strategies, wherein agents move completely at random to explore and disseminate their beliefs. All experiment were replicated 20 times.

Figure 3.6 shows the time taken by the swarm to achieve consensus on the dominant feature in the environment, when agents of the swarm were denied communication in half the environment, $r_c = 0.5$. Any replicates where the swarm failed to reach a consensus within 9000 steps, or reached an incorrect consensus are disregarded in the figures. The feature fill ratio for this experiment is $r_f = 0.65$. The results show that the CA approaches, regardless of the coordination (CA-G and CA-Co), greatly improve the speed at which the swarm converges to a consensus, i.e. every member of the swarm holds the same belief at a given timestep, with our proposed CA approaches achieving consensus over 300% faster across all decision-making strategies.

The results suggest that the algorithm without coordination, CA-G has the best performance, particularly under the DMMD strategy, however, for the continuous communication-denied environment, the CA-G approach reached the incorrect consensus in 9 of the 20 replicates, compared to the CA-Co approach which was incorrect in 5 of the 20 replicates, as shown in figure 3.7(B). All other runs for these approaches with the DMMD strategy were successful. The BR approach failed to reach a consensus in 5 of the replicates using the DMMD mechanism. Another interesting observation here is that the CA-G approach failed to reach any consensus at all in all but 2 of the replicates that used the DC mechanism over both communication-denied environments. This was caused by the agents forming clusters of agents throughout the task that would very rarely mix. If all of the agents in one cluster disseminated



(A)



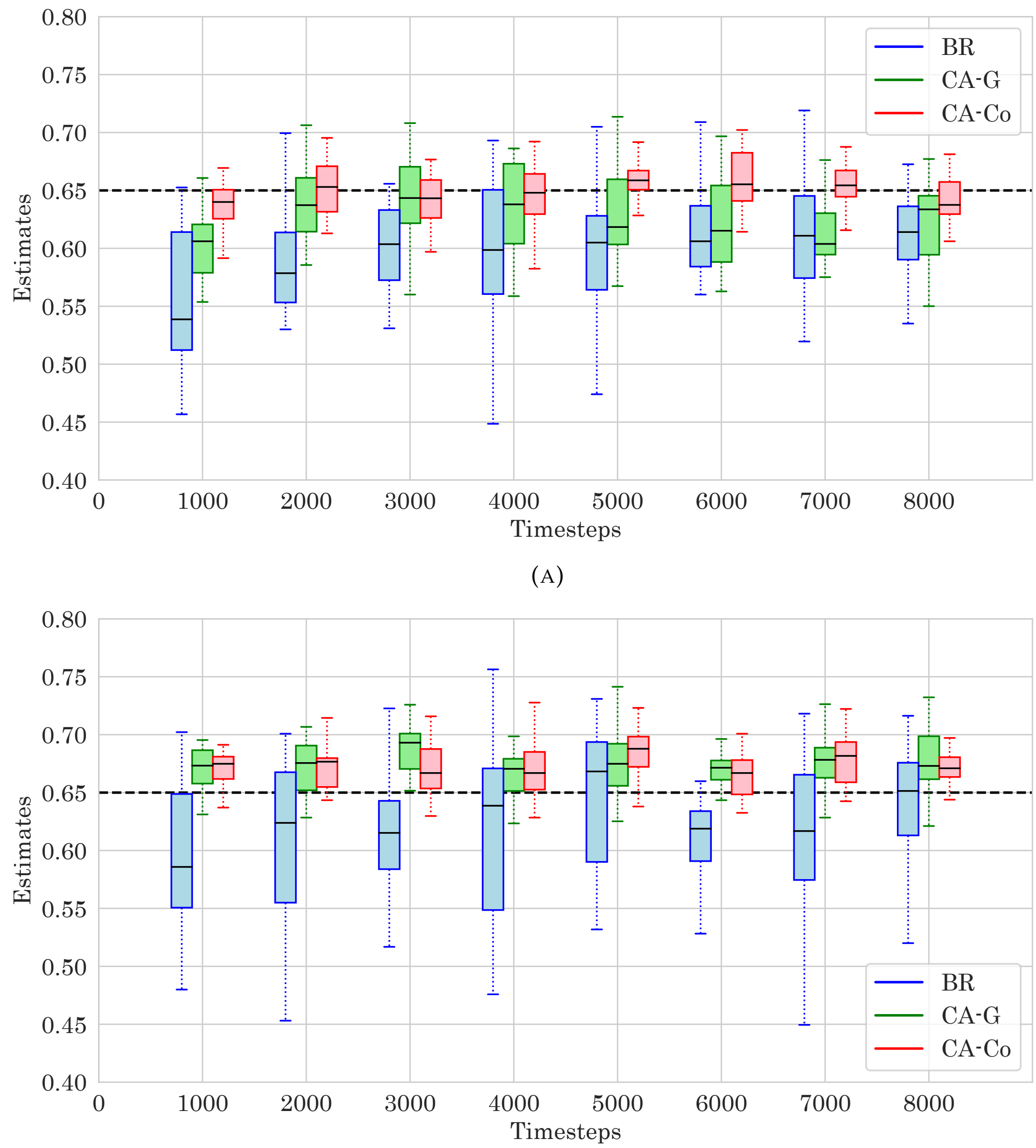
(B)

FIGURE 3.7: Number of experimental replicates that reached the correct consensus (shown in blue), incorrect consensus (shown in red) and that reached no consensus at all (shown in yellow) in an environment where $r_c = 0.5$ and $r_f = 0.65$ for (A) continuous communication denied environment and (B) distributed communication denied environment.

incorrect beliefs, the swarm would not reach a consensus. This clustering issue was mitigated by the CA-Co approach which successfully reached the correct consensus on all 20 replicates in both communication-denied environments for the DC mechanism.

While we have demonstrated that the speed of the decision is greatly increased by the communication-aware approaches, another important factor is the accuracy of the decision that has been made. Figure 3.8 shows the estimates of the ratio of the dominant feature for the DC strategy with $r_c = 0.5$ and $r_f = 0.65$. It can be seen that the estimates for BR, CA-G and CA-Co are close to the ground truth, confirming that the swarm is on average estimating the ratio of the dominant feature in the environment, with a median error of no more than 13%. Interestingly, while the CA-Co approach does not improve the time taken for the swarm to reach a consensus over the CA-G approach (Figure 3.6), it does improve the median estimate in the distributed communication denied environment (Figure 3.8(B).

We also tested the time taken for the swarm to reach a consensus for a feature ratio of $r_f = 0.65$ and communication environment with $r_c = 0$, meaning there are no communication denied areas in the environment and at any point in the environment, an agent is certain that any messages sent will be received by any agents in

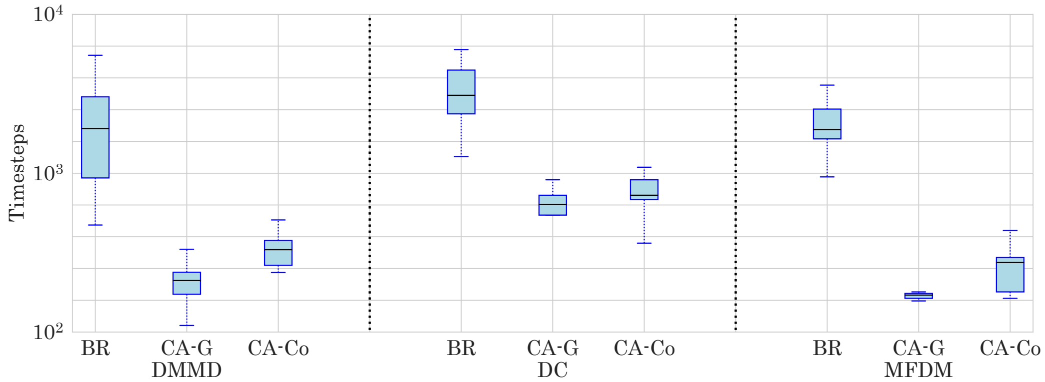


(A)

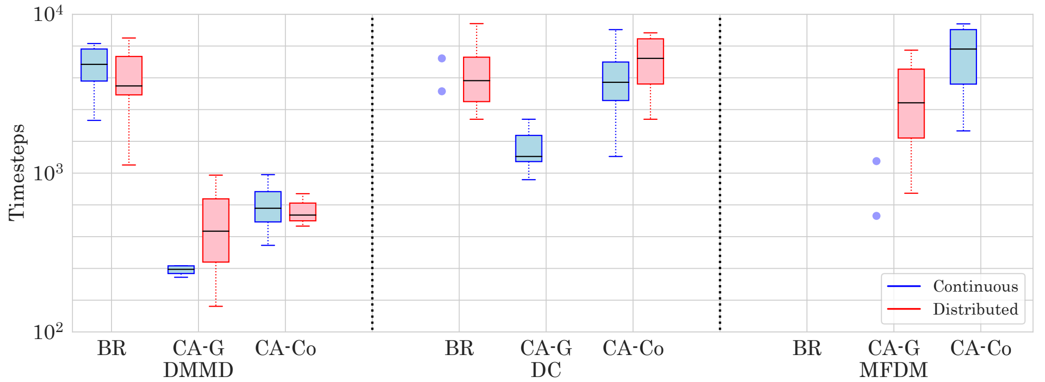
(B)

FIGURE 3.8: Estimates for DC decision-making strategy in a (A) continuous environment and (B) distributed environment, with $r_c = 0.5$ and $r_f = 0.65$. The black dotted line indicates the ground truth.

communication range. This mimics the task proposed by Ebert et al. [48]. Figure 3.9(A)) gives the results for this experiment. They show that the fully coordinated approach continues to outperform the RB strategy. Not only is the median time to convergence considerably faster, the spread of the time taken is greatly reduced, making the CA-Co approach far more predictable and reliable for a collective decision-making scenario. This is due to the agents exhibiting some level of aggregation when in the dissemination state with agents moving towards the same regions as some of their neighbours. This allows for a faster mixing of agents' beliefs and thus reaching a consensus more quickly. In the BR approach, this aggregation is not present and as such, the beliefs are spread around the swarm more slowly.



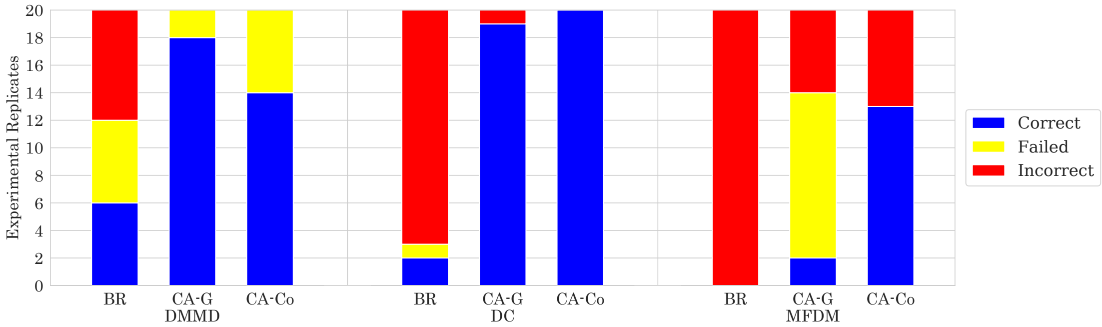
(A)



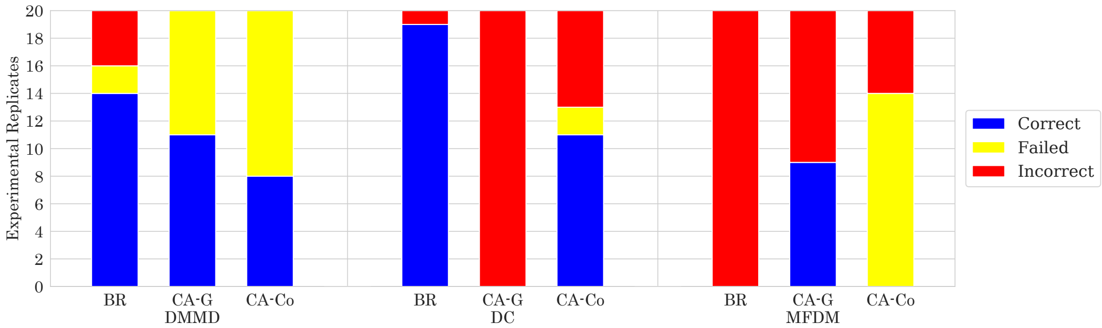
(B)

FIGURE 3.9: Time to convergence for each decision-making strategy, (DC, DMMD and MFDM) with (A) no communication denied regions present and (B) $r_c = 0.5$ and $r_f = 0.53$ in continuous communication denied region (blue) and distributed communication denied region (red).

Finally, we increased the difficulty of the task by using environments with a ratio of the dominant feature, r_f, closer to 0.5, in this case 0.53. This presents a more challenging task since it is harder for individual members of the swarm to individually distinguish which feature is the most dominant at any point. Figure 3.9(B)) shows that the CA-Co approach improves the median time taken to converge to a decision over the BR approach. The CA-Co and CA-G approaches also improve the reliability for the swarm to converge towards a consensus in under continuous communication constraints while using the DMMD and DC strategies. When using the MFDM strategy for decision-making, none of the BR replicates reached any consensus at all. In contrast, the CA-Co approach reached the correct consensus in 13 of the 20 replicates for the continuous communication denied environment, failing to reach a consensus in the other 7 replicates. The CA-G approach also reached the correct decision in 9 of the 20 replicates for the distributed communication environment, failing to reach any consensus at all in the other 11 replicates.



(A)



(B)

FIGURE 3.10: Number of experimental replicates that reached the correct consensus (shown in blue), incorrect consensus (shown in red) and that reached no consensus at all (shown in yellow) in an environment where $r_c = 0.5$ and $r_f = 0.53$ for (A) distributed communication denied environment and (B) continuous communication denied environment.

3.6 Conclusion

In this chapter, we presented a communication-aware algorithm for a swarm to perform collective decision-making in harsh communication-limited environments. Our approach has demonstrated to improve the ability of a swarm to reach a consensus on the feature of interest in the environment, not only in communication-limited environments but also in environments where the communication conditions are homogeneous across the environment. This improved on random movement which is a common method for determining how agents may move in collective decision-making problems. Not only does our communication-aware approach reduce the time it takes the swarm to reach a consensus, but this is achieved while retaining the accuracy of the decision. We have not performed an extensive search for parameter selection and this is left for future work. We further plan to investigate the effect of dynamic communication-denied areas that change in number and size during the experiments.

One challenge that was encountered during this work, was the problem of improving the efficiency and performance of the coordination algorithm. The decentralised method quickly plans a few timesteps ahead at any point. This results in little waste in computation when plans are required to change. A drawback of this approach is that agents plan with incomplete information. The aggregation of information with a more centralised controller would likely improve the performance and capability of the coordination. As has been discussed, a drawback of such a system is its vulnerability to single points of failure.

In the next chapter, we therefore consider a hybrid swarm system which is a relatively new line of research for robot swarms. We examine whether it is possible that such a system can maintain communications while performing at a high level in a given task.

Chapter 4

Hybrid Swarm Control

4.1 Introduction

In Chapter 2, we examined how a swarm's decision-making capabilities were affected when operating in a communication-constrained environment and proposed an approach to help restore a swarm's ability to share information in such a situation. It was found that making the swarm communication aware could overcome the challenges that are imposed by communication constraints in the environment and could even improve the decision-making performance of the swarm in those conditions.

The swarm was equipped with a communication environment mapping capability. This guided agents to areas of high communication potential when they were attempting to disseminate information. When collecting information about the environment, this capability pushed agents to areas of uncertainty. The swarm was also able to make use of a simple coordination algorithm which also aided in improving the performance of the swarm in a collective decision-making task and allowed the swarm to reach a consensus more quickly. This coordination algorithm was run in a decentralised manner, meaning that each agent performed planning locally, while assuming neighbouring agents were sharing the same beliefs of the state of the environment.

