Efficient Control of Drones Communications in IoT

---

by
Yannis Spyridis

EFFICIENT CONTROL OF DRONES COMMUNICATIONS IN IOT

Yannis Spyridis

A dissertation submitted to

The University of Sheffield

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

March 2024

Yannis Spyridis
Efficient Control of Drones Communications in IoT
PhD Dissertation, Wednesday 20th March, 2024
Supervisors: Dr. Thomas Lagkas
Prof. Jie Zhang
Prof. George Eleftherakis

The University of Sheffield
Department of Electronic and Electrical Engineering
Portobello Street
Sheffield, S1 3JD

Acknowledgements

First and foremost I would like to express my heartfelt thanks to all my supervisors and especially Dr. Thomas Lagkas, with whom it has been a pleasure to work. I am also sincerely grateful to Prof. Vasileios Argyriou for his continued interest and guidance. Lastly, I would like to thank my friends and family for their support and encouragement.

I, Yannis Spyridis, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This doctoral thesis explores the development of efficient drone control methods in the dynamic landscape of drone networks within the Internet of Things (IoT). As drones become increasingly integrated into the IoT ecosystem, addressing the complexities and challenges inherent in their coordination becomes paramount for ensuring reliability and efficiency.

The thesis starts with a thorough exploration of IoT concepts alongside drone networks, outlining key application domains and describing state-of-the-art solutions, particularly in localisation and tracking. Additionally, it examines advanced drone route planning strategies, highlighting the opportunities they present and the critical challenges they entail.

The main body of the thesis introduces novel cooperative algorithms, drawing from deterministic principles and artificial intelligence (AI) techniques. Inspired by natural phenomena such as flocking birds, these algorithms enable drones to collaboratively determine their routes for tracking mobile sensors within dynamic IoT environments. As the efficacy of these methods is demonstrated, it becomes apparent how they enhance drone cooperation and significantly improve tracking efficiency.

Building upon this foundation, the thesis next introduces an innovative deep reinforcement learning (DRL) scheme, empowering autonomous drone agents to efficiently develop optimal data collection strategies within an IoT network. By harnessing DRL, drones continually acquire insights from their environment and actions, adapting to changes and making intelligent decisions to optimise their data collection policies. The scheme adapts state-of-the-art algorithms to effectively scale to high-dimensional state-action spaces commonly encountered in real-world IoT applications.

This research contributes to the ongoing discourse surrounding drone-IoT integration, offering novel approaches to drone control. The introduction of these methods opens up new avenues for creating more efficient, and autonomous drone networks within the IoT paradigm, highlighting the untapped potential of AI in this context, and setting the stage for future development in the field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

The Internet of Things (IoT) is seen as one of the most rapidly advancing technologies in the area of computing and networking nowadays. It constitutes a breakthrough environment in which cutting-edge science is utilised to offer solutions towards the improvement of diverse human activities. The vision of the IoT aims at the seamless integration of several objects that act as smart devices with existing networks, allowing the embedment of information from diverse communication systems in the environment [1]. The distinct objectives and the specialised behaviour of such devices is captured by appropriate software inside the IoT infrastructure which allows their interconnection and global operation. Towards this vision, IoT devices are starting to become a thriving aspect of everyday life, with the number of uniquely identified "things" increasing at an exponential rate [2].

Unmanned aerial vehicles (UAVs), also referred to as drones, is a technology that has exhibited a lot of research interest and is expected to enter the grand scheme of the IoT, offering vast potential in public and civil applications. The most common definition of a drone is that of an engine-powered aerial vehicle which uses aerodynamic forces to provide lift and operates without a human pilot onboard. Drones can be categorised based on the technology they utilise into fixed-wing aircraft, single or multi copter rotorcraft, while their size can range from very small (nano/micro drones) to

(a) Quadcopter drone mid-flight. Image source: Unsplash. [6]
(b) Fixed-wing drone mid-flight. Image source: Quantum-Systems. [7]

Figure 1.1: Multicopter and fixed-wing drones.

large, often resembling small airplanes. According to the use case, they can be either controlled remotely or fly autonomously. Figure 1.1 illustrates an example of different types of drones.

The use of drones, started as part of military operations that included a single, large unmanned vehicle, communicating with a centralised ground controller as part of an unmanned aircraft system. The key issues in this architecture are the need for persistent connection with the control station and the base station as well as the provision of access functionality. More importantly, a possible failure in the drone of such a system, leads to an unsuccessful operation. Nowadays, the proliferation of sensor technology has enabled drones to act as sensing devices that are being utilised in cooperative aerial communication networks [3]. This allowed further widening in the capacity of potential applications. It has been showcased that the use of several small drones organised in formations called swarms, offers novel solutions in various areas, such as agriculture, environment inspection, disaster detection, search and rescue operations, infrastructure maintenance, and more [4].

For instance, in the domain of agriculture, specifically in the field of smart farming, the use of drones is considered a significant breakthrough technology [5]. In this context, drones are expected to greatly assist in a more efficient utilisation of the land, providing useful insight regarding the health of the plants with respect to the quality of production. In addition, drones can be utilised to monitor the farming area and identify the driest fields thus providing insight towards a more efficient water allocation, resulting in reduced waste.

In applications of security, environment monitoring and disaster detection, drone networks can be utilised to monitor a designated area, using their equipped sensors to detect abnormalities and immediately call for help. For instance, in search and rescue operations drones can use dedicated sensors to search for victims, so that a rescue party can be dispatched to their location. Forming a cooperative network, drones can share a common map and communicate the areas already monitored, effectively increasing the efficiency when searching [8].

Efficient control algorithms are critical for optimising the performance of drone swarms, especially in scenarios involving remote environments or drone-assisted IoT networks. Despite the rapid advancements in this domain, significant challenges persist in fully realising the potential of drones within IoT networks. These challenges encompass technical limitations and strategic considerations, which are crucial for the successful deployment and operation of drone networks.

One of the primary challenges is the management of drone routes in prolonged missions. Current battery technologies impose constraints on operational time, limiting the duration of flights. Therefore, optimising route planning becomes essential to efficiently utilise the available energy to successfully complete the mission. Addressing these challenges is important, as it enables the resolution of various real-world problems and also facilitates the seamless deployment of drone networks in remote environments. In addition, another significant obstacle in the efficient control of drone swarms lies in devising solutions that leverage simple algorithms and sensors onboard the drones. While advanced algorithms and sensor technologies exist, incorporating complex systems can lead to increased computational overhead and energy consumption, thereby limiting the scalability and practicality of drone swarm deployments. Therefore, developing streamlined algorithms that can effectively govern drone movements and data collection strategies using minimal computational resources is essential for realising the full potential of drone networks within IoT environments.

In light of these challenges, the domain of control algorithms for drone swarms represents a rich research area with several opportunities for exploration and advancement. Unraveling these challenges can lead to an assortment of refined applications that contribute to solving diverse problems across numerous fields. Moreover, overcoming these obstacles is instrumental in enabling the widespread adoption and effective utilisation of drone networks, thereby offering transformative benefits to society at large.

This thesis aspires to present sophisticated drone control methods to address challenges, with key focus in providing efficient navigation mechanisms inside drone networks in the context of IoT infrastructures. Along the journey towards this goal, a series of novel solutions uniquely tailored towards specific applications will be proposed and adaptive shemes will be introduced, designed with an inherent flexibility to adjust to real-world problems.

1.2 Aims and objectives

The main aim of this doctoral thesis is to innovate the sphere of drone control within IoT networks, specifically in the realm of efficient route planning. This aim can be categorised into three primary objectives.

- **Investigate and devise deterministic techniques for cooperative drone control:** The first objective is to delve into deterministic strategies of drone control. This involves an examination of structured methods based on rules, their performance and potential drawbacks. This aspect of the research aims to identify potential areas of improvement and set the stage for the application of more advanced methodologies. The key goal is to identify, understand, and evaluate the possibilities of enhancing deterministic control mechanisms with state-of-the-art techniques, such as AI.

- **Investigate and devise deep learning methods for swarm coordination and cluster formation:** The second objective is to investigate the application of deep learning methods to drone control, with a specific focus on cluster formation. This aspect of the research aims to understand how AI can contribute to more efficient and effective management of drone fleets, and to evaluate how it can facilitate the deterministic strategies. The formation of clusters is critical, as it can offer ways of enhancing the efficiency and effectiveness of mission execution.

- **Investigate and devise a deep reinforcement learning framework for optimising drone route planning:** The third objective focuses on the use of deep reinforcement learning for route planning optimisation. The key goal is to take drone control a step further by learning optimal policies towards achieving the mission goal and to continually adapt to changes in real-time. Focusing on the design of systems that can learn from their environment and conform to induced constraints, can help reduce the need for human intervention and increase the overall control efficiency.

Through the pursuit of these objectives, this thesis intends to offer a series of innovative solutions to the specific challenges encountered in the control of drones within IoT infrastructures. By introducing adaptive schemes with the inherent flexibility to adjust to real-world problems, the ultimate goal is to contribute to the evolution of efficient, intelligent drone control systems, bringing tangible benefits to a wide range of applications and sectors.

1.3 Outline of the thesis

Following in this thesis, there are five distinct chapters: a review of the associated literature explored by the research effort in the context of the thesis, three chapters - each dedicated to analysing and tackling one of the primary defined research objectives, and a chapter that discusses the findings, evaluates the achievement of goals and concludes the thesis.

Chapter 2 delves into the background of the work and is structured to provide an exhaustive analysis of the relevant research and literature that establishes the foundation for this thesis. The chapter begins with a comprehensive overview of the IoT paradigm, establishing its critical role in current technological landscapes. The focus then shifts to drone networks, discussing their distinct characteristics, operational applications with a focus on localisation and tracking methods, and the challenges and current techniques in optimisation of route planning. This forms a solid understanding of the capabilities and limitations of current drone networks. The latter part of the review examines the prospective role of AI in drone swarm management. It begins by assessing the application of machine learning in drone control before exploring how deep reinforcement learning techniques can be used to achieve efficient drone navigation.

Chapter 3, titled "Development of a new Deterministic Technique for Drone Control" involves the study of a deterministic method in the realm of cooperative drone control. This chapter sets the stage by introducing the foundation of a new deterministic technique, followed by an extensive examination of its application in the control of drones. It provides an in-depth analysis of how this technique is leveraged to enhance cooperation among drones in a mobile IoT sensor tracking application for search and rescue operations. Furthermore, it evaluates the strengths and weaknesses of the method, shedding light on potential challenges and areas for improvement. The findings of this investigation pave the way for the subsequent exploration of AI in drone control, setting up a comparative framework for different control strategies.

Chapter 4, titled "Advancing Drone Control: Deep Learning in Cluster Formation", signifies a transition from traditional deterministic techniques, delving into the application of deep learning methodologies within the scope of drone cluster formation and swarm coordination. In this chapter the design and implementation of a deep learning model that can facilitate the formation of drone clusters is introduced, highlighting its ability to create efficient and adaptable swarm formations, further improving the mobile IoT sensor tracking performance of the purely deterministic scheme. The exploration of deep learning leads into the next step of the research: the use of deep reinforcement learning for optimising the drone route planning.

Chapter 5, titled "Multi-Agent Drone Route Planning Optimisation" signifies the apex of this research journey, integrating the understanding gained from previous chapters to tackle a different more complex problem, that of optimising multi-agent drone route planning for efficient data collection in an IoT context. The chapter primarily focuses on the introduction of a novel deep reinforcement learning framework, demonstrating its capacity to manage the dynamic nature of multi-agent systems and optimise drone routes given multiple constraints. Detailed investigation and analysis reveals how the proposed framework can result in efficient and adaptable drone networks that are equipped to handle intricate real-world scenarios. This chapter signifies the importance of intelligent systems in drone route planning optimisation, but also illustrates their potential to greatly enhance the field of drone control within IoT infrastructures.

Finally, chapter 6 concludes the research with a review of key findings, their implications, and future prospects. It analyses research outcomes, acknowledges limitations, and suggests future research directions. It ends by highlighting the untapped potential in intelligent drone control optimisation, inspiring further innovation in the domain.

1.4 Scientific publications related to this thesis

International peer-reviewed journals

1. Y. Spyridis, T. Lagkas, P. Sarigiannidis, and J. Zhang, "Modelling and simulation of a new cooperative algorithm for UAV swarm coordination in mobile RF target tracking," *Simulation Modelling Practice and Theory*, vol. 107, p. 102232, 2021.

2. Y. Spyridis, T. Lagkas, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, G. Eleftherakis, and J. Zhang, "Towards 6G IoT: tracing mobile sensor nodes with deep learning clustering in UAV networks," *Sensors*, vol. 21, no. 11, p. 3936, 2021.

3. Y. Spyridis, V. Argyriou, T. Lagkas, J. Zhang, and P. Sarigiannidis, "Towards optimal multi-agent UAV route planning in IoT networks: a cooperative deep reinforcement learning solution," *IEEE Internet of Things Journal*, 2023. (under review)

International peer-reviewed conference proceedings

1. Y. Spyridis, T. Lagkas, P. Sarigiannidis, and J. Zhang, "Rule-based autonomous tracking of RF transmitter using a UAV swarm," in *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. IEEE, 2020, pp. 1–3.

Chapter 2

Background

2.1 Internet of Things

The core concept of the Internet of things revolves around the discrete identification of an assortment of smart objects inside a global network. The virtual behaviour and functionality of these objects has to be captured under an overlay network that communicates with the Internet itself, while their physical attributes need to be seamlessly integrated into this network [2]. Inside the Internet of Things, architectures are characterised by self organising capabilities and utilise correlative communication protocols for the device interconnection, based on standardised intelligent interfaces.

Viewed from an abstract perspective, the Internet of Things incites a vision of the Internet escaping the digital world, embracing reality and becoming an active part of everyday activities. "Things" inside the IoT are becoming dynamic participants in human processes, by accessing Internet services, communicating among each other through information exchange, and interacting with people. The proliferation of sensor technology has enabled smart objects to perceive their context, while reacting to data sensed from their environment, either autonomously or through human intervention [9].

Considering the variety of cases in which humans need to access remote data on a timely manner, or that inter-device interactions are required in the context of IoT, it is evident that deciding on a self-contained architecture, appropriate for all issues, is inherently challenging [10]. The very foundation on which the Internet of Things is based, inevitably leads to scalability issues, since the IoT scope exceeds that of a local environment [9]. Adjacent to this challenge comes the need for IoT infrastructures to support the immediate integration of new smart objects when needed and to assist in their self configuration inside the specific environment [1].

Interoperability is a vital characteristic of the Internet of Things and is seen as one of the most important concerns for the realisation of this paradigm. Smart devices demonstrate a pronounced variety, leading to issues regarding their unique visual representation inside the IoT. Evidently, these issues affect their portrayal as nodes inside a network, but also expand into the way their produced data is communicated and represented in the architecture of the system [9].

Besides the prominent issues mentioned above, the vision of an Internet of Things faces additional challenges that add to the complexity of undertaking this endeavour. Utilising appropriate technologies, the realisation of the IoT should account for security and privacy concerns, provide methods for the required autonomy features [11], take into consideration energy constraints and allow staightforward service discovery mechanisms [12].

In the direction of this evergrowing vision of an Internet of Things, new classes of smart objects are finding their way inside the grand architecture. The insertion of drones in this paradigm as a special type of aerial devices, allows the expansion of the idea and the envisioning of the Internet of Flying-Things [13]. Consequently, there is a growing demand for novel approaches that will enable the seamless integration of drones in the Internet of Things, conceptualising demanding networking techniques, but paving the way for a significant amount of diverse IoT applications in the process [14].

2.2 Unmanned aerial vehicle networks

Research in the realm of swarm coordination in the context of drone networks has proven to be a daunting task, bringing forth numerous issues that need addressing, before such networks can be effectively operationalised. At first glance, conventional frameworks utilised in mobile ad-hoc networks (MANETs) and vehicular ad-hoc networks (VANETs) may appear suitable, however, research efforts in that direction do not fully contend with some prominent aspects of drone behaviour [15]. Thus, a fresh perspective is required, leading to the advent of the flying ad-hoc network (FANET). These networks aim at a more effective intercommunication between the drones, support schemes for the connection among multiple drone systems and facilitate the control of the autonomous movement of drones [16]. The distinctive aspects of FANETs and their dissimilarities with traditional mobile and vehicular networks are discussed below.

2.2.1 Distinct characteristics

Most mobility models employed to characterise the behaviour of nodes in MANTEs or VANETs are largely two-dimensional. The former are typically governed by the "random walk" model [17], while the latter are commonly associated with the "Manhattan" model [18]. However, in many instances, nodes within a drone network may navigate in three dimensions, while the alteration in their position might occur rapidly. Drones often boast velocities reaching up to 100 m/sec, which vastly outpaces the typically low speeds observed in conventional mobile networks. As a result, the direct application of mobility models from previous paradigms to FANETs proves inadequate, demanding substantial modification and adaptations [19].

Network topology also serves as a notable distinguishing element separating FANETs from traditional mobile networks. In a network of interlinked drones, the relative positions of nodes may undergo frequent changes, and thus the topology experiences recurring alterations. This is a common issue in drone networks, as nodes periodically need to disconnect from the network for recharging or due to fault occurrences. When this occurs, they are replaced by new nodes, leading to the continual creation and dissolution of unique network links. In essence, FANET topology is marked by its highly dynamic nature, as drones unpredictably join or disconnect from the network. The literature suggests that a mesh topology may often be suitable among the drones, while a star topology can be effectively used between the drones and the base station [20].

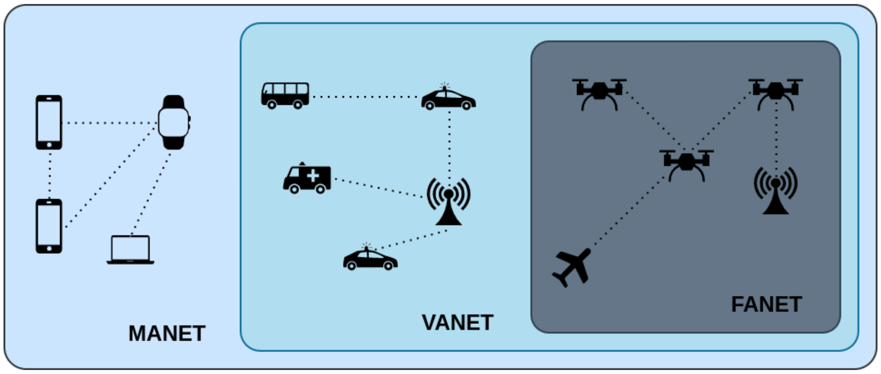


Figure 2.1: Mobile, vehicular and flying ad-hoc network.

The specific application of the deployed drone network significantly influences the network topology. For example, in surveillance or agriculture contexts, the topology tends to remain largely stationary, as drones need to hover over a designated area. However, in sensing applications like search and rescue missions or disaster detection scenarios, the network's topology undergoes rapid changes. There are various use cases where node density is sparse, making network partitioning a frequent occurrence. In such instances, it is crucial for the network to have self-reconfiguring capabilities [21]. This poses substantial challenges in smoothly transferring provided services to a functioning drone and maintaining consistent network communications.

Perhaps one of the most defining characteristic that distinguishes FANETs from conventional ad-hoc networks is their limited energy availability. Drones are battery constrained, and the flight time of small UAVs is typically limited. This is in contrast to nodes in mobile and vehicular networks, which are equipped with energy sources that can last considerably longer or be recharged during operation. Therefore, enhancing energy efficiency is paramount in drone networks, and it is crucial to develop efficient control methods to optimise power usage within the swarm [20]. For instance, efficient path planning is a pivotal aspect that can significantly enhance network longevity and minimise energy consumption in drone networks. By optimising travel routes, the distance and complexity of maneuvers drones need to undertake are effectively reduced, leading to lower energy expenditure [22]. Furthermore, well-planned paths contribute to the balanced energy usage across all drones, preventing overuse of individual units and thus prolonging the collective operational

Figure 2.2: Drone equipped with camera in a forestry application. Image source: Unsplash. [24]

time of the network.

2.2.2 Operational applications

Applications of drone networks span across diverse domains, including agriculture, disaster management, surveillance, and environmental monitoring. The interplay of multiple drones in a networked ecosystem allows for enhanced operational capabilities, efficiency, and robustness, presenting novel opportunities for exploration and innovation. For instance, research in the domain of forestry has recently experienced a growing remote sensing initiative, with the incorporation of drones in sensing systems providing a versatile and appealing solution, while at the same time reducing costs. In essence, deployed drone networks can offer benefits in a diversity of forest applications, including canopy height estimation, monitoring wildfires, and sustaining forest management. Existing approaches demonstrate high potential, however extensive research is still required to identify proper methodologies that can prove effective in the diverse conditions encountered in forests, and that satisfy each unique case [23]. An example of a camera equipped drone in a forestry application is shown in Figure 2.2.

A noteworthy strategy within UAV-assisted forestry involves the fusion of drone photogrammetry and structure from motion methodologies [25,26]. The approach is examined in [27] which harnesses an open-source photogrammetric toolbox to model the canopy surface of deciduous trees. Imagery collected from a fixed-wing drone is integrated with a LiDAR model, with the resultant hybrid system utilised to determine forest elevation. The findings from this approach have shown to be similar with existing data and measurements acquired from manned LiDAR aircraft.

The concept of using drones for fire discovery as well as real-time wildfire monitoring exists at the core of the drone application domain, and the immense potential it can offer has been perceived since the early stages of this notion. In contrast to traditional methods that include the use of human-piloted aircraft, small drones constitute a more affordable alternative that can prove essentially useful in hostile terrains, by avoiding possible risks a pilot may face. Detection and tracking of a forest fire can be achieved using video sequences obtained from drones [28], as well as through infrared sensors by performing histogram segmentation and optical flow deconstruction [29]. Once a fire has been detected, a group of drones can identify the optimal path to each fire spot by using heuristics [30] or other methods.

Remote sensing has proven to be highly valuable in precision agriculture applications as well, offering a wide range of benefits. UAV-assisted networks are extensively utilised to aid in various agronomic use cases, including crop area measurement, yield assessment, detection of infestations or weather-related damage, and soil condition inspection [31]. These applications commonly involve equipping drones with multi-spectral visual sensors and employing autonomous monitoring techniques. This approach is showcased in the 3D imaging system presented in [32] which effectively extracts essential data towards the enhancement of crop management. Similarly, drones can be employed for livestock monitoring, as demonstrated in [33]. The described approach utilises an artificial neural network combined with a clustering algorithm to analyse remote images captured by a drone, exhibiting high accuracy in livestock counting, providing reliable results even in moderately crowded areas. Overall, the integration of drone remote sensing technologies with precision agriculture has led to the development of advanced analytics tools that can process relevant data to provide actionable insights. These tools are crucial in optimising irrigation schedules, identifying nutrient deficiencies, and enhancing the overall productivity of agricultural operations.

2.2.3 Localisation and tracking

In the context of this thesis, the area of mobile target tracking through drone networks holds high relevance, closely aligning with the conducted research. Over the years, this field has attracted significant interest, with numerous approaches proposed in the literature. In particular, the deployment of drone swarms equipped with diverse sensors presents opportunities for innovative solutions in localisation applications. While robust localisation techniques typically rely on visual sensors for effective detection and tracking of moving objects, these methods often require real-time execution of energy-intensive image recognition algorithms, thus posing challenges to the crucial aspect of energy preservation in UAV-assisted networks. Therefore, there is a pressing need to develop efficient algorithms that leverage alternative sensor types with lower energy demands [34].

In the field of localisation, triangulation has long been a popular and widely adopted method, relying on calculated angles between known locations and incorporating distance estimates from the target to establish its position through the formation of triangles. Trilateration, a similar approach, relies on distance measurements using the intersection of formed circles to determine the target's location. A network of drones equipped with electronic surveillance sensors that provide Received Signal Strength Indication (RSSI) of a radio frequency (RF) emitter may be deployed to calculate the distance from it based on a log-normal shadowing model. A fusion center can then collect this information from the drones and utilise trilateration to calculate the position of the RF source [35]. Such methods are effective, particularly when the target emits at high frequencies, as simulation analyses indicate.

Extensive research has been devoted to utilising optimal control techniques for localisation and tracking problems, leading to the proposal of various control schemes in recent literature. In this field, the Kalman filter [36] is a powerful mathematical tool used to estimate the state of a system based on noisy measurements. At its core, the Kalman filter is based on a recursive estimation algorithm that maintains an estimate of the system state by iteratively incorporating new measurements and updating the state estimate. This estimation process leverages two fundamental principles: the system's dynamics model and the measurement model. By combining these models

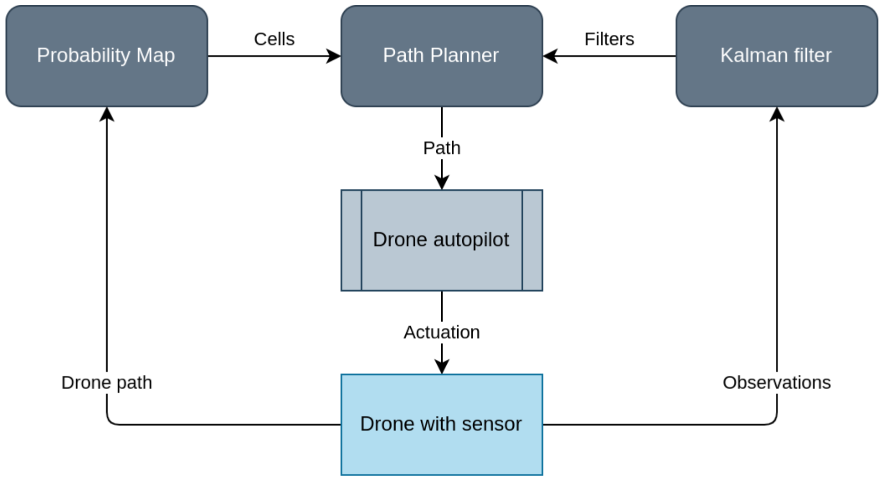


Figure 2.3: Diagram illustrating the control process, where the probability map and Kalman filter are independently built and subsequently merged within a path planner. [37]

with statistical assumptions, the Kalman filter provides a principled framework to compute the optimal state estimate and its associated uncertainty.

