Computational Intelligence for Cooperative Swarm Control

Author:
Sherif, Reem

Publication Date:
2023

DOI:
https://doi.org/10.26190/unsworks/25067

Computational Intelligence for Cooperative Swarm Control

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy at the
School of Engineering & Information Technology
University of New South Wales at
Australian Defence Force Academy

ORIGINALITY STATEMENT

> ☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

COPYRIGHT STATEMENT

> ☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).
>
> For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

> ☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

- The candidate contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. they were responsible primarily for the planning, execution and preparation of the work for publication.
- The candidate has obtained approval to include the publication in their thesis in lieu of a Chapter from their Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

☑ The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement**.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

> The work reported in this chapter 3 has been partially published in the following articles:
> RE Mohamed, S Elsayed, R Hunjet, H Abbass (2021), A Graph-based Approach for Shepherding
> Swarms with Limited Sensing Range. 2021 IEEE Congress on Evolutionary Computation (CEC).
> RE Mohamed, S Elsayed, R Hunjet, H Abbass (2022), Connectivity-Aware Particle Swarm Optimisation for Swarm Shepherding.
> IEEE Transactions on Emerging Topics in Computational Intelligence.
>
> The work, reported in chapter 4, has been partially published in the following article:
> RE Mohamed, R Hunjet, S Elsayed, H Abbass, Deep Learning For Noisy Communication System. 2021 31st International Telecommunication Networks and Applications Conference (ITNAC), 40-47
>
> The work reported in this chapter has been partially published in the following article:
> Reem E Mohamed, Saber Elsayed, Robert Hunjet, Hussein Abbass (2022), Reinforcement Learning for Solving Communication Problems in Shepherding. IEEE Symposium Series On Computational Intelligence.

Candidate's Declaration

I declare that I have complied with the Thesis Examination Procedure.

Abstract

Over the last few decades, swarm intelligence (SI) has shown significant benefits in many practical applications. Real-world applications of swarm intelligence include disaster response and wildlife conservation. Swarm robots can collaborate to search for survivors, locate victims, and assess damage in hazardous environments during an earthquake or natural disaster. They can coordinate their movements and share data in real-time to increase their efficiency and effectiveness while guiding the survivors. In addition to tracking animal movements and behaviour, robots can guide animals to or away from specific areas. Sheep herding is a significant source of income in Australia that could be significantly enhanced if the human shepherd could be supported by single or multiple robots.

Although the shepherding framework has become a popular SI mechanism, where a leading agent (sheepdog) controls a swarm of agents (sheep) to complete a task, controlling a swarm of agents is still not a trivial task, especially in the presence of some practical constraints. For example, most of the existing shepherding literature assumes that each swarm member has an unlimited sensing range to recognise all other members' locations. However, this is not practical for physical systems. In addition, current approaches do not consider shepherding as a distributed system where an agent, namely a central unit, may observe the environment and communicate with the shepherd to guide the swarm. However, this brings another hurdle when noisy communication channels between the central unit and the shepherd affect the success of the mission. Also, the literature lacks shepherding models that can cope with dynamic communication systems. Therefore, this thesis aims to de-

sign a multi-agent learning system for effective shepherding control systems in a partially observable environment under communication constraints.

To achieve this goal, the thesis first introduces a new methodology to guide agents whose sensing range is limited. In this thesis, the sheep are modelled as an induced network to represent the sheep's sensing range and propose a geometric method for finding a shepherd-impacted subset of sheep. The proposed swarm optimal herding point uses a particle swarm optimiser and a clustering mechanism to find the sheepdog's near-optimal herding location while considering flock cohesion. Then, an improved version of the algorithm (named swarm optimal modified centroid push) is proposed to estimate the sheepdog's intermediate waypoints to the herding point considering the sheep cohesion. The approaches outperform existing shepherding methods in reducing task time and increasing the success rate for herding.

Next, to improve shepherding in noisy communication channels, this thesis proposes a collaborative learning-based method to enhance communication between the central unit and the herding agent. The proposed independent pre-training collaborative learning technique decreases the transmission mean square error by half in 10% of the training time compared to existing approaches. The algorithm is then extended so that the sheepdog can read the modulated herding points from the central unit. The results demonstrate the efficiency of the new technique in time-varying noisy channels.

Finally, the central unit is modelled as a mobile agent to lower the time-varying noise caused by the sheepdog's motion during the task. So, I propose a Q-learning-based incremental search to increase transmission success between the shepherd and the central unit. In addition, two unique reward functions are presented to ensure swarm guidance success with minimal energy consumption. The results demonstrate an increase in the success rate for shepherding.

Acknowledgement

This thesis would not be possible without the encouragement, support, and guidance of the people around me. I would like to express my appreciation to my supervisors, Dr. Saber Elsayed, Prof. Hussein Abbass, and A/Prof. Robert Hunjet. Their guidance, unwavering support, and constructive criticism served as life teachings and helped me grow as a researcher and a person. They ingrained in me a sense of appreciation for my PhD experience and a view of it as an essential learning journey.

Dr. Saber has taught me time management and planning through examples, which has resulted in my ability to submit my thesis on time and an enhancement in my professional performance. Throughout my PhD, he has provided me with invaluable support to begin challenging tasks and has enhanced my analytical skills to deal with a variety of circumstances, lessons that will remain with me for life. During the period of COVID19, he assisted me in adhering to my plan by revising it as required to achieve the best results possible given the constraints imposed by the COVID situation.

Throughout my PhD, Prof. Hussein has been an inspiration to me. He has demonstrated to me a variety of coping strategies for unanticipated situations that have arisen during my PhD due to the COVID19 restrictions. He has been supportive and has taught me how to approach each problem with a researcher's mindset. These fundamentals of critical thinking assisted me in getting my work published in a high-impact journal and presented at prestigious conferences. Moreover, his presentation skills have always inspired me, and I continue to learn from him by watching his videos and following his outstanding steps in his career.

A/Prof Robert Hunjet has been teaching me how to put my ideas into action, which includes innovative presentations and problem solving. With his presence, I felt like I was with my family. His constant encouragement to think creatively within the constraints of each given circumstance has been invaluable. A/Prof Robert Hunjet has taught me the importance of adhering to my job's requirements with diligence.

I would also like to acknowledge UNSW Canberra for the full scholarship that allowed me to pursue my PhD studies with a passion for research. My communication abilities have been enhanced by the facilities and various support departments. I have taken every opportunity to put what I've learnt from supervisors in to practice by enrolling in various courses. Ms. Angela Markovi, a student councillor for the defence department, is one of the individuals who have taught me various coping mechanisms, notably in stressful situations.

To my father's soul and my mother's efforts that empowered me during my PhD journey. To my life partner, who assisted me in achieving my PhD. He has been my primary source of pleasure and support in every circumstance. I would like to thank my colleagues in my research group "trusted autonomy", specifically Dr. Heba El-Fiqi and Dr. Aya Hussein, for their unwavering support in all aspects of my Ph.D. I'd like to thank all of my peers for making my time at UNSW Canberra genuinely unforgettable.

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Reem E. Mohamed

List of publications

[Journal articles]

1. R. E. Mohamed, S. Elsayed, R. Hunjet and H. Abbass, *Connectivity-Aware Particle Swarm Optimisation for Swarm Shepherding,* Online First, IEEE Transactions on Emerging Topics in Computational Intelligence.
   doi:10.1109/TETCI.2022.3195178.

[Conference papers]

1. R. E. Mohamed, S. Elsayed, R. Hunjet and H. Abbass, *A Graph-based Approach for Shepherding Swarms with Limited Sensing Range,* 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 2021, pp. 2315-2322.
   doi: 10.1109/CEC45853.2021.9504706.

2. R. E. Mohamed, R. Hunjet, S. Elsayed and H. Abbass, *Deep Learning For Noisy Communication System,* 2021 31st International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 2021, pp. 40-47.[**best student paper award**]
   doi: 10.1109/ITNAC53136.2021.9652171.

3. R. E. Mohamed, S. Elsayed, R. Hunjet and H. Abbass, *Reinforcement Learning for Solving Communication Problems in Shepherding,* 2022 IEEE Symposium Series On Computational Intelligence, Singapore.

Contents

List of Figures

List of Tables

List of Common Acronyms

- **AE** autoencoder
- **AI** artificial intelligence
- **ANN** artificial neural networks
- **AWGN** additive white gaussian noise
- **CH** communication channel/ forward channel
- **CO** constrained optimisation
- **COST** COopération européenne dans le domaine de la recherche Scientifique et Technique
- **CU** central unit
- **DARPA** defense advanced research projects agency
- **DBSCAN** density-based spatial clustering of applications with noise
- **DL** deep learning
- **DRL** deep Reinforcement Learning
- **DT** decision trees
- **F-CH** feedback channel
- **FANET** flying ad-hoc network

- **GCM** global centre of mass
- **KNN** K-nearest neighbour
- **LCM** local centre of mass
- **LOS** line of sight
- **LSTM** long short-term memory
- **LTE** long-term evolution
- **MDP** markov decision process
- **MLR** multiple linear regression
- **MSE** mean square error
- **NCS** networked control system
- **NLOS** non-line of sight
- **NN** neural network
- **PR** polynomial regression
- **PSO** particle swarm Optimisation
- **RF** random forests
- **RMSE** root-mean-square error
- **RNN** recurrent neural networks
- **Rx** receiver
- **SER** symbol error rate
- **SGD** stochastic gradient descent
- **SI** swarm intelligence

- **SNR** signal power to the noise power
- **SOHP** swarm optimal herding point
- **SOMCP** swarm optimal modified centroid push
- **SR** success rate
- **SVR** support vector regression
- **Tx** transmitter
- **UDG** unit disc graph
- **VANET** vehicular ad hoc network
- **WMN** wireless mobile network
- **WSN** wireless sensor network
- **i.i.d.** independent and identically distributed

Chapter 1

Introduction

1.1 Background

Over the last two decades, the use of autonomous agents (i.e., robots) to solve complex tasks has attracted the interest of many industries [2, 3]. In such systems, the agents intend to interact with each other and their surroundings to complete a task. Such behaviour of a self-organised system is an umbrella for swarm intelligence (SI); a branch of distributed artificial intelligence (AI) [4].

Using SI in many industrial applications has several advantages, including cost reduction, increased productivity, revenue and automation, fast delivery, and innovation [5, 6]. In many applications, SI can be used for efficient transportation and identifying and monitoring a hazardous or complex environment [7, 8]. Defence organisations used SI (drone swarm tactics) that involves the rapid distribution of many microsensor network robots in combat zones [9]. SI can be utilised to implement real-time monitoring and detect changes in an enemy's situation. It can also be used to help deploy large-scale unmanned aerial vehicles (UAVs), unmanned ships, and robots as substitutes for troops to undertake reconnaissance and warfare in order to reduce losses [8]. According to an estimate of the uses of SI in 2020 [10], less than 20% of system's interactions will be performed by humans.

To implement SI, the swarm usually needs to be supported by a sensor network and a mechanism for controlling its movements. Among swarm control mechanisms, shepherding has become a prevalent and successful approach [11]. Shepherding is a biologically-inspired swarm guidance methodology where single or multiple agents (sheepdogs) act as pressure points to exert forces through some behaviours that influence swarm members (sheep) to move towards a goal [11, 12, 13, 14]. In the literature, the common behaviours carried out by a sheepdog are (1) collecting: gathering scattered flock sheep representing the swarm members into a designated region, (2) herding: steering a flock of sheep from a start region to a goal; (3) covering: driving a flock to explore areas that have not been visited before; and (4) patrolling: protecting an area to ensure that the sheep do not leave it (paddock) or do not enter it (for example, a bush area where sheep could get lost) [15]. Shepherding has demonstrated various advantages over other swarm controlling approaches, including lower costs, better efficiency, distributed detection, and increased reliability [2, 16].

The system introduced in this thesis is concerned with swarm collaboration in a complex uncertain environment. Shepherding necessitates the development of self-organised systems for swarm control, a task that is hampered by many technical obstacles. Guiding a swarm towards a goal in the environment is a non-trivial problem due to the distributed nature of swarm members that need to maintain cohesion during their movements.

Several elements, such as the number of obstacles and the geographical density of the swarm, have also been shown to affect the success rate of the shepherding task [17, 18]. In addition, the partial observability of the environment is another hurdle [19], as it hinders the sheepdog's ability to choose the most influential leading locations. In contrast, knowing the positions of all the swarm members in real time necessitates a bird's eye view (i.e., the shepherd person) from a reasonably considerable distance, but this may result in other communication issues [20, 21].

With the advancement in communication technologies, the view of the environment perceived by an agent with sensing capabilities can be analysed and processed to generate commands for the sheepdog via a wireless channel. The authors in [22] refer to a swarm of drones as a networked control system (NCS), where the entire system is controlled via a wireless communication network to enhance data collection and decision-making. However, automated swarm control is challenging due to unpredictability in wireless, networking, and environmental constraints [23]; for example, in a fighting wildfires scenario, communication failure between an agent and a ground control unit may result in catastrophic consequences in lives [24]. This motivates researchers to develop adaptive approaches employing artificial intelligence that enable communication systems to interact independently of the channel model [25].