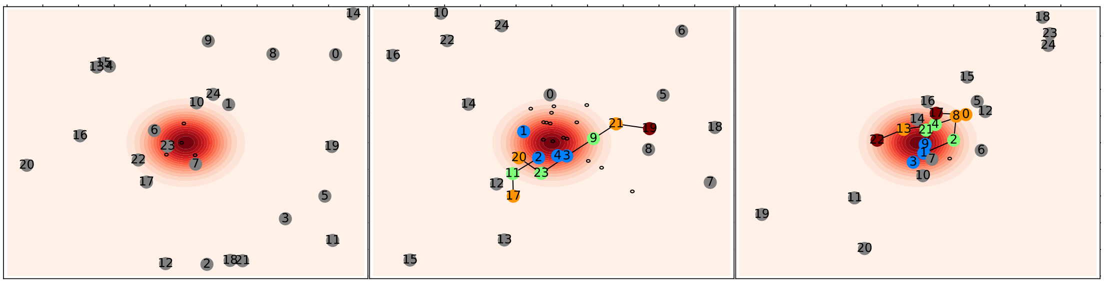
As has been highlighted in the literature, such a decentralised approach has advantages in the form of resilience to single points of failure, meaning that the coordination of the swarm can continue even when individual agents fail. The drawback of such systems is in their relatively slow speed and low long-term accuracy when faced with tasks that require exploration and coverage of an environment [6]. Hybrid approaches have been proposed as a way to provide a control structure that finds the middle ground between a fully centralised and a fully decentralised approach. Recent studies have shown that the introduction of a centralised control unit to facilitate coordination and distribute information around the swarm improves the performance of the swarm over decentralised, hybrid and self-organised control structures [6, 106]. As such, the coordination of the swarm that was seen in Chapter 2, could be improved further by considering a centralised approach at the expense of developing a system that is more vulnerable to failure.

Hybrid approaches have become a promising area of research for swarm systems with the introduction of the MNS with studies that have applied it to robot swarms to enable self-configurable formation control [34, 6]. While the relative performance of hybrid systems has been examined in the context of robot swarms, the relationship and trade-offs that a hybrid system may have with communication, both within the swarm and to a human operator, have not been explored. Further to this, an important aspect of hybrid systems is the swarm's ability to transition fluidly between a centralised and decentralised system [32], which, to our knowledge, has not been studied. A hybrid system that is equipped with this capability may be able to find the optimal level of centralised and decentralised control that is required for a given task. It may also be able to adapt to changing mission requirements relating to communications and performance.

In this chapter, we examine the assumption that a centralised approach always achieves the best performance in a task over a decentralised or hybrid solution. We also consider how a flexible hybrid control system may affect inter-swarm communication and human-swarm communication. To do this, we propose a simple generic hybrid approach which combines elements of decentralised and centralised control systems. Due to its simple nature, it may be used for similar studies that wish to build on top of our work to further examine the costs and benefits of hybrid systems. We show that a hybrid system may achieve higher performance in some tasks (e.g., in our environmental monitoring task) compared to a highly centralised and decentralised approach. Hybrid approaches come with trade-offs in communications, both between a human operator and within the swarm. We show that these trade-offs can be balanced with the performance of the system according to end-user requirements while using our proposed hybrid approach.

4.2 Hierarchical coordination

A simple environmental monitoring task is used to assess each approach. Here, agents are tasked with observing and completing tasks in the environment. To complete an event, agents must observe the events for a set period, in our experiments this is 200s. In our proposed hierarchical coordination algorithm, agents of the swarm dynamically form a forest of trees. The tree structure provides the end-user with a centralised



(A) Decentralised

(B) Highly centralised (Hybrid, $\delta = 1$)

(C) Hybrid (Hybrid, $\delta = 3$)

FIGURE 4.1: Examples the decentralised approach and early tree formations of the highly centralised (Hybrid, $\delta = 1$) and hybrid (Hybrid, $\delta = 3$) approaches during the task. The different coloured agents represent members of a tree at different depths, with root nodes being coloured blue. The small black circles represent existing incomplete events and the underlying event density distribution is shown.

control architecture for the swarm. Moreover, the trees are continually reshaped in response to changes in the operating environment and to configure the agents to observe multiple events at once. Both the task and the coordination we propose are simple in order to ensure that our approach is easy to follow and reproducible. We use the idea of tree formations as a basis for a highly centralised hybrid approach.

For our algorithm, we define a set of n agents in the environment as R, with individual agents given as $r_i \in R$. Each agent, r_i, in a tree keeps track of its parent, its current level in the tree, t_i, and also a list of its followers, F_i, which is the set of descendants of agent r_i. An agent r_i, is said to be in a tree if it is a follower of some other node or a root, $\exists j \in \{1, ..., |R|\} \wedge j \neq i$ s.t. $r_i \in F_j \vee r_i \in L$ where L is the set of roots in the environment which is initially $L = \emptyset$. An agent becomes a root when that agent forms a new tree. When this happens, that agent is not part of any other tree and initially will have no followers. Agents share their lists of followers when they are within communication range of each other, and parents will aggregate these lists to create a list of all of their descendants, $F_j \rightarrow F_j \cup F_i$ where r_j is the parent of r_i.

When an agent, r_i, initially joins a tree, $F_i = \emptyset$. Each agent also maintains a set of scores, for each of its descendants, $S_i = \{s_j \forall r_j \in F_i\}$, which could relate to how well an agent is equipped to complete tasks that are occurring or present at its current location or to know how well the agent is positioned to maximise the swarm coverage.

During a mission, when an event occurs at a location (x, y), $e_n^{(x,y)}$ where $n \in \{1, ..., |E|\}$, the event will be added to the set of events currently in the environment that have not been completed by the swarm, E. Each agent stores a set of events that have been observed or communicated to them, $E_i \subseteq E$. This information is shared between agents when they are in communication range of each other. An agent also defines the set of all events that occur within the sensing range of a location, (x, y)

as $E_i^{(x,y)} \subseteq E_i$. Agents can observe multiple events simultaneously, provided that they are within the sensing range of both events.

When an event occurs in the environment and is observed by an agent in a tree, this information is passed up the tree to the root. This root aggregates all the information about events that are currently being observed by agents in the tree and communicates this as a message to an operator.

Our hierarchical coordination algorithm consists of four elements and aims to form a forest of trees of agents that can act as teams in areas of high activity and importance within the environment:

- Root Formation: This defines where a tree should be centred and aims to decide where a tree would be beneficial for the mission.
- Tree Growth: Adding resources to the tree in the form of other agents, to improve the coverage and detect events in that area.
- Active Recruitment: Reallocating the resources to the most needed areas and directing the growth of the tree to the most appropriate areas of the environment.
- Dissolution: The pruning of unnecessary branches of the tree, so the resources can be put back into the environment for better use.

Each of the four components of our coordination algorithm works as follows:

Root Formation: An agent, r_i forms a root node when the density of events currently occurring within the sensing range of the agent at its current location, (x, y), exceeds an event density threshold, $|E_i^{(x,y)}| \geq \rho$. When this condition is satisfied, the agent stops moving, and declares itself the root node for a new tree at its current location, such that $r_i \in L$. Initially, the set of followers for r_i is empty, $F_i = \emptyset$. When communication is lost with a parent node for 20s, which may be due to faults or dissolution of the parent node, which is described below, the immediate children of that former parent become root nodes, maintaining the sub-tree below that agent. If two different root nodes are within communication range of each other, which may be the case when contact is lost with an agent acting as a parent to the two newly formed root nodes, then the trees are merged, taking the agent with largest sub-tree as the root. In the event that the tree size is tied for the two root nodes, one of the agents is uniformly chosen at random as the root node. Figure 4.2 shows an example of this process where the root node is given by the agent with the star, which is positioned in a high event density area. This agent acts as a centralised leader to all agents that connect to the tree that it forms. This approach takes inspiration from previous works where some agents position themselves in the environment acting as beacons

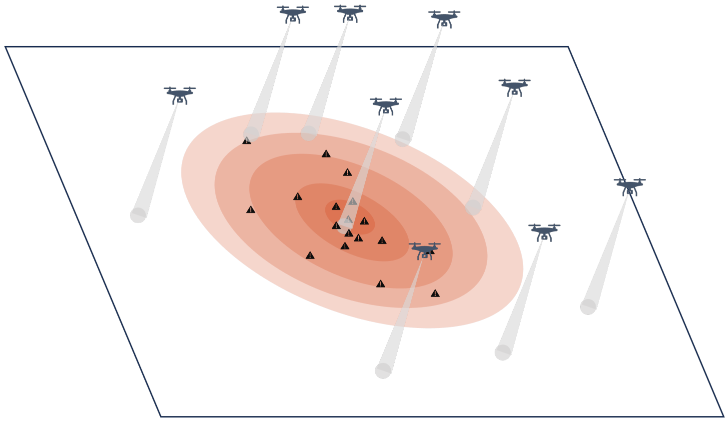


FIGURE 4.2: Example of the root formation process. Here an agent, that is located in an area that has a high density of events present, forms a root node. In this figure, the root node is denoted by the agent with the star symbol.

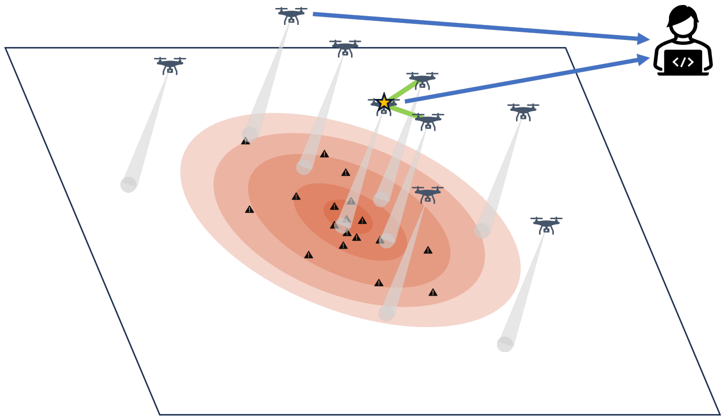


FIGURE 4.3: Example of the tree growth process. Here, two agents that are within communication range of the root node, denoted by the star, join the tree formed by the root node. Agents observing events, whether they are part of a tree or not, send information to a human operator on the events that are being observed.

[107, 108], which has been proven to be an effective and in the case of [107], a very scalable approach achieving a high level of coverage completeness in the task (99.8%).

Tree Growth: Free agents, r_i, are members of the swarm that are not currently part of any tree, such that $r_i \notin F_j \; \forall j \in \{1, ..., |R|\} \wedge j \neq i$. They move around the environment by performing a custom random walk where agents move in a straight line for a uniformly distributed amount of time before turning onto a new randomly chosen heading. This method of movement was chosen to give agents towards the periphery of the environment a better chance of moving towards the area of interest where events were occurring. When a free agent moves within communication range of another member of the swarm, r_j, that is in a tree, such that $|\overrightarrow{r_i r_j}| \geq d_{min}$ so the two agents are at least d_{min} meters apart, then r_i joins the tree, with r_j as its parent and $r_i \in F_j$. This process is shown in figure 4.3 where two nearby agents join the tree formed by the root node.

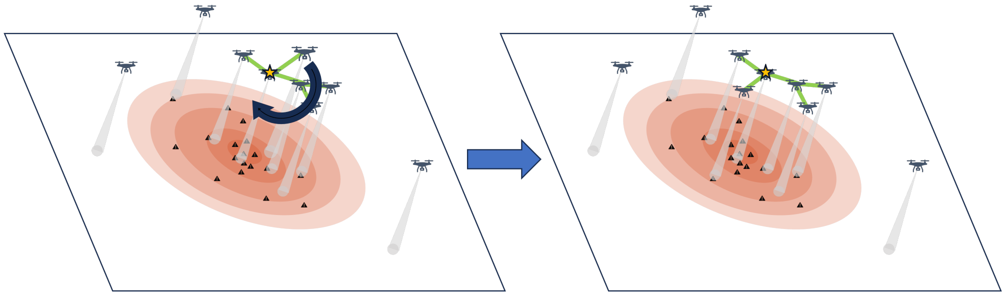


FIGURE 4.4: Example of the active recruitment process. Here, one agent, which is not observing any events, is moved to a new area with an event present.

Active Recruitment: Periodically, in our experiment once every minute, root nodes perform active recruitment to improve the performance of the tree. This is done to relocate any agents that are currently part of the tree at locations where events are not as frequent as in other areas of the environment. It also gives the tree a chance to expand into areas that the root feels are more important to detect a greater number of events than other areas of the environment. The root, r_j first determines the agent that will be moved to improve the number of events or expected number of events monitored by the tree. This is done by selecting a leaf node, r_i, of the tree which is contributing the least, in terms of events monitored, to the tree, such that $\min_{s_i \in S_j}\{s_i | F_i = \emptyset\}$. This is exemplified by figure 4.4, where an agent is moved from an area with no events to observe an area with an event present.

The root considers all possible locations, (x, y), that agent r_i can move to while remaining within the communication range of at least one of the other members of the tree, r_k, and also satisfying the condition for the tree growth, $|\overrightarrow{r_i r_k}| \geq d_{min}$. The root assigns a score to each location, $a_{(x,y)} = |E_j^{(x,y)}|$. The agent r_i then moves to (x, y), changing its parent as appropriate, once the new location is reached.

Since only leaf nodes are considered for active recruitment, there is never an issue of breaking apart an existing branch of the tree, disrupting the overall tree. The active recruitment aims to promote the growth of the tree into areas that have many events occurring.

Dissolution: Agents that are in a tree maintain a dissolution parameter, T, that is set to the maximum value, T_{max} when an event, $e_i^{(x,y)}$, is observed by the agent at their current location, which acts as a stimulus. This parameter is decreases linearlly at each timestep. The rate at which T decreases for a given agent is proportional to the number of parents above that agent in the tree, t_i, given by $T = T - t_i$. When $T = 0$ for an agent, r_i, the agent disconnects itself from the tree and becomes a free agent again. As such, agents that are closer to the periphery of the tree will dissolve quicker than those closer to the root. This allows the tree to remain more dynamic, reducing the potential for having agents remain in unnecessary locations where few events are

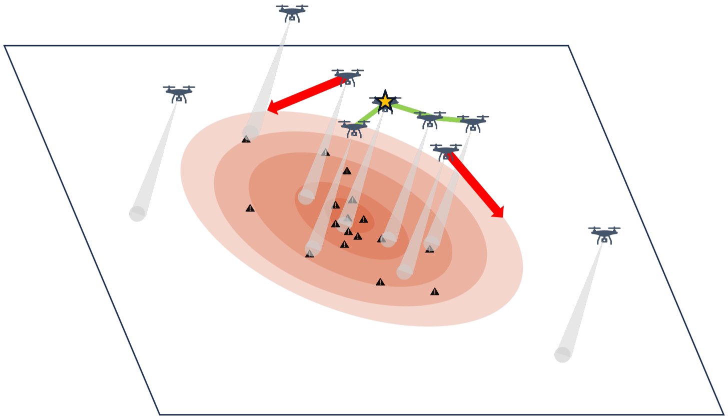


FIGURE 4.5: Example showing the dissolution process. Here, the two agents with red arrows are dissolving from the tree since they have not observed an event for some time proportional to the number of parents above them in the tree.

occurring, while allowing a tree with a well placed agents within the environment to persist. When an agent disconnects itself from the tree, that information is propagated to the rest of the tree by r_i's root so each agent in the tree can update their beliefs of the tree, $F_j \setminus r_i \forall j \in \{1, ..., |R|\}$. Figure 4.5 shows this process where two agents that are no longer observing events, and have not been for some time, leave the tree. They now act as decentralised agents that freely explore the environment.

This parameter is used not only to limit the size of trees but also to ensure that trees are forming in locations where events are occurring with a higher likelihood. If a tree does not detect any events for a long period, it is not necessary to maintain a large tree in that area. The algorithm would dissolve the tree and allow new trees to form in different areas.

Baseline coordination algorithm: We compare our hybrid coordination algorithm against a standard decentralised approach. In the decentralised approach, agents explore the environment via a random walk and exchange information regarding the location and status of events that have been observed, as in the hierarchical coordination algorithm. When an agent observes an event, the agent corrects its heading to move directly towards the event. Once the agent reaches the location of the event, it stops moving and observes the event until it is completed, when the agent continues moving randomly in the environment. The event is considered completed when it has been observed for 200 seconds, at which point that event is removed from the environment. If another event occurs within the agent's sensing range while it is attending to an event, then the agent will immediately move to the new event, once the original event that the agent was observing is complete.