An optimal control localisation algorithm which relies on Kalman filters is described in [37]. The proposed framework consists of a two-layer structure that incorporates two observers in conjunction with a path planning algorithm. The first observer utilises Kalman filters to estimate the states of the objects by integrating observations of objects with simple velocity models. The second observer estimates the state of the open area by employing a probability map, which records the drone's movement within the open area and determines the probable locations of new objects. For drone path planning, it is assumed that an autopilot system is available, enabling the generation of a path instead of directly considering actuator inputs. The path planning algorithm integrates combinatorial optimisation and continuous optimal control techniques to prioritise between exploring uncharted areas and tracking known objects. The control architecture is illustrated in Figure 2.3.

The validation of the algorithm is conducted through simulation, comparing it to multiple base cases and a case where perfect knowledge of object positions is assumed. Monte Carlo simulations are executed with an open area defined by its X- and Y-coordinates, a specified number of objects following a predetermined equation, and a drone with a limited field of view tasked with searching and tracking the objects. The simulations involve varying two parameters: the number of objects and the size of the area. The results of the simulations demonstrate that the algorithm exhibits significantly superior performance compared to the base cases, showcasing an improvement of approximately 5-15%. However, it performs around 20-25% worse than the case where perfect knowledge of object positions is available.

A motion planning algorithm is also presented in [38], where a group of drones cooperatively track a target by optimising their intercommunication and sensing with a remote base station. The described scheme effectively integrates these objectives to accomplish the target tracking mission while ensuring appropriate connectivity with the remote base station. The algorithm takes into account realistic communication environments to generate trajectories for information-gathering by the multi-UAV system. The performance of target fusion and estimation relies not only on the overall information contained in the UAVs' measurement data sets but also on the successful transmission of measurement packets to the base station. Based on this concept, a co-optimised communication and sensing scheme is devised to enhance the overall performance of the system.

The sensing scheme is developed based on a framework of distributed receding horizon optimisation. At each time step, the planned control inputs and trajectories are received by each UAV from the other drones. Subsequently, based on the estimates of the target state and future state predictions of the other drones, each drone optimises its own control inputs and plans its trajectory. The co-optimisation of communication and sensing manifests through the balance between communication reliability and sensing utility, with the objective of maximising the overall information gain transmitted to the base station. Thus, the trajectory planning process of the algorithm incorporates both communication and sensing objectives to achieve an optimal performance in target tracking.

Noteworthy methods in this domain also incorporate model predictive control (MPC) algorithms which allow swarms of drones to cooperatively localise an RF source in a decentralised manner [39]. The algorithm can be executed by the local onboard computers of each drone, structured in a double-layered architecture that ensures software portability. It operates by collecting localised information from neighboring drones and adjusting its velocity accordingly. The algorithm is engineered to attain both a convergence rate and a steady-state regularity of the flock. The ultimate configuration of the flock is determined by the combination of the communication range and the desired inter-drone distance. The utilisation of a decentralised system for multi-drone coordination offers several advantages, including reduced communication complexity and the emulation of anonymity among the drones. The communication process is simplified as broadcasting does not necessitate peer-to-peer handshaking. The interaction range of each drone is determined by the broadcast radius, which establishes a localised sensing area. Moreover, the system maintains anonymity, with all calculations and controls being executed on an onboard minicomputer of each drone. This design enhances security and reduces vulnerability to external interference.

Similarly, the study in [40] adopts a receding horizon control (RHC) strategy for multiple target localisation. The proposed approach, rooted in hybrid system theory, employs a nonlinear MPC algorithm to facilitate real-time motion planning that optimises ergodicity concerning a distribution characterised by the expected information density across the sensing domain. The algorithm's objective is to achieve real-time coverage and target localisation, adapting its actions based on sensor feedback to continually update the expected information density. By focusing on tracking a non-parameterised information distribution across the terrain instead of individual targets in isolation, the approach is fully decoupled from the estimation process and the number of targets. Furthermore, the algorithm is designed to be distributable across multiple agents, allowing each agent to independently compute its own control while sharing coverage statistics across a communication network. The overview of the ergodic control process is illustrated in Figure 2.4.

The findings indicate that this approach has the potential to effectively handle the challenges associated with coverage, search, and target localisation in a comprehensive manner. By directing the robots to track a non-parameterised information distribution across the terrain rather than individual targets in isolation, the approach achieves complete decoupling from both the estimation process and the number of targets. The methodology is validated through simulations and experiments specifically focused on target localisation, demonstrating its capability to operate independently of the number of targets being tracked.

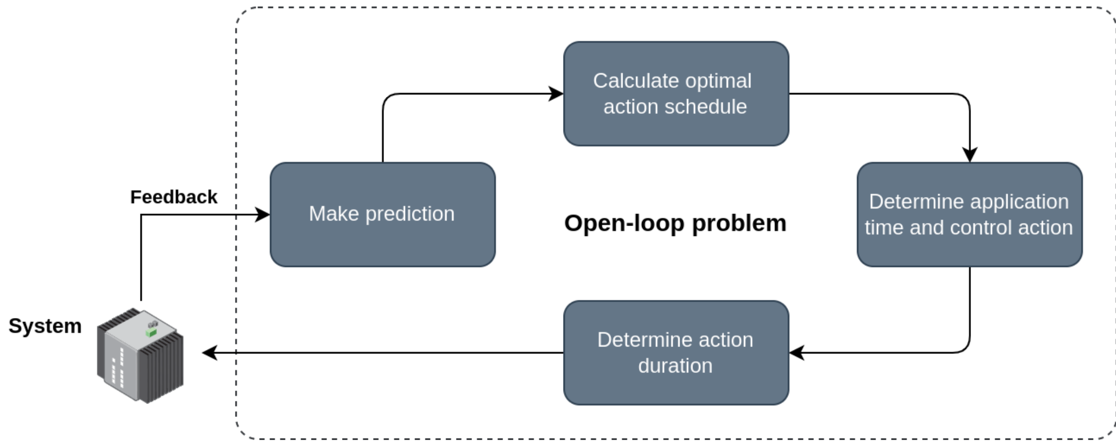


Figure 2.4: Ergodic control flowchart. [40]

The study presented in [41] describes a hierarchical MPC algorithm, which is a modified version of the MPC approach, utilised to streamline the computation of a suboptimal control sequence for addressing the drone control problem. In its basic form, the MPC algorithm dynamically solves an open-loop optimal control problem over a finite horizon, utilising the system's state at a given time as the initial state and the available information at that instant. Building upon this foundation, the hierarchical variant of MPC further divides the problem into a sequence of sub-problems, each characterised by its own time horizon and control objectives. This division facilitates the calculation of a suboptimal control sequence, thereby simplifying the management of the drones.

The study presented in [42] examines a context where the RF source transmits signals intermittently using fixed-wing drones equipped with angle-of-arrival (AOA) sensors. The described system relies in a decentralised architecture whereby each drone determines its path cooperatively based on the information received by neighbouring drones and on the data provided by its sensor. Control actions are taken based on a cost function regarding the distance that each drone needs to cover, the number of active drones assisting in the current target localisation, and the number of adjacent drones. When the cost function goes below zero, the corresponding drone proceeds to assist in the localisation, otherwise it retains its current actions. To accommodate for the intermittent transmission, the algorithm integrates the information of multiple AOA sensors over time. The study investigates the performance of three localisation techniques: triangulation, angle-rate, and

Kalman filtering using simulation experiments and concludes that a combination of these can yield a generally accurate localisation method.

Employing measurements of the Doppler frequency in the received signal, the method in [43] manages to lead a drone to a distant RF emitter, by continuously adjusting the trajectory using bearing estimates. At a first stage the drone follows a random circular trajectory, while storing frequency and bearing measurements. In order to filter the stored values and minimise effects of multipath fading, an outlier rejection technique is applied to the frequency measurements. At the second stage, the drone derives changes in Doppler at regular intervals for a persistent trajectory control that leads it towards the RF source. The study demonstrates the increased performance of this method through simulation experiments; however the required receiver operations have an increased complexity, compared to simply extracting the RSSI.

The issue of frequency drifts over time is effectively tackled by calculating the difference between two sets of frequency measurements. These measurements are obtained by perturbing the bearing around the current trajectory. By adopting this approach, the algorithm succeeds in substantially mitigating the impact of carrier frequency offset and drift. It is worth noting that these offsets and drifts change at a much slower rate compared to the duration of the iteration step. Moreover, the degree of perturbation can be heightened to enhance resilience against measurement noise, albeit resulting in increased travel distance as a trade-off. The algorithm demonstrates convergence towards the proximity of the emitter, whereby the net distance traversed is only slightly larger (approximately 10%) than the initial line-of-sight distance between the UAV and the emitter. Through simulation results, an illustrative UAV trajectory and the corresponding estimated frequency are presented, taking into account the presence of multipath, carrier frequency offset, and frequency drift for that specific trajectory. The received signal power profile along the trajectory, as well as the spatial variations in received power, are also depicted. Lastly, it is shown that this method effectively addresses various technical challenges, including unknown carrier frequency offset, frequency drifts, direction ambiguity, and noise in observations.

Relying on DOA, AOA, or similar sensors is not always convenient, especially when involving small multi-rotor drones. As a result, simpler alternative methods such as passive RSSI sensors, are often preferred. In this context, an RHC algorithm governing the path planning of an RSSI sensor-equipped drone swarm, is described in [44], towards the goal of localising a mobile RF transmitter. At the core of the algorithm, a predictive model for the Fisher information matrix (FIM) is employed to estimate the potential FIM values for the set of potential courses of action. This predictive model serves as the basis for formulating the optimal course of action through local optimisation, enabling the system as a whole to pursue the goal within a finite receding horizon. The state estimation of the target and the FIM optimisation around that point are carried out using an Extended Kalman Filter (EKF). The linear state transition of the target drone state vector simplifies the prediction stage of the EKF. By utilising the EKF estimator, the target location estimation is updated and predictions are made based on the most recent data available.

The application of localising intermittently transmitting mobile RF sources is also examined in [45], leading to a scenario where the RSSI measurements are not continuously available. The employed system model involves a group of drones equipped with omnidirectional RSSI sensors. These sensors receive intermittent, omnidirectional transmissions from a target that exhibits stochastic movement. A general stochastic linear time-invariant model is utilised to describe the target's movement, which encompasses various stochastic movement models that can be represented as linear models. The received power at each RSSI sensor is modeled, taking into account Friis channel attenuation, log-normal fading, and the distance from the target. The measurement model is completed by adding thermal noise to the received power, resulting in an accurate representation of the final measurement. The measurement model characterises the RSSI measurements as unbiased noise added to the expected value of the measurement. Expanding on this model, the FIM for the next estimation cycle is derived, and steepest descent path planning is formulated based on the determinant of the FIM. An EKF is utilised, which incorporates a minimum risk detector to determine if the target was transmitting during the current time step, enabling the estimation to be updated accordingly. The flowchart of the model is illustrated in Figure 2.5.

In the examined scenario, the drones are tasked to patrol a designated area of interest and

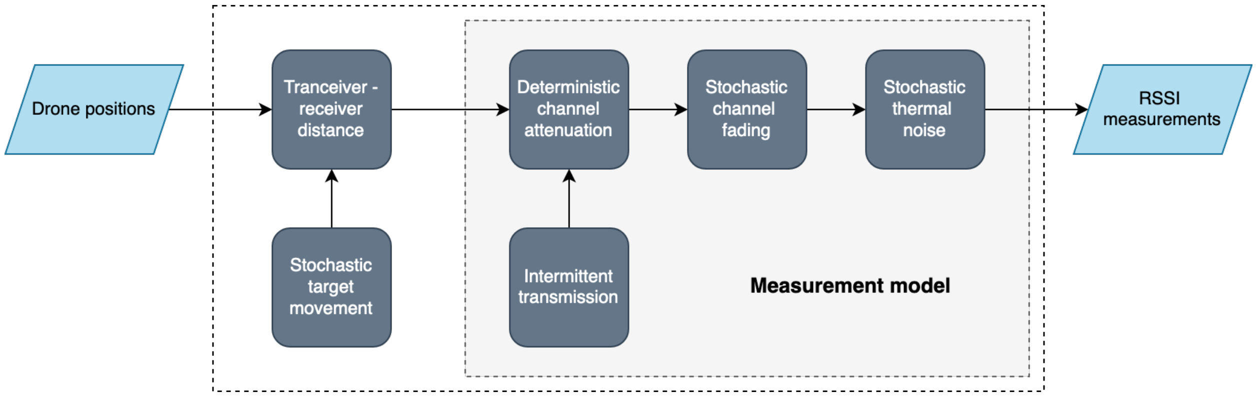


Figure 2.5: RSSI model based on stochastic and deterministic submodels. [45]

pinpoint the sporadic RF source via a two-stage technique. The first phase involves the task of localisation, where the drones estimate the target location, given the previous RSSI measurements, based on a recursive Bayesian estimator. In the second phase, the optimal future trajectory is determined, with the goal of reducing the localisation error by taking into consideration the current estimation, using a steepest descent path planning algorithm. Findings from simulations suggest that the recursive Bayesian estimator slightly outperforms an EKF in the first phase, while the steepest descent algorithm displays major benefits in the second stage, when compared to a bioinspired approach.

The current domain of research in drone localisation methods reveals several notable research gaps that require further exploration. Related to these gaps is the fact that most methods require complex or multiple antennas for direction of arrival, Doppler frequency, or angle of arrival measurements. While effective, such approaches may face challenges in practical implementation for small drones due to size and weight constraints. Additionally, these antenna configurations can introduce complexity in terms of system integration and may not be ideal for resource-constrained platforms. Finally, such antennas might be more susceptible to faults or accumulate errors over time. Contrary approaches utilise RSSI sensors which deal with most of these challenges. However, most existing techniques involve the conversion of RSSI to distance specifically using signal propagation models, which can be imprecise in dynamic environments.

In addition, many prevailing strategies rely on sophisticated algorithms that demand extensive computational resources, often surpassing the capabilities of onboard processing units of drones. This creates a critical bottleneck in real-time applications where swift and efficient decision-making is imperative. Therefore, there is need for methods that rely on simple sensors which are lightweight and small such as RSSI antennas, and for the development of algorithms capable of deriving accurate solutions with minimal computational overhead at the same time.

Furthermore, while deterministic methods have shown promise in addressing certain challenges in drone localisation, they often rely on predefined rules or models, which may not fully capture the complexity of real-world environments. In contrast, AI approaches offer the potential to adapt and learn from data, thereby enhancing the robustness and flexibility of localisation algorithms.

Addressing these research gaps is essential for advancing the field of localisation and tracking using drones, ensuring robustness, efficiency, and feasibility for real-world applications. The research effort of this thesis aims to develop innovative approaches that mitigate these limitations, enabling the development of accurate and energy-efficient drone localisation techniques in the IoT context.

2.2.4 Optimisation in route planning

The problem of multi-UAV route planning is a complex and compelling area of research, which also holds high relevance to the research outcomes of this thesis. It involves the optimisation of paths for a fleet of drones to achieve a set of objectives efficiently and effectively. The problem falls within the broader context of combinatorial optimisation problems, as it requires a solution among a discrete set of possibilities. Being closely related to graph theory, it encapsulates several challenges. Primarily, it includes the optimal determination of efficient paths for individual drones, taking into account their capabilities, current status, and overall mission objectives.

The complexity of this problem scales exponentially with the number of drones and tasks, making it a non-deterministic polynomial-time hard (NP-hard) problem. Hence, traditional algorithms like brute force or exact methods become computationally infeasible for larger instances. From

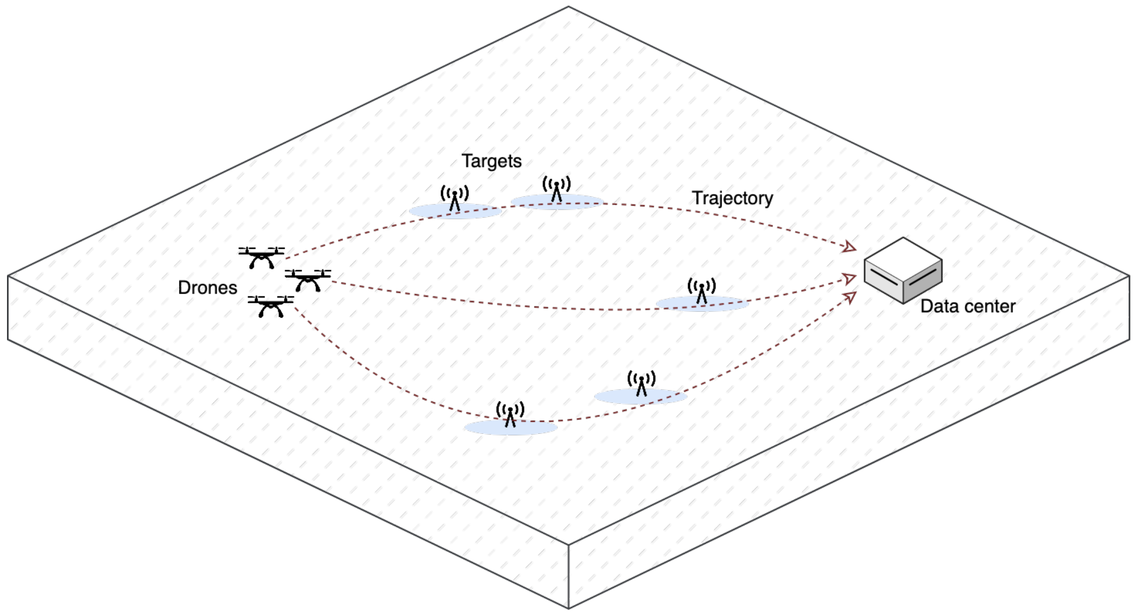


Figure 2.6: Multi-UAV route planning problem. The solution involves optimal target assignment and path planning.

a combinatorial optimisation perspective, the multi-UAV route planning problem is considered a variant of the multiple travelling salesman problem (m-TSP), in which the agents must obey additional constraints. While searching for the shortest path to visit the desired set of locations, the drones need to consider the limited flight time, obstacles, or onboard storage when designing the optimal path plan [46]. Over the years several approaches have been devised to provide solutions to such problems in this context.

Taking inspiration from biological systems found in nature, many methods involve the use of swarm intelligence to govern the collective behavior of agents [47]. These techniques typically involve the design of meta-heuristics to search large decision spaces, without specific assumptions about the optimisation problem at hand. To achieve the decentralised control of multiple drones, the study in [48] describes a particle swarm optimisation (PSO) algorithm using a receding horizon approach. The problem complexity is reduced through discretisation, with focus given on collision avoidance by adding dynamic safe distance constraints. The PSO algorithm, a population-based optimisation method, is utilised to optimise the performance index of the RHC framework, by mimicking the social behaviour of bird flocking or fish schooling. It iteratively updates the position and velocity of each particle in the swarm based on its own experience and the experience of neighbouring particles to find the optimal solution. Ultimately, the described method is specifically designed to enable real-time generation of optimal trajectories for multiple drones, while simultaneously ensuring collision avoidance.

A "comprehensively improved" PSO (CIPSO) method is used in [49] by adaptively tuning the parameters to enhance the convergence speed of the algorithm when designing the optimal formation for the drone path planning. The rapidity and optimality of automatic path planning are enhanced by the described algorithm through several means. Firstly, the particle initial distribution is enhanced by adopting a chaos-based logistic map. Secondly, the commonly used constant acceleration coefficients and maximum velocity are replaced with adaptive linear-varying ones, which can adjust to the optimisation process and improve the optimality of solutions. Lastly, a mutation strategy is introduced to replace undesired particles with desired ones, thereby accelerating the convergence speed of the algorithm. The comprehensive improvements made to the PSO, result in accelerated convergence and improved solution optimality, as evidenced by conducted simulations. In particular, the improved PSO method was evaluated through Monte-Carlo simulations for varying drone formation scenarios. The simulation results were compared with those of other PSO algorithms, and the described algorithm demonstrated higher performance in convergence speed, solution optimality, success rate, and procedure running time. The outcomes regarding the formation path failure rate and the improved percentage of different methods compared to standard PSO, are listed in Table 2.1.

Meta-heuristics such as adaptive genetic algorithms (GA), have also been successfully employed

Table 2.1: Comparison of PSO methods in formation path failure rate and improvement over standard PSO. [49]

Metric	SPSO	LCPSO	LVPSO	CBPSO	PMPSO	CIPSO
Failure rate	5.6%	4.2%	2.2%	3.2%	2.8%	0.8%
Improvement	–	25%	67.71%	42.86%	50%	85.71%

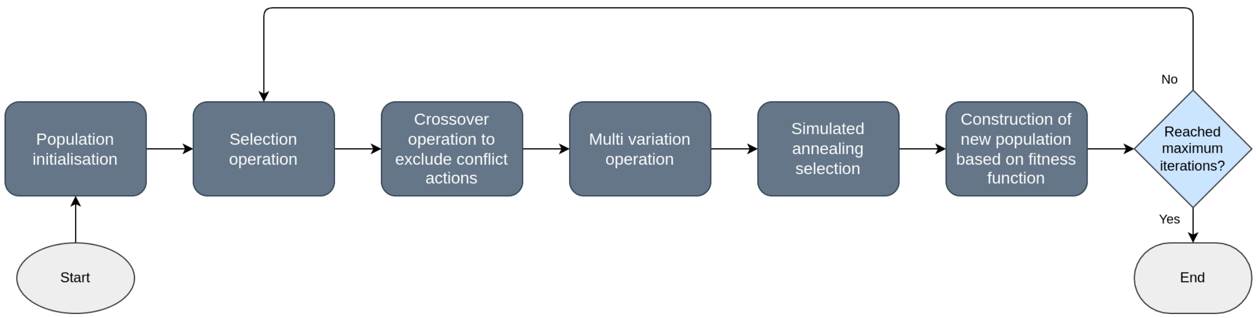


Figure 2.7: Flowchart of the genetic algorithm based on simulating annealing. [51]

in cooperative task assignment inside multi-UAV systems [50] to accelerate the search for good solutions. The efficiency of GA can be improved by using a second selection operation based on simulated annealing (SA) [51], which relies on the path length to reduce the loss in the evaluation function. In the described method, termed ISAFGA and illustrated in Figure 2.7 the acceptance criteria of the SA solution are refined, facilitating the exploration beyond local optima, and a threshold is introduced to determine the suitability of accepting the new solution. Furthermore, the encoding of the drone task sequence and the modification of the GA selection operation through the double selection process contribute to improved performance. The study argues that the integration of the enhanced SA method in the mutation and subsequent selection operations increases the diversity of individuals and significantly enhances the algorithm's efficiency in determining the task allocation order.

Further improvements in GA algorithms can also include population enhancement methods by generating alternative paths through ant colony optimisation [52]. In this method, the main focus lies in improving the convergence process by providing a good initial population that drives the population to optimality, as this choice determines how efficiently the optimisation algorithm converges to a local or global optimum. This is achieved through the utilisation of an ant colony optimiser, Voronoi diagrams, and clustering methods. The described enhancements in the initial population include both random and controlled diversity. After evaluating this technique in checkpoint-based drone path planning problems in varying environments, such as rural, urban, and spatial type terrain models, the study argues that this approach provides effective paths for a drone, while demonstrating a reduction in computational time. However, the comparison of compute times with other methods is not clearly specified, leaving uncertainty regarding the specific methods used for benchmarking.

2.3 Artificial intelligence in UAV swarns

Artificial Intelligence in drone control involves the utilisation of machine learning algorithms and other AI methods to guide the navigation and actions of drones autonomously, negating the necessity for human interference, even in terms of pre-programming rules for the said control. Equipped with AI, drones can process environmental data collected through different sensors and cameras, formulating decisions pertaining to their flight path and mission objective realisation. Tasks traditionally deemed complex and hazardous for humans, such as infrastructure inspection, surveillance, or search and rescue in remote zones, can be accomplished efficiently and securely by AI-integrated drones.

When compared to heuristic or alternative traditional approaches, AI offers several unique advantages. The key difference lies in its ability to learn from experience and adapt the behaviour of the drone according to accumulated data, gradually enhancing its performance. This characteristic also facilitates the effective operation in dynamic and unpredictable environments, demonstrating robustness in challenging situations by being able to manage uncertain or noisy data. In addition to the above, AI can be more effective when handling high-dimensional data and solving intricate problems, unlike conventional algorithms that often struggle with complicated, multidimensional decision-making scenarios. For this reason it is usually the optimal choice for tasks such as path planning or cooperative control of multiple drones.

2.3.1 Machine learning in drone control

Building on the previously mentioned characteristics of AI in drone swarms, several studies have examined the feasibility and effectiveness of ML in managing drone control. The work in [53] explores the control of a drone swarm in GPS-denied environments, based on a dead reckoning

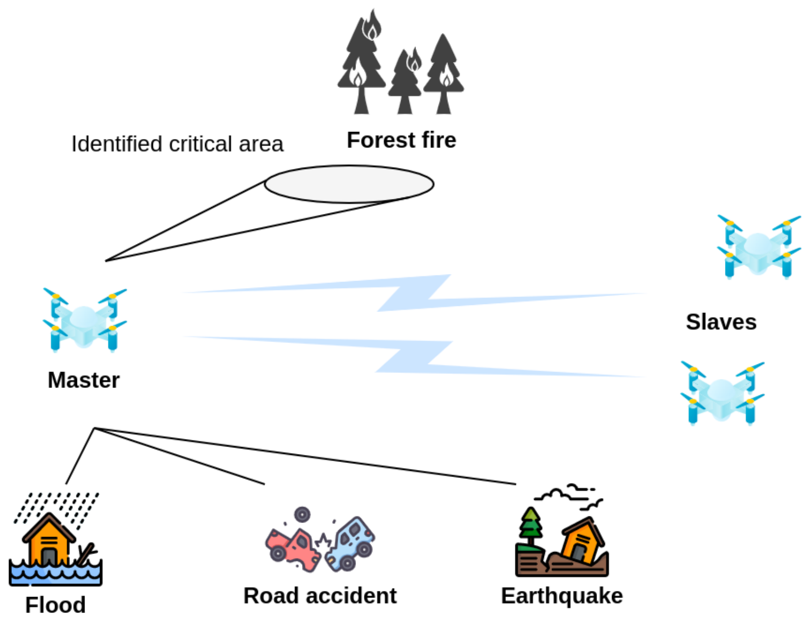


Figure 2.8: Master-slave drone configuration for emergency response. [54]

approach. To address the challenge of environmental deviations that introduce errors, the study incorporates an ML model trained on a spatio-temporal dataset, derived from the drone's location history and swarm network structure, to predict and correct for these disturbances. The evaluation of the model demonstrates the effectiveness of the method and the enhancements it offers compared to structure-based techniques.