The existing literature on flocking assumes that each swarm member knows the location of all or the vast majority of other swarm members. This suggests that the global centre of mass, or GCM, anticipated by every swarm member are the same or comparable. However, this is an unrealistic assumption for physical systems because a swarm member can only perceive its neighbours within a limited radius and estimate a local centre of mass (LCM) that may be significantly further away from the LCMs of other swarm members. This restricted sensing range of the swarm causes the flock's cohesiveness to weaken as swarm members are drawn to various LCMs, which may result in a long completion time. Furthermore, due to the dynamic nature of the sheepdog as an autonomous system with finite battery life, the success rate of a task decreases as the task duration increases. Therefore, identifying the optimal force points for the sheepdog is essential to minimise the swarm's dispersion, which will lead to mission success.

Moreover, modelling the sheepdog or the shepherd as a mobile autonomous agent with a small size and low cost acting as the sheepdog, observing the movements of the swarm in real-time is not possible [19]. This shortcoming urges a sensing system (i.e., a central unit (CU)) to provide a bird's eye view from a relatively large distance from the swarm. At the same time, the sheepdog applies forces within a close vicinity [26]. This is similar to using a sensing system that sends control signals to actuators in autonomous systems [20, 21]. This transition from the standalone performance of the sheepdog to a more distributed solution preserves the ability of the sensing system to perceive the environment without interfering with the shepherd's task [22]. However, due to the physical system constraints, this distributed system may result in unpredictable channel variations (inaccurate information) during the task time [27], which, in turn, may lead to the sheepdog misguiding the sheep. This issue may be hard to fix quickly due to other shepherding constraints, i.e., (a) limited time to minimise energy consumption, (b) the limited number of transmissions to minimise bandwidth usage, and (c) ensuring the cohesion of the swarm.

Regarding wireless communication channels, the distortion in the data due to system noise and wireless signal fading may be minimised by positioning the CU close to the sheepdog during the shepherding task. In other words, dynamic positioning of the CU during the shepherding task may maximise transmission success and, consequently, the success of the shepherding task. However, this solution has a drawback: exerting a significant amount of energy. Therefore, improving the success rate of transmissions while minimising the CU's movements to limit its energy consumption is required to ensure a high success rate in the shepherding task.

1.2 Research Questions and Aims

The effectiveness of shepherding as a strategy for swarm guidance in a real-world situation is not just dependent on the performance of the sheepdog in guiding the swarm, but also on the dog's ability to do so with the least amount of communication possible. Therefore, this thesis aims to advance the current swarm guidance literature by advancing communication models in shepherding.

From the aforementioned discussion, the main research question and sub-questions are as follows:

- how can a robust multi-agent learning system be designed for effective and efficient swarm control of agents in a partially observable environment under communication constraints?

  - What is the optimal herding point in different shepherding problems in the presence of a low-sensing range? How can the shepherding process be improved in partially observable environments?
  - By modelling shepherding as a distributed system, how does the noisy communication channel affect the progress of the shepherding task? How can the shepherding task be improved under a noisy communication channel?
  - How can the shepherding process be improved in dynamic communication systems?

To answer these questions, the following four specific objectives are pursued:

- The first sub-question is answered by designing geometric and optimisation-based approaches to find the near-optimal herding point and path for the shepherd that improve shepherding under low sensing range constraints [Chapter 3],
- The second sub-question is answered by designing collaborative deep learning-based techniques to improve communication between transmitter (Tx) and receiver (Rx) under the effect of challenging communication system noise [Chapter 3],
- The third sub-question is answered by improving the adaptability of the receiver at the shepherd to receive the herding points from the CU under the effect of time-varying noise in the communication channel [Chapter 4] by deploying neural network that can remove the effect of channel noise, and
- designing new mechanisms to improve shepherding in dynamic communication systems by deploying Q-learning based approach in the CU to improve the

success rate of transmissions to the sheepdog during the shepherding task [Chapter 5].

1.3 Contributions To Scientific Knowledge

The following is a synopsis of the primary contributions made in the thesis, as well as its organisation:

1. **A new method for finding near-optimal herding points:** By utilising the sheep's sensing range, unit disc graph (UDG) is used to describe the sensing induced graph among uniform agents equipped with 360-degree viewing sensors in Chapter 3. This allows building a geometric way to define a subset of sheep to be impacted by the shepherd. The geometric principles are merged with meta-heuristic optimisation and clustering to determine the near-optimal herding point while considering flock connectivity. On top of the sheep sensing-induced graph, an optimisation-based technique is utilised to find near-optimal waypoints between the shepherd and a herding point. This process gathers all the sheep and prevents dispersion. During real-time operations, the influence of the shepherd's locations on the sheep graph to redefine a new herding point and an adaptive route to this point are analysed. The proposed method improved the task success rate and decreased the task time.

2. **A cooperative deep-learning approach for shepherding in time-varying and noisy communication channels:** To increase the success rate of transmissions of herding points from the decision maker (i.e. the central unit UC) to the sheepdog (i.e. the actuator), in Chapter 4, deep neural networks were designed at the Tx and the Rx to overcome internal device noise. Then, an effective two-phase training strategy for communication end-systems is proposed to enable the Tx and Rx to obtain low transmission errors in a short learning time, even in tough transmission instances without modelling the communication channel. Then, the learning-based method is improved so that the Tx

and Rx can tackle demanding noise levels with little communication channel utilisation during the training phase. This leads to a hybrid demodulation approach at the receiver where 4-QAM modulation assisted by the use of trained neural network is designed to increase transmission success rate. The communication of the suggested strategy in reducing corruption from sent data is verified on a shepherding scenario, where noise fluctuates owing to Rx mobility. The success rate of the shepherding task is doubled to ensure a similar SR of the task under the assumption of ideal communication channel.

3. **A Learning-based Mobility Model for CU:** The difficulty of dynamic agents interacting over a stochastic channel as Markov Decision Process is characterised in Chapter 5. This paved the way to propose a Q-learning-based incremental search to increase transmission success between the shepherd and the central unit. In addition, two novel reward functions are designed to ensure swarm guidance success with minimal energy consumption. The results demonstrate an increase in the success rate for shepherding.

1.4 Thesis Organisation

This thesis consists of six chapters. Chapter 2 discusses a literature assessment of the shepherding problem, its applications, and its challenges as a physical system. Then, in Chapter 3, a strategy based on optimisation is created to enhance the performance of the sheepdog under the influence of a limited sensing range on the dynamic agents. Then, in Chapter 4, the primary wireless communication issues in the backbone system for the shepherding task under the same physical system constraints are examined. This compels further examination in Chapter 5 of the significance and challenges brought by the dynamic system. In the final chapter, the key findings of the thesis are summarised, its conclusions are drawn, and potential future study areas are identified. It is worth noting that the appendix provides the source data for Chapters 3-5, where the full experimental data are not presented in the main body of the thesis.

Chapter 2

Literature Review

This chapter discusses the background of the topics related to this thesis. It begins with a brief overview of swarm intelligence. It, then, details the shepherding model and related work, followed by a discussion of fundamentals of communication systems. Next, existing work and challenges of using shepherding in dynamic systems and the need for communication models in shepherding are explored. Finally, a summary of the chapter is reiterated.

2.1 Swarm Intelligence

In many biological systems, a group of agents (i.e., birds, ants or fish) behaves together to achieve a task. This group of agents is called a swarm, and their actions form swarm's behaviours [28]. Generally speaking, the agents need to self-organise by using some rules to govern their actions. This collective behaviour can be used to define Swarm Intelligence (SI). In other words, SI can be described as "*a collective behaviour of a decentralised or self-organised group of agents*" [4]. As depicted in Figure 2.1, SI can be seen as several autonomous units (agents) supported by a sensor network. They need to share information to be able to self-organise. This information could be as simple as agreeing to behave using the same internal logic.

SI has been applied in many domains, such as transportation, defence [9], and

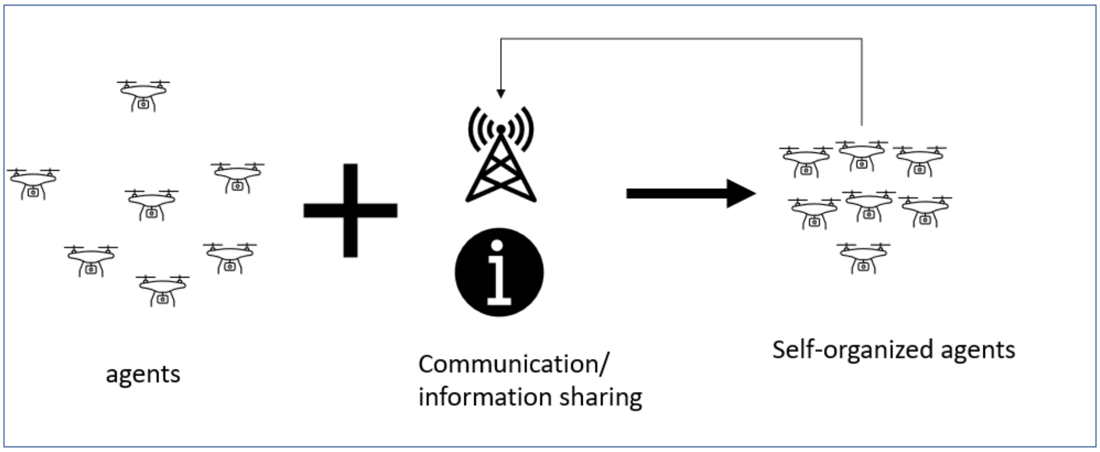

Figure 2.1: Components of an SI system

robotics [8]. As reported in [29], an SI system was introduced with a low-cost, small-scale mobile robotic platform designed for educational purposes to facilitate teaching swarm-robotic concepts. In [30], an SI system enabled a single user to operate many swarm members for collective programming, powering on, and charging. The authors in [30] demonstrated the ability of small autonomous robots to mimic the behaviour of ants in building their colones and birds in flying. Elasticity as a form of robot intelligence capability was considered in designing a soft robot that deforms and absorbs energy in case of a collision [31, 32, 33] damage in robots under the effect of environmental uncertainty[31, 34, 35]. The National Aeronautics and Space Administration (NASA) created "Swarmies" to power space mining operations by using four robots whose motions and behaviour were inspired by ants. It also plans to grow and integrate RASSOR (a digging robot) [36, 37].

In the field of parallel and distributed computer networks, herds and schools are presented as instances of self-organizing, resilient distributed systems [38]. Using the concept of "bird-like objects", or boids for short, the famous work of Reynolds [39] models the polarised motion of clusters of directed particles. If no impetus is supplied, boids will wander rather erratically; hence, an external leading force is required for proper flocking behaviour [40]. Shepherding as a biologically inspired task where the mobility of passive agents is controlled by one or more active agents is a well researched state-of-the-art method that may be used to regulate the mobility of the swarm that is modelled as boids [12, 13].

As highlighted earlier, the effectiveness of a SI system in a real-world situation depends on the leading agent's capability in guiding the swarm (i.e., the swarm model to use), the communication model, and the information shared among the agents. In the following sections, these elements are reviewed.

2.2 Swarm Guidance Models

A swarm displaying boids [39] behaviour acts chaotically in the absence of a global direction, force, or leader. If random motion is added, they either discover a configuration that is best for the given set of parameters or they move randomly. This swarm may fluctuate or develop consistent patterns.

The applied forces on the swarm that aim to control their directions have been modelled differently in the literature on swarm control. Two main models have been considered in the literature. The following two models are discussed in the following subsections.

- **Leader-follower approach**, which involves some of the swarm members being the leaders of the other swarms as they are informed about the required orientation and directions.
- **Shepherding approach**, which involves an external force applied by an external leader that influences the swarm by transferring fear through the swarm members within limited vicinity, as demonstrated in the biologically-inspired shepherding tasks.

2.2.1 Leader-Follower Models

Computational techniques that rely on SI concepts were inspired by natural methods where a group of agents performs a variety of tasks [41]. Guiding a robot with the help of another robot was initialised in [42]. Then, leader-follower control concepts were applied where the leader robot has prior knowledge about the path that the follower robots should be guided in while the follower robots are following basic flocking rules or receiving commands through communication from that leader robot [43]. The leader-follower approach in dynamic systems can be used to provide swarm guidance by allowing some of the swarm members to be informed about the required swarm directions and lead the swarm. The set of swarm leaders may be chosen initially and remain until the end of the task or be elected periodically based on different criteria.

A self-adaptive robot swarm mechanism where all the swarms have been informed about their final destinations has been researched in 2D and 3D space [44]. In [44], the authors proposed collective motion algorithms to allow robots to travel along a pre-planned 3D route. The robots employ just one-hop neighbour information, maintain a network connection for information exchange, maintain a desired adjacent distance, and may circumvent obstacles without dividing the swarm (i.e., member loss). The main challenge in [44] is designing dynamic roles for the swarm system to maintain stability during this collective motion. Commonly, the mathematical model for mobility is composed of non-linear differential equations that describe the mobility of each entity in the system, either the leader or the follower.

The n-dimensional Euclidean vector space is denoted by $\mathbb{R}^n$ and the non-negative real numbers are denoted as $\mathbb{R}_+$. Thus, the system state trajectory can be expressed as a real-valued function $x(.) : \mathbb{R}_+ \to \mathbb{R}^n$, such that $x(t)$ denotes the value that a function $x(.)$ takes at time $t \in R_+$. Moreover, the state of the leader or the follower x can be representing the relative separation and bearing angle between the leader and follower following the given function $\dot{x} = f(x, u, w)$. Where the external disturbance$w(.) : \mathbb{R}_+ \to \mathbb{R}^l$ is essentially bounded and there exists a positive $W \in: \mathbb{R}_+$ such that $|w(t)|_{l_\infty} < W < \infty$. The input signal $u(.) : \mathbb{R}_+ \to \mathbb{R}^m$is generated by a controller generally through a wireless channel. The approaches of the leader-follower concept vary according to the method of obtaining the control values in $u(.)$ as summarised in the following:

- The authors in [43] studied how a self-organised swarm of mobile robots may be guided in a desirable direction by externally influencing some of its members, with the control signals relayed to the mobile robots through ideal communication channels.