If a randomly moving agent, which is not observing an event, moves within communication range of an agent that is observing an event then the randomly moving agent will, again, alter its heading to move towards the event and assist in the observation of the event. Agents also communicate information about events that they are observing to an operator.

Hybrid coordination: The hybrid approach makes use of the hierarchical coordination algorithm coupled with the decentralised behaviours. This is achieved by having agents switch between acting as a decentralised system or a centralised system, in response to the conditions in the environment. In our experiments, the level of centralisation or decentralisation is controlled by the parameter δ. This value corresponds to the number of events that an agent must be monitoring in order to initiate a switch from a decentralised to a centralised system. When acting as a decentralised system, agents make use of the baseline coordination algorithm described above. Here agents do not perform complex coordination and instead rely on local interactions to help direct agents towards events, thus making the system a decentralised control system. When agents are acting as a centralised system, they make use of the hierarchical coordination algorithm described in this section. The formation of centralised tree structures allows for greater control of a larger set of agents by the root node, that acts as the central controller. There are two main elements that define the hybrid system. The first is in the ability for sub-trees present within a formed tree structure to recover from disconnections by taking the decision to form a new tree at the disconnection. This allows the swarm to flexibly recover and continue in the operation of sections of a tree structure that would other wise be unable to function without a connection to the parent and root node. This behaviour combines elements of centralised and decentralised systems. The second element comes from an individual agent's ability to switch between operating as a centralised system or a decentralised system in the following way. A hybrid system with a low δ value, close to $\delta = 1$ has root nodes form frequently in the environment with a high chance of roaming, decentralised swarm members encountering and joining a tree. As such, these kinds of systems present a highly centralised system, where agents spend a large proportion of the time in the experiment acting as a centralised system within a tree structure. Conversely hybrid systems with high δ values, such as those with a value of $\delta = 8$, do not form root nodes as frequently in the environment, if at all. This reduces the time agents spend in a centralised system and instead they spend most of their time as a decentralised system. Hybrid systems with δ values in between these two values ($1 < \delta < 8$) varies the level of centralisation smoothly with the time spent by agents during the experiment proportional to the δ value.

When an agent, r_i observes an event, at (x, y), the robot will perform one of two actions. If the density of events currently occurring within the sensing range of the agent at its current location exceeds the event density threshold, $|E_i^{(x,y)}| \geq \delta$, then the agent will act as a centralised system and form a root node. If this condition is not met, $|E_i^{(x,y)}| < \delta$, then the agent will behave as a decentralised system and will not form a root node but will change its heading to move towards the event and it will proceed to observe the event.

4.3 Experimental Setup

Agents move at a maximum speed of 1 m/s in a bounded environment, $Q \subseteq \mathbb{R}$ which is discretized into a set of points, $(x, y) \in Q$. For our experiments, we fix the density of agents in the environment to 0.0025 agents per m^2. So for the experiments with 25 agents, the environment is 100m $\times$ 100m with 20 replicates of 9000 seconds. We assume that there exists an unknown density function $\phi(x, y) : Q \rightarrow \mathbb{R}^+$ which defines the probability that an event, $e_i^{(x,y)}$ occurs at location (x, y) at time, t. During the experiment, an event have a probability of occurring somewhere in the environment once every second with $pr = 0.1$. The location of the event is then determined by sampling $\phi(x, y) : Q$. For our experiment, in the base scenario, the underlying distribution takes the form of a Gaussian distribution, with mean at $(50, 50)$ and covariance of $\left(\begin{smallmatrix} 50 & 0 \\ 0 & 50 \end{smallmatrix}\right)$. Events are persistent in the environment until they have been observed by agents a set number of times, when they are completed and removed from the environment. Each distinct agent that observes an event contributes towards the event's completion, reducing the number of times an event needs to be observed before it is completed by 1. As such, having multiple agents observe an event will result in the event being completed more quickly and this increase in speed is linear to the number of agents observing the event. Agents can also observe multiple events at the same time, provided they are within the sensing range of $5m$. For our experiments, we do not consider collision avoidance.

The δ value was varied during some of the experiments to examine the effects of different levels of centralisation on a number of metrics including the performance and the communications present in the swarm. Another important parameter is the tree dissolution parameter, T_{max} which dictates how long an agent that is in a tree, should remain part of that tree, while not observing any events. For our experiments, this was set to $T_{max} = 200s$. This parameter also plays a role in the level of centralisation present in the swarm. A shorter dissolution time will cause agents to leave their respective tree more frequently, resulting in a lower level of centralisation present within the swarm. A greater value will result in agents remaining in their tree for longer, thus increasing the overall level of centralisation of the swarm. For the work presented here, the main focus was on examining how the communication requirements and the performance of the swarm vary with a flexible hybrid control system. The key aspect of our proposed hybrid system that distinguishes it from a purely decentralised system is in the tree growth and active recruitment. If the value of T_{max} is too small, neither the active recruitment or the tree growth have time to make a difference on the operation of the system. As such, we decided to fix this parameter for these experiments and only change the level of decentralisation by varying the δ parameter.

Agents use a random walk routine to move within the environment, similar to one implemented in chapter 3. Using this kind of random walk, with a low probability of performing a long walk, gave agents a chance to "escape" the influence of the tree structures when they dissolved from their tree structures. It was found that when shorter walks were used for agents' movements, the agents would aggregate close to tree structures and would have little to no opportunity of exploring the rest of the environment when they dissolved from the tree structure. The routine is as follows. An agent moves in a straight line for a random distance, d, with a mean distance of 5m that is sampled from an exponential distribution, given by:

$$d = \lambda e^{-\lambda x} \tag{4.1}$$

Where λ is the rate parameter, and for our experiments, $\lambda = 0.2$. After travelling this distance, agents perform a turn. This turn has equal probability of being performed in a clockwise or counter-clockwise direction. The agents turn for a period of time that is uniformly distributed between 0 and 2 seconds. Then repeats these steps for the duration of the experiment. For these experiments, we did implement a collision avoidance algorithm. This decision was made in light of the fact that agents could easily become trapped within sections of the tree structures formed by the swarm, leaving them unable to move if considering collisions. As such we do not consider collisions within this work and assume that agents are able to alter their height in order to avoid neighbouring agents. This limits the findings of this chapter to swarms that are composed mainly of aerial robots. A more substantial path planning or relocation algorithm would be required to allow agents to successfully navigate through some parts of the tree structures that are formed.

The simulation environment used in Chapter 3 was adapted for this environment monitoring task. As such, agents share the same communication model, where agents can communicate with any other agents that are within communication range. In these experiments, the range is set to 5m. The environment monitoring task that we adopt is similar to the task studied in several robot coverage problems [109, 110, 111, 112, 113]. In these approaches, an underlying density function is present in the environment. The swarm is responsible for maximising the coverage of this environment to optimally spread over the region with a density that reflects the density function. The underlying density function defines the frequency of events that occur at any location in the environment. As a result, this function cannot be sensed directly by agents and instead relies on the observation of events to build a belief over the underlying density function. This scenario has a range of applications including natural disaster prediction and monitoring, search and rescue in disaster regions, enemy intrusion detection and target tracking [114, 115].

We evaluate the proposed swarm coordination approaches using the following metrics: i. Waiting time for events before being observed by a member of the swarm; this metric shows how responsive the swarm is to changes in the environment and how well the swarm can cover areas where events are occurring with a high probability; and ii. Number of messages received by a human operator at each timestep from distinct agents; this gives a measurement for the cognitive load that an operator may experience while controlling the swarm. Messages are received when an agent is observing an event if it is operating as a decentralised agent and from the root of a tree where multiple agents may be monitoring events within the tree.

4.4 Results

4.4.1 Performance and cognitive load in swarm coordination

Figure 4.6 shows the mean waiting times taken for events to be observed by members of a swarm of 25 agents coordinated with a highly centralised system, shown by Hybrid $\delta = 1$, the decentralised system and a Hybrid system between the two, given by Hybrid $\delta = 3$. Data is aggregated over 10 separate replicates. The highly centralised (Hybrid $\delta = 1$) approach has a median$\pm$IQR waiting time of 34.0$\pm$12.0 seconds. The approach improves on the performance in monitoring environmental events over the decentralised approach with a median$\pm$IQR waiting time of 42.1$\pm$4.7 seconds. This shows a 19.2% reduction in the waiting time. The hybrid approach (Hybrid $\delta = 3$) also outperforms both the Hybrid $\delta = 1$ and Decentralised approaches with median$\pm$IQR waiting time of 26.4$\pm$8.8 seconds, a 37% reduction in waiting time over the decentralised approach and a 22% reduction over the Hybrid $\delta = 1$ approach.

The hybrid approach can improve the performance of the swarm for this task, over that of the highly centralised and decentralised approaches due to its flexibility in switching between acting as a centralised and decentralised system. Furthermore, the value of δ dictates how much time an individual should spend as each of the approaches. As such, at the beginning of the experiment, agents with a higher δ value will remain flexible, being more likely to act as a decentralised system and not commit to forming trees until there is more certainty over the distribution of events. Alternatively, agents with lower δ values will form trees more quickly but may form them in sub-optimal locations which will take longer to adjust with the active recruitment. This can be seen in figure 4.1(B) and (C) where the highly centralised (Hybrid, $\delta = 1$) approach has initially formed trees away from the centre of the event density function, while the hybrid approach has trees that have formed closer to the

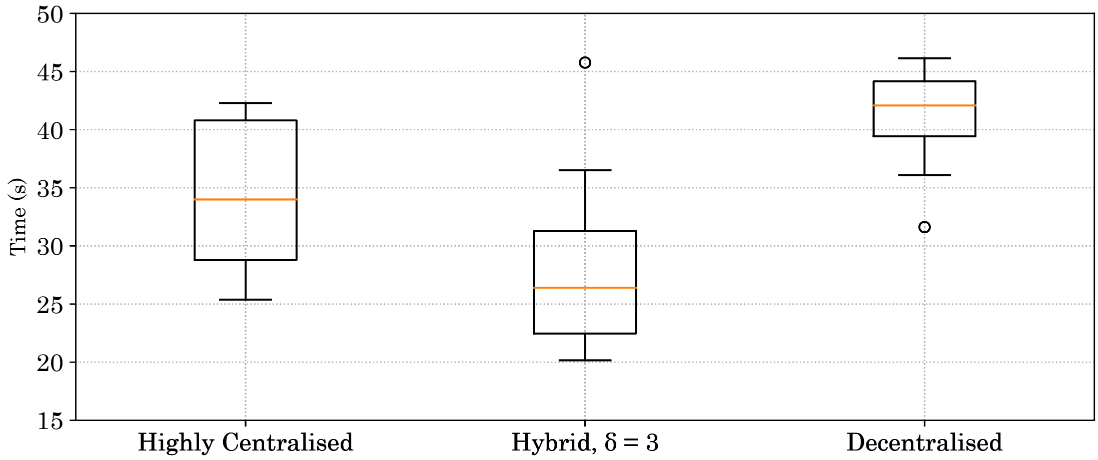


FIGURE 4.6: Box plot of waiting times for environmental events to be observed by a swarm of 25 agents coordinated with a Highly centralised (Hybrid, $\delta = 1$), a Decentralised and a Hybrid (Hybrid, $\delta = 3$) approach. Event waiting time data was averaged across each replicate, and aggregated across 20 replicates.

centre. However, agents that have δ values that are too high and act as a decentralised system for a large proportion of time will not gain the benefits of the coordination.

When we consider the cognitive load (see Figure 4.7), which shows the mean number of messages sent by different agents of the swarm to a human operator at each timestep, we see that the hybrid (Hybrid, $\delta = 3$) and highly centralised (Hybrid, $\delta = 1$) approaches, with median$\pm$IQR of 3.7$\pm$0.30 and 4.6$\pm$0.54, reduce the number of messages sent to the operator by over 50% compared to the decentralised approach, with median$\pm$IQR of 10.0$\pm$0.61. The lower load on the operator overseeing the highly centralised and hybrid swarms is consequent to the information sent from the relatively few dynamically formed leaders of the trees has been aggregated; the multiple environmental events spanning the area covered by a tree do not require the communication of multiple messages as in the decentralised approach.

It is likely that these reductions in the waiting time and the number of messages sent to an operator for the Hybrid approach over the highly centralised and decentralised approaches are a factor of the level of centralisation within the hybrid approach, controlled by the value δ (see Section 4.2, Hybrid coordination).

4.4.2 Scalability of coordination algorithms

We examine the scalability of the proposed Hybrid approach within swarms of 25, 100, and 500 agents. Figure 4.8 shows the mean waiting times for events in the environment by each of the approaches. As in the base case of 25 agents, we see that the hybrid approach provides reductions in the mean waiting time of events over the decentralised approach of 26% and 14% for swarms of 100 and 500 agents,

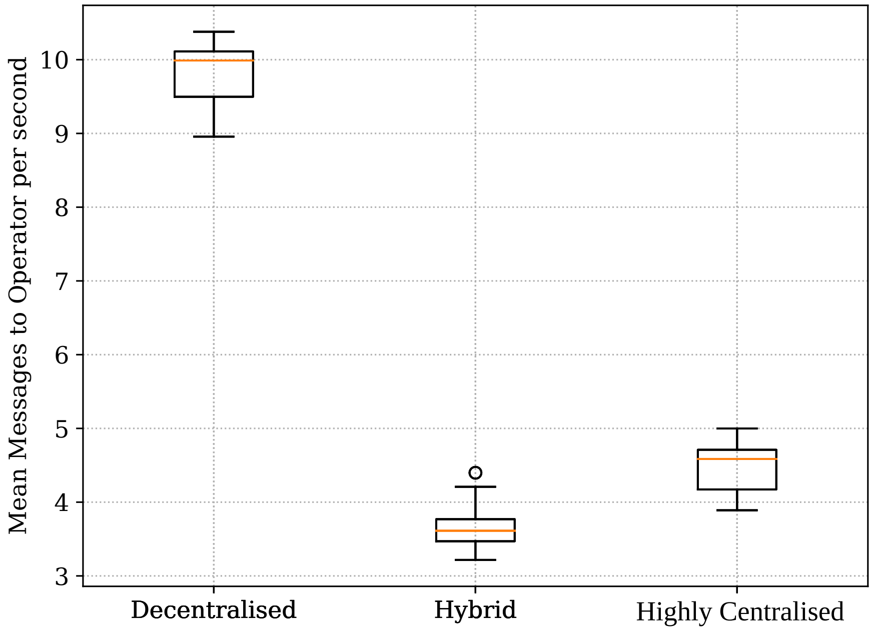


FIGURE 4.7: Box plot of mean number of messages sent by different agents in a swarm of size 25 to a human operator at each timestep with the highly centralised (Hybrid, $\delta = 1$), decentralised and hybrid (Hybrid, $\delta = 3$) approaches. Communication data was aggregated across 20 replicates.

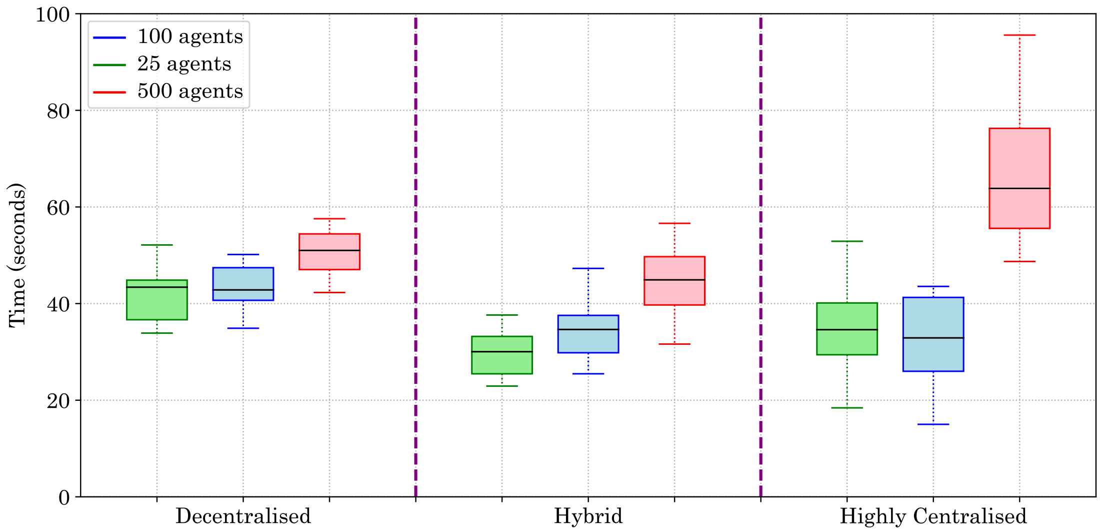


FIGURE 4.8: Box plot of mean waiting time for events to be observed by swarms of sizes 25, 100 and 500 agents coordinated with the highly centralised (Hybrid, $\delta = 1$), decentralised and hybrid (Hybrid, $\delta = 3$) approaches. Data was aggregated across 20 replicates.