Similarly, the study in [54] examines the effectiveness of search and rescue activities during emergencies, considering a drone swarm as a candidate for the emergency communication network. The employed swarm operates in a self-organised manner and relies on an ML model for the drone communications based on the path-loss profile. The work uses a 2D swarm control model that considers the drone velocities, to generate a dataset based on multiple triangular swarm formation techniques and applies K-means clustering to predict the swarm cluster formation. In addition, the study investigates the prediction of RSS based on the drones' reallocation in the swarm. After the training process generates the path-loss profile, the prediction of received signal strength is utilised to determine the reallocation of drones, and a dataset is generated for clustering using a triangular-based swarm formation concept.

The described framework relies on a master-slave configuration, in which the leading drone is the master, while two following drones are considered slaves, as illustrated in Figure 2.8. In the master–slave network, there is a need for a radar sensor on each element of the unit swarm. Every master drone of the unit swarm has the ability to communicate with its two subsequent drones. Consequently, a sub-cluster is formed by joining multiple unit swarms and the expansion of this sub-cluster is the formed swarm. The study demonstrates a strong agreement between the proposed swarms and swarm distances, indicating the effectiveness of this method in predicting received signal strength and power loss for drone swarms, signifying its applicability for search and rescue operations.

Leveraging the periodic probe requests broadcasted by Wi-Fi devices, the study presented in [55] outlines a strategy for inferring user locations during search and rescue missions. The employed method involves a drone extracting the RSSI and physical address of a Wi-Fi device from the transmitted probe requests, while traversing through specific, GPS-identified locations. Then, an ML algorithm based on the random forest model is deployed to classify the device's position into one of several pre-determined location zones. Despite the decent accuracy achieved in designating the correct zone, the associated geographical region must be pre-defined and segmented into suitable zones to appropriately train the algorithm in advance through this method.

2.3.2 Deep reinforcement learning for efficient drone navigation

Deep reinforcement learning (DRL) is a subfield of machine learning that combines deep learning (DL) and reinforcement learning (RL). In contrast to traditional ML, which often requires explicit labeling of data for supervised learning, or deals with pre-existing data in unsupervised learning, DRL agents operate by interacting with an environment to learn optimal actions through trial-and-error, thus learning from the consequences of their actions. The core premise of RL is the concept of "reward", by which the agent makes decisions within an environment, based on its current state,

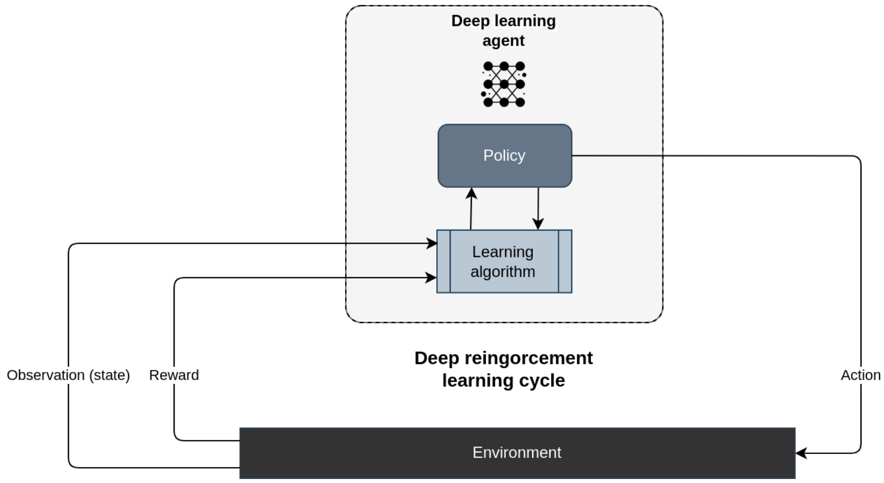


Figure 2.9: The deep reinforcement learning cycle.

to maximise a cumulative reward. At each time-step, the agent tries an action, and the environment responds with a corresponding reward and the new state into which it has progressed. The DL aspect is utilised to interpret complex, high-dimensional inputs. In essence, DRL combines neural networks that interpret large inputs with RL to build sophisticated action policies. The DRL learning cycle is depicted in Figure 2.9.

In the field of combinatorial optimisation, route planning problems for multi-UAV operations have been traditionally solved by mathematical optimisation algorithms and heuristic methods. However, these are often deemed not very suitable in large-scale scenarios, due to the complex and dynamic nature inherent in such tasks [56]. Heuristics are often designed based on assumptions and simplifications that may not hold in real-world scenarios, leading to suboptimal solutions. In addition, such methods require significant computational resources and are often not scalable for large-scale problems. DRL approaches have emerged as a promising solution to address these challenges. These methods leverage neural network architectures to learn optimal policies directly from perceived data, enabling more scalable and efficient solutions [57]. DRL algorithms can effectively handle complex, high-dimensional state-action spaces, and are capable of dealing with dynamic and uncertain environments.

In recent research, policies derived from DRL have been successfully employed to address

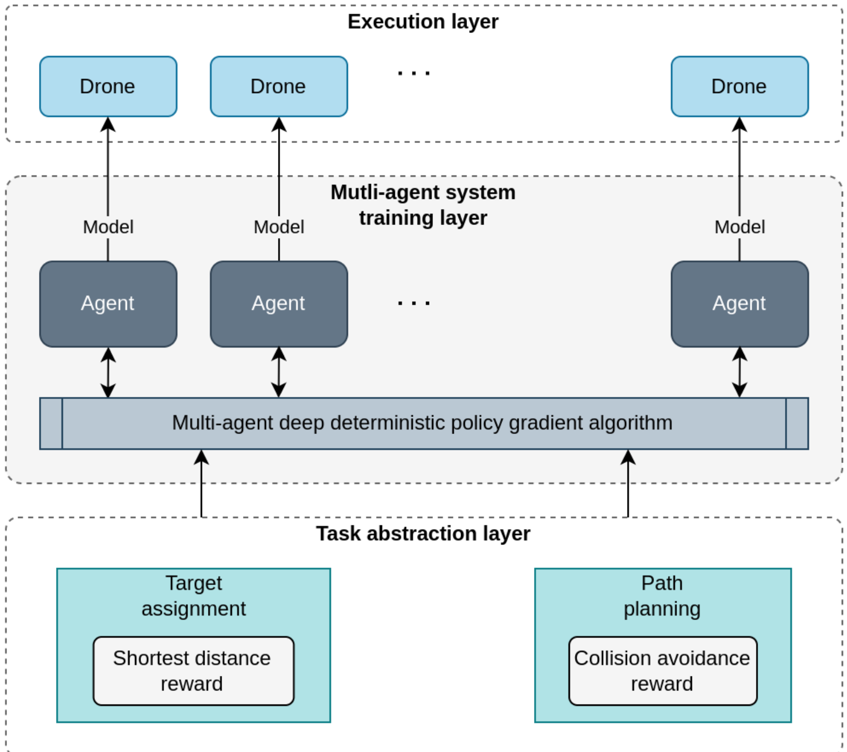


Figure 2.10: The structure of the multi-agent deep deterministic policy gradient framework. [58]

multi-UAV route planning problems without relying on predetermined heuristic rules. For instance, in [58] the problem of task assignment within a fleet of drones is formulated as a multi-agent RL system, and target selection and path planning are solved concurrently by utilising a reward-based approach, based on a three-layered structure, as shown in Figure 2.10. The first layer is the task abstraction layer, which transforms the task optimisation process into the corresponding reward structure convergence procedure. The second layer is the multi-agent system training layer, which trains the agents to learn the optimal policy using the reward structure. The third layer is the policy execution layer, which executes the learned policy to achieve the task goals.

The experimental evaluation of the framework involves the navigation of drone agents, while avoiding collisions with each other and designated threat areas. Various metrics are considered, including the collision probability between agents, the collision probability between agents and threats, and the task completion rate, with the performance investigated for different widths of the threat areas. The results indicate the efficacy of the multi-agent framework, as the collision rate between agents decreases and the task completion improves with longer training times. When the width of the threat areas increases, the performance of the agents naturally deteriorates. Interestingly, the baseline scenario with a threat width set to zero performs worse than the subsequent case with a minimum positive width value.

The flight formation of multiple drones through dynamic spectrum communications is investigated in [59]. The optimal communication strategies are derived from a combined DRL - long-short-term memory (LSTM) network approach to accelerate the convergence speed. The evaluation of information exchange among drones incorporates the concept of Quality of Experience (QoE) to assess the communication system's performance, considering the number of successful transmissions and the associated delay. The packet loss of the drones is modeled using the M/G/1 queuing model, determining the communication priority. To account for the involved delay, the reward function is designed to consider both Quality of Service and QoE.

The experimental analysis illustrates the advantages of the combined DRL approach, compared to traditional reinforcement learning in terms of throughput rate. Furthermore, the use of the priority mechanism on the queuing model, helps to classify the importance of the drone task and ensure successful exchange of formation data in time. Through the use of this mechanism the algorithm is able to reduce delay by 83% based on simulation results.

By optimising the data-transmission scheduling and hovering time, the study in [60] optimises the power usage in a UAV-aided communication network, illustrated in Figure 2.11. A stochastic scheduling algorithm is utilised based on the actor-critic DRL architecture, to optimise data-transmission scheduling and drone hovering time to minimise the total energy consumption. Within an actor-critic architecture, the stochastic policy generated by the actor network maps states to actions, while the quality of the policy is assessed by the critic network through the estimation of the expected cumulative reward. The presented algorithm utilises various techniques to restrict the action space, which otherwise experiences exponential growth due to the combinatorial nature

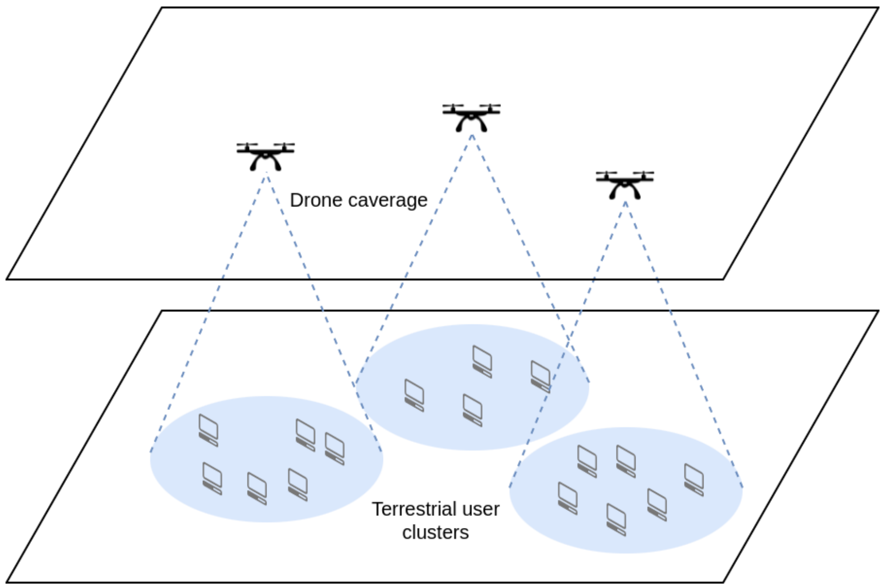


Figure 2.11: A UAV-assisted communication network. [60]

of the problem. Numerical evaluations reveal that the described method yields energy savings of approximately 25-30% compared to conventional DRL algorithms and significantly reduces computational times from the second-level to the millisecond-level.

The study in [61] presents a drone navigation recommender system that combines sensor data with DRL. The system uses two deep learning techniques: proximal policy optimisation for DRL, enabling navigation learning with minimal information, and LSTM networks to provide memory for overcoming obstacles. The system works in a partially observable environment, adopting a step-by-step approach to navigating dynamic and potentially hazardous conditions. The algorithm uses data from sensors on-board the drone, including obstacle detection data from the collision avoidance mechanism, as well as the drone's current direction obtained either via the on-board navigation system or from a compass mounted with the sensors. This data is used as input to an off-policy deep learning model to recommend the direction of travel for the drone according to the current prevailing conditions, surroundings, and sensor readings.

In the study, a simulator was employed to replicate the drone's environment and generate sensor data that faithfully corresponded to real-world data, mimicking the inputs sensed by actual sensors in the given scenario. The accuracy and efficiency of the learned model were then evaluated against these test cases to demonstrate its performance. The study focuses on the significance of testing the learned model in the simulator, as it enhances confidence in the safety of the navigation recommender system and allows for the identification of any limitations or challenges that may arise when implementing the algorithm in real-world scenarios. Nonetheless, it is acknowledged that testing the model solely in the simulator has inherent limitations in terms of the level of assurance that can be demonstrated. To obtain additional assurance and provide evidence of system performance unaffected by integration with other components, real-world testing is deemed crucial.

Chapter 3

Development of a new Deterministic Technique for Drone Control

This chapter provides an in-depth exploration into a deterministic method proposed for effectively controlling drone swarms. This area of research has been steadily gaining traction, driven by the potential applications of drone swarms in numerous fields, including surveillance, search and rescue, agriculture, and more. The deterministic approaches underscore a level of predictability in system behavior, using predefined rules or mathematical models to dictate the actions and interactions within the drone swarm. These techniques offer a structured way of governing drone movements, interactions, and task execution, ensuring a degree of certainty amidst complex, multi-agent systems. The focus of this chapter will be to present, analyse and evaluate a new deterministic technique for drone swarm control, and discuss its applicability, strengths, limitations, and prospects for further improvements. While following chapters investigate the potential of AI-based solutions, this chapter emphasises the value and relevance of deterministic methods, particularly in scenarios where computational efficiency is crucial and where the system's behaviour needs to be thoroughly understood and meticulously controlled.

3.1 Introduction

As discussed in the previous chapter, the proliferation in drone technology has catalysed the emergence of new solutions in areas such as environment [62], surveillance [63], disaster management [64], and search and rescue operations [65–67]. In these contexts, detecting and tracking of individuals become critical in providing necessary assistive services and drones can serve as vital tools towards this goal. Upon locating an individual, they have the ability to provide communication services and transmit the person's location, thereby facilitating the dispatch of a rescue team.

Given that communication can frequently become impaired in disaster scenarios, conventional centralised solutions may not always be accessible, thus necessitating the deployment of decentralised approaches. In light of this, the concept of IoT has undergone a swift progression towards the realisation of intelligent solutions that are practical in rapidly changing environments lacking established infrastructure [68]. IoT frameworks can provide services such as context awareness [69], localisation, and tracking [70], which can be crucial in emergency circumstances, and are particularly valuable in situations where centralised solutions are not optimal.

Various strategies exist for addressing the challenge of localising and tracing a moving target. These include methods involving visual sensors [71], RF time of arrival [72], AOA [73], time difference of arrival [74], Doppler and direction of arrival [75], and RSSI [76] sensors. While algorithms employing visual features can effectively track an object in numerous tracking scenarios, they are often unsuitable for long-range search and tracking operations due to their computational intensity and the requirement for a pre-engineered, centralised architecture. Moreover, time of arrival or related techniques operate on intricate antennas as opposed to simpler RSSI antennas, they frequently encounter synchronisation issues and may impede the mobility of the drones. Given the vital consideration of energy constraints in drone operations, simpler approaches are often favoured to extend flight time and ensure mission completion within the available timeframe. RSSI techniques present promising solutions in this context. However, they come with the significant issue of signal loss in the communication pathway, which is caused by multi-path fading and shadowing.

This study presents a new strategy, whereby a swarm of multi-rotor drones that incorporate RSSI antennas, collaborates to track a mobile target of interest. Leveraging a deterministic algorithm, this approach is able to coordinate the movement of the drones to achieve increased efficiency when tracking. The described strategy can orchestrate the mobility of the fleet, relying solely on the RSSI values measured at every drone. The drones exchange information and synchronise their movements to track and closely follow the mobile RF source. The signal propagation model used in this method is adopted from the report presented in [77]. This model is demonstrated to offer more precise estimations of the signal strength within high-velocity mobile networks, in contrast to the log-distance or the free-space model.

The proposed algorithm offers a real-time scheme that allows drones to perform tracking autonomously, functioning in the absence of centralised solutions or pre-existing infrastructure. The technique's nature offers the advantage of keeping the fleet in the vicinity of the target without necessitating awareness of its precise location or distance calculations that are frequently unreliable. Unlike similar methodologies, the introduced scheme can operate in wide-ranging areas, with limitations only governed by the power of the measured signal and the RF antenna's sensitivity equipped at the drones (e.g., a range of up to 3 km in diameter). As demonstrated by evaluation simulations, this strategy outperforms trilateration based solutions.

The methodology proposed distinguishes itself from common practices as it addresses critical aspects pertinent to real-world scenarios. In pursuit of this objective, it employs simple RSSI sensors equipped on the drones, instead of more complex antennas like DOA or AOA sensors, as suggested by other approaches. Moreover, it integrates the use of agile and compact multi-rotor drones, as opposed to the less flexible and cost-effective fixed-wing drones. Finally, with respect to the tracking scheme, the described method differs from related studies as it does not involve the conversion of the received signal's power into a distance from the target, a process that inevitably introduces errors due to signal fluctuations.

3.2 System description

This section describes the conceived system that addresses the cooperative target tracking problem. It primarily outlines the components that constitute the system, while the following part provides an analysis of the path-loss framework employed for the RSSI model.

3.2.1 System components

The system's key components are as follows:

- **Mobile Target**: Within the context of the proposed application, this is typically an individual lost in a large environment, equipped with an IoT device, or any mobile target attached with a sensor device.

- **IoT Device**: This constitutes an embedded system or the IoT equipment held by the target. The device is integrated with a sensor, that utilises a wireless interface to relay the sensor's data.

- **Tracking Agents**: The multi-rotor drones assigned to locate the moving target and keep it in vicinity. The drones incorporate antennas to receive signal strength and a network interface enabling inter-agent communication. Each drone is capable of identifying its own location via a GNSS sensor and is fitted with a basic flying command apparatus. This system enables them to uphold persistent altitude and velocity, and navigate to specified GNSS locations through controlled steering angles.

- **On-board Processing Unit**: The onboard mechanism of the drones, that establishes inter-drone connectivity via an appropriate network interface. This unit also possesses the computing capability necessary to implement the collaborative tracking scheme and govern the drones' navigation in real-time.

3.2.2 Path loss model

Empirical evidence indicates that, owing to additional reductions from environmental factors, the signal is not correctly modeled by the free-space model in reality. To accurately represent the change in measured signal strength in this study, the loss is calculated through the adoption of the signal propagation model presented in [77]. Applicable to a network of mobile entities moving in free space at high velocities and long distances, this model aligns with the drone speed range in the scenario under consideration. Figure 3.1 showcases the difference of the adopted and the free-space model.

According to the analysis in [77], the signal loss in moving rural networks, can be represented as follows:

$$PL = 41.1 log_{10}(d) + 17.2 + 20 log_{10}(f/5), \tag{3.1}$$

where d represents the distance between the transmitter and receiver, and f represents the frequency of the signal in GHz.

By adding the target transmission power (P_{Tx}), as well as the transceiver's and receiver's antenna gains (G_{Tx} and G_{Rx}), the RSSI at each drone is calculated as follows:

$$RSSI = P_{Tx} - [41.1 log_{10}(d) + 17.2 + 20 log_{10}(f/5)] + G_{Tx} + G_{Rx}, \tag{3.2}$$

Finally, by substituting the power of the transceiver (10 dBm), the transceiver's and receiver's antenna gains (2 dBi), and configuring the frequency to 2.4 GHz, the RSSI can be calculated as follows:

$$RSSI = 3.17 - 41.1 log_{10}(d) \tag{3.3}$$

The scenario under consideration involves drones operating in an unobstructed setting with few obstacles that might induce multi-path propagation; nonetheless, the problem of shadowing

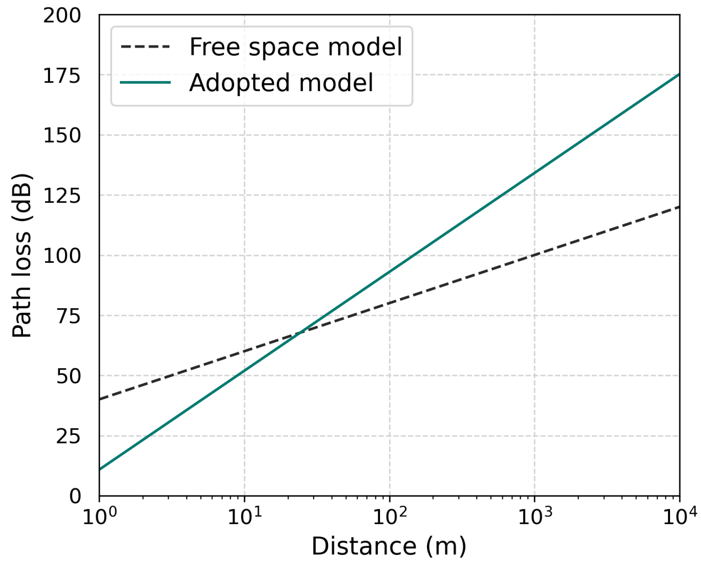


Figure 3.1: Contrast between the free-space and the employed model. The dashed line illustrates path loss in line with the free-space propagation model. The green line showcases path loss based on the utilised scheme.

persists. To account for the impact of slow fading that arises from this problem, the final RSSI value is obtained by adding a Gaussian random variable with mean $\mu = 0$ and standard deviation σ.

3.3 Control algorithm

As previously stated, the aim of the drones within this study is to approach the target and maintain a proximity to provide assistive services. Leveraging the data on the signal strength at each drone, the proposed method entails a straightforward and reliable way to coordinate the collective movement of the swarm in response. The algorithm unfolds in two stages, which are examined in subsections 3.3.1 and 3.3.2.

In the initial stage, termed the "individual search phase", each drone's objective is to survey the area for the most powerful signal. Upon identifying this signal, it summons the remaining drones to its location, indicating the start of the tracing process. The drones then form a swarm and advance to the second stage, referred to as the "cooperative tracking phase". Here, a control algorithm is employed to synchronise their movements, ensuring efficient tracking. In both stages the RSSI sampling takes place at a frequency of 0.5 seconds.

3.3.1 Individual search phase

During this stage, the search area is initially segmented into a number of sectors, determined by the available drones. Presuming the swarm consists of R drones, with $R \geq 3$, the area is divided into R circular sectors, each spanning an angle of $\theta = 360/R$ degrees and assigned to a single drone. Each drone begins its movement in alignment with the bisector of its corresponding segment, continuing until the criterion for transitioning to the cooperative phase is fulfilled. This condition is deemed satisfied when at least one drones achieves an RSSI threshold as calculated by Equation 3.3, which is tenfold stronger than the RSSI at the next nearest drone to the target (difference of 15 dB). An

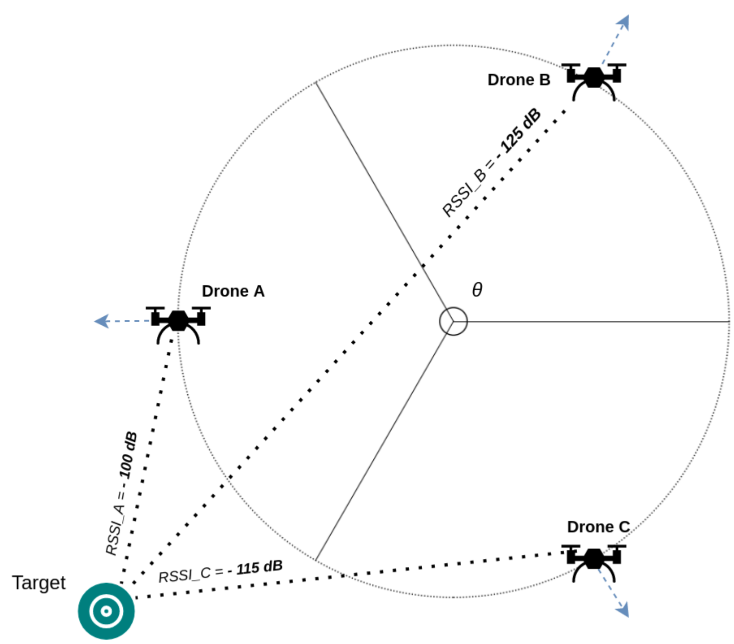


Figure 3.2: Instance of an individual search phase with three drones. Drone A satisfies the condition for the fleet to form and commence the cooperative tracking stage.

example of when this condition is met is illustrated in Figure 3.2. It is worth noting that minor discrepancies in the initiation time of the cooperative phase have negligible impact on the overall efficacy of the tracking scheme.

3.3.2 Cooperative tracking phase

When a drone meets the terminating criteria of the individual search phase, all drones transition into the cooperative tracking phase and commence their swarm movement towards the target. Through the utilisation of RSS measurements, the employed strategy ensures that the drones successfully approximate the target and sustain proximity.

The fundamental principle of the algorithm lies within the changes in RSSI recorded over time. A gradual raise in signal strength suggest that the drone is approaching the target, indicating to sustain its current trajectory. On the contrary, reductions in signal strength indicate that the drone is diverging from the target, thus suggesting the need for a revised directional movement. The adjustment of direction is influenced by the awareness of each drone's current proximity to the target and their corresponding positions within the swarm, as described further below.

To mitigate the signal fluctuations resulting from shadowing in the wireless channel, an augmented strategy is employed. Prior to a decision for change in direction, several consecutive RSS measurements are accumulated in a sample window, and the ultimate value is computed as the average of these samples. However, the considerable distances associated with the context of the examined scenario may render a fixed window size unsuitable. Therefore, the size of the sample window is adaptively modified in response to the RSSI. The correlation between the RSSI value and the dimension of the sample window was decided based on an iterative process of trial and error which determined that high window sizes are appropriate for low RSSI, with exponentially decreasing sizes giving better results as the RSSI gets higher.