- The authors in [45] studies the target enclosing problem using a predefined time-varying circular formation using one leader and three follower UAVs depending on the nearest neighbour local information. In this system, the leader is responsible for sending control signals to its followers while maintaining the stability of the dynamic system.

- The authors in [46] study the time-varying formation tracking problem for linear multiagent systems with multiple leaders and nonholonomic dynamics. The control signals were transmitted from leaders to their neighbouring followers considering the nonholonomic system dynamic while system stability was maintained.

- Swarm systems were addressed in [47], where all agents keep a time-varying formation while tracking a time-varying reference. In that case, the control signals are generated individually for each follower based on their observations.

To guide self-organised flocks in real and virtual mobile robots, the authors in [43] expand a previously developed flocking behaviour [48], where some agents in the swarm are informed about the required destination while the rest are not. The performance of the suggested behaviour is evaluated over a range of model parameters using three measures: (a) the information gain metric from Information Theory, which measures the information exchanged between members during steering; (b) the accuracy metric from directional statistics, which measures the angular deviation of the flock's direction from the intended direction; and (c) the biggest aggregate ratio, which measures the proportion of the flock that is following the largest aggregation.

However, in the leader-follower problem, an assumption is made of the prior knowledge of the agents about the target that may be stationary or dynamic. In [46], the target is the leader who is constantly linked to at least one follower with a broad enough communication radius to cover its followers. The target was the stable leader of the three follower UAVs. A communication link between the leader and at least one follower was guaranteed. The followers were indirectly tied to the leader via indirect relationships to maintain consistent formation tracking. While in [47], no leader was introduced, but a circular formation was desired. The authors assumed the flock's formation shape, size, and target places. The authors offer a cooperative term in translation control laws to stabilise their circular team formation. The findings may be used for time-varying formation tracking, target enclosing, and consensus tracking in linear multiagent systems with one or more targets/leaders.

In [44], the authors studied a collective motion problem in which a swarm of up to 30 robots was designed to move along a pre-planned path that is only known to their leader, from a source to a destination. The proposed self-adaptive collective motion algorithms for swarm were tested with Matlab 2015 simulation in free-space and cluttered environments. the leader could only communicate the control information to a limited number of its nearest neighbours that can apply attraction and repulsion forces on their nearest neighbours. The authors in [49] introduce a technique of collective movement for distributed control that accommodates for the limits of the corresponding dynamics and formation control model. Using a microscopic perspective of thermodynamics applied to a swarm robot, it is feasible to sustain a formation and adjust it to its surroundings via interactions based on local information given by a single leader with its neighbours. The leader and follower agents communicate with neighbouring agents within their respective sensing ranges. Nevertheless, the leader has the global communication capability to accept an order from the operator.

The proposed method controls the leader through ideal wireless communication signal to control the internal energy and knowledge of the phase transitions of the swarm of followers [50] to achieve solidity in the shape, which represents cohesion, and liquidity, which represents flexibility. The number of robots used in the experiment is 217 (1 leader, 216 followers), and the initial shape was an ideal hexagonallattice with the leader as the centre of gravity. The limitations include the existence of collision among agents and obstacles due to the use of maximum velocity in all agents. One possible method is to virtually enlarge the agent's diameter. In addition, the shape-shifting mechanism and its operational parameters were not investigated in this work.

In collective movement tasks, the whole swarm is expected to travel as one unit from one location to another. In contrast, each agent inside the swarm desires reorganisation in a spatial organisation task. As evidence of the efficacy of basic guidelines to repeat an enhanced collaborative foraging behaviour, Talamali et al. [51] demonstrate a cooperative foraging system of up to 200 physical agents supported by virtual binary pheromone sensors. The system consists of individual controllers of the swarm that can replicate normal foraging practises driven by actual ants that recognise pheromone buildup and seek its gradient. The fundamental quality of the controllers is a control parameter that stabilises the assignment between distance selectivity and the quality selectivity of individual fodders. The authors practically tested the system's ability to approximate the optimum foraging strategy for realistic swarm populations.

Despite the variety of leader-follower techniques in the literature, their applicability to physical systems or crowd management is limited by restrictions that can be summarised as follows:

- Leaders and their followers are assumed to be properly linked, either by a short-range communication channel or the leader's physical proximity to their group's centre of mass. This prevents the leader from multitasking, especially if two tasks need them to be in different locations.

- When a sensing system is used to notify the leader about swarm and obstacle locations, it is assumed that the communication connection between the sensing system and the leader robot is ideal, which may not be the case in practice.

- It is assumed that leaders have knowledge of the whole environment and the location of their followers, which may not be the case in open spaces with dynamic obstacles.
- Followers are expected to be initially connected in a reasonably resilient network so that they may influence each other throughout the task's length. This assumption limits the system's validity when there is no mutual effect between followers.
- Leader-follower tasks are evaluated only for formation control or route following, which resemble herding tasks under optimistic assumptions on flock cohesiveness. While real-time route planning and cohesion enhancements, in addition to other sub-tasks associated with shepherding, are not explored.

These assumptions make the leader-follower approach not suitable for herding biological systems or physical systems without communication capabilities with leaders or low connectivity among one another. Thus, we explain the shepherding approach to illustrate its ability to be applied to a wider range of applications.

2.2.2 Shepherding Models

Shepherding is a well-researched strategy for regulating the motion of a swarm of autonomous agents. It is a biologically-inspired swarm guiding strategy in which a single or more agents (sheepdogs) function as pressure points to apply forces via certain behaviours that push swarm members (sheep) towards a goal [11, 12, 13, 14]. As an example of swarm guidance, it offers several benefits, such as cheaper production costs, improved efficiency, dispersed detection, and higher dependability.

In the literature [2, 16], two major types of agents are considered, sheepdog and sheep. Each agent has an initial location, and the controlling agent (sheepdog) aims to change the swarm (sheep)'s location to a goal location. According to [15], the common tasks carried out by a sheepdog are (a) collecting: gathering scattered flock members into a designated region; (b) herding: steering a flock of sheep from a start region to a goal; (c) covering: driving the flock to explore areas that have not been visited; and (d) patrolling: protecting an area to either ensure that the sheep do not leave it (paddock) or do not enter it (the bush where sheep could get lost).

Herding has three main phases which are based on the shepherd's location with respect to the flock and the target position. The first phase is approaching the flock when the shepherd moves from its initial location towards the flock in order to start herding it. The second phase is maintaining flock formation as defined in [52] when the shepherd wants to maintain the flock formation or aggregate the flock to be able to drive it. The third phase is herding which includes both driving the flock and collecting if the flock violates the formation requirements. In this thesis, the main focus is on the herding phase which focuses on swarm guidance.

Applications of autonomous shepherding for guidance have a direct impact on human life, such as fighting wildfires, and crowd evacuation in disasters [53]. According to [54], the mobility of Muslim people during their pilgrimage in Mina/Makkah may have high density flows during the pilgrimage that can turn "turbulent" and cause people to fall. Thus, using guiding robots that can choose suitable pressure points may be helpful in this application of crowd management.

In smart agriculture, modern farms are under the heat from consumer advocacy organisations to become more productive while also improving animal well-being. According to [55], herding endangers the lives of farmers, puts strain on farm dogs, and increases the likelihood of neglect if it is not conducted often and wisely. In [56], research efforts were made to build a more autonomous sky-shepherding system that is considerate of animal welfare and trusted by farmers and consumers. To guide a swarm of agents to achieve its mission, several swarm guidance models have been proposed. The following subsection covers some of them.

2.3 Review of Existing Shepherding Methods

The first herding algorithm in the context of robotics literature used mathematical specifications to formulate a force model of flocking and herding behaviours and was verified using a robotic vehicle herding a flock of ducklings [48]. Then, simulation attempts paved the way for studying herding biological systems like animals with robots. The authors in [57] animated big flocks of fuzzy creatures where a sheepdog manages a herd of sheep while observing their cohesiveness and separation. They detail their attempts to produce realistic motions for the dog and sheep using insights from animal locomotion research and an approach to depict the enormous woolly flock in real-time utilising multi-view impostors with colour variation. The challenge a sheepdog brings to this setting is that the decisions made by the dog influence the network within the flock, and as such, the network is not under the flock's control alone.

Recently in [58], a combination of the leader-follower control approach with the boids flocking behaviour [39] is proposed. The system is composed of minority leaders that have knowledge of a desired trajectory and need to track it while the majority of followers avoid collision and move to the flocking centre. The followers neither know the leaders nor the desired trajectory, but they aim to follow the mobility of their centre of mass as suggested in the boids mobility rules [39]. According to [39], the swarm are affected by three steering vectors: (a) cohesion to stay in the centre of the flock, (b) alignment to smooth their velocities, and (d) separation to avoid mutual collisions. The guiding force of each swarm member is the weighted sum of these forces as modelled in Equation 2.1

$$F_{total}^{t} = W_{\pi\Lambda}\overrightarrow{P^{t}\Lambda^{t}} + W_{\pi\pi}\sum_{i\in n_{rep}}\overrightarrow{P_{i}^{t}P^{t}} + W_{\pi\upsilon}\overrightarrow{P^{t}P^{t1}} \tag{2.1}$$

where a swarm member π experiences repulsive force with weight $W_{\pi\pi}$ to repel from neighbouring sheep within a radius of R_{π} to avoid collision. The set of sheep within R_{π} is a repulsive set n_{rep} of that swarm member, while the set of swarm members within $R_{\pi\pi}$ of a sheep π is the neighbouring set n of that member where $R_{\pi\pi} >> R_{\pi}$. The cohesion force that is in Reynold rules [39], $R_{\pi\pi}$, covers the whole area of the flock. Moreover, $W_{\pi\Lambda}$ is the attraction strength towards the flock centre of mass Λ. Finally, $W_{\pi\upsilon}$ is the strength of the force π exerts in the previous direction, which represents the alignment force.

Inspired by the Reynold's rules for boids [39], Strombom et al. [59] introduced empirical results of shepherding mathematical model that may relate to the model introduced in [60] with a completely different objective of guiding the flock without fragmenting it. The model in [59] is based on observations of the behaviour of a trained female Australian Kelpie working-farm-dog while herding 46 three-year-old female merino sheep in South Australia in March 2010. The model could be used in other more complex tasks as it was found scalable to herd a flock of up to 300 sheep demonstrated experimentally.

The results obtained by GPS systems attached to the sheep allowed the authors to model and understand the animals' collective behaviour during a threat, as discussed in [60], which envisaged a similar set of rules of predators preying on a flock. The model in [59] can be categorised as a centroid push-based model since it depends on influencing the centre of mass of the swarm members. The centre of mass for a flock is called a global centre of mass (GCM) if it represents the mid location among the locations of the swarm in the flock. It may also be called a local centre of mass (LCM) if it represents the mid location among the locations of a set of swarm members.

The mathematical model presented in [59] includes an effect of repulsion force when the swarm of sheep are agitated by an external sheepdog that has prior knowledge about the home location. It uses this fear from the swarm modelled as sheep to herd them towards that location by placing itself within the agitation range $R_{\pi\beta}$ of some of them. This heuristic model is formulated as follows

$$F^t_{total} = \begin{cases} W_{\pi\pi} \sum_{i \in n_{rep}} \overrightarrow{P^t_i P^t} + W_{e\pi_i} \overrightarrow{P^t rand}, & d(P_\pi, P_\beta) > R_{\pi\beta} \\ W_{\pi\Lambda} \overrightarrow{P^t \Lambda^t} + W_{\pi\pi} \sum_{i \in n_{rep}} \overrightarrow{P^t_i P^t} + W_{\pi\beta} \overrightarrow{P^t_\beta P^t} + W_{e\pi_i} \overrightarrow{P^t_{rand}} + W_{\pi\upsilon} \overrightarrow{P^t P^{t-1}}, & d(P_\pi, P_\beta) \leq R_{\pi\beta} \end{cases} \tag{2.2}$$

where $W_{\pi\beta}$ is the repulsion strength between π and the sheephdog, and $W_{e\pi}$ is the strength of the π's angular noise and $rand$ is a random position representing a jittering effect.

The model in [59] highlights two important behaviours of the shepherd: collecting the flock in case of dispersion away from GCM and driving the GCM of the flock when no sheep is not further away from the GCM than a predefined threshold distance. The shepherd goes to the collecting position relative to the dispersed sheep and to the driving position relative to the calculated global centre of mass (GCM) or local centre of mass (LCM) for collecting and driving, respectively.

In [61, 62], the flocking of a group of agents that evaluated sheep connection in obstacle-free and crowded environments was addressed where homogeneous agents were assumed. This subject of agent connection has been examined by measuring the algebraic graph connectivity of multi-agent systems in diverse circumstances. For example, in [63], the challenge of edge selection to maximise graph connectivity was addressed, and in [64], the work was expanded to a multi-agent security problem.

The following subsections cover the existing shepherding approaches that are based on Reynold's rules [39] for boids since they are applicable in biological and physical systems. I divide the existing work into three categories: (1) work done to handle obstacle-free environments, (2) work done to handle cluttered environments. (3) the methods designed with the use of learning-based methods.