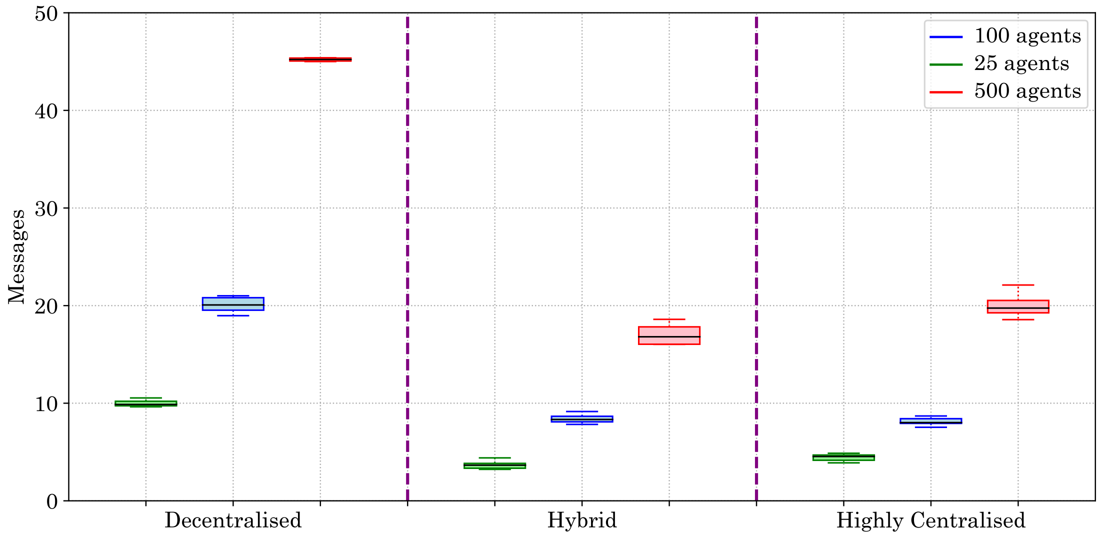


FIGURE 4.9: Box plot of mean number of messages sent by different agents of the swarm to a human operator at each timestep, with the highly centralised (Hybrid, $\delta = 1$, decentralised and hybrid (Hybrid, $\delta = 3$) approaches for swarms of 25, 100 and 500 agents. Communication data was aggregated across 20 replicates.

respectively. These reductions come despite a drop in the performance of the hybrid system as the number of agents in the swarm increases. For this experiment, we kept the density of agents within the environment constant, along with the area over which events were likely to occur as a fraction of the entire environment. This meant that as the number of agents increased, so did the size of the environment and the area where events occurred. For the 25 agents case, the size of the environment was set to 100m $\times$ 100m with a Gaussian distribution with the mean at $(50, 50)$ and covariance of $\left(\begin{smallmatrix} 50 & 0 \\ 0 & 50 \end{smallmatrix}\right)$ meaning events occurred with 99.7% probability within an area that was 14.1% of the total area of the environment. As such the density of agents present was 0.0025 agents$/m^2$. In the case of the 100 agents experiments, the environment was then scaled up to 200m $\times$ 200m, keeping the density of agents at 0.0025 agents$/m^2$. The underlying Gaussian distribution was also scaled to have the mean at $(100, 100)$ and covariance of $\left(\begin{smallmatrix} 199.5 & 0 \\ 0 & 199.5 \end{smallmatrix}\right)$, thus keeping the area where events occurred with 99.7% probability at 14.1% of the total area of the environment. This scaling was also done for the experiments with 500 agents. This was done to examine how the number of messages received by an operator would change, given a larger swarm with a greater number of events to monitor.

For the number of messages to an operator, the results from swarms of 25, 100 and 500 agents (see Figure 4.9), we see that the hybrid and highly centralised approaches reduce the number of messages sent to a human operator at each timestep throughout the mission. The hybrid approach reduces the mean number of messages sent to an operator by over 50% for all swarms of size 25, 100 and 500 agents. There is an increase in the number of messages sent to an operator in the decentralised approach of approximately 24 messages per second as the swarm size scales from 100 to 500

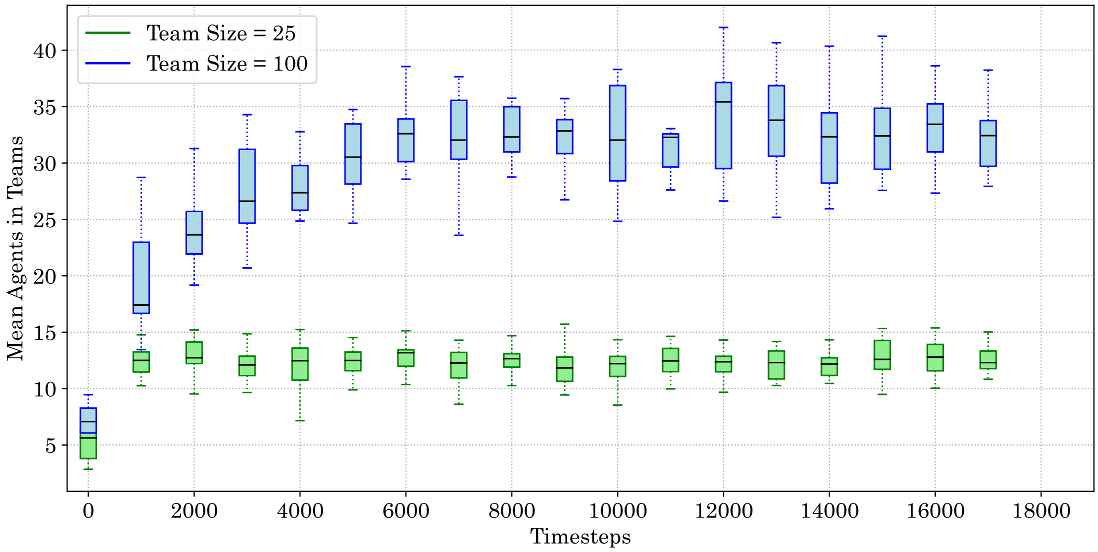


FIGURE 4.10: Mean number of agents that are part of trees by swarms of sizes 25 and 100 agents coordinated with the hybrid approach with $\delta = 3$. Data was aggregated across 20 replicates.

agents. This is over two times the increase in messages per second to the operator that is observed for both the hybrid system, with an increase of 10 messages per second, and the highly centralised system, with an increase of 11 messages per second. In the case of the hybrid approach, this is because the decentralised aspect of the approach draws more members of the swarm towards areas where events are occurring and incentivises them to remain close to that area for longer, as such the density of the swarm increases in these areas for the hybrid approach, with $\delta = 3$ more than it does for the highly centralised approach, with $\delta = 1$. As such, events are more likely to be initially registered by a member of the swarm more quickly.

We also consider the number of agents present in trees throughout the experiments. This is shown in Figure 4.10 for swarm sizes of 25 and 100 agents. In these experiments, the environment was scaled proportionally to the increase in the size of the swarm sizes to keep the density of agents present in the environment equal. The event density function and frequency of events were also scaled to account for the increase in the size of the environment and the swarm size. The swarm of size 25 quickly converges to having 12 agents present in teams at any time within the first 1000 timesteps, while the swarm size of 100 converges to 33 agents in teams at any point in time after 6000 timesteps. This difference in convergence time is likely due to the increase in the size of the environment and the event density function, while the sensing of the range of agents remains the same. As a result, agents spend more time behaving as a decentralised system initially. This is due to events having a greater chance of occurring over a larger area. This means that initially, the density of events in the environment do not exceed the swarm's δ threshold value. As such, trees do not begin to form for some time.

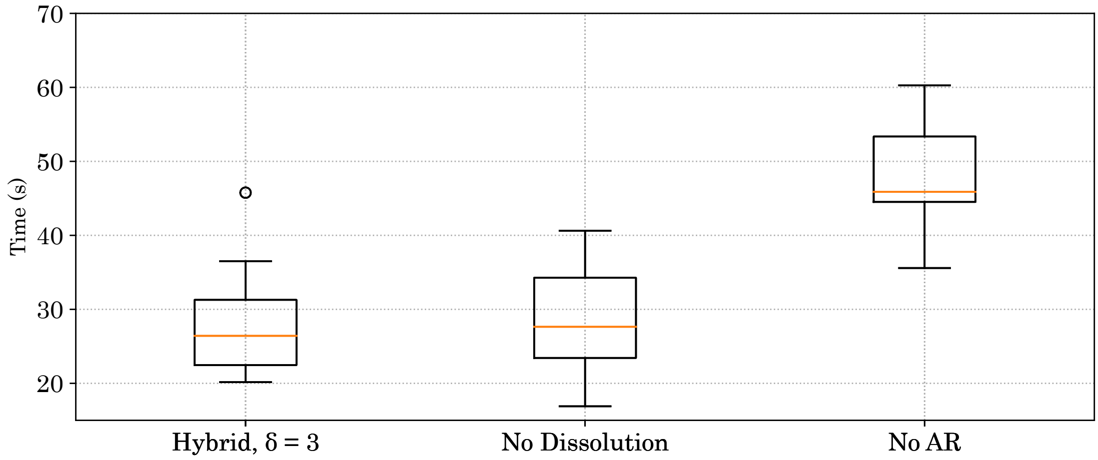


FIGURE 4.11: Ablation analysis of mean waiting times for events to be observed by a swarm of 25 hybrid coordinated agents following the removal of the active recruitment (AR), and the dissolution elements.

This shows the hybrid coordination algorithm remains stable and predictable for swarms with different sizes and for larger environments. This is validated further by considering the coverage of the two swarm sizes. The upper limit of the possible coverage by teams of size 12 and 33 agents, assuming a sensing range of 5m and no overlap, is 942m^2 and 2591m^2 respectively. The actual coverage of the swarms is less than these totals, with a mean coverage of 672m^2 and 1521m^2 for a swarm with a team of 12 agents and 33 agents respectively. These totals exceed the area of the environment where 95% of the events occur. This portion of the event density function for a swarm of size 25 and 100 is a circular area of 628m^2 and 1256m^2 centred at the midpoint of the environment. As such, both swarm sizes can converge to teams that can cover 95% of the events that occur in the environment.

4.4.3 Ablation study

Figure 4.11 compares the performance of the hybrid approach, with $\delta = 3$ against the approach when there is no active recruitment and when there is no dissolution of the trees. The median±IQR waiting time for the hybrid approach is 26.4±8.8 seconds and we see a 73% increase in the waiting time when the active recruitment is removed to 45.9±8.8 seconds. Even though active recruitment only occurs periodically, it has an important effect on performance. We also observe a 5% increase in the waiting time when the dissolution of the trees is removed with a median±IQR time of 27.6±10.8 seconds. This suggests that both the dissolution and the active recruitment help to keep the trees flexible, even when the other is not present, with the active recruitment affecting the performance more than the dissolution element which occurs more frequently.

It is clear that the coordination provided by the active recruitment for the highly centralised and hybrid approaches is an important factor in improving the performance over that of the decentralised approach. When there is no active recruitment present, the performance of the highly centralised approach, with a median±IQR waiting time of 45.9±8.8 seconds, is close to that of the decentralised approach, with 42.7±5.2 seconds.

4.4.4 δ Sensitivity analysis

Figure 4.12 shows the mean number of messages sent to an operator per second against the number of inter-swarm messages (between members) per second over the course of 20 runs for a swarm of 25 agents. Here, we vary the levels of centralisation, these being decentralised, highly centralised (with $\delta = 1$) and hybrid with δ values ranging from 2 (more centralised) to 10 (more decentralised). The points for each of the replicates are shown with ellipses that represent the mean and one standard deviation from the mean for each of the approaches. We can see that a Hybrid approach with a low δ value acts close to that of a more centralised system and as the delta increases that hybrid system behaves closer to that of a decentralised system.

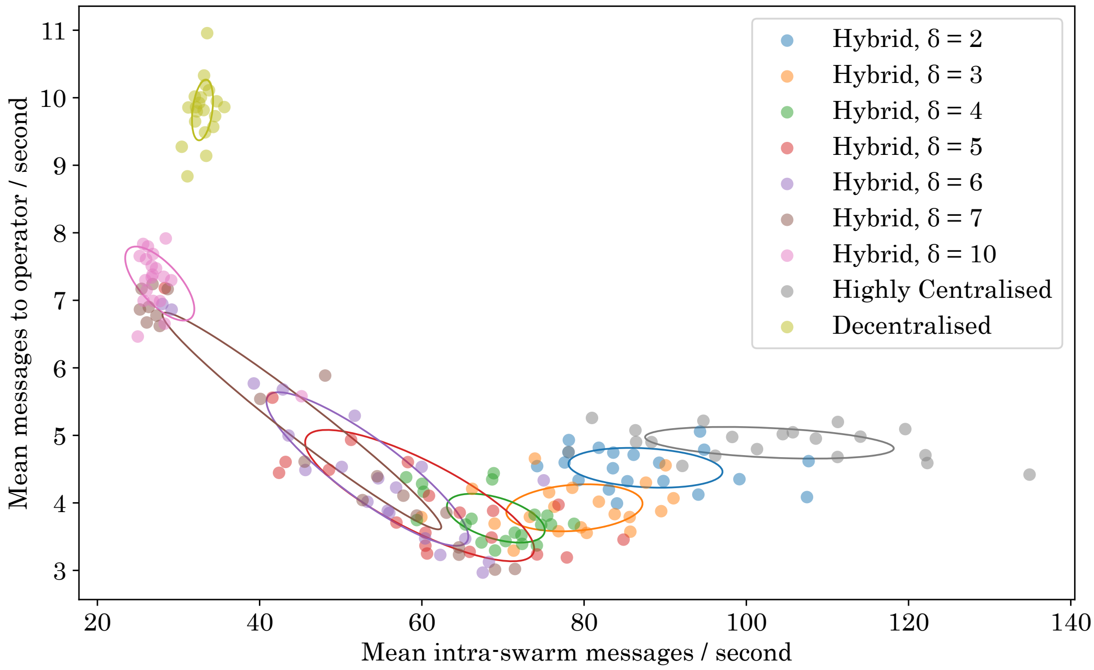


FIGURE 4.12: Mean number of messages sent by different agents in a swarm of size 25 to a human operator against the mean number of messages sent at each timestep within the swarm.

When we consider the cognitive load, we see that the hybrid $\delta = 3$ and highly centralised $\delta = 1$ approaches, with median±IQR of 3.9±0.50 and 4.5±0.42 messages to an operator per second, reduce the number of messages sent to the operator by over

50% compared to the decentralised approach, with median±IQR of 9.9±0.38 messages to an operator per second. The lower load on the operator overseeing the highly centralised (Hybrid, $\delta = 1$) and hybrid (Hybrid, $\delta = 3$) swarms is consequent to the information sent from the relatively few dynamically formed leaders of the trees has been aggregated; the multiple environmental events spanning the area covered by a tree do not require the communication of multiple messages as in the decentralised approach.

We see the best performance of the system for this scenario for $\delta = 3$ and $\delta = 4$. When a member of a Hybrid system senses events in the environment, it must check whether the density of events exceeds δ before deciding to behave as a decentralised system or centralised system by forming a root. Smaller values of delta increase the chance of an agent forming a root node since a lower density of events is required. This results in multiple trees being formed in sub-optimal locations within the environment where the density of events is lower than the maximum density present in the environment. This leads to more leaders being present in the environment and as a result more messages being sent to the operator. Larger δ values reduce the probability of a root forming since the required density of events is much greater. This means fewer trees form and more agents continue to operate as a decentralised system. If an appropriate delta value is chosen to reflect the maximum density of events present in the environment, then we see a minimal number of trees with roots focused near the points in the environment with the maximum density, allowing for fewer leaders and therefore fewer messages sent to the operator, along with a well placed tree to cover the area of interest.