Upon determining two successive mean RSSI values, the algorithm calculates the difference, as outlined in Algorithm 1, to evaluate the drone's current status relative to the target. A positive result categorises the drone as "approaching the target", while a negative value signals the "diverging"

Algorithm 1: Determine Drone Status

```
while cooperative_tracking_phase = true do
    samples_average ← 0 ;
    j ← 1 ;
    while j ≤ 2 do
        RSSI ← 0 ;
        i ← 1 ;
        while i ≤ sample_window_size do
            is_halting ← false ;
            RSSI[i] ← GetRSSI ;
            samples_average[j]+ = RSSI[i] ;
            if RSSI[i] ≤ halt_threshold then
                is_halting ← true ;
            end
            i + + ;
        end
        samples_average[j]/ = sample_window_size ;
        j + + ;
    end
    if is_halting = false then
        if samples_average[1] > samples_average[2] then
            approaching ← false ;
        else
            approaching ← true ;
        end
    end
end
```

status. This data regarding the status is broadcasted to the swarm, governing the upcoming decision of every drone according to Algorithm 2.

The process begins by determining the drone associated with the highest RSSI value across the swarm, designating it as the "nearest". Then, when a drone's present status is identified as "diverging", its RSSI is compared to the swarm's maximum RSSI. If this drone's RSSI is lower, it adjusts its trajectory towards the "nearest" drone. However, if the RSSI is higher, it executes a predetermined rotation. The degree of this rotation, as validated in [78] via a geometrical and

Algorithm 2: Determine Drone Direction

```
RSSI ← GetRSSI ;
max_RSSI ← Max ;
nearest_drone ← GetNearest ;
if approaching = false then
    if RSSI < max_RSSI then
        FollowNearest ;
    else
        PerformRotation ;
    end
end
sample_window_size ← GetWindowSize;
```

numerical assessment, guarantees the drone's progressive approach towards the target.

Beyond following the tracking algorithm, the drones integrate a strategy for collision avoidance, utilising their respective GNSS coordinates. As part of this strategy, each drone verifies its separation from the rest in the swarm upon every RSS measurement, and pauses movement should this separation drop below five meters. The adherence of the drones to the cooperative algorithm gives rise to a flocking behavior within the swarm. This emergent behaviour enables the drones to tail the target effectively, by incrementally progressing towards a "nearest" zone and sustaining proximity.

3.4 System evaluation

An extensive assessment of the described tracking algorithm was carried out through the development of a simulator based on the Processing development environment and Java. The simulation environment, is designed to support different tracking algorithms and offers a graphical display of the simulated scenarios. The primary objective of these simulations was to examine the efficacy of the proposed algorithm through its performance comparison with a standard tracking method. The trilateration technique was selected as the benchmark, given its foundational role in the majority of tracking schemes. To enhance the accuracy of the trilateration process, insights from the study in [79] were incorporated. The coordination of the drones was optimised following the study's

Table 3.1: Key simulation parameters.

Parameter	**Value**
Environment area	28 km^2
Swarm size	3 - 20
Target speed	1.4 - 3 m/s
Drone speed	11 - 15 m/s
Sample window size	3 - 17
Drone altitude	100 m
Drone halting distance	115 m
Target TX power	10 dBm (10 mW)
Drone RX sensitivity	-92 dBm
Target TX antenna gain	2 dBi
Drone RX antenna gain	2 dBi
Signal frequency	2.4 GHz

recommendations, ensuring a triangular formation was maintained during the RSSI sampling, thus preventing the occurrence of collinear readings. The results obtained in the context of the research conducted in this chapter are published in [80].

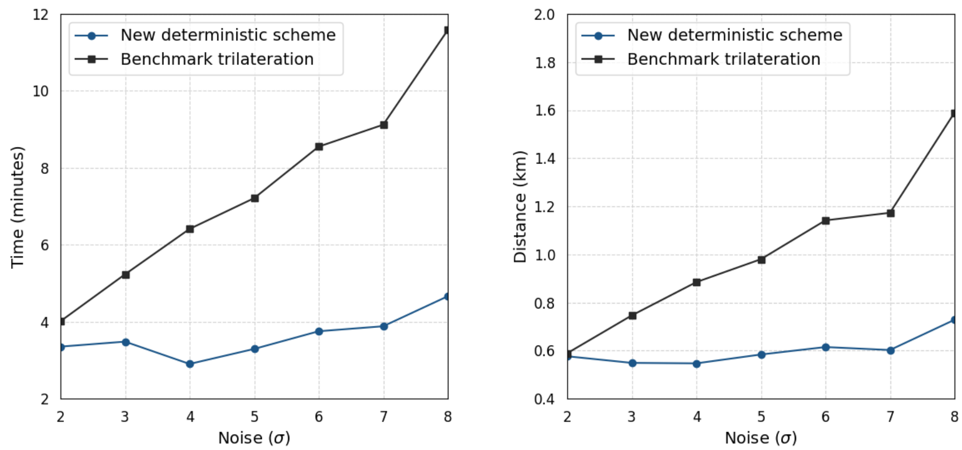
In the conducted simulations, the drones depart from an identical initial position, traversing the search area with a uniform velocity and established altitude. In every instance, the target is randomly positioned on a spot within a circle that has a 3 km radius. To ensure that the target initial positions remain consistent, identical random seeds are employed across both algorithms. The target's movement is modelled after a random waypoint pattern, set at a pace of 1.4 m/s, which corresponds to a human's walking speed. Table 3.1 outlines the key parameters of the simulations. The "halting distance" denotes the maximum proximity at which the drones should approach the target.

In the first part of the evaluation, the performance of the proposed method is compared to that of the reference algorithm with respect to the increase in standard deviation of the noise due to fading (σ). In the next part, the two methods' efficiency is investigated under a practical σ value expected in real conditions, with respect to the increase in drone velocity. Finally, the concluding phase of the assessment focuses on the evaluation of the proposed method with respect to higher target velocities and increased number of drones. For all evaluations, emphasis is given on the key performance indicators below:

- **Minimum time**: The duration is quantified from the start until the first drone achieves the proximity of the halt distance to the target.
- **Average distance**: This metric calculates the mean distance from the target to the nearest drone.
- **Halting cycles**: This denotes the percentage of simulation cycles where the nearest drone remains static because its RSSI lies within the halting threshold.
- **Sustained proximity cycles**: This refers to the percentage of simulation cycles during which the nearest drone maintains a position within the halting distance.

Figure 3.3a plots the minimum time required by the two methods to approach the target against the standard deviation σ of the additive noise. The target's speed is set to 1.4 m/s and the drone velocity is 14 m/s. The graph indicates the proposed algorithm's resilience to the increasing values of σ, illustrating its robustness under diverse conditions. This trend continues in Figure 3.3b, which plots the mean distance of the nearest drone from the target. Increased noise appears to have little impact on the proposed algorithm. However, in the trilateration algorithm, the accuracy declines substantially under noisy conditions, causing its tracking performance to falter significantly, particularly when σ surpasses values of 5.

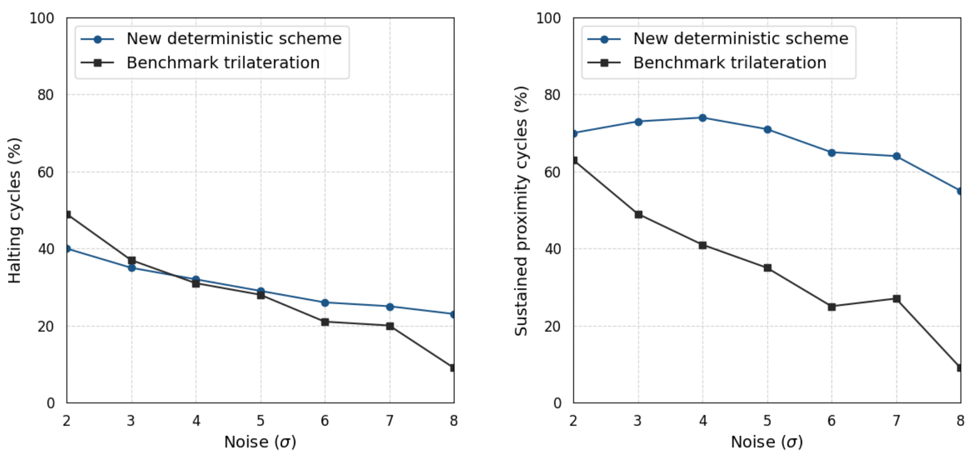
The diagrams illustrated in Figure 3.4a and Figure 3.4b showcase the percentage of simulation cycles where the nearest drone remains static or stays within halting distance respectively. Examining the static cycle outcomes, both algorithms exhibit comparable results under conditions of low noise. However, as the noise level increases towards realistic or higher values, the proposed solution demonstrates an edge. When considering the duration of time the drones can sustain proximity, the proposed method clearly outperforms the trilateration method.



(a) Minimum elapsed time until the nearest drone reaches the target (lower is better).

(b) Average distance between the nearest drone and the target (lower is better).

Figure 3.3: Minimum time and average distance versus the standard deviation of noise due to fading (σ).

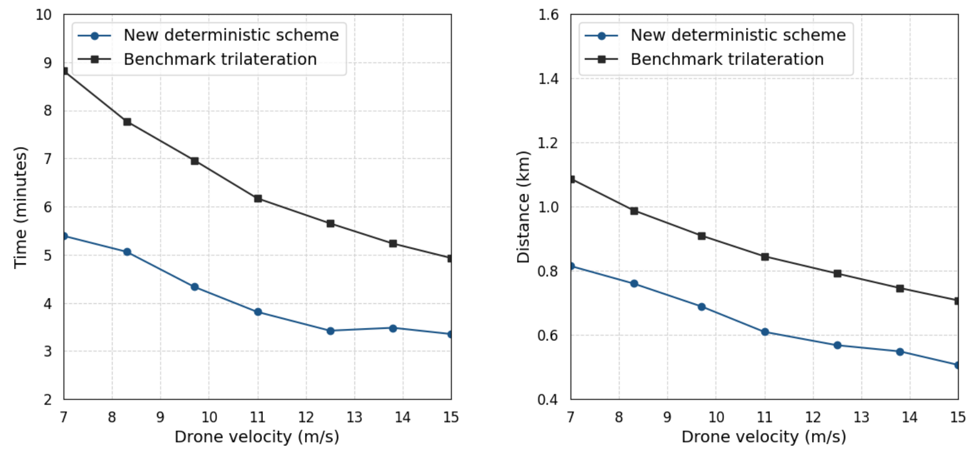


(a) Percentage of cycles where the drone remains static because the RSSI is in halting threshold (higher is better).

(b) Percentage of cycles during which the drone manages to stay within halting distance (higher is better).

Figure 3.4: Halting and sustained proximity cycles versus the standard deviation of noise due to fading (σ).

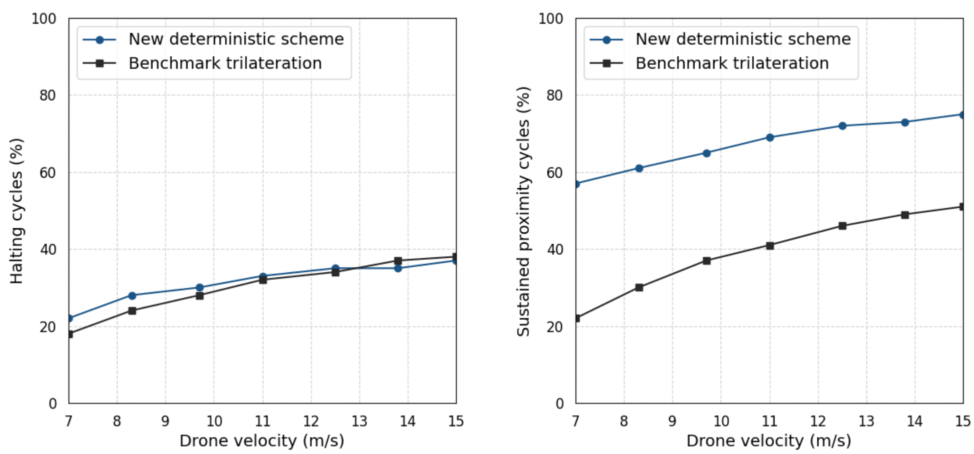
The plotted graphs reveal that the proposed method excels over trilateration, even demonstrating robust performance in highly noisy settings. The key to its efficiency is the strategic avoidance of direct distance calculations using the RSS. Rather, the decision-making process relies on identified



(a) Minimum elapsed time until the nearest drone reaches the target (lower is better).

(b) Average distance between the nearest drone and the target (lower is better).

Figure 3.5: Minimum time and average distance versus the drone velocity.



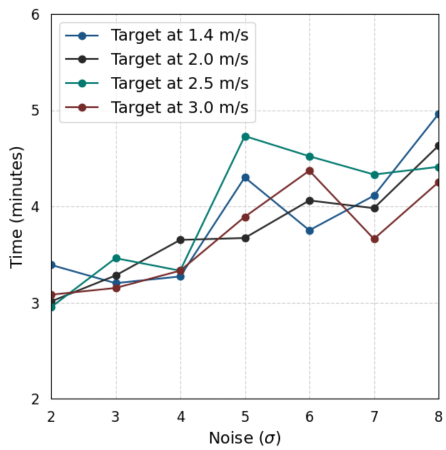
(a) Percentage of cycles where the drone remains static because the RSSI is in halting threshold (higher is better).

(b) Percentage of cycles during which the drone manages to stay within halting distance (higher is better).

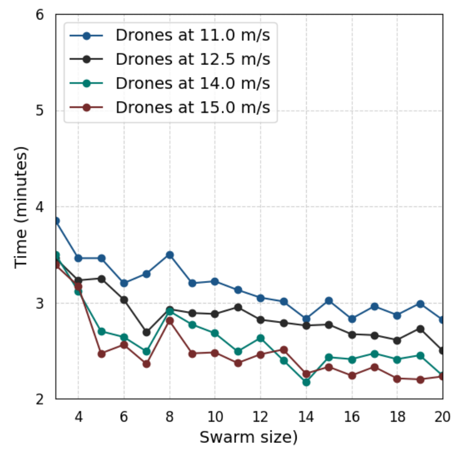
Figure 3.6: Halting and sustained proximity cycles versus the drone velocity.

deviations in the RSSI thereby displaying an enhanced capacity to withstand signal strength fluctuations.

In the next stage of the assessment, the impact of drone velocities on the performance of the



(a) Minimum elapsed time until the nearest drone reaches the target, versus the σ of noise due to fading.



(b) Minimum elapsed time until the nearest drone reaches the target, versus the size of the swarm.

Figure 3.7: Minimum time for different target and drone speeds.

two methods is evaluated. In these evaluations, the standard deviation of fading noise is fixed at a realistic value (σ=3) as proposed in [81], and the drone speeds vary between 11 and 15 m/s. Figure 3.5a depicts the results regarding the minimum elapsed time. Predictably, a reduction in time required for the drones to approach the target is observed as the speed increases. The average target distance as a function of the drone's speed is visualised in Figure 3.5b, which indicates that increases in drone speed also result in decreases in the average distance the drones maintain from the target. These findings underscore the enhanced efficiency of the proposed algorithm in target tracking under realistic noise levels, regardless of drone velocity.

When examining the time duration that the drones remain static due to being in a halting RSSI threshold, both methods achieve marginal improvements with higher speeds, as demonstrated in Figure 3.6a. However, the proposed algorithm has a clear advantage over trilateration in maintaining proximity as the drones' speed increases, indicated by the results of Figure 3.6b. Interestingly, the proposed method exhibits high effectiveness even at slower speeds, potentially offering advantages in energy-constrained or low-speed requirement scenarios.

The evaluation concludes by assessing the efficiency of the proposed method against increased target velocities and inspecting the impact of the swarm's size on the overall tracking process. The minimum time needed to approach the target against the standard deviation σ of the additive noise is plotted in Figure 3.7a. In the examined scenario, the target's velocity varies from 1.4 to 3 m/s, while the drone velocity is set at a constant 14 m/s. The graph reveals minor fluctuations in the algorithm's performance across different target speeds, a predictable outcome given the significant contrast in speed between the target and the drones. Figure 3.7b visualises the minimum time needed by the nearest drone to approach the target, plotted as a function of the swarm size. For lower drone speeds, the outcomes illustrate a higher performance relative to higher number of drones. The algorithm's efficiency however, does not appear to be significantly influenced by a swarm size exceeding seven drones.

Figure 3.8 shows the graphical interface of the developed simulator that executed the above experiments. The interface provides data regarding the speed of both the target and the drones, the current RSSI, and critical parameters including the standard deviation of noise and the dimensions of the sample window. It also displays the actual distances between the target and each drone. By rendering such information accessible, the simulation facilitates a comprehensive understanding of the algorithm's operation, fostering a deeper comprehension of its operational mechanics. To facilitate the analysis of the algorithm's performance, the simulator is exporting all the simulation metrics into spreadsheet files. Therefore, patterns and trends can be discerned visually within the simulation, and a more extensive and in-depth analysis is then carried out using the exported data, augmenting the comprehensive assessment of the algorithm's performance.

3.5 Discussion

In the scope of this chapter, a novel cooperative algorithm, facilitating a swarm of drones in locating and following a mobile target was presented. The ability to provide robust tracking by solely leveraging measurements of the RF signal power radiated by IoT equipment held by the target,

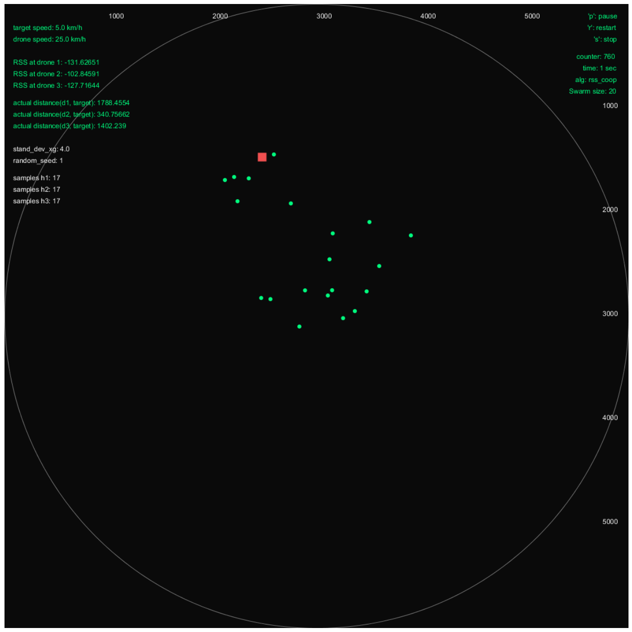


Figure 3.8: Depiction of the simulator's graphical interface, wherein the red dot indicates the target, and the green dots represent individual drones.

proved challenging but effective. This accomplishment was derived from the decomposition of the tracking task into two distinct stages: the individual search phase and the cooperative tracking phase. The system was assessed in a comprehensive evaluation through a series of diverse simulations, which suggested the enhanced efficiency of the presented strategy in comparison to trilateration based algorithms. Additionally, the evaluation results confirmed the algorithm's capability to retain effectiveness under increased levels of signal attenuation due to slow fading.

Figure 3.9 portrays the navigational path pursued by three drones under a scenario of realistic slow fading. The depicted trajectory underscores the effectiveness of the swarm's approach to the target utilising the deterministic algorithm. This level of performance is attained without any knowledge about the RF source's location, relying on the successful implementation of appropriate directional adjustments when deviating from the target.

The prospective applications of this tracking scheme center on search and rescue use cases, and on delivering localisation and relay services in dynamic environments, particularly when conventional network infrastructure is absent. The following chapter contemplates the exploration

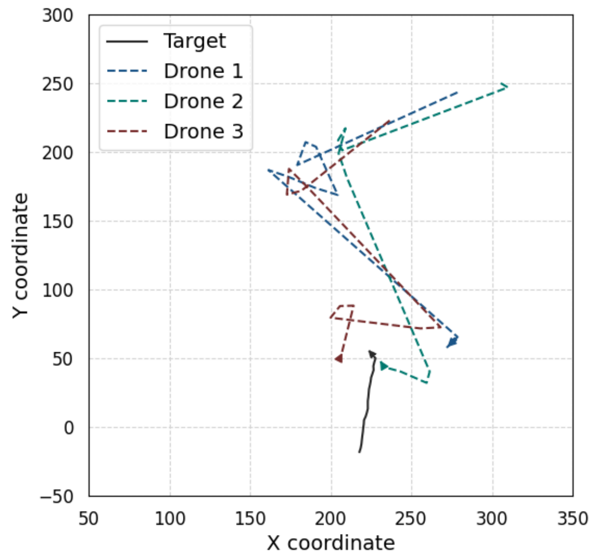


Figure 3.9: Trajectory of a swarm of three drones navigating based on the RSSI from the target, under realistic slow fading conditions.

of AI techniques to augment the described deterministic method.

Chapter 4

Advancing Drone Control: Deep Learning in Cluster Formation

This chapter explores the interaction between AI and drone swarm coordination. It investigates how group formations within the swarm are shaped by deep learning techniques for clusterisation, aiming to enhance the organisational efficiency. This approach moves beyond conventional deterministic methods, enabling dynamic adaptation of the swarm to improve overall effectiveness. By merging the predictability of structured techniques with the adaptive potential of AI, the chapter highlights the potential for integrating deep learning into swarm control.

4.1 Introduction

Ever since the advent of machine learning as a key component of artificial intelligence, a vast body of research has been invested in its comprehensive incorporation within IoT networks, aspiring to devise efficient and resilient algorithms [82, 83]. Fundamentally, ML involves building models which aim to solve knowledge acquisition challenges, employing learning methodologies that augment system capacities by discerning patterns within the supplied data [84, 85]. Implementing ML techniques within IoT networks poses significant advantages, largely owing to the dynamic characteristics of the associated settings. For example, in scenarios such as environmental monitoring, the nodes may be mobile, or even if fixed, their position might shift gradually as a result of environmental alterations (e.g., soil erosion or sea turbulence). In smart city concepts, AI methodologies can be leveraged to improve transportation in the evolving mobility paradigm [86]. Furthermore, IoT networks might be deployed for data collection in locations that are difficult to access or hazardous, often marked by uncertain conditions, or in settings where precise mathematical models are challenging to develop [87]. In such instances, robust systems supported by ML can be applied, empowering the network to self-adapt, relying on low-complexity approximations. Therefore, ML strategies are anticipated to serve as an essential technology enabling the efficient operation of IoT networks, facilitating a plethora of novel applications [88].

Given the energy restrictions inherent in IoT nodes in the investigated context, drones can effectively reduce the transmission power needed by sensor devices through dynamic placement, while ensuring reliable coverage [89]. Therefore, they provide a flexible and efficient approach localising sensors and offering relay services to distant locations within an IoT infrastructure. The methodology presented in this chapter aims to facilitate such applications, empowering a swarm of drones to locate an IoT sensor node in an unspecified position, solely based on the RF broadcast signal emitted by the sensor. Enhancing the deterministic algorithm of the previous chapter, the system employs a DL model which conducts clustering of the swarm based on the RSSI and current coordinates of each drone. Based on this model, the new algorithm identifies remote drone groups at predetermined intervals, recalling them to the base to conserve energy while maintaining the clusters that appear to have more effectively approximated the target sensor. The control algorithm is augmented through the following key contributions:

- The introduction of a graph representation of the drone network, enabling the use of a graph convolutional network (GCN) architecture.
- The implementation of regular interval clusterisation within the network, according to the RSSI and current drone locations, featuring a dynamic number of clusters at each interval.

- The introduction of a deep learning loss function to optimise the clusters.
- The augmented method still operates without explicit knowledge of the target sensor location or reliance on distance estimates.

4.2 Methodology

In the scenario under consideration, the target node is characterised as an IoT gateway that might be positioned on, or in proximity to an IoT device. The engaged drones function as mobile relays, tasked with collecting information or ongoing status reports from the target device. This collected data is intended for a data center, which receives it through the support of the drones. The objective is to track the IoT source node by harnessing wireless signal observations, facilitated by omnidirectional RSSI sensors installed in the drones. This section introduces the GCN concept and its use in clustering within the drone network, to enhance the swarm coordination in efficiently approaching the target sensor.

4.2.1 AI-assisted control

The nature of GCN architectures gives them the ability to capture complex, non-Euclidean structured data, such as the drone networks in the context of this chapter. These networks have non-grid structures, thereby making them challenging to address using traditional convolutional neural networks, which excel at handling grid-like data such as images or time-series [90]. GCNs, as a part of the broader family of geometric DL, have emerged as a powerful tool to address this shortcoming. They utilise graph structures to detect object relations in high-dimensional data, enabling learning on irregular data structures. They leverage the principle of local connectivity, also known as neighbourhood aggregation, to generate representations of nodes based on their neighbours, thereby capturing both local and global graph structures [91]. Essentially, in a GCN, the feature representation of each drone in the swarm network is recursively updated based on the representations of its local neighbours. Thus, a message-passing mechanism is employed where a node gathers information from its neighbours and updates its representation accordingly.

Similar to the previous chapter, the aim of the control algorithm here is to guide the drone fleet to the unidentified location of the mobile sensor with high efficiency. Given that the drones are exclusively equipped with RSSI sensors, their navigation is wholly dependent on the information regarding the signal power detected at their antennas throughout the operation. For the path loss model, the scheme described in 3.2.2 is utilised once again, to accurately measure the signal attenuation in the highly mobile network, thus the RSSI is calculated based on Equation 3.2.

In the scenario under consideration, all drones are initially deployed from a single location and begin to move in random, fixed directions until a pre-determined threshold of RSSI difference between the two nearest drones is registered. Periodically, the employed GCN model performs a network clustering operation, segregating the swarm into drone groups based on their proximity to the target and their spatial locations. Leveraging the unsupervised learning paradigm, the model adeptly partitions the drones into optimal clusters without the need for pre-training at any specific location or labeled data.

Subsequently, the algorithm performs ordering of the clusters according to their proximity to the target, eliminating those that are deemed distant. The drones associated with the distant clusters then execute a return trajectory to the base. This strategic clustering and pruning approach enables the method to retain drone groups that are more likely to converge on the target effectively while minimising the overall energy expenditure by grounding unnecessary drones.