2.3.1 Shepherding Methods in Obstacle-free environments

Strombom et al. [59] proposed a self-propelled particle model of local attraction-repulsion for one shepherd herding a group of interacting agents towards a predefined destination. The model was based on observations of the behaviour of a trained female Australian Kelpie working-farm-dog while herding 46 three-year-old female merino sheep in South Australia in March 2010. The model could be used in other more complex tasks as it was found scalable to herd a flock of up to 300 sheep demonstrated experimentally. The results obtained by GPS systems attached to the sheep allowed the authors to model and understand the animals'collective behaviour during a threat, as discussed in [59] and [60], which envisaged a similar set of rules of predators preying on a flock.

The models in [59, 60, 65] are suitable for a swarm of biological agents, including the sheep in a free-space environment and normal weather conditions. The assumptions may also be valid in human control during crises in indoor scenarios. However, these assumptions may not be suitable for a swarm of robots with a limited sensing range. Given that the members of the swarm have a limited sensing range, this might result in the fragmentation of the swarm, which would then lead to the failure of the mission. In these models, each swarm member was assumed to view a maximum number of its nearest neighbours regardless of their distances and sensing range.

The driving and collecting rules modelled in [59] were improved in CADSHEEP method in [17], such that the behaviour of collecting dispersed sheep from the flock is only applied when it serves in pushing the flock towards home. Recently, these rules were learnt by the shepherd as an agent in [66], where the authors designed a curriculum-based reinforcement learning model for the shepherd to efficiently learn an effective shepherding policy by dividing the shepherding task into two sub-tasks; driving the sheep to home and collecting the dispersed sheep from the flock.

The forces modelled in [39, 59] are applied on each swarm member assuming that each swarm member has real-time information about the locations of all other swarm members and thus is able to compute the flock centre of mass accurately so that the flock remains cohesive. The assumptions of large $R_{\pi\pi}$ for maintaining flock connectivity have extra hurdles on the success of the shepherding task [24].

The authors in [58] studied the effect of the inability of each member to have an accurate estimation of the flock centre of mass which may lead to the dispersion of the flock as they are guided by the leaders. They suggested a consensus algorithm to allow the followers to make a more accurate estimation of their locations. As a result of the swarm members' restricted field of vision, the assumption of connectedness in [39] is broken, which in turn leads to fragmentation and ultimately the failure of the mission. Furthermore, the sheepdog's ability to watch the flock in motion, make a decision about where to apply pressure, and then physically get to that spot is crucial to the success of the shepherding task.

The authors of [67] explored the influence of two forms of signal disturbances on dynamic systems such as autonomous shepherding. They analyse shepherd sensing sheep-location noise and sheepdog actuation noise. This study simulates the empirical model in [67] under actuation and perception noises to prove the algorithm's parameterisation delivers stable performance. According to their findings, reducing actuation noise is more important for the shepherding agent than sensor noise, and various noise levels need different parameterisation. To achieve effective shepherd performance, the threshold required to collect scattered sheep should fluctuate with noise levels

In [68], the authors discuss shepherding huge clusters of passive agents modelled as continuous coherent and sparse areas by a team of three active agents that are eliminating the passive agents. Agents with similar features were assumed to move cohesively as a group. The grouping strategy of the passive clusters is based on restricted Boltzman machine [69] to train the autoencoder, where PSO is used for feature selection to enhance it. Given the improved grouping strategy, the autoencoder could reduce the dimensionality [70] of graphic representation of the passive agents' clusters. After some iterations of stochastic movements of the active agents considering their influence range of targets, the structure of the group becomes dynamic.

Using dynamic system control laws, a single herder or shepherd was designed using the dynamics of two- or three-dimensional nonholonomic vehicle models to herd a non-cooperative flock in [71]. The problem of dispersion in the flock was addressed in [72] for diverting a flock of birds away from a prescribed area, such as an airport, by determining a specific subset of agents that is needed to be influenced. In this work, an AUV collectively drives the school of birds away from the airport air given their network dynamics by using the dynamic control laws for maintaining the flock stability.

The authors in [73] designed a heuristic algorithm for a shepherd to guide the furthest sheep from home in a shepherding task of a heterogeneous flock that is composed of responsive and non-responsive sheep. Their assumptions are based on the availability of global information at the shepherd including the sheep type either responsive or non-responsive. While in [74], the authors consider normal sheep that are influenced by all the four forces modelled in Equation 2.1 (separation, alignment, attraction, and repulsion from the shepherd) and variant sheep that are only influenced by three of these forces. During the shepherding task, the shepherd uses a predictive control model to discriminate the sheep type by observing their deviation from the predicted trajectory. With this sheep categorisation, the shepherd targets the furthest sheep from the home location to be influenced to the home by applying the proposed furthest agent targeting (FAT) method.

The FAT method can be considered a successful dynamic method adopted by the shepherd to change its destination according to the swarm dynamics [73, 74]. It has been extended to multiple herders in [75], where the herders have no communication abilities. Each shepherd guides the whole flock by chasing its own target sheep independently through FAT where the furthest sheep is selected using a weighted sum of being furthest from the home location and the shepherd. Cooperative behaviour naturally emerged among two to four shepherds as a consequence of the spatial distribution of shepherds in three different scenarios and the task time decreased as the number of shepherds increased.

The occlusion-based method was introduced in [76] to herd flocks of sheep to a home location. The performance of the coordination methods proposed was guaranteed by the Lyapunov stability theory. To validate the work up to 50 sheep of two flocks were herded by 5 shepherds that were assumed to communicate, and know the target positions that the sheep must be herded to as well as their relative distances to the flock's centre of mass (COM). These assumptions were enforced through the use of COSα system [77] for location tracking in validating the occlusion protocol in indoor experiments of 7 sheep and 3 sheepdog MONA robots [78].

The authors in [76] highlighted that in an outdoor environment, the sheep mobility model may be altered and the communication system between the controlling laptop may not be ideal which may affect the performance of the robotic sheepdogs. Recently, in [79], multiple herders relying on dynamic system stability have been used in shepherding tasks. In this work, the authors focused on the motion control of the barking UAVs acting as sheepdogs that operate using a sliding mode-based control approach that teaches the UAVs to follow the moving limits of the animal's footprints and avoids collisions with other UAVs. Powerful computational simulations of animal dynamics simulated as per Reynold's rules [39] demonstrate the efficacy of the proposed approach.

2.3.2 Shepherding Methods in Cluttered Environments

The presence of obstacles is a challenge in flocking and thus creates difficulties in guiding flocks in herding tasks. This is because guiding a well-connected flock requires less time and energy than guiding a flock that is prone to dispersion. This encourages researchers to address the shepherding problem in different environments.

In [12], shepherding was defined as a mix of two behaviours: collecting a flock to reduce dispersion and driving it to keep it moving forward. The three different methods studied were using (a) a straight line, (b) a safe zone, and (c) a dynamic roadmap, whereby the steering point was (a) straight behind the flock, (b) side-to-side behind it, and (c) turning it for driving, respectively. According to Lien et al.'s experiments, the side-to-side approach was the best one resulting in more movements by the shepherd moving to achieve shorter travelling distances by the flock in both open and cluttered environments.

The research study in [72] assumed complete knowledge provided by a satellite central vision system and was extended in [80] to find a feasible trajectory of an Unmanned Aerial Vehicle (UAV) to minimise dispersion. Then, approximations of the distances between an approaching pursuer and a flock were derived based solely on their local interactions.

The work in [80] extended the dynamic method in [72]. The authors in [80] introduced a waypoint method that is based on the mechanical model of a swarm motion in response to herder locations. The goal is to create a UAV trajectory that minimises dispersion while herding birds away from an airport. Then, distances between an approaching pursuer and a flock were estimated based on local swarm interactions. This investigation assumes perfect satellite central vision knowledge.

In [81], the authors proposed a control approach for shepherd-like robots using position-based steering to govern the flock. Each agent estimates its movement by summarizing each rule. The flocking sheep agents notice the guiding agents and strive to avoid them, which moves the flock. Each steering agent must guide the closest flocking agent to its destination. Multiple steering agents create an arc to direct the flock without centralised coordination. Then, we suggest a novel rule for collecting behaviour that consolidates dispersed flocks.

In [82], the authors present deep reinforcement learning techniques combined with the probabilistic roadmaps to training a shepherd to herd agents around barriers utilising noisy but regulated environmental and behavioural data. The simulation results suggest that the proposed strategy is robust, and insensitive to environmental and behavioural model errors in different shapes of obstacles. The learning-based method in [82] has a higher task SR, and shorter task completion time and path length than that of the rule-based behavioural methods in challenging scenarios with groups of passive agents with low sensing ranges and strenuous passages.

Modelling the agents in the autonomous shepherding scenario as physical systems helps in quantifying the influence factors in the autonomous shepherding task. These factors can be used as the measuring metrics for the effectiveness of the autonomous shepherding method. A robot may successfully replace a human shepherd in charge of a flock of swarms modelled as robots if it can do the task in a limited amount of time. Therefore, reducing the amount of time spent on the task results in a reduction in the amount of energy required for the whole task by all the agents. A critical assessment of each of the above-reviewed pieces of research is outlined in Table 2.1. It is worth noting that these research studies made ideal and unrealistic assumptions about the flock's cohesion by assuming large ranges of the field-of-view for the sheep and the sheepdog.

Automatic optimisation and control in an engineering system are correlated with near-real-time data collecting and optimum decision-making [83]. Swarm control and multi-agent systems (MAS) operate together to offer an integrated framework for system optimisation and control. In its simplest form, a system may consist of a single active agent and a single passive agent; nevertheless, a dozen passive agents may be considered in certain situations. However, to minimise the complexity of optimising computations, a small number of passive agents is more often used to develop optimum control algorithms for active agents within constrained time and computing complexity [84]. Maximising quality and performance while minimising flock control expenditures is a shared goal.

The optimisation of autonomous agents' activities proposed in [85] was studied by teaching Bayesian networks. Teaching Bayesian networks by using a genetic algorithm was applied to a discrete problem of agents' activities selection which is a similar problem to shepherding one sheep with one agent on a 4×4 mesh [42]. In [85], a large number of passive agents have been studied through simulation and real-life experiments using heuristic control algorithms. Rule-based techniques have been used in [15] to study the feasibility of a group of shepherds working cooperatively without communication to efficiently control the motion of another group (the flock).

In [86], the authors describe a novel motion planning approach, dubbed Deform, for shepherding in settings involving obstacles. Up to fifteen sheepdogs were in charge of guiding up to two hundred sheep to a predetermined destination. The suggested solution operates effectively in a variety of contexts where shepherds see the flock as an abstracted deformable shape, allowing excellent scaling to bigger teams of shepherds and larger flocks despite the rise in the sheep's stochastic mobility.

This motivated the authors in [87] to teach canine-like robots to herd a flock of recalcitrant sheep-like entities towards a desired point in free space by imposing mathematical relationships among them. The authors proposed a control mechanism for any number of sheep led by two or more dogs, in addition to a projection of the combined dynamics of the dogs and sheep to a simple unicycle robotic system. It was demonstrated that one sheep may be driven to a certain area using two or more dogs. Matlab simulations and hardware experiments with Pololu m3pi robots demonstrated the effectiveness of the proposed control technique.

2.3.3 Learning-based Methods for Shepherding

The majority of the research work evaluated in [14] relies on humans to infer a model and set of rules, which is not an effective solution to complicated problems like shepherding, particularly when considering environmental restrictions like sheep's restricted sensing range. Long et al. outlined three significant challenges as follows

- Human vision may be skewed, as addressed in [88]; if people are unable to observe the whole spectrum of swarming, it is possible that models based on human perception of swarming are flawed or incomplete.
- Human-designed models are not guaranteed to be the most efficient method to lead an artificial agent to shepherd, nor the appropriate and/or only approach under varied swarm limitations on memory, processing, and energy. Moreover, transferring sheepdog behaviour to a robot may not guarantee that the behaviour is beneficial and/or efficient for the robot. Similarly, UAVs have

distinct physical limits than biological sheepdogs.

- Existing approaches are not readily adaptable to a dynamic environment and cannot be generalised to different kinds of shepherding by addressing swarm size in a single model [18, 89, 89]

In the literature, there have been previous approaches for solving the single agent and multi-agent path planning problems [61, 62, 90]. Although path planning is one of the most fundamental difficulties that must be resolved to allow the guiding agent to guide the swarm in a limited time in different environments [91, 92], there are no similar path planning methods for swarm guidance, especially for swarm with limited sensing range.

One of the most common reactive methods is the artificial potential field (APF). With APF, the swarm can apply Reynold's rules [39], and the sheepdog may apply the agitation force of the swarm to guide them in the desired directions following the shepherding method in [59]. Nonetheless, the efficiency of APF relies on observations of each individual swarm member. In [93], the author models robot sheep with reactive behaviours using APF for multiagent systems to assess the difficulty of the shepherding problem and provide a greedy method that solves it in a linear time. The worst-case length of the shepherding task is linear in the sheep's visual range. The authors also examine how well such tactics may be learnt since learning often yields tremendous results.

Artificial Intelligence (AI) technologies may be categorised generically as analytics and autonomy [67]. Analytics focuses on algorithms that provide perception, interpretation, and projection of sensorial data-derived information. Autonomy revolves around making decisions and modifying the environment via action formation. Multiple intelligent systems evaluating and interacting in order to accomplish a number of objectives pose an intriguing challenge for artificial intelligence in the context of dog-shepherding.