This shows that more centralised control does not always provide the best performance for a task. It suggests that a system which is too sensitive to stimuli that cause it to centralise can get trapped in local minima. To overcome this issue, a more flexible solution is required that refrains from centralisation too quickly. This would allow time to assess the scenario and adapt accordingly. While this sacrifices performance in the short term, it guarantees a better performance over time.

4.5 Conclusions and future work

In this chapter, we proposed a control structure that gives flexibility over the level of centralisation or decentralisation for a robot swarm system. The performance of such a system and the effects the system could have on communication with a human operator in an environmental monitoring task have been evaluated. The results suggest that having a swarm system centralise too quickly can be detrimental to the swarm's performance and results in the system becoming stuck in local minima. Finding a balance between centralised and decentralised control can improve this, but comes at a cost of performance in the short term. Future work will consider how an aspect of learning could be added for the hybrid approach that will allow the swarm to adapt the delta value in response to the task and the environment. This would give the swarm control over its own level of centralisation as needed.

The highly centralised (Hybrid, $\delta = 1$) and the hybrid (Hybrid, $\delta = 3$) approaches have both been shown to outperform the decentralised approach in an environmental monitoring task. These approaches reduce the number of messages sent to a human operator, reducing the noise the operator receives. One drawback to these approaches is that they come with inter-swarm communication and message size overheads. However, we have shown that by varying δ for the hybrid approach, different levels of centralisation can be achieved to find an optimal level of inter-swarm and swarm-to-operator communication for a given mission. These communication overheads pose another interesting question for future work in how these approaches might be affected by communication limitations in the environment. Whether this would affect the overall performance of the highly centralised and hybrid approaches could also be explored. Another direction for future work is to allow the swarm to select its δ value during runtime to adapt to the changing environment.

One key limitation of this chapter is that only one kind of task was considered. Considering a scenario where the task clearly favours either a more centralised system or a more decentralised system could highlight new relationships between the performance and communication metrics of the swarm. As such, the results of this chapter are limited, in that the conclusions drawn may not generalise well to new problems. However, we have shown that in this task, that the best level of system control may be one that lies between a highly centralised system and a decentralised one. Another limitation of this work is in the lack of a truly centralised system to benchmark the hybrid systems against. While we consider a highly centralised hybrid system, the inclusion of a control structure that is fully centralised with a single leader for the entire swarm and global communications could shed light on how close the highly centralised system is to full centralisation. Doing so would allow the full spectrum of centralised / decentralised control to be compared for this task.

Chapter 5

Optimised Hybrid Swarm Control Structures

5.1 Introduction

In Chapter 3, we found that hybrid approaches can outperform centralised or decentralised approaches in certain tasks. For our experiments, we studied an environmental monitoring task where agents were tasked with observing events that occurred according to an underlying density function. Agents acted as one of three different types of systems; decentralised, centralised or hybrid.

We also found that different levels of centralisation or decentralisation within a hybrid system can lead to trade-offs in performance and communications to an operator. We were able to do this by varying the δ value for the swarm which dictated the threshold number of events that an agent needed to be observing at any one point in the environment to form a root node. Colloquially, the δ value relates to how easy it is for the swarm to form trees. In these experiments, the δ value was fixed at the beginning of the simulation and kept constant for the duration of that simulation.

However in reality this becomes impractical because it may not be possible to determine a suitable δ value before the swarm enters an unknown environment. Instead, the swarm needs to be able to learn the δ value depending on the environment and according to the mission requirements set by the operator. Doing this would allow the swarm to optimise the level of centralisation it exhibits on the fly, according to the current state and dynamics of the environment it is operating in.

This idea of exploring the mechanisms and the transition that allow a decentralised, self-organised system to become a hierarchical system, and vice-versa, is an important challenge that so far has not been addressed [18]. Much of the work in hybrid systems focuses on the development of the robotic infrastructure to allow such a system [4] and examining the feasibility of such an approach in a swarm setting [5, 33].

In this chapter, we examine the possibility of the swarm being able to learn the optimal δ values and vary this depending on whether performance or communication to the operator is an important factor for the mission. In doing so, we examine the mechanisms required to smoothly transition a swarm system from a decentralised system to a hierarchical system and back again. We propose an extension to the hybrid approach, which was studied in Chapter 3, that gives a swarm the ability to vary its level of (de)centralisation depending on feedback from an operator and the current state of the environment. We find that by using this approach, the swarm can find the optimal parameters to maximise the performance and communication requirements in an environmental monitoring problem.

5.2 Method

As in Chapter 3, we use a simple environmental monitoring task to examine a hybrid system and the mechanisms that affect and enable it to move along the spectrum from a decentralised to a hierarchical system and back again. Events occur within the environment according to an underlying density function which is unknown to the swarm. Events are present until they have been observed for a total of 200 seconds at which point they are deemed completed and disappear from the environment.

We extend the hierarchical approach that was used in the Chapter 4 to enable it with the capability of varying the δ value during the simulation. To do this we use Particle Swarm Optimisation (PSO). The aim of this algorithm is to optimise both the δ values and the dissolution times iteratively, (δ, t_d). These values are initially assigned a random value between a specified range. During this experiment, these ranges were $0 \leq \delta \leq 10$ and $0 \leq t_d \leq 300$. These initialised values make up a population of candidate solutions with each particle knowing the global best solution and the particle's own best solution that it has found so far during the simulation.

In our approach, each member of the swarm is considered a particle, similar to approaches that have previously been proposed to extend PSO for robot swarms in tasks such as control [116, 117], motion planning [118] and target search [119, 120, 121]. Within multi-robot and swarm systems, PSO is used to optimise and tune parameters, often using only local interactions. It is well suited for finding the best solution in a complex space that may have multiple local minima in unsupervised learning problems since it uses fitness functions to evaluate particles that can be updated based on observations, without the need for external input. It also does not require significant computation, is simple to implement and requires only three controlling parameters [122, 123]. It is for these reasons that we decided to use PSO for this problem.

The algorithm for the implemented PSO is as follows. For our experiments, each agent is acts as one particle in the PSO algorithm. Agents share information with neighbours and update their own parameters accordingly at set intervals of 30s. This reduces the speed at which the algorithm converges and gives agents time to evaluate their learned parameters. At each interval, the velocity and position are updated with the particle and the swarm's collective knowledge over the problem space, according to the following equation:

$$V_{t+1}^{i} = V_{t}^{i} + \varphi_1 R_{1_t^i}(p_t^i - x_t^i) + \varphi_2 R_{2_t^i}(g_t^i - x_t^i) \tag{5.1}$$

Where each particle, i, at iteration, t, has velocity, V. φ_1 and φ_2 are the acceleration coefficients which dictate how much an individual particle's velocity should be influenced by the global and individual best value, which are given by g_t^i and p_t^i for particle i, at time t, respectively.

To evaluate the solutions of each particle, we define a fitness function that is maintained by each agent in the swarm. This considers the performance of the individual and the communications to an operator. The fitness function is defined as the following:

$$f_t = u_p P_t + (1 - u_p) C_t \tag{5.2}$$

The performance of an individual agent, P_t, is defined as the number of events monitored by that agent in a set time frame prior to time t. The motivation for this aspect of the fitness function is to ensure that agents are aiming to situate themselves in areas of the environment with a high level of activity. The communications aspect of the fitness function, C_t, is defined as the reduction in the number of messages that needs to be sent to an operator in a set time frame, prior to time t. This is found by calculating the difference in the total number of events observed by an agent's sub-tree and the number of messages that the agent would need to send to an operator to inform them of those events. In this case, it is found by taking the number of events observed by the sub-tree and subtracting 1 (for the single message that would relay information about these events to the operator). The coefficient u_p dictates whether the focus for the swarm is placed on enhancing performance or reducing communication to an operator, where $0 \leq u_p \leq 1$.

$$\text{P-best} = \begin{cases} [\delta_t, T_t] & \text{if } f_t > f_{Pbest} \\ [\delta_{Pbest}, T_{Pbest}] & \text{otherwise} \end{cases} \tag{5.3}$$

$$\text{G-best} = \begin{cases} [\delta_t, T_t] & \text{if } f_t > f_{Gbest} \\ [\delta_{Gbest}, T_{Gbest}] & \text{otherwise} \end{cases} \tag{5.4}$$

At each time step, agents compute their own fitness score using equation 5.2. This individual score is compared to the fitness score corresponding to the personal best set of parameters found by an agent, P-best $= [\delta_{Pbest}, T_{Pbest}]$, and the global best set of parameters that the agent has saved, G-best $= [\delta_{Gbest}, T_{Gbest}]$. G-best is shared by agents and represents the best set of parameters found by the swarm for the experiment. If $f_t > f_{Pbest}$ then the agent updates P-best with its current set of parameters, $[\delta_t, T_t]$. Otherwise, P-best remains unchanged. The agent also compares the current fitness score, f_t, to the global best, f_{Gbest}. If $f_t > f_{Gbest}$, then G-best is also updated accordingly. Otherwise, G-best remains the same. This is described in equations 5.3 and 5.4.

$$\text{G-best} = \begin{cases} [\delta^j_{Gbest}, T^j_{Gbest}] & \text{if } f^j_{Gbest} > f^i_{Gbest} \\ [\delta^i_{Gbest}, T^i_{Gbest}] & \text{otherwise} \end{cases} \tag{5.5}$$

G-best is also updated by each agent when they exchange information with another agent which has found a G-best with a higher fitness score, described in equation 5.5. Agents share information on G-best with neighbours when they are in communication range with each other. When agent i communicates with agent j, they compare the best global fitness score that they have found, given by f^i_{Gbest} and f^j_{Gbest} respectively. Each agents' G-best parameters are then updated to the values with the highest corresponding fitness score.

We also examine the importance of a second parameter in the optimisation process. This is the dissolution time, T, measured in seconds. The significance of this parameter is that it works as the opposite of the δ value. The δ value dictates how quickly or easily an individual agent switches states from a decentralised to a centralised strategy. The dissolution time dictates the reverse, how quickly or easily an agent switches states from a centralised to a decentralised agent. Both values constantly adjust the behaviour of agents in the swarm, allowing them to transition between a centralised system, a fully decentralised system and a set of fluid intermediate states between the two as required. The optimal amount of time spent in any of these states is dictated by the interaction between both parameters, δ and T.

5.3 Results

In figure 5.1 it is possible to see the relationship between T, the time agents wait after observing an event, and δ, the threshold number of events an agent must be observing in order to form a root node. Systems that have a low δ value will form trees more easily during the task and as such, agents will spend more time acting as a centralised system, while the converse it true for systems with a higher δ value, which will spend more time as a decentralised system. When it comes to T values, systems that have a higher value will wait for longer in a tree after they have finished observing all their events within sensing range. This means they will spend more time as a centralised system and are more likely to observe new events at their current location or to be relocated within the tree by the root node. On the other hand, systems with low T values will wait for less time before leaving the tree, if they do not observe any new events at their location. As such, these systems will spend more time as decentralised systems.

Figure 5.1(A) shows the mean wait time of events for swarms using the varying T and δ values. Although the landscape is complex and rugged, the optimal value here occurs with $T = 20, \delta = 4$ with a wait time of $26.87s$, however there are 13 other high performing configurations in the region of $10s \leq T \leq 150s, 2 \leq \delta \leq 4$, with wait times very close to the optimal value (within 5% of the optimal wait time). This shows that for this problem, a system with a high level of flexibility is best suited for minimising the wait time of events in the environment. Systems that have a low δ and T value will very quickly form tree structures in the environment, and will not wait long after events have finished being observed. This overall strategy means that the swarm will potentially form many tree structures but only those that are located in strong positions, with a high density of events will remain. The others will dissolve or merge when events have been observed.

Figure 5.1(B) on the other hand shows the mean messages sent to an operator per second by the swarm for varying T and δ values. In this landscape, which is much smoother than the one shown in figure 5.1(B), the optimal value occurs with a system that has the values of $T = 270, \delta = 7$. This result shows that in order to minimise the number of messages sent to an operator, the system should be strict in its requirements to form a tree (given by a high δ value) but once, a tree has been formed, agents should remain a part of that tree for a longer time. As such, this reduces the number of trees formed during the experiment, but allows time for that tree to grow in size and improve its coverage over the environment.

The difference in optimal configurations for T and δ highlighted in figure 5.1 suggests that both the δ and T values affect the performance of the system and that considering both the transition from centralised to decentralised, controlled by T is just as important as the reverse, controlled by δ. To fully optimise the system, we need to

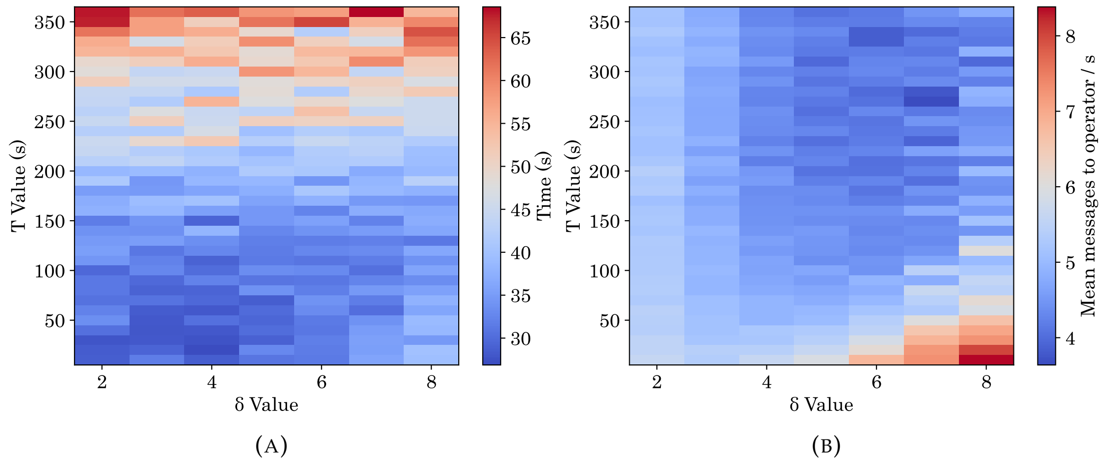


FIGURE 5.1: Heat map showing the (A) mean waiting time of events and (B) mean messages to an operator per second for different T and δ values. T values were varied at intervals of 10s. Results were aggregated over 10 replicates

consider both values. This points to the idea that having a hybrid swarm system has a constant push and pull between a decentralised structure and a more centralised structure can improve the performance of that swarm in certain tasks, which will come with trade-offs to other aspects of the swarm system such as the number of messages sent to an operator. Here the mechanisms we use to determine the switch between both states, δ and T, are event-based and time based respectively. These two parameters both rely on the current state of the environment in that δ depends directly on the number of events present at an agent's location while that agent is in a decentralised state. T on the other hand depends indirectly on whether any events are present at an agent's location since this parameter dictates the time an agent remains at a location in a centralised state after observing all events at that location. If any new events occur at that location again, this timer is reset. The fact that these parameters are both reactive to the environment allows the system to remain flexible to changing conditions which in this task is new events appearing at new locations.