The entire tracking process is depicted in the flowchart shown in Figure 4.1. Upon reaching the RSSI termination threshold, the rest of the swarm transitions into a cooperative mode and navigates based on the deterministic strategy presented in the previous chapter, enabling the drones to closely trail the target sensor. The optimum frequency at which the clustering operation is performed was investigated through extensive simulation evaluations. It was observed that, in general, shorter intervals tend to yield higher performance. However, for the final implementation, a clustering interval of 30 seconds was chosen as it strikes a satisfactory balance between robust performance

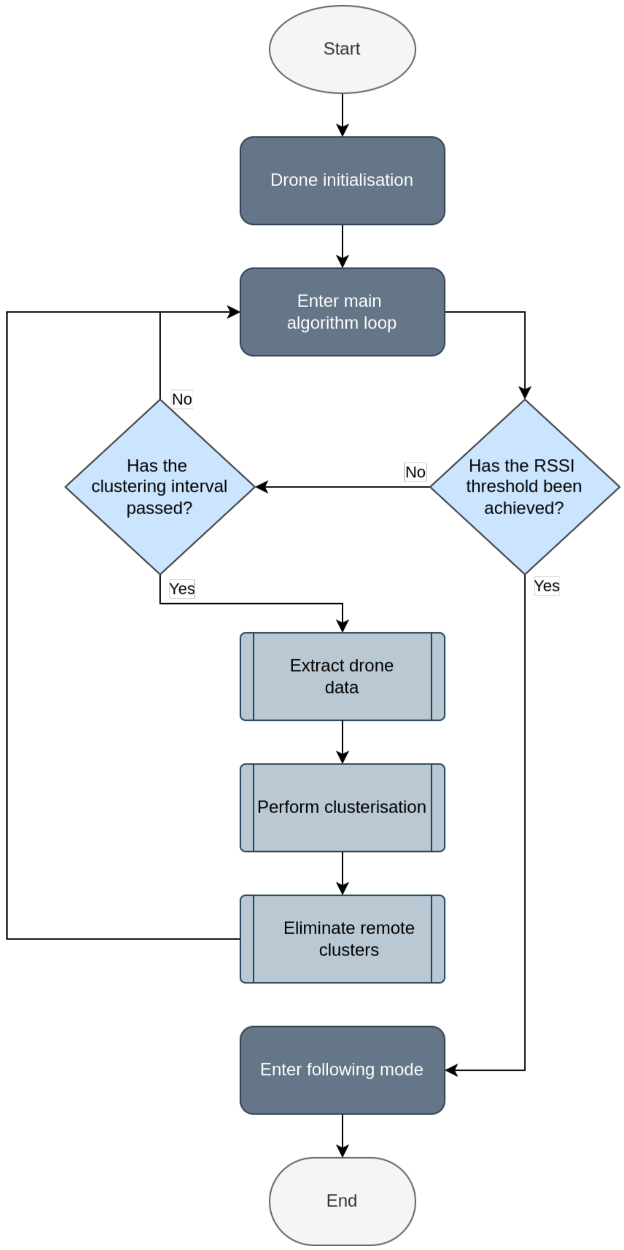


Figure 4.1: Flowchart illustrating the proposed process for tracking the IoT device.

and practical feasibility for real-world applications.

4.2.2 Deep learning clustering for swarm formation

The deployment of GCNs for graph partitioning purposes within drone swarms promises significant benefits. The adjacency matrix of the graph can be used to represent the topology of the swarm network, where each node represents a drone, and the edges represent the communication links. The drone's features include its state information (i.e., position and RSSI from the target). The application of GCNs can facilitate the breaking down of the global swarm into smaller, more manageable sub-swarms or clusters. By learning the optimal clustering, different groups can be controlled based on criteria such as spatial proximity. Furthermore, due to the dynamic nature of drone swarms, GCNs are highly suitable as they can naturally adapt to changes in the graph topology (e.g., drones joining or leaving the swarm).

The method adopted in this chapter merges the "graph partitioning using attention-based pooling" (GAP) framework [92] with a newly designed DL loss function of the RSSI, with the aim of clustering drones based on their proximity to the RF source. Characterised by its ability to generalise, this technique facilitates the joint optimisation of the target loss function, thereby fine-tuning the model for this specific graph structure. After the resulting clusterisation, the algorithm ranks the

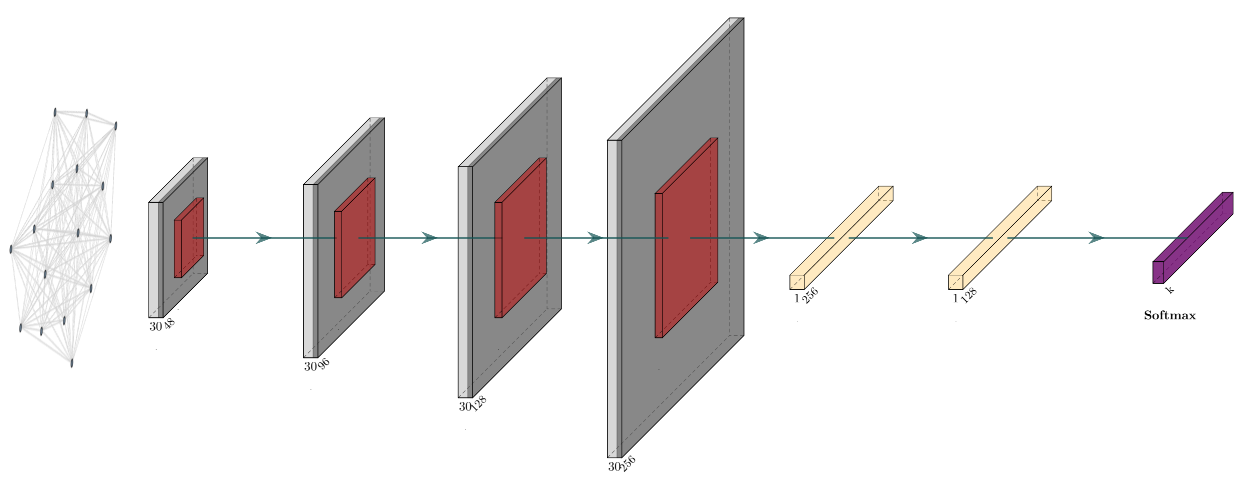


Figure 4.2: Architecture of the employed graph convolutional network. The input dimensions are n x 3, where n is the number of drones in the swarm network. The output layer is of n x 1 dimensions.

clusters and retains those demonstrating the best proximity to the target.

The employed model incorporates four GCN layers to perform convolution on the graph data, with a subsequent application of a Rectified Linear Unit (ReLU) activation function on each. Following in the architecture are two fully connected layers, designed to scale the output to the decided number of clusters. Finally, a softmax function is placed in the end to output the probability distribution of the drones over the multiple clusters. Figure 4.2 illustrates the complete architecture. This arrangement enhances the feature representations and leads to a better correlation between the input parameters in the cluster challenge under consideration. The size of the layers was tuned based on the specific problem and data to avoid issues related to over-fitting.

4.2.3 Graph representation of network

The utilisation of the GCN architecture requires a depiction of the drone network through a graph model to act as input to the DL network. Within the examined network, all nodes are engaged in intercommunication, thus a fully-connected graph is created. The graph represents the drones as vertices and their inter-connections as edges. This representation leads to the deduction of the degree and adjacency matrices, which in turn are used to derive the normalised Laplacian matrix of the graph. The normalised Laplacian with respect to the spectral properties of the graph is calculated by [93]:

$$L_{normalised} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{4.1}$$

where I represents the identity matrix and D, A correspond to the degree and adjacency matrices respectively. Furthermore, the data related to each drone's spatial coordinates and RSSI derived from the RF source are formulated in another matrix. The GCN model is built using the aforementioned matrices, and is trained to output the most fitting clusterisation.

4.2.4 Loss function

The training procedure involves a specified loss function with the objective to optimise for the RSSI within each individual cluster, defined as follows:

$$L_{RSSI} = \frac{1}{N}\sum_{n=1}^{N} |Yr|, \tag{4.2}$$

where Y denotes the output matrix from the neural network, while r represents a vector containing the RSSI values. Through the optimisation of this function, the model constructs the clusters and categorises every node. To maintain a balanced distribution of drones in each partition an additional loss is added, shown in the following equation:

$$L_{\text{balance}} = \max\left(0, \frac{n}{k} - \sigma Y\right), \tag{4.3}$$

where n denotes the number of drones, k the number of clusters, and σ represents a hyperparameter used to adjust the relief degree.

4.2.5 Dynamic calculation of optimal cluster count

The dynamic characteristics of the designed control algorithm result in varying numbers of drones positioned in different locations each time the GCN model is invoked. As such, determining the optimal number of clusters to be generated by the model in advance poses a challenge. This number is critical as it directly influences the algorithm's performance, and therefore it needs to be determined dynamically based on the current conditions. While it may seem intuitive that increasing the number of clusters would improve the performance, at a certain point there are diminishing returns, leading to a phenomenon known as over-fitting. While this point can be identified by several methods, in the context of this chapter, the 'elbow' method was selected.

The efficiency of the control algorithm inherently fluctuates with differing numbers of clusters. Given that every experimental scenario presents a unique setup at each clustering interval, it's

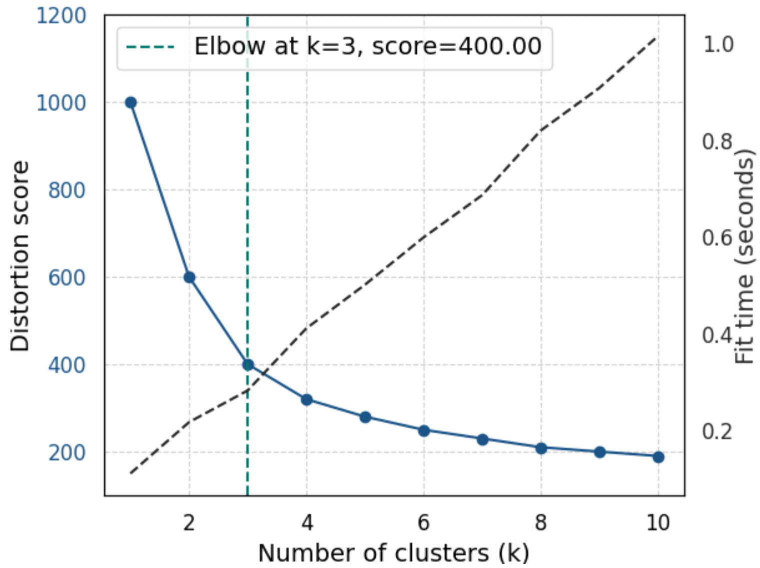


Figure 4.3: Plot illustrating the elbow method, where the evaluation of the distortion score involves computing the summation of squared distances between each point and its designated centre. In this example, the optimal cluster number is 3.

intrinsically challenging to find a universal correlation. In light of this complexity, the 'elbow' method facilitates the automation of the entire procedure, leading to the identification of an optimal cluster count. Although there may be instances where it could produce a less-than-ideal solution, it generally yields optimal results in the majority of cases in the considered scenario.

To employ the 'elbow' method, k-means clustering is performed over a range of varying k values, with the sum of squared errors being computed for each distinct k. The sum for each k is then plotted, resulting in a line chart reminiscent of an arm, as can be seen in Figure 4.3. The 'elbow' point signifies the optimal k value, beyond which the integration of an additional cluster doesn't substantially enhance the model's ability to capture patterns in the dataset. The identification of the 'elbow' point is accomplished mathematically by computing the gradient of the graph between each pair of k values and deducing where the most significant shift occurs.

Table 4.1: Simulation parameters for each investigated scenario. The main difference lies in the drone speed and the starting distance of the target.

Parameter	Scenario 1	Scenario 2
Number of drones	45	45
Clusterisation interval	30 s	30 s
Drone altitude	100 m	100m
Drone velocity	**40 km/h**	**50 km/h**
Target velocity	5 km/h	5 km/h
Target distance	**2 km**	**3 km**
Target TX power	10 mW	10 mW
Target TX gain	2 dBi	2 dBi
Drone RX gain	2 dBi	2 dBi
Signal frequency	2400 MHz	2400 MHz
Total duration	1000 s	1000 s

4.3 Experimental setup

In order to assess the efficacy of the AI-assisted control algorithm, a test-bed featuring two distinct tracing scenarios was constructed:

1. The initial scenario involves drones that maintain a velocity of 40 km/h, while the target sensor is randomly positioned at a distance of 2 km from the designated deployment site.
2. For the subsequent scenario, the target is placed 3 km away from the drone deployment site, and the drones' sustain a speed of 50 km/h.

Under both scenarios, the target sensor's mobility is governed by a random waypoint model, with the drones navigating based on the described algorithm of section 5.3. Each experimental scenario is executed with a variety of noise due to slow fading (simulated with the standard deviation of σ) to evaluate how the algorithm performs under different conditions. The impact of fast fading is deemed negligible due to the vast distances involved and the open environment in which the drones operate, therefore multi-path propagation is not considered. Besides a varying σ, the evaluation includes a varying RSSI threshold at which the clustering procedure is set to terminate. To increase the reliability of the results, each experiment is repeated fifteen times with different random seeds for the target position and the directions of the drones, with the average result considered as the final outcome. The specific parameters of the target sensor and the drones used in the simulation are listed in Table 4.1.

The scenarios were investigated using an enhanced version of the simulator developed in the context of the previous chapter, updated to incorporate the use of the new GCN model. The adjustments allow the extraction of clustering outcomes at every interval based on the new scheme. During each experiment, the simulator is programmed to generate all drone positions and RSSI values at each clusterisation interval, which are used to extract the data that serve as input to the GCN model, which in turn produces a series of integers specifying the indices of the drones that belong to the less efficient cluster. The input dimensions are n x 3, corresponding to each drone with their unique RSSI reading, along with x and y coordinates. Since all drones fly at the same, constant altitude, the z coordinate does not affect the result and is omitted. As before, the simulator provides visual depictions of the drones and target, and displays numerical data pertaining to the parameters and measured metrics. Following the completion of the experiments, the conclusive outcomes are compiled and exported. The simulator during different timestamps is illustrated in

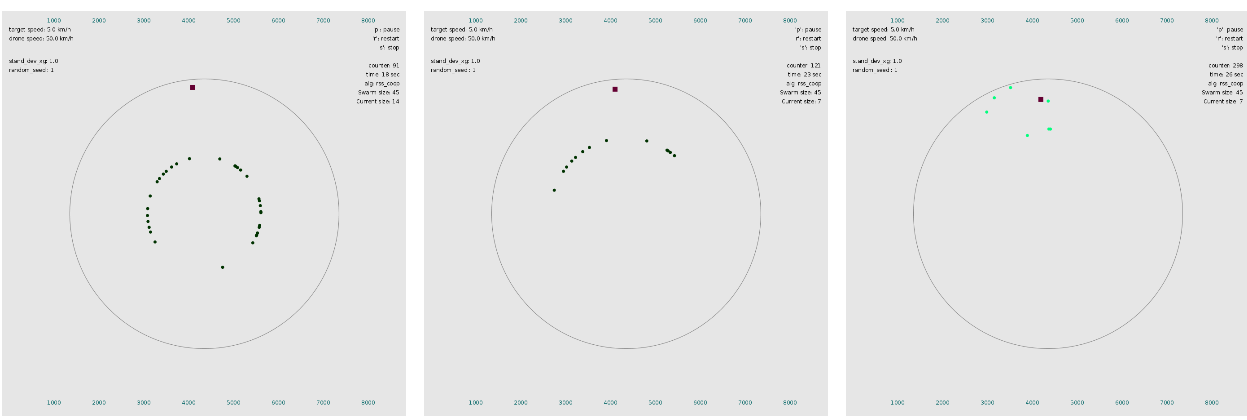


Figure 4.4: Simulation snapshots at different timestamps. The target sensor is depicted by a red square, and the drones are represented by dots. In the left sub-figure, the swarm is depicted 90 seconds after the algorithm initiation. The middle sub-figure is 120 seconds in, while the last depicts the following mode after reaching the threshold that denotes the termination of the clustering process.

Figure 4.4.

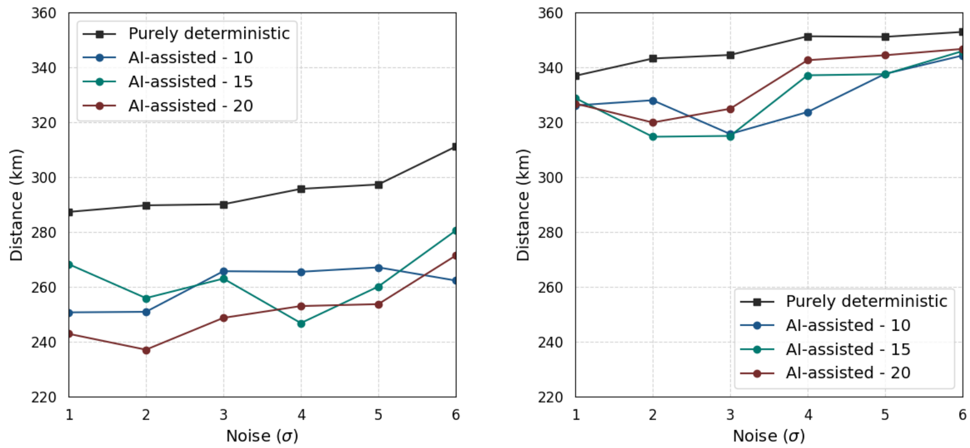
4.4 Evaluation

This section describes the findings from the conducted simulations across both scenarios. The primary focus of these experiments is to assess whether the previously introduced deterministic algorithm could be effectively augmented through AI-assisted control, and to provide a comprehensive comparison of the two methods under varying conditions. The evaluation involves quantifying the performance across three metrics: the time needed for the swarm to approach the sensor, the mean distance each drone traverses during the tracking operation, and the aggregate distance covered by the entire fleet. The results obtained in the context of the research conducted in this chapter are published in [94].

4.4.1 Results

Figures 4.5a and 4.5b showcase the time, in seconds, necessary for the sensor to be located, plotted against the standard deviation σ of the Gaussian noise, for the two scenarios under consideration. Results of the AI-assisted control algorithm at varying clusterisation termination thresholds, are compared with outcomes from the purely deterministic method. When the sensor is positioned 2 km away, the AI-assisted algorithm seems to perform optimally when the clustering stops at a 20 dB threshold. However, an overall average termination threshold of 15 dB appears to offer a more consistent performance across both sensor placements. Regardless of the RSSI threshold, the findings suggest that AI-assisted control surpasses the purely deterministic approach, providing a time advantage of nearly 60 seconds in some instances.

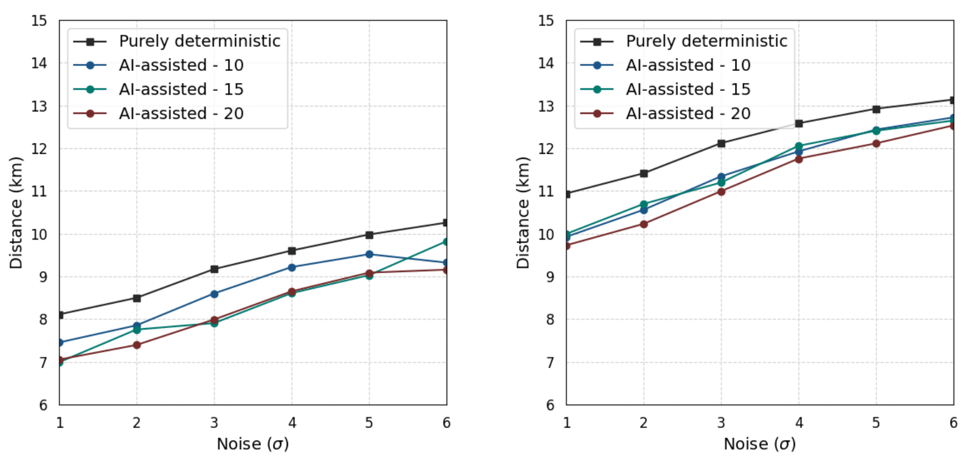
Figures 4.6a and 4.6b effectively present the results, comparing both methods, in terms of mean distance traversed per drone, across both scenarios. The metrics regarding the total distance covered by the entire fleet are similarly illustrated in the graphs of figures 4.7a and 4.7b. Generally, the AI-assisted method performs most efficiently at a threshold of 20 dB, with each drone traversing,



(a) Target positioned 2 km away from the swarm deployment point.

(b) Target positioned 3 km away from the swarm deployment point.

Figure 4.5: Comparison of the time taken for the first drone to locate the sensor, plotted against the standard deviation (σ) of the additive noise. The two algorithms are compared at clustering termination thresholds of 10, 15, and 20 dB of RSSI.



(a) Target positioned 2 km away from the swarm deployment point.

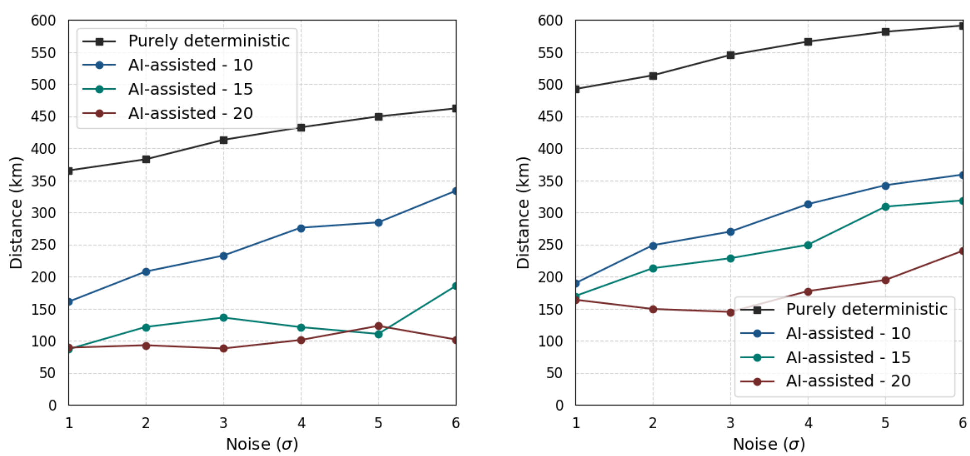
(b) Target positioned 3 km away from the swarm deployment point.

Figure 4.6: Comparison of the mean distance each drone traverses, plotted against the standard deviation (σ) of the additive noise. The two algorithms are compared at clustering termination thresholds of 10, 15, and 20 dB of RSSI.

on average, approximately 1 km less in comparison to the purely deterministic algorithm. When assessing the aggregate distance of the entire fleet throughout each experiment, the benefits of AI-assisted control become even more apparent. This result is further justified given that the new method systematically eliminates distant drone groups, retaining only those that demonstrated the highest effectiveness in reaching the target. A close observation of the values in these plots provides insight into the number of remaining active drones; a lower total distance tends to imply fewer drones present at the end.

Table 4.2 provides a comprehensive overview of the mean results when the sensor is positioned 2 km from the swarm deployment location, examining all values for the standard deviation σ of the additive noise, across all RSSI termination thresholds, for both algorithms. The AI-assisted method consistently demonstrates higher efficiency in all metrics, regardless of the selected threshold, particularly notable in the reduction of total distance traversed by the fleet.

The findings indicate that through the AI-assisted control the augmented algorithm proves to be more efficient in terms of both the time required to reach the sensor and the aggregate distance



(a) Target positioned 2 km away from the swarm deployment point.

(b) Target positioned 3 km away from the swarm deployment point.

Figure 4.7: Comparison of the total distance covered by the entire swarm, plotted against the standard deviation (σ) of the additive noise. The two algorithms are compared at clustering termination thresholds of 10, 15, and 20 dB of RSSI.

Table 4.2: Summary of average results across all standard deviation σ values of the additive noise, for both algorithms, at varying RSSI termination thresholds. Time and distance are quantified in seconds and meters respectively. Sensor positioned 2 km away from the swarm deployment point.

Threshold	**AI-assisted control**			**Purely deterministic**		
	Time	**Average distance**	**Total distance**	**Time**	**Average distance**	**Total distance**
10	260	8661	249032	332	9201	414214
15	268	8351	126716	295	9270	417322
20	251	8222	99191	268	9345	420705

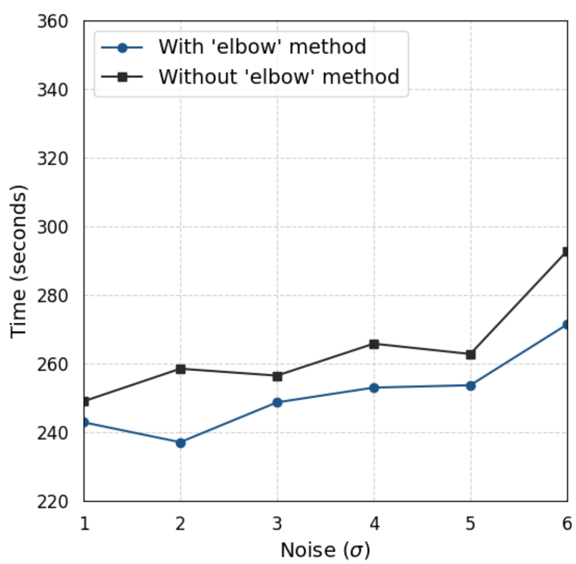


Figure 4.8: Comparison of the time taken by the first drone to approach the target device at a 2-kilometer distance, considering varying levels of additive noise dispersion. Outcomes obtained when employing and not employing the elbow method for identifying the cluster count.

traversed, especially when a 20 dB threshold is applied. Therefore, employing the new method at a real-world implementation is deemed suitable, as it demonstrates greater efficiency in the distances traversed by the swarm, leading to a lower energy expenditure in each mission.

4.4.2 Ablation study

For a deeper understanding of the advantages of applying the "elbow" technique, which dynamically computes the optimal cluster count as opposed to pre-defining a fixed number, the first scenario was examined in an experiment without dynamic adjustment. The time metric evaluating the time taken for the first drone to approach the sensor, is illustrated in Figure 4.8. The cluster count is set at 4 in the non-dynamic setting, and the termination threshold at 20 dB. The graph portrays an enhanced efficiency when leveraging the 'elbow' technique to determine the cluster number, with the swarm reaching the sensor roughly 10-20 seconds quicker, influenced by the present noise. This suggests the high value of integrating a heuristic strategy to calculate the ideal cluster number, as opposed to retaining a fixed count regardless of the swarm size.

4.5 Discussion

This chapter delved into the task of deploying a swarm of AI-assisted drones with the goal of tracking a remotely placed sensor in an IoT setting. The deterministic method described for such tasks in the previous chapter, was augmented by the incorporation of a cluster formation technique using AI. The foundation of the new algorithm relies on modeling the swarm network as a graph structure, and employing a GCN architecture to identify clusters among the drones. The model uses features of the drone coordinates and RSSI readings from the sensor to perform optimal clustering.