In recent years, machine learning has emerged as the best method for this type of challenge. Deep learning is transforming practically every computer science subject by using sophisticated hardware [94]. While deep learning comprises several research branches and is itself a subfield of machine learning, one of the most promising is its combination with reinforcement learning (RL), known as Deep Reinforcement Learning (DRL). DRL was named one of MIT's ten breakthrough technologies in 2017 [95], and recent advancements imply it has endless potential to further artificial general intelligence.

There are several ways that employ learning to computationally enhance traditional SMPs. A Lightning framework [96] saves paths in a lookup table and uses a learning heuristic to construct new pathways and read and repair old ones. Coleman et al [97] experience is an experience-based technique for caching experience in a graph instead of individual trajectories. Although these approaches surpass standard planning procedures in higher-dimensional regions, lookup tables require a lot of memory and are not particularly effective at making generalisations to new planning circumstances. Zucker et al [98] proposed a reinforcement learning-based method to bias samples in discrete-time environments. However, since they need a large number of interactive encounters, reinforcement learning-based approaches are known for their late convergence.

Authors in [99] proposed the DeepSMP1 approach for intelligent neural sampling. It is composed of two different neural units. The initial part of the system is an autoencoder, which integrates point cloud data from the obstacle space by learning a stable and invariant feature space. The second module is a stochastic DNN that generates incremental samples for SMPs during online execution by using barrier encoding, and initial and final values for start and goal parameters. Importantly, this technique can be generalised to unseen scenarios through obstacle space encoding, and any SMP may utilise these informed samples to converge quickly to the optimum solution. This approach may adaptively sample a portion of configuration space that is most likely to include an optimum route solution, integrate with SMP methods, have a short completion time, and be generalised to uncharted settings. However, its potential for generalisation has not been validated.

Reinforcement learning strategies to transfer information to unseen but analogous settings, such as in the shepherding task, are emphasised in [93]. Two methods are examined: State space abstraction creates an abstract state space from "similar" states, whereas function approximation methods aim to approximate the reinforcement learning agent's value function. This study inspired the authors in [100] to use SARSA to simulate a dog herding sheep using reinforcement learning. The robot used reinforcement learning to herd sheep to the goal by first attaining a sub-goal. The dog is awarded for sub-goals and punished for not herding. Stochastic sheep-dog interactions and multiple sub-goals slow agent learning until the 350th episode of the learned shepherding task, when the agent succeeds.

In [56], the physiological and behavioural responses of twelve Dorper sheep (Ovies aries) to UAV are studied in order to adapt mathematical models of shepherding to the new dimension. The authors laid the groundwork for AI to assist farmers and pilots in becoming more self-sufficient in flock management from a bird's eye view. The creators of [101] were inspired to create a robotic dog equipped with a coordination algorithm so that it could herd the sheep using occlusion-based motion control. This control system is more adaptable and efficient than formation-based strategies for herding large numbers of sheep. The suggested method was verified by simulations and lab-based studies using actual robots and a vision-based tracking system.

The authors in [102] proposed an AI-dog that employs influence mapping, state machines, and A* [103] route finding to intelligently react to real-world shepherding directions provided by a high-level shepherd AI controlling a flock of sheep through waypoints on several maps. Human testers considered the AI shepherd to be a formidable opponent in competition (using a point-and-click or voice recognition interface). User testing revealed that the system's AI components contributed to its authenticity and enthralling gameplay. Such a smart autonomous system (SAS) combines analytics with autonomy in order to comprehend, learn, decide, and act autonomously as a watchdog artificial intelligence (WAI) agent that supervises a human and SAS-based ecosystem. Implementing this concept is still in progress due to the complexity of the system which requires not only path planning methods but advanced data perception and sensor fusion methods.

The literature on shepherding has been studied to tackle different challenges in the problem including the feasibility of the use of a robotic system to replace the sheepdog. Then, more challenges have been considered including the heterogeneity of forces among the swarm members, the feasibility of building a stable dynamic system of swarm robots and herders, and the feasibility of robotic systems to herd swarms in a different environment. In the last decades, various methods have been used to deal with these challenges independently by assuming an ideal communication challenge in the herder systems or by assuming an ideal dynamic system in all the agents. Recently, optimisation and deep learning methods have been developed at the sheepdog to be able to successfully complete the shepherding task in a limited time. The most common problems that were addressed in the literature are summarised in Table 2.1.

Since there is no mechanism for teaching a person the reinforcement learning reward function, the authors in [52] researched educating a computer. They create reinforcement learning reward functions using systematic instructional design, a human education technique. A hierarchical evolutionary reinforcement learner employs a neural network to produce a boids-based swarm controller. The approach may help create hierarchical reinforcement learners that learn progressively through a multi-part reward function. The hierarchy includes lesson-specific behaviours and skills. In [66], the authors devised a curriculum-based reinforcement learning model for the shepherd to efficiently acquire an effective shepherding policy by separating the shepherding activity into two sub-tasks: driving the sheep home and collecting the scattered sheep from the flock.

Table 2.1: Shepherding Models

Problem	Method	Single herder
Herding a flock of ducklings	Geometric rules based on the APF[48]	Yes
Collecting and driving behaviours definition	three approaching and steering methods [12]	Yes
Scalability in shepherding	Heuristic method based pushing the centre of a flock of up to 300 sheep [59]	Yes
Scalability of shepherding passive agents with stochastic motion	Autoencoder to reduce the dimensionality of the graphic representation of the problem [68]	No
Scalability for shepherding	sliding mode-based control approach with multiple flying shepherds [79]	No
Herding without dispersion of birds	n-wavefront algorithm to determine the subset of agents to be influenced [72]	Yes
Herding without dispersion of birds in cluttered environment	m-waypoint algorithm used for a UAV to safely herd the flock without fragmenting it [80]	Yes
Artificial intelligence in shepherding	AI watchdog system [102]	Yes
Herding in cluttered environment	Differential evolution-based method [17]	Yes
Restricted field of vision of the flock centre of mass	Consensus among followers for localisation [58]	Yes
Herding initially dispersed flock	Occlusion-based method with multiple shepherds communicating [76]	No
Restricted field of vision of the flock centre of mass	Deep reinforcement learning based method [66]	Yes
Shepherd sheep-location and actuation noises	imitation learning for shepherding parameterisation [67]	Yes
Heterogeneous flock	FAT method [73]	Yes
Heterogeneous flock	FAT and predictive control model for sheep discrimination [74]	Yes
Heterogeneous flock	FAT with multiple shepherds without communication [75]	No

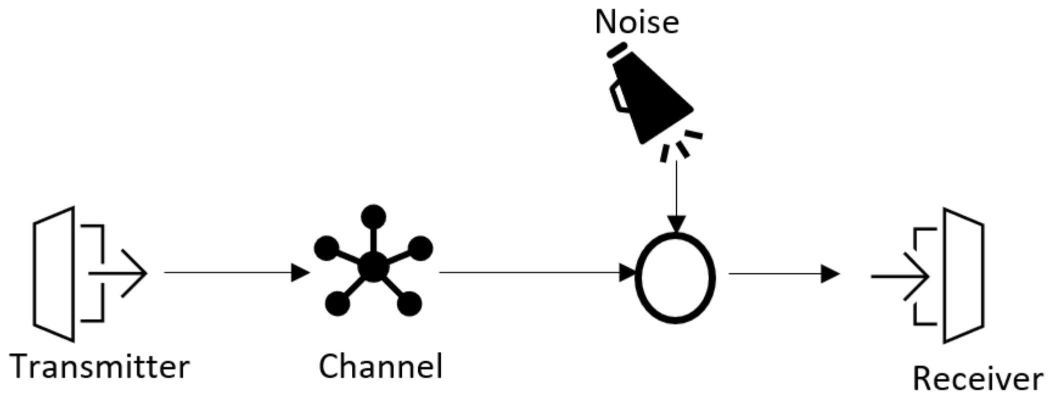

Figure 2.2: Components of a communication system

2.4 Communication Systems

As mentioned earlier, a swarm usually needs a communication system that can help control its movement. In this section, an overview of communication systems and their components is provided.

A communication system models a transmitter-receiver exchange of data through a channel that allows the transmitted signal to move through a transmission medium under the effect of noise, attenuation, and distortion. The basic components of communication systems are depicted in Figure 2.2 that include a transmitter (Tx), receiver (Rx), and a communication channel that allows the signal to move under the effect of channel noise.

Both the Tx and Rx in communication systems are vulnerable to the disturbance caused by internal noise. According to [104], noise in a communication channel may distort the data being communicated, making it impossible for Rx to read the data being delivered accurately. The mobility of the end systems and the changes in the surrounding environment both contribute to the introduction of noise in today's communication systems.

Non-ideal components or currents may cause device noise. These include residual resistance, capacitor loss, leakage current, and material defects. Analysing noise by presuming the gadget works like a textbook is unsuitable [105]. Thermal and shot noise are device-related. These noises indicate oscillations in the resistance or emission inside a device [106, 107]. Shot noise comes from discrete processes like charge unit flow. Thus, thermal noise is controlled exclusively by the device's loss at a certain temperature, but shot noise may be minimised by improving the correlation between events in a single device [106].

The excess noise in devices and the physical source of the noise have been reviewed in [105]. Then, a networked system identification challenge to find mathematical models for control/estimation/filtering systems has been studied in [108] for linear time-invariant (LTI) open-loop processes in a networked context. In order to improve the efficiency of noise measurement systems, technologies, and circuit designs for low-noise applications, the authors in [107] provide specific processes for the noise measurement system, noise parameter de-embedding, noise source extraction, and noise source implementation.

It is possible to establish electronic communication in both wired and wireless systems. The limits imposed by mobility on the usage of wired communication are significant. Due to the need for navigation in an unstructured environment for a swarm of robots, wireless communication was chosen straightaway. There are several methods for wireless communication, such as acoustic propagation, radio-frequency (RF) transmission, etc. As a model of communication, acoustic propagation-based systems have several limitations. As a result of the low transmission frequency, the bandwidth of such a system is constrained [109].

This bandwidth constraint limits the use of acoustic propagation-based systems in many channels for sound transmission. High transmission power causes an overloading issue on the receiving antenna, popularly known as the 'Near and Far' problem [110]. The near-and-far problem occurs when an acoustic receiver simultaneously receives and transmits from the same base system, decreasing the transmitted power and range. In situations where large propagation delays remain in the range of seconds, transmission speed is also an issue.

Bandwidth efficiency maximises spectrum use by allowing more information, whereas data rate indicates the maximum information transfer across a channel. Power efficiency means transmitting trustworthy information with optimal power. All these aspects may not be optimised at once. Power efficiency requires lower-order modulation, which reduces bandwidth efficiency and data. The transmission speed restriction affects the maximum data rate, which is determined by channel capacity. Consequently, there is a trade-off between the expectations of different modulation techniques. In the design of digital radio frequency (RF) systems, the optimization/trade-off of these parameters is application-oriented.

Since the RF stations are hardwired to an external power source, bandwidth efficiency with low bit-error-rate (BER) is given significant attention while designing a terrestrial microwave radio connection. Since just a small number of receivers are needed, neither power efficiency nor the cost/complexity of receivers are a primary consideration. However, due to the constraints imposed by mobile phones' batteries, efforts to improve power efficiency have mostly been directed towards cellular communication. Consequently, in mobile communication, battery efficiency and cost efficiency are more significant restrictions than bandwidth efficiency.

Cordless phones, cellular communication, LAN, MAN, WAN, and PCS, as well as radio and television, radio-frequency identification (RFID), keyless door entry, patient monitoring in hospitals and nursing homes, and keyboards and cordless mouse for PCs all employ radio frequency (RF) or microwave transmissions [111]. While some of these applications have historically employed infrared (IR) technology, contemporary developments are shifting towards radio frequency (RF), since IR needs a direct line of sight connection.

The limitations on acoustic propagation-based systems and the spread of long-range wireless communication systems like mobile phones and satellites motivate the use of RF technology in our swarm communication system. Successful usage of a radio frequency communication system among a swarm of mobile robots and effective navigation to the location of interest is addressed in [112, 113]. Thus, enabling interaction between different physical realms through a common protocol that is compatible with several platforms is essential.

2.4.1 Wireless Communication Channels

The atmosphere's electrical properties may impede or enhance the transmission of electrical impulses. Ionisation of air creates several levels in the atmosphere, both in the ionosphere and the troposphere. A radio frequency signal may either travel through the earth or be reflected off the ionosphere before reaching the receiver, as shown in Figures (a) and (b). These transmissions may be thought of as either ground waves or air waves. Season, time of day, and solar radiation all have a role in how the sky wave behaves. As can be seen in Figure 1.2, microwaves are transmitted with little attenuation by the ionosphere, and signals travel only along direct lines of sight (c). For this reason, a microwave connection can go no farther than around 50 kilometres before it loses signal due to the earth's curvature.

Increasing the range of a microwave link can be achieved through the use of a human-made reflector in the sky, known as a satellite communication system. Another method to extend the range is by placing repeaters at periodic intervals, referred to as a terrestrial communication system.

In satellite communication, putting an artificial reflector in the sky is one approach to improve the range of the transmitted signal. While in a terrestrial communication system, the placement of repeaters at periodic intervals is another method for extending the range of a microwave connection. The air-to-air communication connection assumes no blockage between end-systems while considering air density changes at high altitudes [114]. This model may not match the swarm guidance application requirements because β will have to adjust latitudes while guiding the ground swarms and communicating with the CU.