Figure 5.2 shows the distribution of mean messages to an operator per second against the mean waiting time of events for the 252 different configurations of T and δ. Each point represents the mean of 10 replicates, with some replicates being positioned ahead of the Pareto front. The Pareto front is shown by the red points, joined with a red line. This represents the configurations of T and δ that are not dominated, in terms of the wait time of events or the number of messages sent to an operator per second, by any other configuration. It shows the trade-offs that must be considered when trying to optimise the two objectives, in this case minimising the mean messages sent to an operator per second and also minimising the mean waiting time of events in the environment. This will be an important measure for the quality of solutions and configurations that our PSO approach will be able to find. In this problem, where there are clear trade-offs between the mean waiting time and the mean messages sent

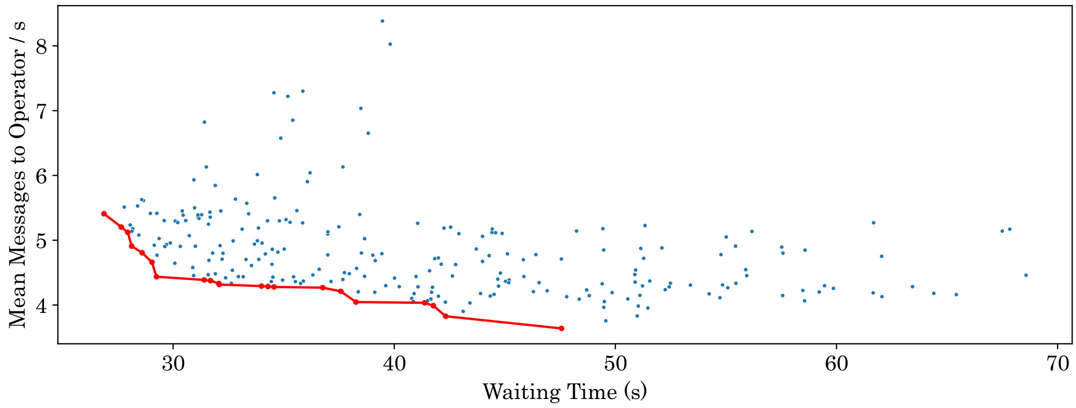
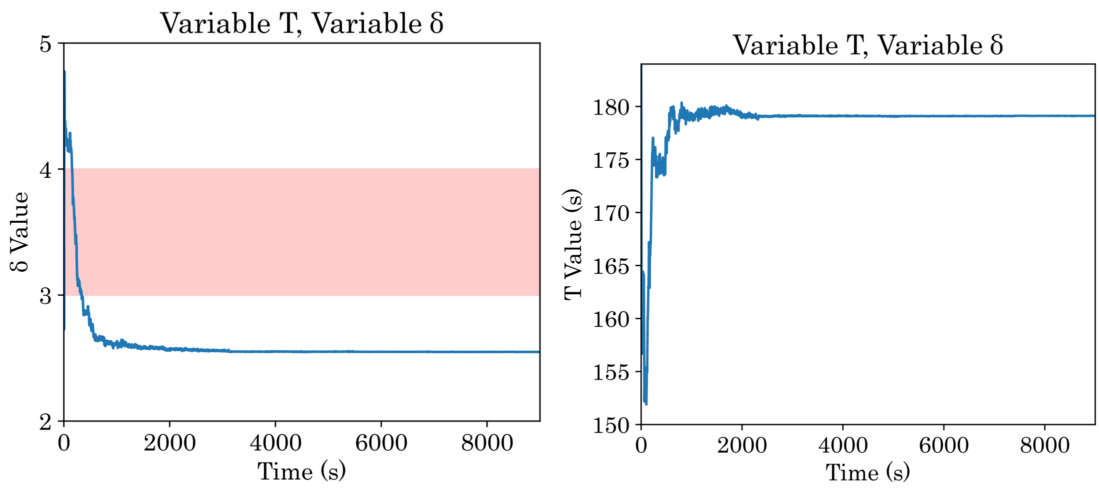


FIGURE 5.2: Plot showing the distribution of mean messages to an operator per second against the mean waiting time of events for different T and δ values. δ values are in the range $2 \leq \delta \leq 8$ with intervals of 1, while T values are in the range $10s \leq T \leq 360s$ with intervals of 10s. Results were aggregated over 10 replicates. The Pareto front is shown in red.

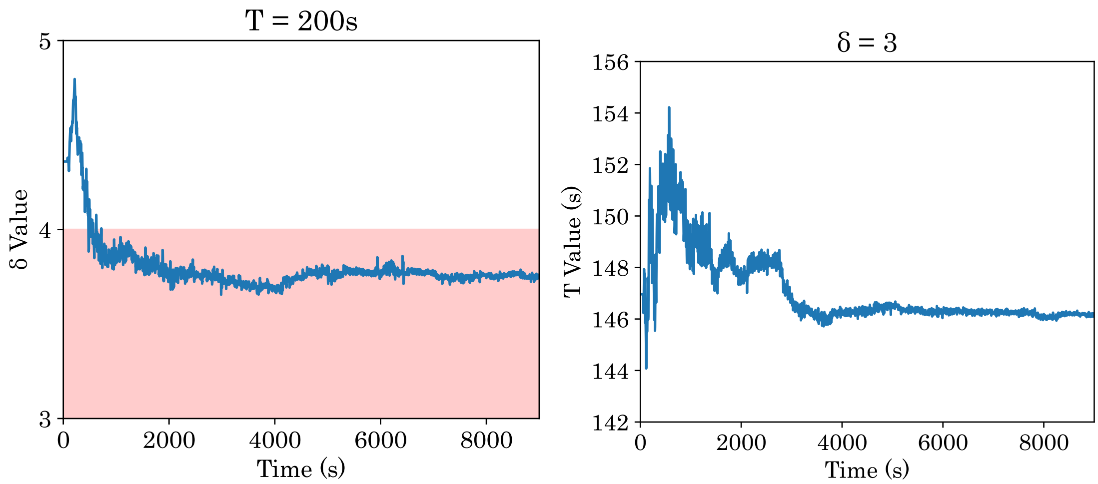


(A) (B)

FIGURE 5.3: Plot showing (A) how the mean global best δ value of the swarm varies over time. The red area shows the area where the optimal δ value was found in the Chapter 4 and (B) how the mean global best T (s) value of the swarm varies over time. In both figures, the two values, δ and T were optimised simultaneously. Both results were aggregated over 20 replicates.

to an operator, the resulting configuration from the optimisation process should be close to or exceed the Pareto front. The solution may lie ahead of the Pareto front due to the changes in values of T and δ during the optimisation process.

Figure 5.3 shows how the global best of the two values, δ and T, change throughout the simulations. We can see that the δ value is not optimised to the optimal value that was found in the Chapter 4, $\delta = 3$, and the T value is found to be lower than the value used in Chapter 4 too, which was set at $T = 200s$. However, figure 5.1 suggests that the values found in Chapter 4 is not the global optimum and as such. Since the T



(A) (B)

FIGURE 5.4: Plot showing how the mean global best δ value of the swarm varies over time, when (A) the T (s) value is kept constant, using the value of T that was used in Chapter 4 of $T = 200$ and (B) the δ value is kept constant, using the value that was found in Chapter 4 of $\delta = 3$. Results were aggregated over 20 replicates.

value was fixed to find the result in Chapter 4, allowing the swarm to optimise and vary this value itself, may mean that the resulting T value is also linked to a different optimal δ value. This may explain the difference in the values to those found in Chapter 4. It's important to note here that any δ value in the range $3 \leq \delta < 4$ would act the same as $\delta = 3$, the optimal value found in Chapter 4. This is highlighted by the red region in figure 5.3. Instead, here, the mean value of $\delta = 2.57$, was found after 1600s, while $T = 178s$ was found after 2000s over the course of 20 replicates.

Figure 5.4 shows the mean δ and T values that are optimised separately and we see this time that the δ value does sit within the range of the optimal value that was found in Chapter 4. This value reaches $\delta = 3.6$ after 5000s, however δ lies within the range of $3 \leq \delta < 4$ after 900s. The value of $T = 146s$ is found at 3500s although the variance of this value is relatively small throughout the experiment with a range of just 10.4s. The motivation for this was to examine how well the swarm is able to optimise a single parameter when the other is set, rather than both simultaneously. This could be a useful tool when some information about the environment is known *a priori*.

Now if we look at figure 5.5 we see the performance of the swarm in terms of the mean waiting time of events for the simulations. Even fixing one value results in a better median performance than decentralised or when both are optimised together. When we combine the results of the optimised values, we find a marginal performance improvement, with the combined optimal values with a median performance time of 29.0s compared to the Fixed T median performance time of 29.8s, the Fixed δ median performance time of 30.0s and when T and δ are optimised together with a median performance of 32.1s. One important point here, however, is

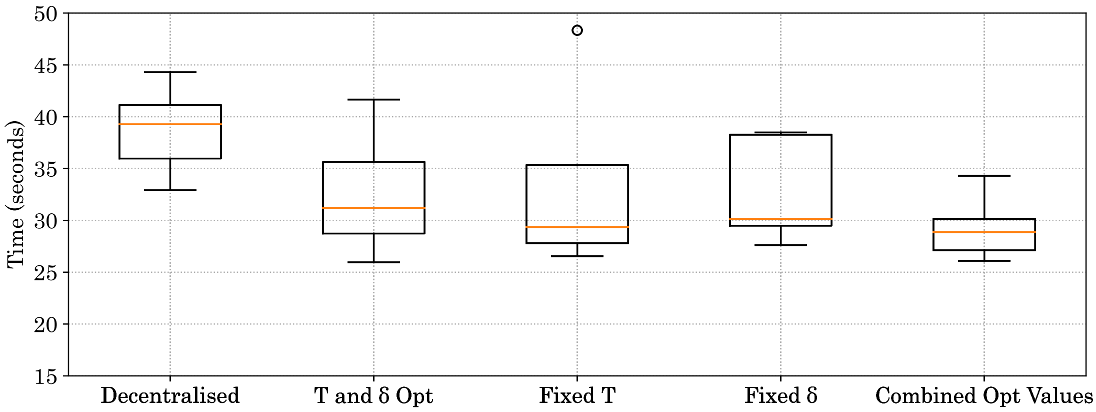


FIGURE 5.5: Box plot of mean waiting time for events to be observed by swarms of sizes 25 agents coordinated with the Decentralised (Decent) and Hybrid approaches with both δ and T being optimised together (T and δ Opt), a fixed T with δ being optimised (Fixed T), a fixed δ value with T being optimised (Fixed δ) and the result of using the found optimised values of δ and T which are fixed at the beginning of the simulation. Data was aggregated across 20 replicates.

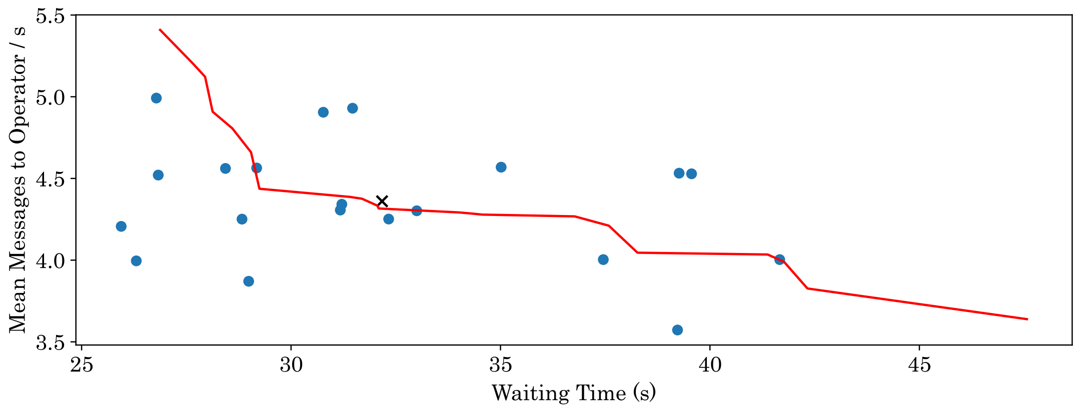


FIGURE 5.6: Plot showing the distribution of mean messages to an operator per second against the mean waiting time of events for 20 replicates of the Hybrid approach with both δ and T being optimised together. The Pareto front, shown in figure 5.2, is also shown here in red and the mean of the 20 replicates is shown by a black cross.

that the distribution of the performance is much more compact for the combined optimal values with an IQR of just 3.2s compared to the Fixed T IQR of 7.9s. Using the optimised values, the swarm has shown a more consistent performance acting to keep the waiting time of events at a lower level more frequently. Even when using the T and δ simultaneously optimised values, does the performance of the swarm improve on a decentralised approach.

Figure 5.6 shows the distribution of mean messages to an operator per second against the mean waiting time of events for 20 replicates of the Hybrid approach in which both the T and δ values were optimised simultaneously. The red line shown here represents the same Pareto front shown in figure 5.2. The mean of the 20 replicates is also shown here by a black cross. Here, 15 of the 20 replicates are ahead of or on the

Pareto front that is shown. The mean is also positioned very closely to the Pareto front, with a value of $(32.17s, 4.36)$. This value has an L2 distance of 0.09 from $(32.09s, 4.32)$ which corresponds to the point that represents the results of a system with $T = 150s, \delta = 7$ which is also on the Pareto front. As such, we can see that when trying to optimise both the T and δ parameters together, the resulting configuration and performance of the system while optimising is, on average, as good as possible without increasing the waiting time of events or the mean messages to and operator.

5.4 Discussion

The results presented in this chapter provide evidence that there are two main mechanisms involved in allowing a hybrid system to fluidly transition between a centralised and fully decentralised system; The first is the transition from centralised control to decentralised control which is governed here by parameter T. The second is the transition from decentralised control to centralised control, governed here by the parameter, δ. Each parameter is important in determining an agent's level of centralisation or decentralisation within the swarm. An important indicator of this can be seen in fig 5.1 where varying the T and δ values influences the mean waiting time of events and the mean messages to an operator. While we have seen previously, in Chapter 3, that the δ value played a crucial role in optimising the performance of the hybrid system and its communications with an operator, the T value was not explored. In this chapter, we did consider the T value as an additional parameter and aimed to optimise this alongside the δ value for this task.

We implemented a PSO algorithm to allow the swarm to learn and optimise the values of δ and T, individually or simultaneously. Two separate experiments were run where one parameter was kept constant, while the other was optimised by the swarm. This gave converged values for each parameter separately. These values are shown in figure 5.4, of $T = 146s$ and $\delta = 3.7$ (which performs the same as $\delta = 3$ in our implementation). When given these values as constants, the resulting simulation gave the best performance of the swarm compared to all other methods that were tested. This can be seen in figure 5.5.

While optimising both values of T and δ, there were a number of issues that impacted the effectiveness of the PSO approach in being able to reach a solution quickly. The first issue stems from the interaction each parameter has on one another; Firstly, the swarm begins in a decentralised state with randomised δ and T values. Over time, agents begin to form trees and initially, the root nodes are composed of agents with lower δ values. This is a result of a low density of events present in the environment at the beginning of the simulation. The agents' fitness function will begin to increase when they start to observe events. This in turn stimulates the optimisation of both the

δ and T values. Due to the nature of the fitness function, the formation of trees and recruitment of sub tree members greatly increases the agents' fitness score. Agents that form root nodes have the highest fitness scores compared to other members of the swarm. As a result, the fitness scores from these root nodes, are shared around the swarm as the global best score. In doing so, the δ value is initially optimised to lower values. As each member of the swarm optimises towards these lower δ values, the range of possible δ values decreases. This results in the swarm becoming stuck in a local optima. This is supported by figure 5.3(A) where the δ value optimises early to a value below the optimum range (highlighted in red).

The second issue is related to the optimisation of the T value. In its current implementation, the T value does not exert enough influence on the fitness score of an agent to make a significant difference to the overall score. Further more, the δ value initially dictates what value of T provides a strong fitness function. This is because when a member of the swarm forms a root, it begins to increase it's fitness score greatly as the tree builds. With a greater number of agents in the sub-tree, there is a greater reduction in the number of messages sent to the operator. As has been established, at the beginning of the simulation, the swarm will optimise towards lower δ values. At this time, the T value is also being optimised, despite not affecting the fitness score. The speed at which the δ value optimises dominates its effect on the fitness function and thus dictates the value towards which T is optimised. Despite having had a minimal effect on the fitness score, the T value is optimised towards the random initialised value of a root node agent. This value of T is also shared with the rest of the swarm as the global best. As such we have the same problem that we have already outlined for the δ value, which is also occurring for the T value. In it's current form, it seems the fitness function is dominated by the communication element of the function. One way this could be addressed would be to study the Pareto front of P and C to determine values of the u coefficient which does not result in the function being dominated by the C term.