By prioritising the clusters that are more effective in advancing towards the sensor and dismissing those deemed inefficient, the control algorithm demonstrates a remarkable effectiveness in leading the swarm near the target. The changing dynamics in the drone fleet necessitate the use of the heuristic 'elbow' method which plays a critical role in deducing the ideal cluster count at each interval, with these clusters being optimally formed by optimising the designed loss function.

The simulation experiments that serve as the empirical validation of the augmented algorithm, illustrate the higher performance of this approach over the purely deterministic method. The findings highlight the high potential of incorporating AI and specifically GCNs in the context of swarm coordination.

However, it should be noted that deploying AI models, and in particular GCNs, onto drones presents challenges as they require significant computational resources for training, which may exceed the capabilities of resource-constrained drones. While training in simulators can mitigate this issue, disparities between simulated and real-world environments, such as sensor noise, environmental dynamics, and communication constraints, can affect the model's performance when deployed in real applications. Addressing these challenges requires careful consideration of computational efficiency, model adaptability, and the fidelity of simulation environments to real-world conditions.

To conclude, with the knowledge gained from this investigation, the vision of using dynamic, AI-based approaches in real-world scenarios is significantly enhanced, offering a robust foundation for the development of next-generation drone control solutions in challenging environments. Particularly in settings where traditional network infrastructure might be lacking or entirely absent, these findings could lead to significant advancements in search and rescue operations and in providing communication and location services.

The study in this chapter provides a first glance into the workings of AI and lays a solid groundwork for the next chapter, which aims to harness the full potential of deep learning by developing an advanced deep reinforcement learning scheme. The new scheme targets the solution of complex decision-making problems in the context of IoT applications, involving the development of policies that allow the drones to act optimally towards their goal given several constraints.

Chapter 5

Multi-Agent Drone Route Planning Optimisation under Constraints

The previous chapters explored the critical role of drones in the field of IoT and investigated deterministic and AI-assisted methods for swarm coordination in IoT sensor localisation contexts. Building on that foundation, this chapter is dedicated to the development and optimisation of multi-agent drone route planning strategies, which is an area of paramount importance when dealing with more complex operations and larger scale environments.

This chapter delves into the inherent complexities involved in multi-agent systems, where the main challenge is not only the design of an optimal route for each individual drone but also the coordination among multiple drones to ensure mission success while satisfying certain constraints. Therefore, the importance of each drone as an independent decision-making entity is addressed under the prism of it being part of a broader, coordinated system. The proposed algorithmic framework leverages novel computational strategies and optimisation methods, aimed at improving UAV-assisted data collection efficiency in IoT applications. In the developed system, each drone builds a distinct neural network which is trained through complex trade-offs between exploration and exploitation to develop effective action-selection policies.

5.1 Introduction

The field of UAV-assisted data collection has sustained a growing research initiative, particularly within the framework of IoT networks [95]. In the context of the applications examined in this thesis, these networks encompass sensing devices with small battery capacities, restricted to emit low-power broadcasting signals [96]. Drawing near to IoT devices and ensuring robust line-of-sight connections to terrestrial sensors, multi-UAV systems serve as adaptable data acquisition and relay platforms [97, 98]. The realisation of the required autonomy in these applications, necessitates an intricate control system, capable of learning efficient trajectories and guiding the drones based on intelligent decisions [99].

The realm of multi-UAV route planning can be seen as a variation of the m-TSP [100], which lies within the domain of NP-hard problems. Because of their inherent intricacy, an optimal solution to such problems is rarely achieved within a manageable computational timeframe. Solutions that are generally accepted for large-scale NP-hard problems can be produced by approximation algorithms [101], while typical approaches often involve heuristic algorithms such as PSO [102], genetic algorithms [103, 104], or the use of quantum annealing [105]. These methods are highly regarded for their ability to provide satisfactory outcomes efficiently. Despite this, an increased complexity of combinatorial optimisation problems, is associated with high-dimensional spaces, in which the volume of data necessary to provide satisfactory solutions often expands exponentially, giving rise to what is known as the "curse of dimensionality," a phenomenon that poses significant challenges to even the most advanced computational methods.

In response to the complexity of high-dimensional CO problems, recent studies have shown that the primary issue can be broken down into more manageable sub-problems, more easily solved through heuristic algorithms [106]. For instance, in the case of multi-UAV task scheduling, a problem partition might entail distinct task allocations for each drone, achieved through simulated annealing algorithms [107] or ant colony optimisation [108]. However, the generalisation abilities of these strategies are often limited by the need for human-crafted rules to supplement the heuristic frameworks in relation to the specific problems they address. Reinforcement learning emerges as a potential substitute, due to its capability to automate the search conducted by heuristic methods, via the self-supervised training of agents, without need for labelled data.

Nevertheless, despite the successful application of RL methods in combinatorial optimisation problems, the "curse of dimensionality" persists when additional constraints are introduced and the space complexity is increased [109]. Considering the history of reinforcement learning in this context, the introduction of deep neural network architectures has resulted in the favourable deployment of deep reinforcement learning methods within the field. With the aid of advanced training techniques, DRL allows the involved agents to develop decision-making policies, capable of offering solutions efficiently, that do not rely on human-engineered rules, thus enhancing flexibility and adaptability.

Considering the aforementioned points, the focus of this chapter lies in the introduction of a multi-agent deep reinforcement learning (MADRL) strategy that is designed for efficient route planning in UAV-assisted data collection scenarios. The problem under consideration involves drones, assigned to an end-to-end task of collaborative data acquisition from terrestrial IoT sensors, which concludes with offloading their payload at a designated server facility. To prevent data redundancy, each sensor should be visited exactly once, and the drones must adhere to a constrained storage capacity. In addition, they are permitted to offload at the data center only upon confirmation that the collection task has finished. This leads to a significant increase in the complexity of the decision space, particularly with the incorporation of multiple drones.

The agents' learning ability is significantly enhanced by the integration of a double deep Q-learning [110] strategy. This strategy harnesses two distinct neural networks per agent - the first devoted to devising the final action selection policy, and the second employed for the evaluation of actions, thus eliminating overestimation issues.

To ensure the effective evaluation of the proposed method within the context of the defined problem, a suitable RL environment is developed, able to encode unique mission instances into sequential states that are utilised by the training algorithm. The constructed environment features a tailored reward function designed to guide the training procedure in alignment with the DRL strategy.

The core contributions of the proposed approach are summarised as follows:

- A novel cooperative MADRL strategy is introduced, tasked with building efficient action-selection policies for individual drones. This strategy empowers each drone to develop a unique policy, thus facilitating distinct action-taking capabilities in every state of the environment. The resulting policies leverage collective environmental information, avoiding reliance on individual observations for the action selection.

- A custom RL environment is designed specifically to facilitate data collection use cases. It accommodates multiple agents and supports the encoding-decoding of mission instances to sequential states.

- A new reward function is described, designed to effectively train agents for optimal action selection within the environment. The reward function incorporates elements of reward shaping and uses a global reward scheme, facilitating cooperative learning among the drones.

5.2 Mathematical model and framework description

5.2.1 Problem formulation

This chapter considers an environmental monitoring application in which a wireless sensor network acts as the integral part of an IoT architecture. In the considered application, terrestrial IoT sensors transmit data which is collected by drones deployed as aerial data collection stations. The drone set is expressed as $U = 1, 2, ..., N$, where N corresponds to the total number of drones, and the sensor set is expressed as $S = 1, 2, ..., K$, where K denotes the total number of sensors. A square grid is used to represent the environment, where each drone or sensor position corresponds to a 2D matrix, in which every location l_{ij} is identified by its row index $i \in \mathbb{Z}, 0 \leq i < X$ and its column index $j \in \mathbb{Z}, 0 \leq j < Y$.

To successfully complete the mission, a certain number of drones, denoted by n, must depart from the starting location, collect data from a specified number of sensors (k), and navigate to the data center located at the ending location to offload their payload. The number of steps that drone n needs to arrive at sensor k is denoted as t_{nk} and calculated as $|i_n - i_k| + |j_n - j_k|$. The drones have a limitation on the amount of data they can store due to their onboard storage capacity, denoted as C. Lastly, a binary decision variable v_{nk}, is used to keep track of which drone has loaded data from which sensor as follows:

$$v_{nk} = \begin{cases} 1, & \text{drone } n \text{ collects data from sensor } k \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

The goal set by the objective function is to minimise the number of steps required for each agent to reach the solution:

$$min \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} v_{nk} \tag{5.2}$$

Subject to the following constraints:

$$\sum_{n=1}^{N} v_{nk} \leq 1, \forall k \in S, \tag{5.3}$$

$$\sum_{k=1}^{K} v_{nk} \leq C, \forall n \in U, \tag{5.4}$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} v_{nk} = S, \tag{5.5}$$

that guarantee that each sensor is visited by no more than one drone (5.3), the capacity of each drone is not exceeded (5.4), and data from all sensors have been loaded (5.5).

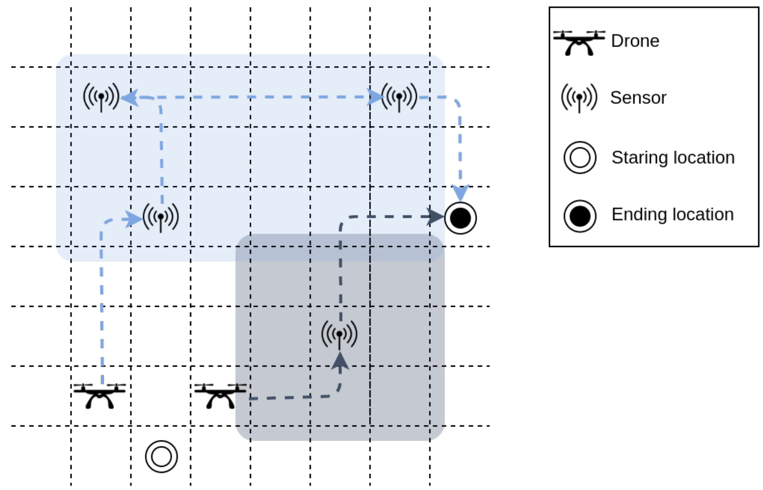


Figure 5.1: Example of environment with 2 drones and 4 sensors.

5.2.2 Custom environment

In order to examine the solution to the problem under consideration, a new RL environment is devised and developed, utilising the OpenAI Gym library [111]. The environment is built in accordance to the core Gym API functions, and is able to incorporate multiple agents. The cooperative interaction of the agents within the environment utilises a sequential decision-making process. This process is designed based on a Markov Game, which extends Markov Decision Processes to multi-agent settings. The aim is to collectively maximise the obtained reward and successfully complete the mission. An illustration of the environment can be seen in Figure 5.1.

A Markov Game utilises a set of states $\mathscr{S}$, to express all possible arrangements of the environment and the N agents participating. To control the behaviour of the agents, the environment incorporates a set of actions A and a set of observations O. With respect to these factors, an action is chosen by each agent u based on a learned policy $\pi_u : O_u \times A_u \to [0,1]$, which determines the subsequent state of the environment through a transition function $\mathscr{T} : \mathscr{S} \times A_1 \times ... \times A_N \to \mathscr{S}$. Throughout this process, each agent is awarded a unique reward based on a reward function $r_u : \mathscr{S} \times A_u \to \mathbb{R}$, and acquires a new observation $o_u : \mathscr{S} \to O_u$. The aim of the agents in the Markov Game is to maximise the total reward obtained, within the designated time-step horizon T:

$$R_T = \sum_{t=0}^{T} \gamma^t r_u^t, \tag{5.6}$$

where γ is a discount factor, which ranges from 0 to 1. The purpose of γ is to reduce the value of future rewards, making them less valuable than immediate rewards.

5.2.3 State space

In order for the agents to act effectively within the environment, a clear representation of the environment's state is essential. To achieve this, discrete states are introduced as a way to abstract the navigation in a continuous space. These states are expressed by an integer, encoding the positions of the agents, their available storage, and the data collection status (i.e., the sensors from which data has been loaded). Each state is mapped to a distinct integer through a bijective mathematical function, which also allows the recovery of the state, given the corresponding value. The agents make decisions within the discrete time horizon, where, at each time step, a state transition function (see Section 5.2.5) is employed to update the environment state based on the actions of the agents and to provide a reward (see Section 5.2.6). The state space encompasses the set of unique states that define the range of possible configurations in which the environment can exist.

5.2.4 Action space

Assuming that the drone agents possess a flight control system that enables basic navigation, the environment allows for a set of actions denoted as A, which comprises four fundamental movement actions, along with two supplementary actions specifically designed for loading or offloading data from the sensors. The collection of actions defining the action space of the environment is listed in Table 5.1.

Table 5.1: Action space of the developed environment.

Action No.	Action Description
0	Move south
1	Move north
2	Move east
3	Move west
4	Load data
5	Offload data

5.2.5 Transition function

Through the transition function, the environment is able to progress from state s_t at time-step t to state s_{t+1} at time-step $t+1$, after each agent has selected a discrete action. Given a subset of actions A_t and a state s, the transition function $T(s, A_t)$ is employed to calculate a 4-dimensional vector $t = [p, s', r_G, f]$, with p representing the probability that the actions in A_t will result in a transition to state s', yielding a combined reward r_G. The variable f is a boolean that indicates if the problem has been solved (i.e., mission completed).

5.2.6 Reward function

The reward function provides a way to encode the quantitative criteria that assess the impact of every agent's action on the environment. This function is utilised to calculate an integer value for every state transition under a corresponding action, with $R(s_t, a_t, s_{t+1})$ expressing the reward obtained by an agent at state s_t when selecting action a_t, and updating the environment to state s_{t+1}. The received reward r_L is computed based on the following equation:

$$r_L = r_{load} + r_{storage} + r_{offload} + r_{step} + r_{done}, \tag{5.7}$$

where r_{load} is provided according to the success of the loading action as follows:

$$r_{load} = \begin{cases} +10, & \text{loaded from new sensor} \\ -5, & \text{attempted to load from same sensor} \\ -5, & \text{attempted to load from empty location} \end{cases} \tag{5.8}$$

In addition, when attempting the load action, $r_{storage}$ is assigned the value of -5 if the drone's storage capacity is exceeded, or 0 otherwise. Similarly, for the action of offloading data, a negative value of -5 is allocated to $r_{offload}$ if the drone attempts to offload data at an incorrect position, or 0 otherwise. Lastly, the agents receive a reward of -5 at each time step through r_{step}, as an incentive to promptly progress towards the solution, and a highly positive reward (r_{done}) is received, when the mission is successfully completed.

In summary, as per the reward function, when a drone agent successfully loads data from an unvisited sensor, a positive reward (+10) is granted, while if the drone attempts to load data that has already been collected or at a location without an available sensor (i.e., lacking an established communication link with a sensor), a negative reward (-5) is received. Similarly, when a drone attempts to offload data at a position away from the data center, a negative reward (-5) is also gained. A notably high positive reward is earned when a drone offloads the data at the data center, while ensuring all sensors have been visited and their corresponding data collected, to strongly incentivise the correct approach to accomplish the mission. Furthermore, in a technique known as reward shaping [112], each drone obtains a negative reward (-5) after each time-step to encourage achieving the solution in the minimum possible time. The selection of the granted rewards was refined iteratively and subjected to sensitivity analysis to evaluate its impact on agent behaviour and effectiveness.

It is important to note that the reward function assesses the actions chosen by individual agents, from which they acquire a local reward (r_L). Within the framework of the cooperative multi-agent setting of this environment, this local reward serves to shape the global reward (r_G), which acts as a learning signal within the transition function, enabling the calculation of the total combined reward for all drones and reflecting the overall performance of the system.

5.3 Cooperative deep reinforcement learning control

The process of reinforcement learning revolves around iterative and exploratory interactions between the agent and the environment, which aim to effectively train the agent to approach a solution to the problem. In order to maximise the expected total reward, the Q-learning algorithm [113] can be applied to any finite Markov Decision Process in this context. This algorithm produces a strategy, known as a ’policy’, which maps each tuple ’(state, action)’ to a corresponding Q-value. In simple and small-scale environments, this mapping process can be efficiently handled using a Q-table. However, in complex environments where the state space or the number of agents increases, the effectiveness of Q-tables diminishes exponentially. Given the complexity of the problem introduced in subsection 5.2.1, the method introduced in this chapter incorporates a nonlinear Q-function approximator in the form of an artificial Deep Neural Network, known as the Deep Q-network (DQN) [114], which utilises given states to derive the corresponding Q-values for each potential action.

5.3.1 Multi-agent double DQN framework

DQNs are recognised for their ability to effectively handle high-dimensional spaces by employing multiple interconnected layers of neurons in their architecture. However, when using artificial neural networks as Q-function approximators, it is important to consider the potential divergence that can occur during training. This divergence primarily arises from the fact that in basic architectures, future Q-values are evaluated using the same policy that determines the action selection. To mitigate the overestimation of Q-values in Deep Q-learning, the introduced method adopts the Double Deep Q-learning concept [110] to develop the training architecture. The approach involves the utilisation of a second distinct neural network, known as the ’target network’, to evaluate Q-values. The target network is synchronised with the main network at specific intervals during training, enabling the main network to converge towards stable targets.

Based on this architecture, a cooperative double DQN strategy, referred to as CoopD2Q, is proposed. In this strategy, each agent develops their own main and target networks, as well as an independent memory buffer. The detailed architecture and training strategy are presented in subsections 5.3.2 and 5.3.3. It is worth noting that in addition to Double Q-learning, there are other extensions of the DQN algorithm, such as Dueling DQN [115], which separates the Q-value estimation into two processes. However, given the nature of the proposed system model, a dueling architecture is found to be less efficient, as evidenced by the associated experiments.

In Double Q-learning, the calculation of Q-values for the target network is accomplished by applying the Bellman equation in the following manner:

$$y_t = r_{t+1} + \gamma max Q(s_{t+1}, a'; \theta^{-}), \tag{5.9}$$

where θ^{-} represents the target network parameters and γ indicates the discount factor. Therefore, the gradient loss is computed as follows:

$$L_t(\theta) = \mathbb{E}_{s,a,r,s_{t+1}}[(r_{t+1} + \gamma max Q(s_{t+1}, a'; \theta^{-}) - Q(s, a; \theta))^2], \tag{5.10}$$

where $Q(s, a; \theta)$ represents the predicted Q-value and $\mathbb{E}_{s,a,r,s_{t+1}}$ indicates the Q-function of the current policy.

Although the target network effectively addresses the challenge of converging to unstable targets, the issue of generalisation over correlated input data still persists. To mitigate this concern, a memory buffer, commonly referred to as 'experience replay', is employed to store recent tuples of (s, A, s', r). At specific update intervals, a mini-batch of these samples is selected from the memory buffer to update the parameters using the gradient descent calculation of Equation 5.10. In the devised multi-agent DQN architecture, each agent is trained using their own individual main and target network, alongside a distinct memory buffer, as illustrated in Figure 5.2.

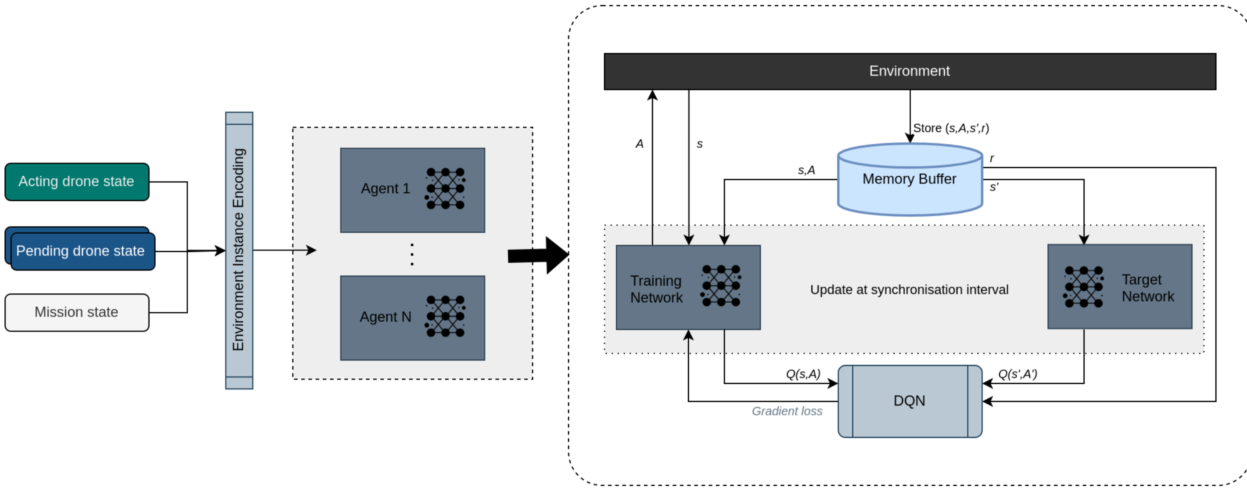


Figure 5.2: The proposed cooperative double DQN framework for drone route planning.

5.3.2 Model structure

In order to accurately estimate the Q-function, the training and target networks of each agent are constructed utilising the sequential model of the Keras API, as depicted in Figure 5.3. The input dimensions corresponding to the cardinality of the environment's state set are then reduced to a lower space using an Embedding layer. This is succeeded by a Reshape layer that transforms the preceding output into a 1-rank tensor of the same size. Following this, four Dense layers are applied, whose parameters are trained and updated during the back-propagation process to facilitate network convergence. Each Dense layer employs the ReLU activation function, except for the final layer, which utilises a linear activation function, having output dimensions equivalent to the action space of the environment. Utilising Dense layers enables the network to acquire knowledge from the combined features of the preceding layers, albeit at the expense of increased computational complexity.

5.3.3 Training strategy

In order to enhance the learning capability of the DQN agents, this study utilises an ϵ-greedy training strategy, where the agents initially choose random actions, but gradually shift towards selecting

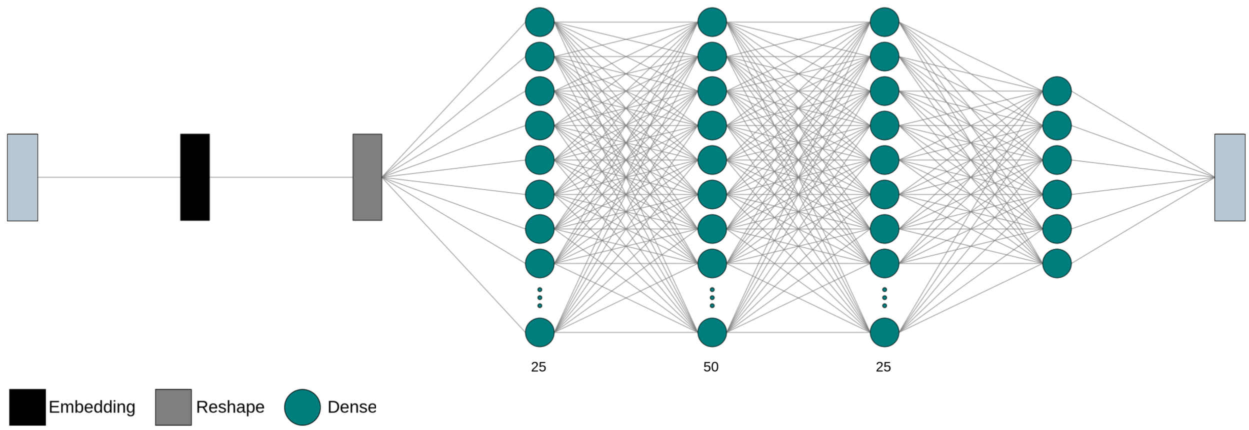


Figure 5.3: The structure of the employed DQN model. The Embedding layer is denoted by the black colour, the Reshape layer is shown in gray, and the Dense layers are represented in teal.

actions with the maximum known Q-value, as expressed in Algorithm 3. A control hyperparameter is employed to regulate how ϵ is gradually annealed to 0.1, ensuring the balance between exploration and exploitation during the learning process.

Algorithm 3: Action selection strategy

Input : The predicted Q-values for each agent: Q_i;
the current time-step of training: $timestep$;
the number of random time-steps: max_random_steps;
the ϵ-greedy variable: ϵ; number of agents: num_agents;
Output : The set of selected actions of each agent:
$A = \{a_1, a_2, ..., a_i\}$;

$p \leftarrow random(0, 1)$;
if $timestep < max\_random\_steps$ **or** $\epsilon > p$ **then**
  for $i \leftarrow 0$ **to** num_agents **do**
    $a_i \leftarrow random(0, 6)$;
    $A \leftarrow (A, a_i)$;
else
  for $i \leftarrow 0$ **to** num_agents **do**
    $a_i \leftarrow argmax(Q_i(s, a))$;
    $A \leftarrow (A, a_i)$;
return A;

Rather than training the networks for a fixed number of epochs, the proposed strategy continues indefinitely until a satisfactory global reward is achieved, as depicted in Algorithm 4. At the start of each epoch, the environment is initialised to a random state, while in each epoch, the training persists for a specific number of steps (max_train_steps) to enhance efficiency. During each step, the agents employ the ϵ-greedy strategy outlined in Algorithm 3 to select actions, and the environment is updated accordingly using the transition function. Following this, each computed tuple (s, a_i, s', r_G) is stored in the agents' memory buffer, which at any given time, retains the last $memory\_size$ tuples.

To increase training stability, the target network is updated only after a certain number of actions, by utilising a random mini-batch sampled from the memory, as regulated by the hyperparameter ($target\_update\_interval$). Moreover, the learning process is controlled by the $main\_update\_interval$, which determines the frequency of updating the weights of the main

Algorithm 4: Multi-agent DQN training strategy

```
r_G ← −∞;
while r_G < optimal_reward do
    s ← randomEnvState();
    for train_step ← 0 to max_train_steps do
        get set A of actions using Algorithm 3;
        decay ϵ;
        transition env based on s and A, get s' and r_G;
        for i ← 0 to agent_number do
            memory_i ← [s, a_i, s', r_G]);
        if timestep % actions_before_update = 0 then
            for i ← 0 to agent_number do
                get a random sample from memory_i;
                update target Q_i-values using Eq.5.9;
                calculate gradient using Eq. 5.10;
        if timestep % target_update_interval = 0 then
            for i ← 0 to agent_number do
                network_i ← target_network_i;
```

training network with the target network. The main hyperparameters governing the learning process are detailed in Table 5.2.