For the most part, current UAV channel models assume that the velocities and directions of both the transmitter and receiver (Tx and Rx) remain constant. However, as was explored, in practical communication settings, the Tx and Rx of UAV may encounter variations in both speeds and trajectories, as studied in [115]. On the other hand, a ground-to-ground communication link that suffers from excessive levels and causes of channel fading may not match the assumption of complete real-time observation of the environment that is available at the CU.

The air-to-ground communication model consists of two end systems with a considerable difference in their heights [113]. The higher-height system experiences less signal fading than the lower-height system. In the conventional mobile communication models, the base station is immovable and equipped with sufficient power sources, while the mobile devices are rechargeable and powered by batteries devoted only to the communication process.

Recently, the notion of micro-UAvs communication was created inside military usage and has seen an astounding development of UAVs for civil and academic uses. This development is being fueled by the wide variety of applications for this technology. Some examples include fire detection, search and rescue, monitoring, construction assessments, agricultural monitoring, remote sensing, weather services and more. The increasing utilization of UAVs has been facilitated by advancements in low-cost control solutions, the progression of microelectronics with readily available sensors and components, as well as the growth of a worldwide community of developers with various UAV-related open-source projects [116].

Telemetry data, control directives, and other information must be sent from the UAV to a ground control station over a secure communication connection for the drone to operate safely. Many solutions have been investigated and put into practice; some of them are mentioned in [117]; nevertheless, most of these systems have either limited operating range or high implementation complexities. The range constraint may be circumvented with decreased complexity by using existing broad coverage mobile communications infrastructures.

The research presented at [118] explored the potentials of mobile networks with their fully implemented infrastructures, extensive radio coverage, high throughputs, decreased latencies, and the broad availability of radio modems. According to the authors, the unmanned aerial system (UAS) may be built in a modular fashion that enables it to use a combination of wired and wireless components (such as unmanned aerial vehicles and ground control stations). This paper offered a UAS architecture backed by flight testing to demonstrate the viability of depending on 4G networks for the operation of vehicles in semi-automatic or fully-automatic modes with minimum jitter and packet loss. Extreme difficulties arise due to the fact that the propagation environment of UAV-aided communication systems is different from that of conventional systems. In order to effectively build and implement these communication systems, a precise knowledge of the UAV wireless channels is required. Furthermore, environmental issues constitute obstacles to peer-to-peer implementation in a truly dispersed form, as will be explained below.

2.4.2 Channel Models For Noise

There are several sources of natural noise that result in wideband noise, such as thermal vibrations of atoms in conductors (commonly referred to as thermal noise or Johnson-Nyquist noise), radiation emitted by heated objects including the Earth, and astronomical sources such as the sun [119]. In contrast, narrowband noise encompasses shot noise, which can result in power spectral density peaks that rise above the background noise by 50 dB and can be modelled through the use of modulated sinusoids [120].

In [121], the authors suggest a relatively simple model that combines ambient noise and impulsive noise, making it useful for predicting analytical performance in worst-case scenarios. This model involves adding Middleton's Class-A noise to the received signal without noise. Class-A noise is a sample from an i.i.d. discrete-time complex random process, the probability density function of which is an infinite weighted sum of Gaussian densities with decreasing weights and increasing variances. Additive White Gaussian Noise (AWGN) is a fundamental noise model used in information theory to simulate the impact of various natural random processes, and its characteristics are denoted by modifiers:

- Additive since it adds to all the system-intrinsic noises.

- The term "white" refers to the uniform power across the frequency band of the information system. It is analogous to the colour white, which emits uniformly at all visible frequencies.
- Gaussian since it has a normal time domain distribution with a zero mean. Based on the central limit theorem of probability theory [122], the aggregate of many independent random processes has a normal or Gaussian distribution.

Transmission in AWGN channels is hindered by the linear addition of wideband or white noise with constant spectral density (watts per hertz of bandwidth) and Gaussian amplitude. However, the model overlooks fading, frequency selectivity, interference, nonlinearity, and dispersion. Before examining these other phenomena, AWGN channel model constructs simple mathematical models to comprehend a system's behaviour. Satellites use AWGN channels, while in terrestrial route modelling, AWGN replicates the background noise of the channel under examination. AWGN amplitude diminishes the SNR which increases uncertainty in the received signal over time.

For the AWGN channel, a theoretical limit to the maximum data rate R_b that can be transmitted in a channel with a given bandwidth BW is given by the Shannon theorem [123] as $R_b < B\log_{10}(1 + SNR)$ where the signal power to the noise power expressed in decibels (dB) is $SNR_{dB} = P_{signal}/N_o$, A ratio higher than 1 indicates more signal than noise. This means that for the same bandwidth, the data rate is inversely proportional to SNR, which is also affected by the changes in the strength of the signal power.

Hartley-Shannon law [123] establishes an upper limit on the capacity of a communication channel (C) with a given bandwidth (B), signal-to-noise ratio (NSR), and gaussian noise $C = 3.32 \times B\log_{10}(1 + SNR)$.

2.4.3 Channel Fading Models

Fading in wireless communications involves signal attenuation that varies with time, location, and radio frequency. Since the 1980s, phenomenological and statistical studies have characterised the fast fading "wave interference" that is associated with time-varying propagation [124, 125]. In multipath propagation, weather (especially rain), and shadowing from barriers may cause fading in a wireless channel. Since the channel may be modelled as a linear time-invariant (LTI) system, fading can be described as changes in the channel transfer function within a fraction of a second or multiple seconds [125].

Statistical models of fast fading affect modulation, diversity, and coding choices. The channel fading models and their predictions match system performance analysis studies and laboratory "channel fading simulators" for flexible testing of novel ideas or designs [126]. In the 1950s and 1960s, fading channel processes were studied for over-the-horizon communications over several frequency bands. The 300 MHz-3 GHz UHF and 3-30 GHz SHF bands are employed for tropospheric scatter and ionospheric communications, respectively. Early models may assist quantify fading effects in mobile digital communication systems, even though mobile radio systems have distinct fading effects than ionospheric and tropospheric channels [126, 127].

In obstacle-free environment and within a limited distance between the Tx and Rx, the transmitted signal suffers from free-space path loss that is proportional to the distance between the Tx and Rx, as described in Friis formula [128] in Equation (2.3)

$$P_r(d_{CU-\beta}(t_h(i))) = \frac{P_t G_t G_r}{(4\pi/\lambda)^2} \times \frac{1}{(d_{CU-\beta}(t_h(i)))^2} \tag{2.3}$$

Assuming that the Tx is the CU and the Rx is the sheepdog, $d_{CU-\beta}(t_h(i))$ represents the distance between the Tx and Rx at the time of transmission of the sheepdog heading $t_h(i)$. Moreover, the notations P_r and P_t are the received and transmitted powers, respectively, and the P_t is assumed to be constant as well as the transmitter and receiver gains G_t and G_r for isotropic antennas [129] at the Tx and Rx, respectively. Thus, at constant transmission frequency, the wavelength λ is constant, which means that the received power is inversely proportional to the square distance between the CU and β.

In cluttered environments, reflectors around transmitters and receivers produce many signal paths. The receiver perceives several copies of the transmitted signal travelling various paths with various levels of attenuation, delay, and phase shift. This may increase or decrease the received signal power. Deep fade, or strong destructive interference, may cause communication to fail to owe to a large decline in channel SNR. Wireless communication signals suffer Wireless communication signals suffer the following:

- Reflection happens when a propagating electromagnetic wave hits a smooth surface with unusually large dimensions relative to the RF frequency ($f = c/\lambda$).

- Diffraction happens when the radio channel between the transmitter and receiver is blocked by a dense body with large dimensions relative to the signal wavelength, resulting in the formation of secondary waves behind the obstructing material. Diffraction allows RF energy to travel from transmitter to receiver without line-of-sight. Shadowing occurs because the diffracted field may reach the receiver even when shaded by an impenetrable obstacle.

- Scattering happens when a radio wave impinges on a large, rough surface or any surface with dimensions causing the reflected energy to disperse (scatter) in all directions. In metropolitan areas, lampposts, street signs, and greenery disperse signals.

Cellular networks, underwater acoustic communications, and broadcast communication employ fading channel models to simulate electromagnetic transmission over the wireless channel [130]. David Tse [104] roughly divided the types of fading into two:

- Large-scale fading from path loss and shadowing by large objects like buildings and hills. This frequency-independent phenomenon happens when the mobile passes across a cell size.
- Small-scale fading owing to constructive and destructive interference of the signal with its reflections. This happens on a spatial scale comparable to the wavelength of the carrier wave and is frequency dependent.

Physical conditions determine the density of barriers between transmit and receive antennas. Outdoor plains have few impediments, whereas inside surroundings have numerous obstructions. Shadowing captures this environmental unpredictability by representing obstacle density and absorption behaviour as random integers. It differs from multipath fading significantly since shadow fades endure in many seconds or minutes [104].

Large-scale fading is common in either air-to-air and air-to-ground transmissions due to atmospheric conditions of rain or gases. Rainfall as well as the mobility of the shepherd may lead to small-scale fading. For the rain, it absorbs, scatters, and diffracts the propagated wireless signal into multiple paths that are received at different times; while for the mobility of the shepherd in a cluttered environment causes the same effect [131, 132].

Moreover, the operating frequency may decrease the resilience of the transmitted signal; for example, lower frequency bands have higher penetrating capabilities into obstacles compared to high frequency carriers [131]. Small-scale multipath fading is more relevant to the design of reliable and efficient communication systems. The mobility of the Tx or Rx or both present doppler shift in the peaks of the signal at the received signal compared to the original time gaps between the peaks of the transmitted signal[133]

However, attenuation and propagation delays normally fluctuate slowly with frequency. Time-varying route lengths and frequency-dependent antenna strengths affect these fluctuations. We can eliminate this frequency dependency when broadcasting across narrow bands relative to the carrier frequency. Due to path delays, the total channel response may vary with frequency even if the individual attenuation and delays are expected to be independent of frequency. This has been proven through recording median signal levels at 800 MHz for small-scale and large scale signal changes in 4-foot square regions inside and around eight suburban residences from different sites using a van in [134].

The authors of [135] compared three indoor propagation measurements at frequencies of 28, 73, and 142 GHz for mm-Wave to sub-THz radio propagation in a consistent indoor office environment at the NYU WIRELESS centre. The experiments measured wide bandwidth signals (e.g., 100 MHz) in 5G and future wireless communication systems over distances up to 40 meters. The results showed that the simplest model for channel filter taps assumes a large number of statistically independent reflections and dispersions with random amplitudes in the delay window of a tap. When there are numerous reflections and no line-of-sight signal, the envelope of the received signal is statistically described by a Rayleigh Probability Density Function (pdf). However, if a strong nonfading signal, such as a line-of-sight transmission path, is present, the small-scale fading envelope is described by a Rician pdf.

Researchers have been measuring and estimating propagation path loss for different applications in different environments [115, 136, 137, 138] where X_σ and n are set accordingly. In [139] the Rice factor (X_σ) was measured versus the distance between Tx and aircraft Rx near the airport in an air-to-ground channel for 5 GHz. In [140], it is observed that Rician fading occurs when the envelope amplitude due to small-scale fading has a Rician probability density function and the received signal contains many reflected rays and a strong line of sight component. The specular components are non-faded and their amplitudes approach zero where the Rician pdf approaches a Rayleigh pdf [127].

It can be noted that small-scale fading is shown in two phenomena; (a) time-spreading of the signal's underlying digital pulses; and (b) motion-induced channel time-variation (e.g., a receive antenna on a moving platform). In typical cellular scenarios with a limited number of reflectors, the Rayleigh fading model [127] is adopted because of its simplicity. In general, a mobile radio travelling across a vast region must interpret signals that suffer both forms of fading: small-scale fading layered a top large-scale fading.

The Okumura model [141] is a radio propagation model based on data measurements conducted in Tokyo. The model works well in cities with numerous urban buildings but few towering blocking structures for frequency ranges ($[50, 1920]$ MHz) working on mobile station antenna at 1 to 3 m height and base station (BS) antenna at $[30, 100]$ m height, where the link distance is in range $[1, 100]$ km. The model serves as the foundation for the Hata model, a radio propagation model for anticipating the path loss of cellular communications in outdoor contexts that is valid for microwave frequencies ranging $[150, 1500]$ MHz. It is also known as the Okumura-Hata model since it is based on data from the Okumura model [142]. The model simulates city structure-induced diffraction, reflection, and scattering using Okumura model visuals. The Hata Model rectifies suburban and rural applications based on measurements collected in Oman and Egypt [143, 144].

COST (COopération européenne dans le domaine de la recherche Scientifique et Technique) is a European Union Forum for cooperative scientific research that produced the COST Hata (COST 231) model using experimental results in several European cities [145]. This empirical-deterministic model estimates urban route loss spanning 800 to 2000 MHz. COST 231 tested the Walfisch Ikegami model (normally used for frequencies up to 2 GHz) in the 3.5 GHz WiMAX deployment. Field WiMAX network signal power measurements are compared to model predictions. Root mean square error compares line of sight (LOS) and non-line of sight (NLOS) circumstances.

The work in [146] provides a theoretical comparison between the Okumura model [141], Hata model [142], and COST 231 [145]. In [147], the authors use image to show the difference between the effect of different channel modes. In [147], the random noise in AWGN channel marginally degrades the picture in the AWGN channel; random noise and block noise considerably impair the image in the flat fading channel; and random noise severely degrades the image in the frequency selective fading channel. The Rayleigh fading model [127] is representative for scattering mechanisms where there are many small reflectors, but is adopted primarily for its simplicity in typical cellular situations with a relatively small number of reflectors. In general, a mobile radio roaming over a large area must process signals that experience both types of fading: (a) small-scale fading superimposed on (b) large-scale fading.