These issues are exacerbated by the fact that, in a swarm of 25 agents, there is relatively little diversity in the initial values of δ and T. This comes despite using Latin Hypercube sampling to increase the coverage of chosen values in the parameter space. As such, the outcome of the optimisation is heavily dependent on whether or not an agent forming an early tree is initially given δ and T values close to that of the optimal values. Despite this limitation, placing bounds on the maximum and minimum possible values for δ and T may provide a viable approach. Setting appropriate bounds on the δ and T values limits the parameter space while keeping it broad enough to provide diversity in the initial values. PSO may be suitable for this task once the changes that have been suggested are taken into account. These changes being alterations to the fitness function to avoid domination of one term and the bounding of possible values for δ and T. However, the results in this chapter have given us an outlook that it is possible to learn optimal values of δ and T for a given task.

Another limitation of this work is in the lack of real robotic experiments. While the MNS has been proposed as an implementation of a hybrid control system [4], a hybrid control system that can change its level of centralisation on the fly has not been considered. As such, there are several practical challenges that such systems face which have not been explored. A key practical challenge facing such a system is the presence of sensor noise. In our approach agents can detect events present in the environment. In a real-world scenario, these events may correspond to a variety of phenomena, such as the outbreak of a fire, or the emergence of a human from a shelter in a search and rescue mission. The sensors used to detect such phenomena will be vulnerable to several noise sources, causing the observed values to deviate from the true value present in the environment. Sensors that measure physical quantities in the environment can be affected by factors such as temperature and polling rate which may be unable to detect small fluctuations in the true value of the observed phenomena [124]. In cases where techniques such as object detection and classification are used, the system will be affected by the accuracy of the implemented detection model and the quality of the images used. In our approach, both mechanisms that dictate the transition between control structures for a single individual, rely on the detection of events via a sensor. Introducing noise into this system may affect how accurately the swarm perceives the environment and how individuals switch between a centralised and decentralised state. Noise present in the positioning system of the swarm could also affect the swarm's ability to successfully maintain coordination in our proposed hybrid system. Agents rely heavily on being able to accurately track their own position in the environment. This enables them to ensure they remain within the communication range of their leader and local sub-tree members when operating as a centralised system. Noise in this system could result in disconnections of the branches in sub-trees, affecting the overall performance of the swarm.

Another hardware challenge that a hybrid control system may face, revolves around the communication technology used. This is a particular concern for the hybrid control swarm's ability to share information on the events that have been detected and the coordination within the centralised structures that have formed. Here, there would be two possible limitations. The first is in the bandwidth capabilities of the wireless channel being used [23] which affects how much information can be relayed efficiently, with high bandwidth channels being reserved for transmitting information to the human operator. The second is the latency that the swarm may experience in receiving messages from neighbours as a result of the duty-cycle that is imposed on the communication channel [23]. In the proposed hybrid control system, this may impact the swarm's ability to react to changes in the density of events present in the environment and reduce the performance of the system as a whole. To address and further understand the challenges highlighted here, future work must focus on the implementation of hybrid control systems into real robotic platforms.

5.5 Conclusion

In this chapter, we have explored some of the important factors affecting a hybrid swarm's ability to switch states from a centralised to a decentralised system and back again. In our experiments, these two transitions were controlled using two parameters, δ, which is an environmental observation value and T which is a time-based value. We have shown that both transitions are vital to the operation of a hybrid system and affect the performance of that system in its task.

We have also shown that these parameters can be learned for a specific task. In our experiments we used PSO. The δ and T values were learned simultaneously and separately in our experiment. When learned simultaneously, the approach outperformed a purely decentralised or hierarchical system and showed that in unknown environments, that such a system can be applied to give the swarm a method of optimising its level of centralisation according to its environment and the requirements of an operator. The optimised approach performed in a way, close to approaches that made up the Pareto front for this task, with an L2 distance from the Pareto front of 0.9. When learned separately, the system using the converged δ and T values performed the best compared to all other approaches that were used. This showed that in certain tasks, the optimal system can lie somewhere between a fully centralised and fully decentralised system. Alterations to the fitness function and the use of other reinforcement learning techniques such as Deep Q-Networks (DQNs) may improve the ability of the swarm to optimise these values simultaneously. The fitness function could also be studied to find the Pareto front which would enable coefficients to be optimised, avoiding the outcome where the function is dominated by one term.

One key limitation of this work is that only one kind of task was considered. Exploring how the swarm is able to optimise its level of centralised and decentralised control for other tasks needs to be considered in order to examine how the underlying mechanisms are affected in such tasks. Scenarios that benefit specifically a centralised or decentralised approach more, or a task that requires adaptation to changes in the environment would present a more complex challenge for the swarm and the optimisation process. As such, the results of this chapter are limited, in that the proposed approach may not generalise well to new problems. However, we have shown that in this task, that it is possible to optimise the level of centralisation according to operator requirements and find a strong solution to the configuration of a system that can manage the trade-offs between performance and communication to an operator in this task.

Chapter 6

Conclusions and future work

In this chapter, we present a summary of our conclusions, future work and research directions that may come from the work in this thesis.

In Chapter 3, we examined how the decision making performance of the swarm is affected in communication-constrained environments. To do this, the performance of a swarm system in a collective perception problem under communication constraints was considered. We extended the collective perception problem proposed in [52], to include a communication environment that defined where agents could successfully communicate. This performance was tied to the time taken for the swarm to reach a consensus when communications were denied in a proportion of the environment. The resulting communication quality distribution differs from most of the studies that are present in the literature which consider homogeneous communication environments [54, 53, 52, 48, 33]. This resulted in the proposal of a new benchmark scenario to study the collective perception of a robot swarm in a communication-constrained environment. The results showed that the performance is higher than the state-of-the-art collective decision-making algorithms in communication-constrained setups. This was thanks to the communication-aware approach that was used by the swarm along with a simple coordination algorithm. This coordination algorithm made use of decentralised planning and considered the uncertainty of the environment. Agents did not make explicit plans with other members of the swarm which may require replanning in the future. This addresses gaps in the literature of multi-robot coordination where *a priori* knowledge was required for agents to reconnect in the future [50, 45, 44]. Instead, agents plan a few time steps in advance based on their own information. This forms a map over the communication environment and the associated uncertainty, which is supported by beliefs they have received from neighbours. If the information held by an agent changes in the future, the agent can adapt appropriately, without having to consider previously arranged plans. Here, agents assume that neighbours have similar beliefs to them over the communication environment.

Future work in this area may focus on examining the scalability of the communication-aware approach. Considerations were made with regards to reducing the complexity of the GP and the communication overheads required for sharing data relating to it. However, it is important to see if these techniques can scale to larger environments and swarm sizes. Another case that was not considered was when the features present in the environment correlated with the communication-denied environment. A study of the proposed communication-aware algorithm in such an environment would provide insight into the robustness of the algorithm in more challenging environments.

In Chapter 4 we also considered how varying a swarm's control structure affects the performance, the inter-swarm communications and swarm-human communications. We explored the assumption that a centralised approach always achieves the best performance in a task over a decentralised or hybrid solution. While previous work has compared the performance of centralised, decentralised and hybrid approaches [6, 125], this is the first time that the level of centralisation (or decentralisation) has been varied. Previous approaches have considered hybrid systems based on the MNS [4], which makes use of centralised and decentralised aspects to form self-organised formations [4, 33, 6]. In contrast, our approach allows the swarm to flexibly vary the swarm-wide level of centralisation (or decentralisation) according to the state of the environment. It is also the first study to consider factors of hybrid swarm control systems that may affect human operator cognitive load. Here, the number of messages sent to an operator was used to represent the information being shared with a human. It has been shown that the volume of messages sent to an operator can affect the performance of a human-swarm system [42]. To do this, a simple generic hybrid approach which combines elements of decentralised and centralised control systems was proposed. It was shown that the proposed hybrid system could achieve higher performance in an environmental monitoring task compared to a simple centralised and decentralised approach. The centralised approach used here was not the most sophisticated and could be improved upon. However, it presents a case which suggests that the optimal performance of a system, in terms of mission requirements, and communications may depend on the control structure present in that system. It was also shown that this performance increase comes with trade-offs in inter-swarm and human-swarm communications. We showed that these trade-offs can be balanced with the performance of the system according to end-user requirements while using our proposed hybrid approach. This gives a system the ability to answer the question; what is the optimal point of performance and communication where both of these conditions can be met satisfactorily for the task and the operator requirements?

The final area of work examined in Chapter 5 was the possibility of being able to learn the optimal values of parameters that dictated the level of (de)centralisation for the best performance of the swarm in a given task. This extended the work carried out in

Chapter 4. Here, the swarm learned and optimised a set of parameters that were crucial in defining its behaviour and control structure. This allowed the mechanisms required for a swarm to transition between a centralised and decentralised control architecture during a mission. Such a technique has not been explored in previous work considering hybrid control systems [4, 5, 33, 6]. This is an important factor in understanding how such systems operate and how they can adapt to their environment [32]. By using this approach, it is possible to optimise the parameters for a swarm to maximise the performance and communication requirements in an environmental monitoring problem. It was found that it is possible to learn the optimal parameters in our simple proposed hybrid system when considering the two parameters separately. However, the method for doing so was unable to find the optimal values simultaneously during the experiments. The limitations with this approach for optimising the values were discussed.

Future directions for this work may focus on improving the approach seen in Chapter 5 for optimising the parameters used by the swarm for varying the level of (de)centralisation. While the technique used in this thesis was unable to optimise the values simultaneously, this may be achieved by studying and honing further the fitness function of the swarm. Analysis of the Pareto front of the function could allow for a more refined selection of coefficients, reducing the dominance of the communication (C) term over the performance (P) term. Other techniques could be explored for more effective optimisation of the parameters, such as deep learning approaches like DQNs. Another interesting direction of future work would look at the robustness of the hybrid system to failure. Our implementation gave agents the capability to continue to function and adapt when a parent node or leader loses contact with the sub-tree. However, the robustness was not studied here, with a greater focus placed on the examination of the trade-offs when the level of centralisation (or decentralisation) is varied. These are key limitations related to the work done in these chapters.

A particularly important future direction for this work is to implement the algorithms proposed in Chapters 3, 4 and 5 into a real robotic platform. Doing so would allow the assessment of the feasibility of having the algorithms proposed in those chapters implemented into a real system and address a key limitation of this work. For example, having the swarm suffer from sensor and communication noise would provide insight into whether the uses of the approaches proposed could be integrated into real-world scenarios. Another factor that would be highlighted in a real-world platform implementation would be the robustness of the proposed approaches to the failure of individual robots. In Chapter 3, a real-world implementation would be particularly important to examine the feasibility of using a Gaussian Process to map the communication environment. We integrated several techniques to reduce the complexity of the GP computation and communication overhead of sharing data related to the GP generation. Implementing these approaches in a real swarm platform would allow the examination of the true communication, energy and computational complexity costs when using such an approach. It would also give insight into the communication overheads that may be present in a swarm using the communication-aware approach. Having a large number of agents communicate within a relatively small area could result in communication collisions, which is not considered in our simulations. In Chapters 4 and 5, a real swarm platform could allow similar considerations. One example would be the study and integration of energy trade-offs into the optimisation of the swarm system. Doing so may allow further fine tuning of the system according to mission requirements that may be defined by a human operator.

One final limitation of this work that may be addressed in future works is the lack of different scenarios for the work done in chapters 4 and 5. As such the results and conclusions drawn from these chapters are limited to the task considered. Examining how these approaches and claims extend to more general problems and cases will be important in addressing the implementation of hybrid control systems for swarms into real applications. Understanding better the relationship and trade-offs between the communication of the swarm and its performance in a task is important in enabling a swarm to improve the situational awareness of a human operator. As such, it is important that this limitation is discussed.

Other avenues of work could focus on providing insight into the effects that hybrid systems, and particularly adaptive hybrid systems, have on human operators. While fully centralised and decentralised systems have a wealth of knowledge in the field of Human-Swarm interaction, newer hybrid systems do not. Our work focused on the performance and the number of messages sent to an assumed human operator. However, user studies examining the real-world effects of a swarm that changes its control structure on the fly may provide important insights into the true challenges faced by humans operating these systems which will become an important area of future swarm research. Ensuring a swarm system can effectively work and interface with human operators is a vital capability for such systems. This is due to the current necessity to include a human in-the-loop during the operation of swarms. Considering these kinds of challenges will aid in the deployment of robot swarms into a broader number of real-world applications.

References

[1] T. G. Kelly, M. D. Soorati, K.-P. Zauner, S. D. Ramchurn, and D. Tarapore, “Collective decision making in communication-constrained environments,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7266–7271.

[2] T. G. Kelly, M. D. Soorati, K.-P. Zauner, S. D. Ramchurn *et al.*, “Trade-offs of dynamic control structure in human-swarm systems,” *arXiv preprint arXiv:2408.02605*, 2024.

[3] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Autonomous Robots*, vol. 8, pp. 345–383, 2000.

[4] N. Mathews, A. L. Christensen, R. O’Grady, F. Mondada, and M. Dorigo, “Mergeable nervous systems for robots,” *Nature communications*, vol. 8, no. 1, pp. 1–7, 2017.

[5] W. Zhu, M. Allwright, M. K. Heinrich, S. Oğuz, A. L. Christensen, and M. Dorigo, “Formation control of uavs and mobile robots using self-organized communication topologies,” in *International Conference on Swarm Intelligence*. Springer, 2020, pp. 306–314.

[6] A. Jamshidpey, M. Wahby, M. Heinrich, M. Allwright, W. Zhu, and M. Dorigo, “Centralization vs. decentralization in multi-robot coverage: Ground robots under uav supervision,” *arXiv preprint arXiv:2408.06553*, 2024.

[7] W. Zhu, S. Oğuz, M. K. Heinrich, M. Allwright, M. Wahby, A. L. Christensen, E. Garone, and M. Dorigo, “Self-organizing nervous systems for robot swarms,” *Science Robotics*, vol. 9, no. 96, p. eadl5161, 2024.

[8] T. K. Kaiser, T. Potten, and H. Hamann, “Evolution of collective decision-making mechanisms for collective perception,” in *2023 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2023, pp. 1–8.

[9] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, and N. R. Jennings, “Decentralized coordination in robocup rescue,” *The Computer Journal*, vol. 53, no. 9, pp. 1447–1461, 2010.

[10] G. Best, M. Forrai, R. R. Mettu, and R. Fitch, “Planning-aware communication for decentralised multi-robot coordination,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1050–1057.

[11] S. Murata and H. Kurokawa, *Self-organizing robots*. Springer, 2012, vol. 77.

[12] D. J. Sumpter, “The principles of collective animal behaviour,” *Philosophical transactions of the royal society B: Biological Sciences*, vol. 361, no. 1465, pp. 5–22, 2006.

[13] R. S. Anderson, “Eolian ripples as examples of self-organization in geomorphological systems,” *Earth-Science Reviews*, vol. 29, no. 1-4, pp. 77–96, 1990.

[14] R. Groß and M. Dorigo, “Self-assembly at the macroscopic scale,” *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1490–1508, 2008.

[15] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” *Swarm Intelligence*, vol. 7, pp. 1–41, 2013.

[16] H. Hamann, *Swarm robotics: A formal approach*. Springer, 2018.

[17] G. Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, “The grand challenges of science robotics,” 2018.

[18] M. Dorigo, G. Theraulaz, and V. Trianni, “Swarm robotics: Past, present, and future [point of view],” *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.

[19] M. Kozai, T., Niu, G., Takagaki, *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*. Elsevier, 2016.

[20] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld, “Airshield: A system-of-systems muav remote sensing architecture for disaster response,” in *2009 IEEE International Systems Conference Proceedings*, 2009, pp. 196–200.

[21] L. Abraham, S. Biju, F. Biju, J. Jose, R. Kalantri, and S. Rajguru, “Swarm robotics in disaster management,” in *2019 International conference on innovative sustainable computational technologies (CISCT)*. IEEE, 2019, pp. 1–5.

[22] K. A. Roundtree, M. A. Goodrich, and J. A. Adams, “Transparency: Transitioning from human–machine systems to human-swarm systems,” *Journal of Cognitive Engineering and Decision Making*, vol. 13, no. 3, pp. 171–195, 2019.

[23] D. Tarapore, R. Groß, and K.-P. Zauner, “Sparse robot swarms: Moving swarms to real-world applications,” *Front. Robot. AI 7: 83. doi: 10.3389/frobt*, 2020.