5.4 Evaluation

In order to evaluate the effectiveness of the proposed multi-agent DQN strategy within the defined system model, a range of experiments are conducted in two unique phases. During the first stage, three DQN algorithms are employed to train a single agent, with the objective of assessing the convergence performance of each model. During the second stage, given the considerable time required for the models to converge, two scenarios are devised in which two cooperative agents are trained simultaneously. In the first scenario, the conventional DQN algorithm is employed, and in the second scenario, the introduced Double DQN strategy is utilised to train both agents. For both scenarios a spatial environment was designed as an 8 x 8 grid, with each cell representing a length of 100 meters.

5.4.1 Experimental environment

The training for all scenarios is performed on a Linux system, utilizing the CUDA API on an Nvidia GPU with a computation capability of 6.1. In the initial stage, a custom environment is generated, featuring the placement of 3 IoT sensors at predetermined locations. For each algorithm, a single

Table 5.2: The key hyperparameters that control the learning process.

Hyperparameter	**Description**
lr	Learning rate
max_train_steps	Training steps per epoch
exploration_steps	Steps for exploration
greedy_steps	Steps for exploitation
memory_size	Length of the memory buffer
target_update_interval	Target network update frequency
main_update_interval	Main network update frequency

agent undergoes training for a duration of $250k$ time-steps. The Adam optimiser is utilised to update the network parameters every 750 steps based on a mean squared error (MSE) loss, employing a learning rate of 10^{-3}. Lastly, the memory size is constrained to $50k$, while the maximum number of training steps per epoch is established at 300.

In the second evaluation stage, the employed environment features the presence of 4 IoT sensors. In the two considered scenarios, the two cooperative agents undergo training for a total of $10k$ epochs or until a sufficient global reward is attained. During this stage, the target network is updated at intervals of 1000, while utilising a batch size of 32. The memory size and the maximum steps per epoch remain consistent at $50k$ and 300, respectively, while the exploration steps are set to 1000, and the greedy steps to $40k$. The parameters of the networks are updated using the Adam optimiser with a learning rate of 5×10^{-4} in this case. In addition, these scenarios employ the Huber loss for the back-propagation in the target network as it demonstrates increased sensitivity to outliers compared to MSE. The Huber loss quantifies the error between the predicted and target Q-values, which are computed based on the Bellman equation (5.9), as follows:

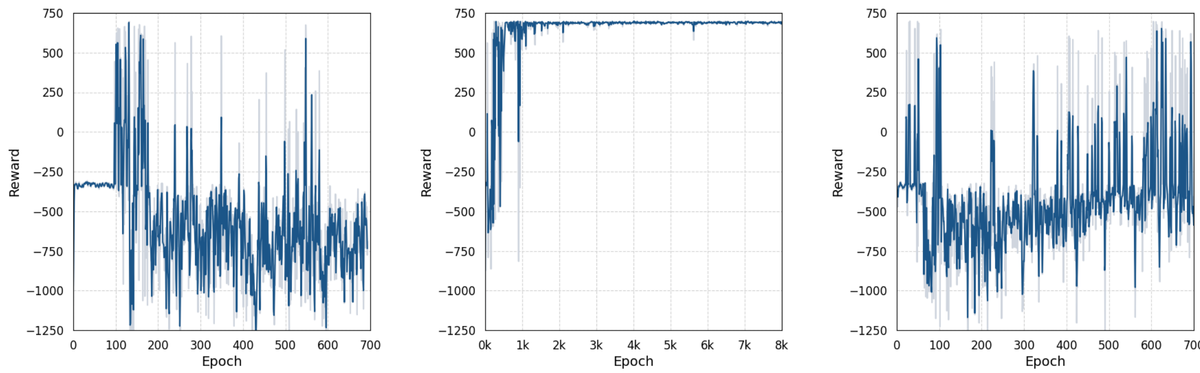
$$L_\delta = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & for \left|(y-\hat{y})\right| \leq \delta, \\ \delta(|y-\hat{y}| - \frac{1}{2}\delta), & otherwise \end{cases} \tag{5.11}$$

where δ represents a hyperparameter that governs the threshold at which the function shifts from quadratic to linear.

5.4.2 Learning performance

Table 5.3 outlines the respective training durations for each algorithm. While the single agent scenarios reach completion in a relatively quick manner during the training process, the multi-agent setting demands a significant amount of time for the models to converge due to the exponential expansion of the state space. Figure 5.4 illustrate the training results during the initial evaluation stage, corresponding to the standard DQN, Double DQN, and Dueling DQN algorithms.

The conducted experiments reveal that the DQN algorithm falls short of achieving convergence



(a) DQN (b) Double DQN (c) Dueling DQN

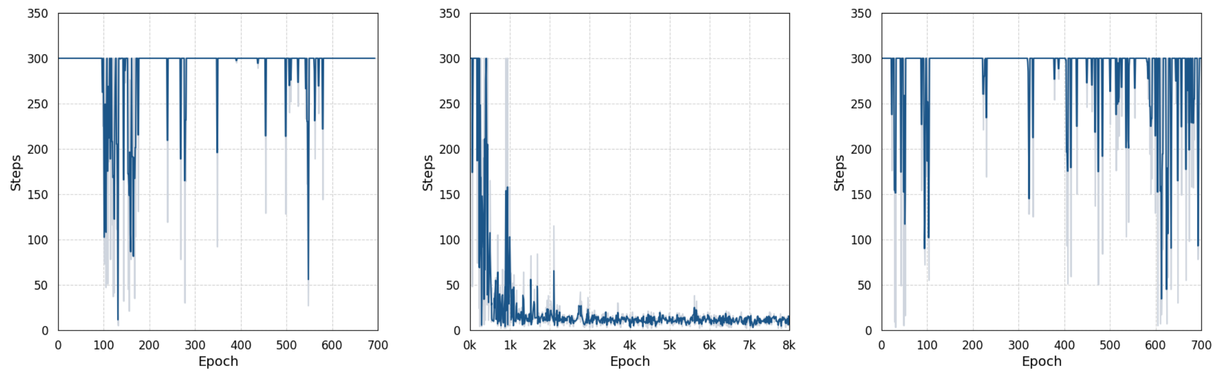
Figure 5.4: Cumulative reward obtained by the three DQN algorithms in the single agent environment.

and even fails to sustain a positive reward after completing $250k$ time-steps. On the other hand, Dueling DQN exhibits the ability to maintain a consistent positive reward, although it also falls short of full convergence. By contrast, the Double DQN algorithm offers the highest level of stability, as observed through its early convergence and the sustained maintenance of a maximum reward throughout the training duration.

The outcomes of the experiments are also reflected in the relationship between the number of training steps and the number of epochs, as depicted in Figure 5.5. In the standard DQN case, the training is continued for max_train_steps in each epoch until the end. In contrast, the Dueling approach successfully reduces the required steps to approximately 200 towards the later epochs. Remarkably, the Double DQN algorithm demonstrates very high efficiency, necessitating a minimal

Table 5.3: Time required for each algorithm to finish training the agents.

Algorithm	Training Time
DQN (single-agent)	29 min
Double DQN (single-agent)	28 min
Dueling DQN (single-agent)	29 min
DQN (multi-agent)	30 hours
CoopD2Q (multi-agent)	11 hours



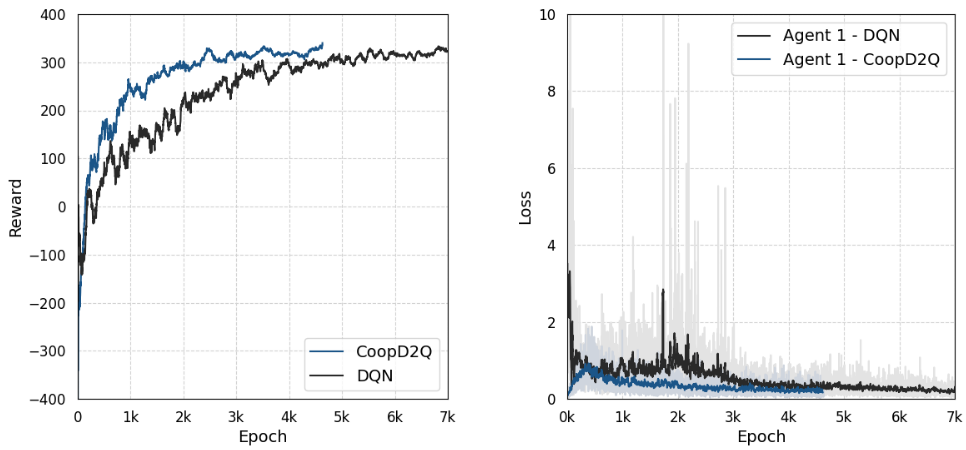
(a) DQN (b) Double DQN (c) Dueling DQN

Figure 5.5: Required training steps per epoch by the three DQN algorithms in the single agent environment.

number of steps to conclude training in each epoch, even as early as 1000 epochs have passed.

The findings obtained from the experiments provide support for the hypothesis that the simple DQN algorithm is susceptible to overestimating Q-values, whereas the Double DQN algorithm effectively addresses this issue. In contrast, while the Dueling DQN model eventually achieves convergence, the training process demonstrates comparatively lower efficiency. This can be attributed to the fact that the Dueling architecture aids in distinguishing between situations where the choice of action is insignificant and cases where action selection is crucial. However, in the environment under consideration, the actions tend to exhibit similar Q-values, thus diminishing the informative value of the estimations provided by the Dueling DQN approach.

When examining the performance of the standard DQN algorithm in comparison to the introduced Double DQN approach in a multi-agent context, a similar trend emerges. The global running reward attained by both agents throughout the training duration is depicted in Figure 5.6a. CoopD2Q establishes a significant lead over standard DQN early on and achieves the target reward prior to 5,000 epochs. On the other hand, exhibits slower growth, reaching the target reward values at over 7000 epochs. The higher efficiency of CoopD2Q is further demonstrated when evaluating the training loss for each method, as illustrated in Figure 5.6b which depicts the loss of the agents versus the epoch number, for the two algorithms. Despite the steady decrease in loss observed



(a) Global cumulative reward for the standard DQN and CoopD2Q algorithms.

(b) Training loss of the CoopD2Q and the standard DQN algorithms.

Figure 5.6: Running reward and agent loss during training in the multi-agent setting.

over time in both algorithms, the DQN algorithm shows significantly higher and more variable loss values, highlighting a significant instability in learning performance.

The disparity in performance between the two algorithms, although still significant, is less evident when examined in the multi-agent setting. This observation can be attributed to the substantial level of cooperation among the agents and the relatively predictable and stable nature of their interactions. As a result, the overestimation of Q-values is less pronounced. Therefore, while the inclusion of a distinct target network in CoopD2Q proves advantageous in scenarios where Q-values are prone to overestimation, the increased complexity in the architecture (requiring an additional network per agent) slightly impedes the convergence of the model.

5.4.3 Testing performance

The evaluation of the trained agents' performance was carried out in both a single-agent and a multi-agent setting, encompassing a total of 3000 tests. The environment was configured to a random state for each test to ensure a diverse range of initial conditions. The results of these evaluations are presented in Tables 5.4 and 5.5, providing insights into the average reward attained by each algorithm, the average number of steps taken per episode (limited to a maximum of 50), and the success rate of completing the mission.

In the context of the single-agent evaluation, exceptional performance is demonstrated by the Double DQN algorithm, successfully completing all missions with maximum reward and minimum steps. In contrast, both standard DQN and Dueling DQN achieve a low reward, with the latter slightly surpassing the former by successfully finishing approximately 17% of the missions. It is worth noting that although standard DQN and Dueling DQN exhibit subpar performance, it is reasonably possible that with an increased training limit, they would eventually converge towards adequate performance levels. This comparative analysis primarily aims to provide an informative insight into the performance potential of these methods within the threshold at which the Double DQN reaches optimal outcomes. Therefore, it provides a guiding indication of their prospective efficiency in the more time-intensive context of the multi-agent setting.

Within the multi-agent setting, CoopD2Q outperforms the standard DQN method and demonstrates higher performance, even when examining only the cases of successful missions, as depicted in Table 5.5. Although the DQN algorithm offers satisfactory outcomes, the significant difference

Table 5.4: Agent performance in the single-agent setting.

Algorithm	**Average Reward**	**Average Steps**	**Success Rate**
DQN	-68.84	50	0%
Double DQN	690.97	10.02	100%
Dueling DQN	47.11	42.47	17.46%

Table 5.5: Agent performance in the multi-agent setting. The 'Overall' metrics represent performance across all test cases, while the 'In successful' metrics focus specifically on the cases where the mission was completed successfully.

Algorithm	**Overall**			**In successful**	
	Average reward	**Average steps**	**Success Rate**	**Average Reward**	**Average steps**
DQN	226.71	17	66%	343.50	12
CoopD2Q	256.92	14.25	71.5%	359.33	11

in training duration (close to 3x times higher) highlights its considerably lower efficiency.

5.4.4 Ablation study

An ablation study is conducted on the multi-agent environment, to examine the impact of the onboard storage capacity limit. The environment under consideration involves the use of 4 IoT sensors and 2 drones, excluding the storage constraint specified in Equation 5.4. The study entails training the cooperative agents by utilising the introduced CoopD2Q method, to investigate variations in learning capability. Remarkably, the training duration demonstrates a notable decrease, reaching the optimal global reward in just 1.6 hours, as compared to the 11 hours required previously. The attained cumulative reward, and the mission success rate, are presented in Figure 5.7. The outcomes demonstrate a highly stable training process, leading to a rapid increase in the success rate. These findings highlight the negative impact of an additional constraint on the decision space's size and the resulting challenge in training multiple agents with distinct decision-making policies.

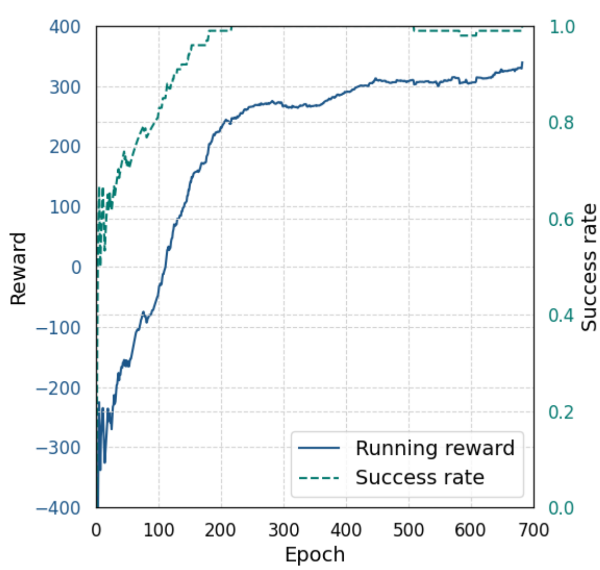


Figure 5.7: Global cumulative reward and success rate in the multi agent setting, without the storage constraint. Note the two lines are plotted using different y-axes.

5.5 Discussion

This chapter built upon the knowledge gained in the previous chapters and focused on harnessing the full potential of deep learning techniques to solve complex decision-making problems in the context of IoT applications. Specifically, the chapter investigated a multi-agent deep reinforcement learning approach and introduced a new framework to address the route planning problem in a devised UAV-assisted IoT data collection scenario.

The problem under consideration was formulated as an extension of the m-TSP with additional constraints, such as considering the drones' onboard storage capacity. The objective was to develop efficient decision-making policies that allow multiple drone agents to cooperatively tackle the challenge of jointly optimising their routes, while ensuring that each sensor is visited only once and all data are collected without exceeding the storage limits. In response to this need, the chapter proposed the CoopD2Q framework, which is based on a Double DQN architecture.

To facilitate the application of the multi-agent DRL approach in the devised problem, a new environment capable of encoding and decoding unique mission instances was designed and developed. This environment, in conjunction with a novel reward function, provided a global learning signal to the agents, aiming to maximise their long-term rewards. The reward function was carefully designed to guide the agents towards optimal decision-making and route planning strategies, with high efficiency.

The experimental analysis served as empirical validation for the effectiveness of the CoopD2Q approach in solving the complex route planning problem in the UAV-assisted data collection context. The results highlighted several advantages of the proposed method over different DQN methods. Through an enhanced training stability, the training process in CoopD2Q contributes to more reliable and consistent model convergence. In addition, the models achieve the convergence significantly quicker compared to their counterparts. Lastly, the agents trained through CoopD2Q showcase improved performance when solving problem instances in evaluation scenarios. They are able to achieve higher success rates, obtain larger rewards, and arrive at the optimal solution in fewer time-steps, indicating the higher efficiency of the introduced method.

One potential weakness in this approach is the synchronisation problem that may arise during the agent deployment. If the drones fail to synchronise effectively, it can lead to inefficiencies and suboptimal route planning outcomes. Without proper synchronisation, drones may inadvertently visit the same sensor multiple times or miss certain sensors altogether, resulting in incomplete data collection or redundant data acquisition. To address this, it is crucial to incorporate such cases into the training process, simulating errors or delays in the communication, and adjusting the learning process accordingly. However, this induces additional complexity during training.

The findings from this chapter highlight the significant contributions of applying DRL techniques in UAV-assisted IoT architectures. The successful implementation of the CoopD2Q approach signifies a leap forward in addressing the complexities associated with route planning, especially when considering additional constraints. This research has direct implications for areas such as environmental monitoring, disaster response, and surveillance systems, where efficient route planning and data collection are crucial.

By harnessing the full potential of DRL, this chapter paves the way for the development of sophisticated decision-making schemes in the context of advanced IoT applications. The integration of DRL techniques with UAV-assisted IoT networks opens up possibilities for improving efficiency, scalability, and adaptability in various domains. The findings provide valuable insights into the dynamics of multi-agent DRL but also establishes a solid foundation for further advancements in the field, as it can be easily adapted to an assortment of optimisation problems in IoT.

Chapter 6

Conclusions

6.1 Summary of research

The introductory chapter of the thesis laid the groundwork for the research on efficient drone control algorithms, by providing the essential context and outlining the involved objectives of the study. The chapter discussed a brief background on the Internet of Things, and offered a concise overview of the history and significance of drone technologies, with an emphasis on their diverse application across various industries, such as agriculture, remote sensing, or emergency response. The motivation behind the research stemmed from the growing demand for drone applications, with requirements for improved drone performance. The pressing challenges faced by existing drone control algorithms, including high energy consumption, limited flight time, complex navigation decisions, and suboptimal performance in varying environmental conditions, were identified. Based on these challenges, specific thesis aims and objectives were established, to address them effectively. The core aim was to develop and optimise efficient control algorithms to govern the collective behaviour of drones organised in swarms, to enhance the overall performance, and ensure reliable operation under different environmental scenarios.

Chapter 2 was dedicated to exploring the background of the research and was structured to offer a comprehensive analysis of the relevant literature and research that formed the foundation of the thesis. The chapter commenced with an overview of the Internet of Things paradigm, establishing its crucial role in the current technological landscape. The focus then transitions to drone networks, covering their unique features and practical use-cases. This includes an in-depth look at localisation and tracking, alongside the challenges and current methods employed in route planning optimisation. This analysis facilitated a profound comprehension of the present capabilities and limitations of drone networks inside the IoT. The latter part of the chapter delved into the role of AI in drone swarm management. Initially, it evaluated the implementation of machine learning in drone control and subsequently explored the utilisation of deep reinforcement learning techniques to achieve efficient drone navigation. Through this analysis, the potential advantages and challenges of integrating AI-driven approaches in drone swarm management were unveiled. By synthesising an extensive body of literature, the second chapter facilitated a profound understanding of the background pertaining to IoT, drone networks, and the prospective applications of AI in drone control. This critical analysis served as a strong basis for the ensuing chapters, wherein novel approaches to optimise drone control algorithms were proposed, and the prospects of efficient drone swarm management through AI-driven methodologies were further explored.

Chapter 3 offered a thorough investigation into the application of deterministic methods for effectively controlling drone swarms. This research domain has been steadily gaining prominence due to the vast potential of drone swarms across diverse fields, including search and rescue, which this research targeted. Deterministic approaches emphasise predictability in system behavior by employing predefined rules or mathematical models to govern the actions and interactions within the drone swarm. These techniques provide a structured framework for managing drone movements, interactions, and task execution, thereby ensuring a level of certainty in complex multi-agent systems. The main focus of this chapter was to introduce, analyse, and assess a novel deterministic technique for drone swarm control, exploring its applicability, strengths, limitations, and potential for further enhancements. While the following chapters explored AI-based solutions, this chapter emphasised the significance and relevance of deterministic methods, particularly in scenarios where computational efficiency and meticulous control over system behavior are paramount considerations.

Chapter 4 explored the integration of AI with drone swarm coordination, specifically focusing on group formations within the fleet. The chapter delved into the powerful capabilities of deep learning in cluster formation, providing an efficient organisational structure for the drones. Unlike conventional deterministic methods, this approach enabled dynamic adaptation of the swarm, enhancing efficiency dynamically. By combining the predictability of deterministic techniques discussed in the previous chapter with the adaptive potential of AI, the chapter advocated for the integration of deep learning in swarm control. This synthesis presented a compelling argument for the synergistic utilisation of both approaches to advance drone control and enhance the overall performance and effectiveness of drone swarms.

Chapter 5 focused on the crucial aspect of multi-agent drone route planning optimisation, which becomes imperative in complex operations and larger environments. After exploring the significance of drones in the IoT context and investigating deterministic and AI-assisted methods for swarm coordination in the previous chapters, this chapter emphasised the need for efficient route planning when dealing with multiple drones, in complex combinatorial optimisation problems. The research delved into the complexities of multi-agent systems, where optimising routes for individual drones must be synchronised with overall mission success and constraints satisfaction. The algorithmic framework proposed in this chapter employed innovative computational strategies and optimisation techniques based on deep reinforcement learning, to enhance the UAV-assisted data collection efficiency in IoT applications. Each drone in the system operated as an independent decision-making entity, but was also a part of the coordinated fleet. The developed system leveraged neural networks, with each drone building its unique network, trained through intricate trade-offs between exploration and exploitation, enabling the development of effective action-selection policies. By employing these advanced computational approaches and optimisation methods, the chapter significantly improved the coordination and efficiency of multi-agent drone route planning for IoT applications, resulting in more efficient data collection and mission accomplishment.

6.2 Key findings

6.2.1 Deterministic drone control

The first core findings of the research involved the evaluation of the new deterministic control scheme. The introduced algorithm utilised readings of the RF signal strength transmitted by an IoT device held by the moving target, to effectively lead the drones to close distance. By decomposing the tracking task into two stages – individual search and cooperative tracking – the algorithm demonstrated enhanced efficiency compared to trilateration based methods, particularly under high signal attenuation conditions. Throughout the evaluations, key performance indicators were used to assess the system's performance:

- **Minimum time**: The introduced control scheme exhibited resilience to increasing values of noise, showcasing robustness under diverse conditions. In contrast, the trilateration algorithm's accuracy declined substantially under noisy conditions, resulting in a significant faltering of its tracking performance and the time required to reach the target.

- **Average distance**: The proposed algorithm demonstrated little impact from increased noise, maintaining consistent performance, while the trilateration algorithm suffered from decreased accuracy as noise levels rose, leading to inability in maintaining close distance to the sensor.

- **Halting cycles**: Under conditions of low noise, both algorithms achieved comparable results in the duration of time the nearest drone remained static. However, the proposed algorithm demonstrated an edge as noise levels increased towards realistic or higher values.

- **Sustained proximity cycles**: The proposed algorithm clearly outperformed the trilateration method in sustaining proximity to the target over time.

In the next stage of the assessment, the impact of drone velocities on the performance of the two methods was evaluated. The proposed algorithm exhibited enhanced efficiency in target tracking under realistic noise levels, regardless of drone velocity. Lastly, the efficiency of the deterministic control scheme was assessed against increased target velocities and the impact of the swarm's size on the overall tracking process. The algorithm's performance showed minor fluctuations across different target speeds, while the number of drones in the fleet had a varying influence on the system's efficiency, with higher performance observed for higher swarm sizes at low drone speeds, when the total size does not exceed seven drones.

Overall, the cooperative deterministic scheme presented in the first part of the research offered a promising solution for efficient drone swarm tracking of mobile IoT targets. Its reliance on RF signal deviations instead of distance calculations, strategic decomposition of the tracking task, and ability to maintain accuracy in noisy environments make it a valuable advancement in the field of drone control algorithms. The findings contribute to enhancing drone swarm management capabilities, showcasing its potential for a wide range of practical applications, particularly in the field of search and rescue.

6.2.2 Cluster formation through deep learning

The second core part of the research outlined the key findings of evaluating the potential effectiveness of integrating AI-assisted control into the previously introduced deterministic algorithm, to govern cluster formation inside the fleet. A comprehensive comparison of the two schemes was performed under diverse conditions, measuring three essential metrics: the time needed for the swarm to approach the IoT sensor, the mean distance covered by each individual drone during tracking, and the overall distance traveled by the entire fleet.

The outcomes revealed that the AI-assisted control algorithm exhibited notable advantages over the purely deterministic approach. Specifically, it showcased improved time efficiency in locating the sensor and demonstrated higher overall fleet efficiency. Moreover, the AI-assisted method resulted in reduced average distance covered by individual drones during the tracking procedure, indicating enhanced operational efficiency. Notably, the analysis indicated that optimal performance was achieved when the AI-assisted algorithm utilised a specific termination threshold for clustering, showcasing the greatest efficiency in distances traversed by the swarm, leading to reduced energy expenditure in each mission.

Finally, this stage of the research explored the advantages of employing the "elbow" technique, dynamically determining the cluster count, rather than using a fixed value. The investigation highlighted the superiority of the dynamic approach, with the swarm reaching the sensor more efficiently, emphasising the value of adaptive strategies in achieving optimal tracking outcomes.

6.2.3 Multi-agent route planning under constraints

In the third and final part of the research, the evaluation involved the investigation of a new multi-agent cooperative DRL policy to optimise UAV-assisted data collection in IoT networks, under constraints. At first, the convergence performance of three DQN algorithms was assessed by training a single agent in a custom environment with three IoT sensors. The Double DQN algorithm exhibited the highest stability, achieving early convergence and sustained maximum reward throughout the training duration. In contrast, the standard DQN algorithm failed to achieve convergence and sustain a positive reward. The Dueling DQN approach demonstrated consistent positive reward, but its convergence was comparatively less efficient.