In the following subsection, the different effects of the wireless communication signal strength are discussed. Then the basic information theory methods that were designed to deal with the wireless channel challenges are explained in the subsequent section.

2.5 Information Theory

A practical radio receiver design must address many essential characteristics including gain, selectivity, sensitivity, and stability. The modulated information is first impressed on the radio carrier signal, and the detector in the receiver is responsible for demodulating that signal [148]. Modulation and coding are transmitter activities that provide efficient and accurate information transmission [149]. The complimentary demodulation technique recovers the message by reversing the modulated wave, which "carries" the message information. Optimal techniques for detecting the presence or absence of a signal in noise and signal extraction from a noisy backdrop are of paramount practical importance [150].

2.5.1 Channel Coding

Shannon's basic block diagram of unidirectional transmission comprises a message source (Tx), a coding device that encodes the message for transmission, the transmission channel (CH), a decoding device at the receiver (Rx), and a noise source (N) that indicates the disturbance in the channel [151].

Long-distance digital communication may need coding and modulation. Encoding converts digital messages into new symbols. Decoding returns an encoded sequence to the original message, sometimes with transmission contamination errors. Uncoded message transmission from a source would need M distinct waveforms, one for each symbol. A K-bit binary codeword may represent each symbol. To encode M source symbols, we require $Klog2\ M$ digits per codeword. If the source generates r symbols per second, the binary code will contain Kr digits per second, requiring K times the bandwidth of an uncoded signal. Error-control coding increases bandwidth and device complexity but yields practically error-free digital transmission despite low SNR.

A basic solution for ensuring reliable communication across a noisy medium is creating an encoded vector with systematic redundancy. The decoding system leverages this redundancy to extract the source vector and channel noise from the received vector [149]. Source-coding methods use source signal statistics for effective encoding whereas channel coding aims to minimise redundancy for improving bandwidth efficiency. Demodulation and decoding need correct synchronisation parameter estimations to maximise channel code performance.

Turbo codes and LDPC codes are contemporary channel coding developments. Whereas capacity-approaching codes allow the receiver to function at very low SNR levels, making parameter estimation more difficult with minimal redundancy. Pilot symbols enhance standard data-aided (DA) and non-data-aided (NDA) algorithms [152, 153, 154].These approaches decrease throughput or increase acquisition time. Using soft information at the decoder output to solve this issue was examined [155, 156]. Code-aided (CA) algorithms use the coding structure to estimate parameters [155, 157]. Only linear, nontime dispersive, rician fading, additive white Gaussian noise (AWGN) channels were examined for this method.

2.5.2 Signal Modulation

Modulation allows the designer to insert a signal in a frequency range that is not limited by technology. To keep hardware costs and difficulties to a minimum, fractional bandwidth should be limited between $1 - 10\%$ of the absolute bandwidth divided by the central frequency. The basic goal of modulation in a communication system is to provide a modulated signal that is appropriate for the transmission channel's behaviour. It consists of two waveforms: (a) a modulated signal representing the message and (b) a carrier wave appropriate for the application. A modulator changes the carrier wave systematically in response to differences in the modulating input. Modulation has several practical uses, including the following:

- Modulation for efficient transmission over appreciable distance.
- Modulation to overcome hardware limitations
- Modulation to reduce noise and interference

An electromagnetic wave that travels through space, with or without a medium, is referred to as a modulated signal. The transmission efficiency depends on the frequency of the signal being transmitted. By leveraging the frequency-translation property of coded wave modulation, message data can be embedded in a carrier frequency that is chosen based on the desired transmission mode. The design of a communication system can be impacted by the cost and availability of hardware, which often relies on operating frequencies.

Modulation has also been utilised to restrict the requirement for raising the transmission power, which is a result of its effectiveness in reducing noise and interference. Increasing the signal strength until it overwhelms the contamination is a brute-force strategy for countering noise and interference. High power, on the other hand, is expensive and may harm equipment. Fortunately, FM and other kinds of modulation have the useful virtue of decreasing both noise and interference. This capability is known as wideband noise reduction because it demands a transmission bandwidth that is significantly larger than the modulated signal's bandwidth. Wideband modulation enables the designer to trade off expanded bandwidth for reduced signal strength, as described by the Hartley-Shannon equation. It should be noted that a higher carrier frequency may be required to provide wideband modulation.

In digital modulation, the modification of the transmitted signal's amplitude, phase, or frequency with respect to the digital message signal is used to categorise it. If the broadcast signal's amplitude or phase changes in relation to the message signal, the resulting signal is called amplitude shift keying (ASK) or phase shift keying (PSK), respectively. ASK and PSK are known as linear modulation methods since they employ the principles of superposition and scaling. In contrast, frequency shift keying (FSK) involves fluctuation of the transmitted signal's frequency with respect to the message signal. As a non-linear modulation technique, FSK is less spectrally efficient compared to linear modulation methods.

As a result, linear modulation methods are often used in wireless communications [140]. Another advanced modulation approach is quadrature amplitude modulation (QAM) and quadrature phase shift keying (QPSK), which modify both the amplitude and phase of the transmitted signal with respect to the digital message stream. QAM codes the in-phase (I) and quadrature (Q) channels individually, while PSK codes the complex sent symbol directly. Square QAM constellations are distinguished by their intrinsic spectrum efficiency and simplicity of implementation [158, 159].

Because of their bandwidth and power economy, QAM and FSK are commonly employed in communication protocols. Furthermore, M-ary QAM uses less power than M-ary PSK modulation [140] and is therefore commonly used in recent wireless communication protocols. Where M is the number of distinct values per symbol in M-ary QAM (M-QAM). At constant transmission power, as M increases, the data rate increases while the resistance to noise decreases.

The challenge of achieving power-efficient high-data-rate transmission while optimally utilizing limited bandwidth is a crucial aspect of modern and future wireless communication systems. The objective is to reduce the average transmit power of the signal while maintaining a certain bit error rate (BER) for power efficiency. However, high data rates require a higher transmit power to maintain the same level of performance within a limited bandwidth. One solution for this issue is adaptive modulation, which provides spectrally efficient high-data-rate transmission with optimal power utilization in current and future wireless communication systems.

In wireless communication, the received signal-to-noise ratio (SNR) can vary due to multipath fading. To improve system capacity and make the best use of limited bandwidth, adaptive modulation is crucial by assigning high data rates to channels with good channel conditions (high SNR) and low data rates to channels with poor channel conditions (low SNR). This way, the appropriate modulation scheme with variable constellation order can be selected based on the channel conditions.

A major challenge in wireless communication is to achieve high data-rate transmission while also being power efficient and using the bandwidth optimally. To enhance power efficiency, the average transmitted power of the constellation must be reduced at a certain bit error rate (BER). However, high data rates with restricted bandwidth require high transmit power to maintain the same performance. To tackle this challenge, adaptive modulation is a promising solution for spectrally efficient, high data-rate transmission with optimal power economy in contemporary and future wireless communication systems[104].

Constant SNR is maintained in adaptive modulation by altering different parameters such as transmitted power, data rates, and modulation orders [160, 161]. In [162], square QAM modulation was introduced to increase the delay spread immunity four times compared to the conventional QPSK systems at $BER = 10^{-3}$. In [163], variable constellation was studied on baud rates of 32 kB and at a carrier frequency of 1 GHz where the block size was found to be not effective on the efficiency of varrying constellation. While the range of $BER = [10^{-2}, 10^{-5}]$ had approximately 5 dB improvement in the channel SNR in the range of $[25, 40]$ dB compared to a fixed 16-level. Similarly, in [164], the authors investigated the BER performance as well as the delay spread immunity of a proposed adaptive-modulation system transmitting data under multipath fading conditions. More performance analysis on varying modulation systems has been studied in [165, 166, 167].

This adaptive modulation has been commonly used in high-speed modems [168, 169], satellite connections [170], and applications requiring high QoS [171, 172]. Similarly adaptive modulation has been adopted in the wireless standards of the cellular networks including the third generation (3G) [173, 174, 175], fourth generation (4G) [176], and fifth generation (5G) [177]. Furthermore, Wifi networks including IEEE 802.11n, IEEE 802.11ac, and IEEE 802.11a use adaptive modulation to optimise the data transmission based on the changing channel conditions [178, 179].

Both the electronic equipment that makes up the transmitters (Tx) and receivers (Rx) in communication systems are sensitive to both internal and external noise sources. These noise sources come from the electronic devices themselves as well as the communication channels. According to [104], noise in a communication channel might distort the data that is being transmitted, which makes it impossible for Rx to accurately comprehend the data that is being sent.

In contemporary communication systems, noise can also be caused by the mobility of the end systems as well as changes in the surrounding environment. In order to maintain error-free communication in dynamic settings, both Tx and Rx must be able to successfully adjust to varying degrees of background noise. However, existing frameworks have difficulties throughout the adaptation process, which results in poor communication. In the next section, the methods to increase the flexibility of the Tx and Rx in a manner that is dependent and independent from the channel model have been discussed.

2.6 Existing Solutions to Wireless Communication System Challenges

The adaptive transmission was initially suggested in the late 1960s [161]. It implies maintaining relatively high and consistent channel capacity and bit rate by adjusting transmitted power level [161], constellation size [162], coding rate/scheme [180], or any combination of these factors [155]. These systems maximise average spectral efficiency by transmitting at high rates during good channel circumstances and lowering bandwidth as the channel deteriorates. Due to decreased hardware limitations and channel estimate approaches in modern systems, adaptive modulation methods became more popular due to the increased need for spectrally efficient communication.

In time-varying fading channels, high-performance demodulation involves either an implicit or explicit estimate of the multiplicative distribution (MD) process of the noise and transmitted signal [181]. Based on the channel models given in Section 2.5, different modulation and coding methods have been proposed [182] inspired by the work to address Rician [183], and Rayleigh fading channels [184, 185]. Channel estimate works well for dynamic channels.

Channel estimate works well for dynamic channels. A well-studied channel estimation method is pilot symbol assisted modulation (PSAM) [127, 183, 184] or tone calibration techniques (TCT) [186, 187]. By comparing the received pilot samples to the known pilot symbols, one may estimate the channel's gain and phase changes. These channel estimations are filtered using an estimated fading process filter to decrease the noise that can be summarised as follows:

- The first method interpolates channel estimations for intermediate data carriers without considering the underlying channel model. The ideal 2D Wiener interpolation is challenging to implement, hence simple methods such as spline interpolation and linear quadratic have been proposed in [188, 189, 190]. The fundamental disadvantage of these low-complexity interpolation techniques is the performance error in bit error rate (BER) floor at a high signal-to-noise ratio (SNR).

- In the second technique, the channel model is considered to be sparse, meaning it can be modelled with a minimal number of channel taps. This assumption holds true in the majority of actual situations. Pilot carrier observations may be used to calculate channel response at intermediate data carriers by estimating channel taps. The sparse channel model technique lacks a performance error floor, unlike the interpolation scheme. The sparse pilot-assisted channel estimation (PACE) performance is determined by the choice of the pilot carrier position.

The work in [191] was the first to attempt to optimise the placements of pilot carriers. However, the focus was on the scenario when pilot carriers split the total number of carriers, resulting in equispaced pilots. In [192], the authors optimise the distance between adjacent pilot carriers, and then they suggest a cubic parametrization of pilot carriers and convex optimisation in [193]. In [194], the authors offer a pilot design based on the norm of the MSE of the channel/symbol estimation to optimise a preamble and pilot placements. The channel estimation MSE is not minimised by any of these methods. Only the authors in [192, 193, 194] consider guard bands dispersed over the frequency band.

Despite the effectiveness of adaptive transmission techniques and the accuracy of the recent channel characterisation methods via studies, there are various practical limits that govern whether adaptive modulation may be utilised, including (a) the rate of channel change compared to the robustness of channel estimate through feedback channel to the Tx; (b) the availability of perfect feedback channel to assure the variability of the Tx to adapt; and (c) the sensitivity of the Rx to change In other words, if the channel changes quicker than can be anticipated and transmitted back to the transmitter, adaptive approaches will fail. Learning-based methods like filtering and Neural Networks (NNs) have been researched in communication systems to increase Tx and Rx adaptability to wireless communication channels.

2.6.1 AI-based methods

Different statistical and learning-based methods have been used to estimate channel behaviour to improve the quality of the communication process. For dynamic physical systems, transceivers are able to sense the channel during the communication process. The data received by the transceivers have been processed by different methods including but not limited to: (a) Kalman filtering, (b) Artificial Neural Networks, and (c) Autoencoders. In this subsection, the state-of-the-art methods used for improving channel quality with the help of these methods are presented.

Kalman filtering, also known as linear quadratic estimation (LQE), employs a sequence of measurements collected across time, including statistical noise and other imperfections, to estimate a joint probability distribution over the variables for each period, resulting in more accurate estimates of unknown variables than those based on a single measurement alone [195, 196]. This digital filter is sometimes named after Stratonovich–Kalman–Bucy, which is a specific example of Ruslan Stratonovich's nonlinear filter, developed previously [197].