[24] J. Y. Chen, K. Procci, M. Boyce, J. Wright, A. Garcia, and M. Barnes, “Situation awareness-based agent transparency,” Army research lab, Aberdeen proving ground, MD Human Research and Engineering, Tech. Rep., 2014.

[25] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.

[26] A. S. Rao, M. P. Georgeff *et al.*, “Bdi agents: From theory to practice.” in *ICMAS*, vol. 95, 1995, pp. 312–319.

[27] J. D. Lee and K. A. See, “Trust in automation: Designing for appropriate reliance,” *Human factors*, vol. 46, no. 1, pp. 50–80, 2004.

[28] M. L. Cummings, “Human supervisory control of swarming networks,” in *2nd annual swarming: autonomous intelligent networked systems conference*. Citeseer, 2004, pp. 1–9.

[29] P. Walker, S. Nunnally, M. Lewis, A. Kolling, N. Chakraborty, and K. Sycara, “Neglect benevolence in human control of swarms in the presence of latency,” in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012, pp. 3009–3014.

[30] S. Nunnally, P. Walker, A. Kolling, N. Chakraborty, M. Lewis, K. Sycara, and M. Goodrich, “Human influence of robotic swarms with bandwidth and localization issues,” in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012, pp. 333–338.

[31] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, “Efficient multi-robot coverage of a known environment,[in:] 2017 ieee,” in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1846, pp. 1846–1852.

[32] M. Dorigo, G. Theraulaz, and V. Trianni, “Reflections on the future of swarm robotics,” *Science Robotics*, vol. 5, no. 49, p. eabe4385, 2020.

[33] A. Jamshidpey, W. Zhu, M. Wahby, M. Allwright, M. K. Heinrich, and M. Dorigo, “Multi-robot coverage using self-organized networks for central coordination,” in *International Conference on Swarm Intelligence*. Springer, 2020, pp. 216–228.

[34] Y. Zhang, S. Wang, S. Oğuz, M. Heinrich, X. Wang, E. Garone, and M. Dorigo, “Self-reconfigurable hierarchical frameworks for bearing-based formation control of robot swarms,” Technical Report TR/IRIDIA/2021-004, IRIDIA, Université Libre de Bruxelles . . . , Tech. Rep., 2021.

[35] P. Walker, S. Nunnally, M. Lewis, N. Chakraborty, and K. Sycara, "Levels of automation for human influence of robot swarms," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57. SAGE Publications Sage CA: Los Angeles, CA, 2013, pp. 429–433.

[36] G. Coppin and F. Legras, "Autonomy spectrum and performance perception issues in swarm supervisory control," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 590–603, 2011.

[37] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2015.

[38] A. Hussein, L. Ghignone, T. Nguyen, N. Salimi, H. Nguyen, M. Wang, and H. A. Abbass, "Characterization of indicators for adaptive human-swarm teaming," *Frontiers in Robotics and AI*, vol. 9, p. 745958, 2022.

[39] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. Di Caro, "Human-swarm interaction using spatial gestures," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3834–3841.

[40] J. Patel, Y. Xu, and C. Pinciroli, "Mixed-granularity human-swarm interaction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1059–1065.

[41] M. Soorati, J. Clark, J. Ghofrani, D. Tarapore, and S. D. Ramchurn, "Designing a user-centered interaction interface for human–swarm teaming," *Drones*, vol. 5, no. 4, p. 131, 2021.

[42] A. O. Abioye, M. Naiseh, W. Hunt, J. Clark, S. D. Ramchurn, and M. D. Soorati, "The effect of data visualisation quality and task density on human-swarm interaction," in *2023 IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2023.

[43] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, vol. 7, p. 36, 2020.

[44] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[45] I. Vandermeulen, R. Groß, and A. Kolling, "Re-establishing communication in teams of mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7947–7954.

[46] W. L. Teacy, J. Nie, S. McClean, and G. Parr, "Maintaining connectivity in uav swarm sensing," in *2010 IEEE Globecom Workshops*. IEEE, 2010, pp. 1771–1776.

[47] N. Majcherczyk, A. Jayabalan, G. Beltrame, and C. Pinciroli, "Decentralized connectivity-preserving deployment of large-scale robot swarms," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4295–4302.

[48] J. T. Ebert, M. Gauci, and R. Nagpal, "Multi-feature collective decision making in robot swarms," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 1711–1719.

[49] P. Bartashevich and S. Mostaghim, "Multi-featured collective perception with evidence theory: tackling spatial correlations," *Swarm Intelligence*, pp. 1–28, 2021.

[50] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.

[51] C. Coquet, A. Arnold, and P.-J. Bouvet, "Control of a robotic swarm formation to track a dynamic target with communication constraints: Analysis and simulation," *Applied Sciences*, vol. 11, no. 7, p. 3179, 2021.

[52] G. Valentini, D. Brambilla, H. Hamann, and M. Dorigo, "Collective perception of environmental features in a robot swarm," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 65–76.

[53] G. Valentini, E. Ferrante, H. Hamann, and M. Dorigo, "Collective decision with 100 kilobots: Speed versus accuracy in binary discrimination problems," *Autonomous agents and multi-agent systems*, vol. 30, no. 3, pp. 553–580, 2016.

[54] G. Valentini, H. Hamann, M. Dorigo *et al.*, "Self-organized collective decision making: the weighted voter model." in *AAMAS*, 2014, pp. 45–52.

[55] MoD, "£2.5-million injection for drone swarms," https://www.gov.uk/government/news/25m-injection-for-drone-swarms, 2019, online; accessed 16 July 2021.

[56] G. Jennings, "UK reveals progress in 'swarming drone' development efforts," https://www.janes.com/defence-news/news-detail/uk-reveals-progress-in-swarming-drone-development-efforts, 2020, online; accessed 16 July 2021.

[57] RAF, "Team Tempest," https://www.raf.mod.uk/what-we-do/team-tempest/tempest/, 2020, online; accessed 16 July 2021.

[58] G. Allison, “Royal Air Force swarming drone project ‘exceeding expectations’,” https://ukdefencejournal.org.uk/royal-air-force-swarming-drone-project-exceeding-expectations/, 2020, online; accessed 16 July 2021.

[59] C. E. Harriott, A. E. Seiffert, S. T. Hayes, and J. A. Adams, “Biologically-inspired human-swarm interaction metrics,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58. SAGE Publications Sage CA: Los Angeles, CA, 2014, pp. 1471–1475.

[60] L. Ruetten, P. A. Regis, D. Feil-Seifer, and S. Sengupta, “Area-optimized uav swarm network for search and rescue operations,” in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0613–0618.

[61] J. H. Reif and H. Wang, “Social potential fields: A distributed behavioral control for autonomous robots,” *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[62] A. Howard, M. J. Matarić, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 299–308.

[63] Q. Wang and H. Zhang, “A self-organizing area coverage method for swarm robots based on gradient and grouping,” *Symmetry*, vol. 13, no. 4, p. 680, 2021.

[64] I. Khoufi, P. Minet, A. Laouiti, and S. Mahfoudh, “Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges,” *International journal of autonomous and adaptive communications systems*, vol. 10, no. 4, pp. 341–390, 2017.

[65] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 2. IEEE, 2003, pp. 1293–1303.

[66] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 500–505.

[67] J. Li, B. Zhang, L. Cui, and S. Chai, “An extended virtual force-based approach to distributed self-deployment in mobile sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 8, no. 3, p. 417307, 2012.

[68] J. Chen, S. Li, and Y. Sun, “Novel deployment schemes for mobile sensor networks,” *Sensors*, vol. 7, no. 11, pp. 2907–2919, 2007.

[69] N. A. B. Ab Aziz, A. W. Mohemmed, and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram," in *2009 international conference on networking, sensing and control*. IEEE, 2009, pp. 602–607.

[70] H. Tan, Y. Wang, X. Hao, Q.-S. Hua, and F. C. Lau, "Arbitrary obstacles constrained full coverage in wireless sensor networks," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2010, pp. 1–10.

[71] R. Stranders, A. Rogers, and N. Jennings, "A decentralised coordination algorithm for maximising sensor coverage in large sensor networks," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010, pp. 1165–1172.

[72] Y. Chen, H. Wang, F. Xiong, A. Li, Z. Qin, and J. Yuan, "Relay selection based on trajectory prediction for uav networks," in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2020, pp. 1212–1217.

[73] G. Valentini, E. Ferrante, and M. Dorigo, "The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives," *Frontiers in Robotics and AI*, vol. 4, p. 9, 2017.

[74] A. Almansoori, M. Alkilabi, and E. Tuci, "On the evolution of mechanisms for three-option collective decision-making in a swarm of simulated robots," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2023, pp. 4–12.

[75] G. Valentini, H. Hamann, and M. Dorigo, "Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1305–1314.

[76] Q. Shan and S. Mostaghim, "Collective decision making in swarm robotics with distributed bayesian hypothesis testing," in *International Conference on Swarm Intelligence*. Springer, 2020, pp. 55–67.

[77] G. Morlino, V. Trianni, and E. Tuci, "Evolution of collective perception in a group of autonomous robots," in *International Joint Conference on Computational Intelligence*. Springer, 2010, pp. 67–80.

[78] A. Almansoori, M. Alkilabi, J.-N. Colin, and E. Tuci, "On the evolution of mechanisms for collective decision making in a swarm of robots," in *Italian Workshop on Artificial Life and Evolutionary Computation*. Springer, 2021, pp. 109–120.

[79] A. Almansoori, M. Alkilabi, and E. Tuci, "A comparative study on decision making mechanisms in a simulated swarm of robots," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.

[80] J. Prasetyo, G. D. Masi, P. Ranjan, and E. Ferrante, "The best-of-n problem with dynamic site qualities: Achieving adaptability with stubborn individuals," in *International Conference on Swarm Intelligence*. Springer, 2018, pp. 239–251.

[81] J. Prasetyo, G. De Masi, and E. Ferrante, "Collective decision making in dynamic environments," *Swarm intelligence*, vol. 13, no. 3, pp. 217–243, 2019.

[82] S. Kernbach, D. Häbe, O. Kernbach, R. Thenius, G. Radspieler, T. Kimura, and T. Schmickl, "Adaptive collective decision-making in limited robot swarms without communication," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 35–55, 2013.

[83] T. Schmickl and H. Hamann, "Beeclust: A swarm algorithm derived from honeybees," *Bio-inspired computing and communication networks*, pp. 95–137, 2011.

[84] N. Mathews, A. L. Christensen, E. Ferrante, R. O'Grady, and M. Dorigo, "Establishing spatially targeted communication in a heterogeneous robot swarm." in *AAMAS*, 2010, pp. 939–946.

[85] N. Mathews, A. L. Christensen, R. O'Grady, and M. Dorigo, "Spatially targeted communication and self-assembly," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2678–2679.

[86] M. S. Talamali, A. Saha, J. A. Marshall, and A. Reina, "When less is more: Robot swarms adapt better to changes with constrained communication," *Science Robotics*, vol. 6, no. 56, p. eabf1416, 2021.

[87] A. F. Winfield and J. Nembrini, "Safety in numbers: fault-tolerance in robot swarms," *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[88] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle *et al.*, "Argos: a modular, multi-engine simulator for heterogeneous swarm robotics," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 5027–5034.

[89] M. D. Soorati, M. K. Heinrich, J. Ghofrani, P. Zahadat, and H. Hamann, "Photomorphogenesis for robot self-assembly: adaptivity, collective decision-making, and self-repair," *Bioinspiration & biomimetics*, vol. 14, no. 5, p. 056006, 2019.

[90] A. Paas, E. B. Coffey, G. Beltrame, and D. St-Onge, “Towards evaluating the impact of swarm robotic control strategy on operators’ cognitive load,” in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2022, pp. 217–223.

[91] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 321–328.

[92] A. Elfes, “Sonar-based real-world mapping and navigation,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[93] ——, “Using occupancy grids for mobile robot perception and navigation,” *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[94] R. Smith, P. Cheeseman, and M. Self, “A stochastic map for uncertain spatial relationships,” in *4th International Symposium on Robotic Research*. MIT Press Cambridge, 1987, pp. 467–474.

[95] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *The international journal of robotics research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[96] M. Kegeleirs, G. Grisetti, and M. Birattari, “Swarm slam: Challenges and perspectives,” *Frontiers in Robotics and AI*, vol. 8, p. 23, 2021.

[97] R. K. Ramachandran, Z. Kakish, and S. Berman, “Information correlated lévy walk exploration and distributed mapping using a swarm of robots,” *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1422–1441, 2020.

[98] J. Hörner, “Map-merging for multi-robot system,” Master’s thesis, Charles University, 2016.

[99] M. Kegeleirs, D. G. Ramos, and M. Birattari, “Random walk exploration for swarm mapping,” in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2019, pp. 211–222.

[100] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, “Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes,” in *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*. IEEE, 2008, pp. 109–120.

[101] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.

[102] W. Luo and K. Sycara, "Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6359–6364.

[103] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM transactions on graphics (TOG)*, vol. 1, no. 3, pp. 235–256, 1982.

[104] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[105] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," *arXiv preprint arXiv:1309.6835*, 2013.

[106] L. Esterle, "Centralised, decentralised, and self-organised coverage maximisation in smart camera networks," in *2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE, 2017, pp. 1–10.

[107] T. Stirling, S. Wischmann, and D. Floreano, "Energy-efficient indoor search by swarms of simulated flying robots without global information," *Swarm Intelligence*, vol. 4, pp. 117–143, 2010.

[108] D. Maftuleac, S. K. Lee, S. P. Fekete, A. K. Akash, A. López-Ortiz, and J. McLurkin, "Local policies for efficiently patrolling a triangulated region by a robot swarm," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1809–1815.

[109] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots." in *robotics: science and systems*, 2006, pp. 49–56.

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

[111] L. Zuo, W. Yan, and M. Yan, "Efficient coverage algorithm for mobile sensor network with unknown density function," *IET Control Theory & Applications*, vol. 11, no. 6, pp. 791–798, 2017.

[112] A. Sadeghi and S. L. Smith, "Coverage control for multiple event types with heterogeneous robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3377–3383.

[113] S. Kia and S. Martinez, "Multi-agent coverage control: From discrete assignments to continuous multi-agent distribution matching," *arXiv preprint arXiv:2407.13890*, 2024.

[114] C. Song, L. Liu, G. Feng, Y. Fan, and S. Xu, "Coverage control for heterogeneous mobile sensor networks with bounded position measurement errors," *Automatica*, vol. 120, p. 109118, 2020.

[115] K. Abbassi, M. H. Jeridi, and T. Ezzedine, "Wsn for event detection applications: Deployment, routing, and data mapping using ai," in *Wireless Sensor Networks-Design, Deployment and Applications*. IntechOpen, 2021.

[116] G. Rossides, B. Metcalfe, and A. Hunter, "Particle swarm optimization—an adaptation for the control of robotic swarms," *Robotics*, vol. 10, no. 2, p. 58, 2021.

[117] J. Pugh and A. Martinoli, "Multi-robot learning with particle swarm optimization," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 441–448.

[118] G. G. Rigatos, "Distributed gradient and particle swarm optimization for multi-robot motion planning," *Robotica*, vol. 26, no. 3, pp. 357–370, 2008.

[119] J. Pugh and A. Martinoli, "Distributed adaptation in multi-robot search using particle swarm optimization," in *International Conference on Simulation of Adaptive Behavior*. Springer, 2008, pp. 393–402.

[120] J. M. Hereford, "A distributed particle swarm optimization algorithm for swarm robotic applications," in *2006 IEEE International conference on evolutionary computation*. IEEE, 2006, pp. 1678–1685.

[121] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira, "A novel multi-robot exploration approach based on particle swarm optimization algorithms," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 327–332.

[122] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *Ieee Access*, vol. 10, pp. 10 031–10 061, 2022.

[123] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," *Archives of computational methods in engineering*, vol. 29, no. 5, pp. 2531–2561, 2022.

[124] F. Bordoni and A. D'Amico, "Noise in sensors," *Sensors and Actuators A: Physical*, vol. 21, no. 1-3, pp. 17–24, 1990.

[125] D. Albani, W. Hönig, D. Nardi, N. Ayanian, and V. Trianni, "Hierarchical task assignment and path finding with limited communication for robot swarms," *Applied Sciences*, vol. 11, no. 7, p. 3115, 2021.