Next, the evaluation focused on training two cooperative agents simultaneously in an environment with four IoT sensors, using the introduced CoopD2Q method. The new DRL algorithm, which relied on the extension of Double DQN for multi-agent contexts, outperformed the standard DQN algorithm, achieving the target reward in fewer epochs. However, the added complexity slightly impacted the model's convergence due to the requirement of an additional network per agent, heavily affecting the training time.

In both single-agent and multi-agent settings, CoopD2Q demonstrated superior performance compared to the standard DQN algorithm. CoopD2Q achieved higher rewards and success rates, completing missions in a more efficient manner. Finally, an ablation study conducted in the multi-agent environment revealed that the introduction of an additional constraint, such as the storage capacity limit, considerably impacts the training duration. Without an extra constraint, the training process demonstrates a rapid decrease in training time, indicating the challenge of training multiple agents with distinct decision-making policies under constrained conditions.

Overall, the findings support the effectiveness of the CoopD2Q algorithm for multi-agent cooperative tasks and highlight the benefits of addressing the issue of Q-value overestimation, and improving training stability through the proposed DRL approach. The study provides valuable insights into the convergence performance of various DQN algorithms in both single-agent and multi-agent contexts, shedding light on their potential applicability and efficiency in real-world scenarios, for various problems of combinatorial optimisation nature, which are often involved in IoT networks.

6.3 Addressing thesis objectives

The primary aim of this thesis was to advance the field of drone control within IoT networks, specifically focusing on optimising route planning for efficient and intelligent operations. By introducing adaptive schemes capable of real-world problem-solving, the ultimate goal was to contribute to the advancement of drone control systems, bringing tangible benefits across various sectors and applications in IoT. This aim has been successfully accomplished through the fulfillment of three key objectives: 1) investigate and devise deterministic techniques for cooperative drone control, 2) investigate and devise deep learning methods for swarm coordination and cluster formation, and 3) Investigate and devise a deep reinforcement learning framework for optimising drone route planning.

Firstly, the thesis extensively explored deterministic techniques for cooperative drone control, delving into pre-defined strategies, evaluating their performance, and identifying potential limitations. By introducing a new deterministic control algorithm, and critically analysing its implementation, a level of certainty was achieved when governing the navigation of drones in unstructured environments, applicable in remote IoT networks. Through this method, the thesis laid the foundation for incorporating cutting-edge methodologies, particularly those driven by AI, to enhance and complement the deterministic control mechanisms. This formed the basis for the investigation and development of more flexible and adaptable drone control systems, aimed at further improving their performance and robustness in real-world scenarios.

Secondly, the thesis examined and developed deep learning methods to optimise swarm coordination through adaptive cluster formation. This in-depth exploration of AI solutions based on deep learning, was essential for understanding how more advanced algorithms can contribute to efficient drone fleet management. By focusing on cluster formation, the research sought to enhance energy efficiency and mission execution effectiveness. The incorporation of deep learning methodologies proved to be pivotal in optimising cluster formation, enabling drones to work cohesively and achieve superior operational efficiency in more complex IoT environments.

The third objective centered on the design and implementation of a deep reinforcement learning framework for optimising drone route planning, under certain constraints. This novel approach managed to take drone control to a higher level by allowing drones to learn and adapt continuously based on real-time feedback, through a constant reward-based approach. Deep reinforcement learning enabled drones to autonomously learn from their interactions with the environment, allowing them to make intelligent decisions and optimise route planning dynamically. By reducing the reliance on human intervention, the research successfully achieved higher levels of autonomy and efficiency in drone operations, which is required to realise the required degree of autonomous operation in data collection applications inside IoT networks.

The integration of advanced AI-driven techniques within traditional deterministic strategies has significantly transformed the capabilities of drone control within IoT networks. Furthermore, through the novel DRL framework, the thesis demonstrated the development of intelligent, adaptable, and efficient drone control systems that can effectively navigate complex environments, optimise cluster formation, and dynamically plan routes based on real-time feedback. These achievements have profound implications across diverse industries, as they enable drones to perform a wide range of tasks more effectively and autonomously, leading to increased efficiency and reduced operational costs in various applications, particularly in the emergency response and remote sensing sectors.

6.4 Strengths and limitations

One of the main strengths of this thesis lies in its comprehensive approach to addressing the challenges in drone control within IoT networks. The investigation of new deterministic techniques, followed by the successful integration of state-of-the-art AI methodologies, demonstrates the research's innovative nature and its contribution to the evolution of drone control systems. The exploration of deep learning methods for swarm coordination and cluster formation showcases the research's practical significance in providing adaptive management, enhancing energy efficiency and mission execution effectiveness, particularly in dynamic environments. Additionally, the development of a novel deep reinforcement learning framework for optimising route planning highlights the thesis's commitment to pushing the boundaries of autonomous drone control, under the presence of limitations and constraints. By achieving the aim of innovating drone control within IoT networks, the thesis exhibits its potential to significantly impact industrial sectors, leading to more efficient, reliable, and intelligent drone missions.

Despite its strengths, this research also exhibits certain limitations. One such drawback pertains to the complexity and computational overhead associated with the AI-driven methodologies introduced. The implementation of deep learning networks inside the DRL framework in real-world applications may need special computational resources and time, requiring intricate planning in resource-constrained environments. Moreover, the evaluation of these techniques through simulations may not fully capture the nature of real-world scenarios, necessitating further validation through physical drone experiments. Additionally, while the research successfully showcases the potential advantages of AI-driven techniques, it is essential to acknowledge the challenges of interpretability and explainability in AI-based drone control systems. The use of AI algorithms often leads to "black-box" decision-making, which may hinder understanding and trust in critical operations. As AI becomes more integrated into drone control, addressing these interpretability challenges becomes crucial for ensuring safe and reliable operations.

6.5 Future directions

The research conducted in this thesis opens up several exciting and promising future directions that can further advance the field of drone control within IoT networks. Firstly, as AI technologies continue to evolve, future research can focus on developing more interpretable and explainable AI-driven drone control systems. Addressing the "black-box" nature of AI algorithms will be crucial for gaining trust and confidence in autonomous drone operations. Techniques such as model introspection, attention mechanisms, and interpretable deep learning architectures [116] can be investigated to provide insights into the decision-making process of AI-based drone control systems. Additionally, investigating the integration of uncertainty estimation methods in AI algorithms can help assess the reliability and confidence of AI-driven decisions, making the systems more transparent and accountable.

Secondly, future research can further explore the synergies between AI and deterministic techniques in drone control. Hybrid methods that intelligently switch between deterministic and AI-based strategies depending on the complexity of the task and environmental conditions may be investigated, to potentially enhance the system's adaptability and performance. Investigating how AI can optimise the parameters and configurations of deterministic algorithms may also further improve their efficiency and effectiveness.

Furthermore, the safety and security of drone operations constitute critical concerns for future research. Addressing issues such as cyber threats and privacy concerns in AI-driven drone control systems will be essential for deploying drones in critical IoT infrastructures Developing robust fail-safe mechanisms and testing the resilience of AI-based control systems against adversarial attacks will be pivotal in ensuring safe and secure drone missions.

Finally, looking ahead, another promising avenue for future research in the context of drone control within IoT networks lies in exploring the potential of quantum optimisation techniques. Quantum computing, with its ability to efficiently solve complex optimisation problems, through properties of quantum hardware has the potential to revolutionise the field of drone route planning and resource allocation. Traditional optimisation algorithms may yet face challenges in handling larger combinatorial problems, which are common in drone swarm coordination and route planning inside complex IoT environments. Quantum optimisation algorithms, such as Quantum Annealing [117] and Variational Quantum Algorithms [118], offer the advantage of leveraging quantum principles like superposition, tunneling, and entanglement to investigate a vast solution space and find optimal solutions rapidly. Algorithms based on these principles can enable the simultaneous exploration of multiple possible drone routes and configurations, resulting in more optimal determination of flight paths. Therefore, integrating quantum optimisation into drone control systems could lead to significant advancements in efficiency and performance. Moreover, quantum optimisation could enhance resource allocation strategies, such as energy-efficient task assignment and dynamic adaptation to changing environmental conditions. By harnessing quantum computing's power, drone control systems may achieve unprecedented levels of scalability and adaptability, supporting more extensive swarm missions and highly dynamic scenarios. However, it is important to acknowledge that as of writing this thesis quantum computing is still in its early stage, and practical quantum hardware capable of tackling real-world problems remain a significant challenge. As such, quantum computing for drone control remains a future possibility rather than an immediate implementation. Nevertheless, as the field of quantum computing matures, exploring its application to drone control will be a captivating and impactful direction for future research.

Bibliography

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé *et al.*, "Vision and challenges for realising the internet of things," *Cluster of European research projects on the internet of things, European Commision*, vol. 3, no. 3, pp. 34–36, 2010.

[3] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[4] M. Campion, P. Ranganathan, and S. Faruque, "A review and future directions of uav swarm communication architectures," in *2018 IEEE international conference on electro/information technology (EIT)*. IEEE, 2018, pp. 0903–0908.

[5] M. Bacco, A. Berton, E. Ferro, C. Gennaro, A. Gotta, S. Matteoli, F. Paonessa, M. Ruggeri, G. Virone, and A. Zanella, "Smart farming: Opportunities, challenges and technology enablers," *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*, pp. 1–6, 2018.

[6] K. Eliason. (2016) Unsplash. [Online]. Available: https://unsplash.com/photos/black-and-gray-quadcopter-drone-in-mid-air-GBaEtTw7UB8

[7] (2024) Quantum-systems vector. Quantum-Systems. [Online]. Available: https://quantum-systems.com/vector

[8] M. Bryson and S. Sukkarieh, "Co-operative localisation and mapping for multiple uavs in unknown environments," in *2007 IEEE aerospace conference*. IEEE, 2007, pp. 1–12.

[9] F. Mattern and C. Floerkemeier, *From the Internet of Computers to the Internet of Things*. Springer, 2010.

[10] S. H. Shah and I. Yaqoob, "A survey: Internet of things (iot) technologies, applications and challenges," *2016 IEEE Smart Energy Grid Engineering (SEGE)*, pp. 381–385, 2016.

[11] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[13] S. W. Loke, "The internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds," *arXiv preprint arXiv:1507.04492*, 2015.

[14] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[15] D.-Y. Kim and J.-W. Lee, "Integrated topology management in flying ad hoc networks: Topology construction and adjustment," *IEEE Access*, vol. 6, pp. 61 196–61 211, 2018.

[16] Z. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 136–142, 2018.

[17] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[18] A. Mahajan, N. Potnis, K. Gopalan, and A. Wang, "Urban mobility models for vanets," in *2nd IEEE International Workshop on Next Generation Wireless Networks*, vol. 33, 2006, p. 38.

[19] A. Srivastava and J. Prakash, "Future fanet with application and enabling techniques: Anatomization and sustainability issues," *Computer science review*, vol. 39, p. 100359, 2021.

[20] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE communications surveys & tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[21] G. Churyumov, V. Tkachov, V. Tokariev, and V. Diachenko, "Method for ensuring survivability of flying ad-hoc network based on structural and functional reconfiguration," *Informacionnye tehnologii i bezopasnost'». Materialy dokladov XVII Mezhdunarodnoj nauchno-prakticheskoj konferencii (27 nojab. 2018, g. Kiev, Ukraina). Kiev*, pp. 145–159, 2018.

[22] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2020, pp. 1–6.

[23] L. Tang and G. Shao, "Drone remote sensing for forestry research and practices," *Journal of Forestry Research*, vol. 26, pp. 791–797, 2015.

[24] J. Blackeye. (2016) White drone mid-flight. Unsplash. [Online]. Available: https://unsplash.com/photos/drone-flying-in-sky-XYrjl3j7smo

[25] P. J. Zarco-Tejada, R. Diaz-Varela, V. Angileri, and P. Loudjani, "Tree height quantification using very high resolution imagery acquired from an unmanned aerial vehicle (uav) and automatic 3d photo-reconstruction methods," *European journal of agronomy*, vol. 55, pp. 89–99, 2014.

[26] J. P. Dandois and E. C. Ellis, "High spatial resolution three-dimensional mapping of vegetation spectral dynamics using computer vision," *Remote Sensing of Environment*, vol. 136, pp. 259–276, 2013.

[27] J. Lisein, M. Pierrot-Deseilligny, S. Bonnet, and P. Lejeune, "A photogrammetric workflow for the creation of a forest canopy height model from small unmanned aerial system imagery," *Forests*, vol. 4, no. 4, pp. 922–944, 2013.

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

[29] ——, "Fire detection using infrared images for uav-based forest fire surveillance," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 567–572.

[30] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Multiple uavs in forest fire fighting mission using particle swarm optimization," in *2017 International conference on unmanned aircraft systems (ICUAS)*. IEEE, 2017, pp. 1404–1409.

[31] A. Urbahs and I. Jonaite, "Features of the use of unmanned aerial vehicles for agriculture applications," *Aviation*, vol. 17, no. 4, pp. 170–175, 2013.

[32] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar, "Devices, systems, and methods for automated monitoring enabling precision agriculture," in *2015 IEEE international conference on automation science and engineering (CASE)*. IEEE, 2015, pp. 462–469.

[33] P. Chamoso, W. Raveane, V. Parra, and A. González, "Uavs applied to the counting and monitoring of animals," in *Ambient Intelligence-Software and Applications: 5th International Symposium on Ambient Intelligence*. Springer, 2014, pp. 71–80.

[34] C. Caillouet, F. Giroire, and T. Razafindralambo, "Efficient data collection and tracking with flying drones," *Ad Hoc Networks*, vol. 89, pp. 35–46, 2019.

[35] J. Liang and Q. Liang, "Rf emitter location using a network of small unmanned aerial vehicles (suavs)," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–6.

[36] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[37] A. Albert and L. Imsland, "Combined optimal control and combinatorial optimization for searching and tracking using an unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 691–706, 2019.

[38] Z. Liu, X. Fu, and X. Gao, "Co-optimization of communication and sensing for multiple unmanned aerial vehicles in cooperative target tracking," *Applied Sciences*, vol. 8, no. 6, p. 899, 2018.

[39] Q. Yuan, J. Zhan, and X. Li, "Outdoor flocking of quadcopter drones with decentralized model predictive control," *ISA transactions*, vol. 71, pp. 84–92, 2017.

[40] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, "Real-time area coverage and target localization using receding-horizon ergodic exploration," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 62–80, 2017.

[41] P. Sarunic and R. Evans, "Hierarchical model predictive control of uavs performing multitarget-multisensor tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 2253–2268, 2014.

[42] D. J. Pack, P. DeLima, G. J. Toussaint, and G. York, "Cooperative control of uavs for localization of intermittently emitting mobile targets," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 4, pp. 959–970, 2009.

[43] M. Faruk Gencel, U. Madhowl, and J. Pedro Hespanhal, "Rf source seeking using frequency measurements," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[44] F. Koohifar, A. Kumbhar, and I. Guvenc, "Receding horizon multi-uav cooperative tracking of moving rf source," *IEEE Communications Letters*, vol. 21, no. 6, pp. 1433–1436, 2016.

[45] F. Koohifar, I. Guvenc, and M. L. Sichitiu, "Autonomous tracking of intermittent rf source using a uav swarm," *IEEE Access*, vol. 6, pp. 15 884–15 897, 2018.

[46] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, 2020.

[47] J. Kennedy, *Swarm intelligence*. Springer, 2006.

[48] D. M. Vijayakumari, S. Kim, J. Suk, and H. Mo, "Receding-horizon trajectory planning for multiple uavs using particle swarm optimization," in *AIAA Scitech 2019 forum*, 2019, p. 1165.

[49] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for uav formation via comprehensively improved particle swarm optimization," *ISA transactions*, vol. 97, pp. 415–430, 2020.

[50] F. Ye, J. Chen, Y. Tian, and T. Jiang, "Cooperative task assignment of a heterogeneous multi-uav system using an adaptive genetic algorithm," *Electronics*, vol. 9, no. 4, p. 687, 2020.

[51] X. Wu, Y. Yin, L. Xu, X. Wu, F. Meng, and R. Zhen, "Multi-uav task allocation based on improved genetic algorithm," *IEEE Access*, vol. 9, pp. 100 369–100 379, 2021.

[52] Y. V. Pehlivanoglu and P. Pehlivanoglu, "An enhanced genetic algorithm for path planning of autonomous uav in target coverage problems," *Applied Soft Computing*, vol. 112, p. 107796, 2021.

[53] W. Power, M. Pavlovski, D. Saranovic, I. Stojkovic, and Z. Obradovic, "Autonomous navigation for drone swarms in gps-denied environments using structured learning," in *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 219–231.

[54] H. Khalil, S. U. Rahman, I. Ullah, I. Khan, A. J. Alghadhban, M. H. Al-Adhaileh, G. Ali, and M. ElAffendi, "A uav-swarm-communication model using a machine-learning approach for search-and-rescue applications," *Drones*, vol. 6, no. 12, p. 372, 2022.

[55] V. Acuna, A. Kumbhar, E. Vattapparamban, F. Rajabli, and I. Guvenc, "Localization of wifi devices using probe requests captured at unmanned aerial vehicles," in *2017 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2017, pp. 1–6.

[56] J. Zhang and X. Jiahao, "Cooperative task assignment of multi-uav system," *Chinese Journal of Aeronautics*, vol. 33, no. 11, pp. 2825–2827, 2020.

[57] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*. Springer, 2017, pp. 66–83.

[58] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning," *IEEE access*, vol. 7, pp. 146 264–146 272, 2019.

[59] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic spectrum interaction of uav flight formation communication with priority: A deep reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 892–903, 2020.

[60] Y. Yuan, L. Lei, T. X. Vu, S. Chatzinotas, S. Sun, and B. Ottersten, "Energy minimization in uav-aided networks: Actor-critic learning for constrained scheduling optimization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5028–5042, 2021.

[61] V. J. Hodge, R. Hawkins, and R. Alexander, "Deep reinforcement learning for drone navigation using sensor data," *Neural Computing and Applications*, vol. 33, pp. 2015–2033, 2021.

[62] J. Burgués and S. Marco, "Environmental chemical sensing using small drones: A review," *Science of the total environment*, vol. 748, p. 141172, 2020.

[63] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[64] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-uav systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86, 2017.

[65] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked uavs as aerial sensor network for disaster management applications," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[66] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of intelligent & robotic systems*, vol. 61, pp. 563–585, 2011.

[67] Y. Karaca, M. Cicek, O. Tatli, A. Sahin, S. Pasli, M. F. Beser, and S. Turedi, "The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations," *The American journal of emergency medicine*, vol. 36, no. 4, pp. 583–588, 2018.

[68] G. Daneels, E. Municio, K. Spaey, G. Vandewiele, A. Dejonghe, F. Ongenae, S. Latré, and J. Famaey, "Real-time data dissemination and analytics platform for challenging iot environments," in *2017 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2017, pp. 23–30.

[69] S. Gite and H. Agrawal, "On context awareness for multisensor data fusion in iot," in *Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 3*. Springer, 2016, pp. 85–93.

[70] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2013.

[71] B. Bethke, M. Valenti, and J. How, “Cooperative vision based estimation and tracking using multiple uavs,” in *Advances in Cooperative Control and Optimization: Proceedings of the 7th International Conference on Cooperative Control and Optimization*. Springer, 2007, pp. 179–189.

[72] J. He, Y. Geng, and K. Pahlavan, “Toward accurate human tracking: Modeling time-of-arrival for wireless wearable sensors in multipath environment,” *IEEE Sensors Journal*, vol. 14, no. 11, pp. 3996–4006, 2014.

[73] C. R. Karanam, B. Korany, and Y. Mostofi, “Magnitude-based angle-of-arrival estimation, localization, and target tracking,” in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2018, pp. 254–265.

[74] F. Fletcher, B. Ristic, and D. Musicki, “Recursive estimation of emitter location using tdoa measurements from two uavs,” in *2007 10th International Conference on Information Fusion*. IEEE, 2007, pp. 1–8.

[75] A. Lin and H. Ling, “Doppler and direction-of-arrival (ddoa) radar for multiple-mover sensing,” *IEEE transactions on aerospace and electronic systems*, vol. 43, no. 4, pp. 1496–1509, 2007.

[76] F. Viani, L. Lizzi, P. Rocca, M. Benedetti, M. Donelli, and A. Massa, “Object tracking through rssi measurements in wireless sensor networks,” *Electronics Letters*, vol. 44, pp. 653–654(1), May 2008.

[77] P. Kyösti, J. Meinilä, L. Hentilä, X. Zhao, T. Jämsä, C. Schneider, M. Narandzić, M. Milojević, A. Hong, J. Ylitalo, V.-M. Holappa, M. Alatossava, R. Bultitude, Y. deJong, and T. Rautiainen, “Winner ii interim channel models,” EBITT, TUI, UOULU, CU/CRC, NOKIA, Tech. Rep., 11 2006, workpackage: WP1 Channel Model. Updated 09/02/2007.

[78] T. Lagkas, G. Eleftherakis, K. Dimopoulos, and J. Zhang, “Signal strength based scheme for following mobile iot devices in dynamic environments,” *Pervasive and Mobile Computing*, vol. 65, p. 101165, 2020.

[79] F. B. Sorbelli, S. K. Das, C. M. Pinotti, and S. Silvestri, “On the accuracy of localizing terrestrial objects using drones,” in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[80] Y. Spyridis, T. Lagkas, P. Sarigiannidis, and J. Zhang, “Modelling and simulation of a new cooperative algorithm for uav swarm coordination in mobile rf target tracking,” *Simulation Modelling Practice and Theory*, vol. 107, p. 102232, 2021.

[81] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, "WINNER II interim channel models," *Radio Technologies and Concepts for IMT-Advanced*, pp. 39–92, 2009.

[82] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[83] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in uav enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2018.

[84] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014.

[85] P. Langley and H. A. Simon, "Applications of machine learning and rule induction," *Commun. ACM*, vol. 38, no. 11, p. 54–64, Nov. 1995. [Online]. Available: https://doi.org/10.1145/219717.219768

[86] A. Nikitas, K. Michalakopoulou, E. T. Njoya, and D. Karampatzakis, "Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era," *Sustainability*, vol. 12, no. 7, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/7/2789

[87] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, 2002, pp. 575–578.

[88] M. Mozaffari, W. Saad, M. Bennis, Y. H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[89] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[90] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[91] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 2, pp. 345–360, 2021.

[92] A. Nazi, W. Hang, A. Goldie, S. Ravi, and A. Mirhoseini, "Gap: Generalizable approximate graph partitioning framework," *arXiv preprint arXiv:1903.00614*, 2019.

[93] D. Spielman, "Spectral graph theory," *Combinatorial scientific computing*, vol. 18, p. 18, 2012.

[94] Y. Spyridis, T. Lagkas, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, G. Eleftherakis, and J. Zhang, "Towards 6g iot: Tracing mobile sensor nodes with deep learning clustering in uav networks," *Sensors*, vol. 21, no. 11, p. 3936, 2021.

[95] T. Lagkas, V. Argyriou, S. Bibi, and P. Sarigiannidis, "Uav iot framework views and challenges: Towards protecting drones as "things"," *Sensors*, vol. 18, no. 11, p. 4015, 2018.

[96] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M.-T. Zhou, "Femto: Fair and energy-minimized task offloading for fog-enabled iot networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4388–4400, 2018.

[97] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, "Energy-efficient data collection and device positioning in uav-assisted iot," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1122–1139, 2019.

[98] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for uav relay networks," *IEEE Communications Letters*, vol. 22, no. 1, pp. 161–164, 2017.

[99] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based uav path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.

[100] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *omega*, vol. 34, no. 3, pp. 209–219, 2006.

[101] D. S. Hochba, "Approximation algorithms for np-hard problems," *ACM Sigact News*, vol. 28, no. 2, pp. 40–52, 1997.

[102] S. Pang, T. Li, F. Dai, and M. Yu, "Particle swarm optimization algorithm for multi-salesman problem with time and capacity constraints," *Applied Mathematics & Information Sciences*, vol. 7, no. 6, p. 2439, 2013.

[103] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *European journal of operational research*, vol. 175, no. 1, pp. 246–257, 2006.

[104] H. Liu, J. Ge, Y. Wang, J. Li, K. Ding, Z. Zhang, Z. Guo, W. Li, and J. Lan, "Multi-uav optimal mission assignment and path planning for disaster rescue using adaptive genetic algorithm and improved artificial bee colony method," in *Actuators*, vol. 11, no. 1. MDPI, 2021, p. 4.

[105] C.-H. Song, K. Lee, and W. D. Lee, "Extended simulated annealing for augmented tsp and multi-salesmen tsp," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3. IEEE, 2003, pp. 2340–2343.

[106] H. Liu, X. Li, G. Wu, M. Fan, R. Wang, L. Gao, and W. Pedrycz, "An iterative two-phase optimization method based on divide and conquer framework for integrated scheduling of multiple uavs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5926–5938, 2020.

[107] L. Huo, J. Zhu, G. Wu, and Z. Li, "A novel simulated annealing based strategy for balanced uav task assignment and path planning," *Sensors*, vol. 20, no. 17, p. 4769, 2020.

[108] S. Gao, J. Wu, and J. Ai, "Multi-uav reconnaissance task allocation for heterogeneous targets using grouping ant colony optimization algorithm," *Soft Computing*, vol. 25, pp. 7155–7167, 2021.

[109] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review," *International Journal of Automation and Computing*, vol. 14, no. 5, pp. 503–519, 2017.

[110] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[111] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[112] S. M. Devlin and D. Kudenko, "Dynamic potential-based reward shaping," in *Proceedings of the 11th international conference on autonomous agents and multiagent systems*. IFAAMAS, 2012, pp. 433–440.

[113] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[114] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[115] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[116] J.-X. Mi, A.-D. Li, and L.-F. Zhou, "Review study of interpretation methods for future interpretable machine learning," *IEEE Access*, vol. 8, pp. 191 969–191 985, 2020.

[117] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, “Perspectives of quantum annealing: Methods and implementations,” *Reports on Progress in Physics*, vol. 83, no. 5, p. 054401, 2020.

[118] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, “Variational quantum algorithms,” *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.