According to [198], Kalman filtering is best suited for multi-input multi-output antenna (MIMO) systems working on multiple frequency ranges using Orthogonal frequency division multiplexing (OFDM). This is because the MIMO system with OFDM allows each receiving antenna component in the antenna array to sense the channel on the different frequency ranges. With the help of the collected data, the Kalman filter will be efficiently used for channel behaviour estimation. The use of the Kelman filter can be narrowed down to the following two categories as follows :

- Training Based Approach where Kalman filtering for MIMO-OFDM system and also using jakes training sequences method can also improve the channel estimation with low complexity [199].
- Pilot Assisted Approach where Tx and Rx symbols are provided in the sent message and message header to highlight channel effects on received messages.

The authors in [200] introduced a new pilot expansion (PE) training approach for MIMO channel estimation. That uses frequency- and time-selective fading. PE estimates channel impulse response length. The method lowers channel fluctuation and Doppler rate. While the work in [201] considers pilot-aided/data-aided Kalman channel tracking for OFDM systems in fast time-varying channels. The pilot-aided technique tracks channel changes.

PSAM uses a Kalman filter for pilot-symbol-aided parametric channel estimation [202]. Moreover, Kalman filter is used in tracking the signal subspace of the channel samples' correlation matrix-enhanced OFDM system work in [203]. It also extends multi-antenna effectively where the experimental results reveal that the suggested technique can monitor Doppler frequency and block fading channel time changes. Random-set theory evaluated a MIMO-OFDM channel with an unknown and time-varying number of paths resulting in Rayleigh fading [127, 204]. In a study published in [205], the authors developed a new soft-output MMSE-MIMO detector based on MMSE-CE. This detector efficiently allocates power between pilot and data symbols, thereby increasing the minimum SINR (signal-to-interference-and-noise ratio) when there are equal or unequal numbers of transmit and receive antennas.

In [206], the minimum mean square error (MMSE) approximation to investigate the optimality of the pilot symbol locations was introduced. The authors found that equi-spaced and equi-powered allocation is optimal for a single antenna system. However, the use of pilot symbols consumes the bandwidth used in communication. Despite the high performance in channel estimation through Kalman filter-based approach to cope with the channel change, bandwidth utilisation can be improved through the use of neural network-based methods.

2.6.2 Neural Networks in Communication System

An artificial neural network (or neural network) has an input layer of neurones (or nodes, units), one or two (or possibly three) hidden layers, and an output layer [207]. Neural networks (NNs) learn by evaluating samples with known "input" and "result" and creating probability-weighted correlations between them. A neural network is trained from an example by calculating the difference between its processed output (typically a prediction) and a target output. The network modifies its weighted associations using this error number and a learning strategy. Adjustments will make the neural network's output more like the trained outputs. After enough changes, training may be ended in the supervised learning by seeing examples, without task-specific rules.

In [208], thresholding neural network (TNN) for noise reduction was introduced in many applications. Smooth soft- and hard-thresholding activation functions are introduced as well as various gradient-based methods. Discussing MSE-optimal soft-thresholding approaches. In TNN, soft-thresholding has one MSE-optimal solution, the best approaches are examined, and gradient-based learning algorithms that are designed to find the best solutions in many scenarios and applications of noise reduction. It may also reduce time-scale or time-frequency noise in real-time.

Supervised and unsupervised batch and stochastic learning techniques are used. TNNs with stochastic learning algorithms may be employed as innovative nonlinear adaptive filters. Ideal circumstances show that the stochastic learning method is statistically convergent. TNN beats other noise reduction algorithms in identifying MSE-optimal thresholding solutions, according to numerical findings. TNN-based nonlinear adaptive filtering surpasses linear adaptive filtering in optimum solution and learning performance. The usage of NNs on mobile devices in video processing applications [209] has pushed researchers to employ a time-delay NN to represent the non-linear behaviours of communication networks. It includes static and pulsed DC characterisation, scattering parameter measurements, real-time load/source-pull at fundamental and harmonic frequencies, and gate and drain time-domain RF waveforms.

First utilised in cognitive radio networks (CRN) communication systems in [210], NNs are a benchmark. In [210], two cognitive terminal learning algorithms predicted the data rate a radio configuration might accomplish if chosen for the operation. In [211], current and new trends in developing highly efficient, reliable, secure, and scalable machine learning architectures for such devices are presented. The authors present a path to tackling the community's main concerns and building scalable, high-performance, and energy-efficient edge machine learning systems.

Autoencoder (AE) is an artificial NN used in unsupervised learning for data encoding [212, 213]. The autoencoder learns a representation of incoming data to accurately recreate it. The autoencoder has two primary components: an encoder and a decoder with neurons sharing a coding layer. Encoders extract data features and the decoder presents well-reconstructed input data to the network output layer. The activity levels of the neurons are the newly-learned representation of data. The decoding layer has fewer neurons than the network inputs, reducing data dimension.

Classical weight initialisation approaches make deep autoencoder training challenging. The learning process cannot determine the true influence of weights in the first layer of the network on output. This involves disappearing or bursting gradient. Exploding gradients may trap the network in a local minimum, resulting in a poor solution, whereas vanishing gradients slow network learning. The authors in [212] proposed a pre-training approach that may help. It enables initialising network weights with values that extract the required characteristics from data. Standard backwards propagation techniques perform better during fine-tuning when starting weights are chosen. Unsupervised, layer-by-layer training of a neural network is performed iteratively during pre-training. The restricted boltzmann machine is used to calculate weights between two network layers [69].

2.6.3 Learning-based Methods

Radio propagation channel characteristics in different conditions are needed for wireless communication network development and deployment. Since radio waves fade, radio propagation in physical settings affects wireless communication systems. Large-scale and small-scale fading affect any communication system's wirelessly delivered messages. Since the receiving antenna(s) receives signals mostly through reflections, diffractions, and scattering processes, air conditions and nearby physical objects induce signal losses and multipath propagation. Multipath effects cause signal power fluctuation and signal power uncertainty. This research develops large-scale path loss models for radio coverage estimation, frequency allocation, base station optimisation, and antenna selection [214, 215, 216].

Due to the ongoing growth of communication technologies and the exponentially rising need for greater mobile data traffic, research has focused on the frequency range above 6 GHz to overcome the congestion of the preceding bands. This frequency range satisfies the requirements of the fifth-generation (5G) wireless system and other high-speed multimedia services [217, 218, 219, 220]. For these systems, large-scale fading models help optimise base station installations, estimate radio coverage, and quantify wireless communication radio performance [216].

The work in [221] assesses multiple linear regression, polynomial regression, support vector regression, decision trees, random forests, K-nearest neighbours, artificial neural networks (ANN), and artificial recurrent neural networks (RNN). Long short-term memory underpins RNNs (LSTM). The top machine-learning-based route loss prediction models are selected from measurement data. This research found that the ANN, RNN-LSTM, and MLR techniques perform best and worst in root-mean-square error. The research demonstrates that these learning algorithms can accurately and stabley forecast route loss in the mmWave frequency range.

Deep learning (DL) approaches have been used to eliminate noise from the broadcast signal at the Rx in a noisy communication channel by modelling a neural network (NN) at the communication end-systems. End-to-end learning, utilising stochastic gradient descent (SGD), was introduced in [222]. Recently, the authors in [223] examined cooperative training across several channels to create a single encoder and decoder that performs well on a class of channels. Joint training emulates non-coherent transmission methods. This work proposes meta-learning to overcome collaborative training's drawbacks: Meta-learning discovers a shared initialisation vector for rapid channel training instead of training a single model. Numerical findings show considerable training speed-ups and effective encoders and decoders after one SGD repetition.

2.7 Communication Systems in Swarm Guidance

The choice of architecture for a wireless network involves fundamental aspects of network design. The primary consideration is whether to use station-oriented networks like mobile networks LTE and 5G or a peer-to-peer network like wifi [224] and bluetooth [225]. Station-oriented networks are Infrastructure-based networks where communication flows from network nodes to a single central hub that has single points of failure and is not re-configurable [226]. On the other hand, in a peer-to-peer architecture, communication flows directly among the nodes in the network, and the end-to-end process consists of one or more individual communication links. Peer-to-peer networks are infrastructure-less adhoc networks that consist of a wireless sensor network (WSN), wireless mesh network (WMN), and mobile adhoc network (MANET), which can be further classified into Vehicle Area Networks (VANET) and UAV Control Networks (UAVCN).

Many application requirements determine whether a peer-to-peer or base-station-oriented architecture is used. Peer-to-peer architectures provide dynamic typologies. DARPA's early 1970s packet radio networks became mobile ad-hoc networks. They remain a popular issue in communication network research. They use many hops to reach their communication partner. Wireless access technologies like WLAN 802.11 [224], Bluetooth [225], and Zigbee [227] are widely available, making ad hoc networks practical. Both standards may construct and manage networks without central institutions, fitting within the infrastructure-free ad hoc network model.

Despite the high data rate of the systems communicating at frequencies (2.4GHz – 5.4GHz) range, they are not impervious to weather conditions like rain, fog, smog, and dust particles, due to the small wavelength of these signals [228, 229], unlike the systems working in the frequency range the (902-928 MHz) frequency band, like TVs and radios. In 1985, the U.S. Federal Communications Commission (FCC) allocated the Industrial Scientific Medical (ISM) 2.4-GHz band for high-speed wireless local area networks (WlANs) to stimulate practical research and development [230]. This encourages the emerging peer-to-peer communication systems that operate in the 2.4 GHz range, like ZigBee and Wifi.

XBee-PRO 802.15.4 is a communication module developed by Maxstream Co, and characterised by its low-cost and low-power consumption [231], but weather-related occurrences may interfere with some broadcasts due to its short wavelengths [232]. ZigBee uses the MAC layers and PHY layers defined by IEEE® 802.15.4, which is the shortest-distance wireless communication standard for 2.4GHz. IEEE® 802.15.4 [233] provides a robust foundation for ZigBee, ensuring a reliable solution to noisy environments. ZigBee-based networks also allow customised topology and protocols [227]. Different navigation-related environmental concerns have been studied in [234] and presented via simulation and experimental findings. The main finding is that ZigBee RF [227], an extendable protocol, can be used to achieve coordination among the swarms for autonomous robotic swarm navigation.

Thanks to the various Mobile Ad-hoc Networks (MANETs), the agents can communicate and thus cooperate for task completion in a fully-distributed fashion. Nevertheless, there are some limitations of sharing information through peer-to-peer networks that depend on the features of the nodes [235]. For example, Ad-hoc network with slow mobile nodes like MANET suffers the least from topology change while the fast dynamics of UAVs in Flying Ad-hoc Networks (FANETs) as an example of dynamic agents impose fast topology change and also high sensitivity to delay. Vehicular Ad-hoc Network (VANETs) lies between both types of networks with sometimes high network demands as MANETs in some situations like avoiding collisions as shown in Table 2.1.

Moreover, peer-to-peer communication technologies restrict the number of transmissions per unit of time and connectivity of each agent due to bandwidth and topology restrictions. The most common peer-to-peer communication technologies used in ad-hoc networks are IEEE 802.15.4 (ZigBee), IEEE 802.11 and its variations,

Table 2.2: Types of Ad-hoc Networks

Network	Node Speed	Access	Topol-ogy change	Sensitivity to delay	Mobility model	Power efficiency
MANETs	Low	Low	Slow	Application dependent	Random	Energy efficient
VANETs	High	High	Fast	Some are Very high[235]	Regular	Not needed
FANETs	Very high	Very high	Fast	Some are Very high[235]	Regular	Energy efficient for mini UAVs

and low-power Bluetooth.

However, Bluetooth is excluded due to its limited number of nodes per piconet which is seven. It is also limited to star typologies which adds more constraints on the agent communication network. While Zigbee and Wifi represent the most prevalent peer-to-peer communication technologies in the market nowadays. Table 2.2 shows a comparison of the three aforementioned Ad-hoc networks showing the FANETs are the most demanding type as a cost of high accessibility to different environments and dynamics. Thus, the technical specifications are given in the table showing their applicability to FANETs as the most demanding type of MANETs. For example using shepherding to herd a large number of UAVs by a fewer number of UAVs, where all the agents share their view of the environment over a FANET.

2.8 Dynamic Communication Systems in Shepherding

Humans can do coordinated activities owing to their actions, perception, and interpersonal understanding as they may alter their minds while doing a job. Unfortunately, modern robots lack this knowledge and the flexibility to adapt to new strategies despite their ability to do complex tasks in controlled situations. In addition, certain tasks, such as working in a mine [236], a nuclear plant/radioactively dangerous zone [237], or extended monitoring missions might be lethal to humans [238]. With human swarm control, a human may monitor the overall development of a project using a computer screen or other data recording methods. A swarm of human-supervised robots doing a search-and-rescue mission is an example of a collaborative task.

The aim is to reduce the amount of human involvement required to complete tasks. The fundamental concept of swarm robotics is to split and efficiently distribute complicated tasks among the members. It is mostly motivated by the observation of insects, such as ants, termites, wasps, and bees. Insects have been seen coordinating their behaviours to perform tasks that are beyond the capacity of a single organism. A robotic swarm is capable of both terrestrial and aerial applications [239, 240].

Swarm robots have several underlying challenges that must be solved in order to produce a functional system. Some of the most fundamental difficulties that must be resolved are communication, path planning, tracking, task allocation, sensor selection, system dependability, and scalability [91, 92, 241].

The work in [242] studies the swarm of drones as Networked Control System (NCS), where the overall system is controlled via a wireless communication network. This is built on a tight interface between networking and computing systems and aims to enable and support the fundamental control functionality of data gathering and exchange, decision-making, and command distribution.

Social insect pheromone has inspired swarm robots in recent decades [243]. By using a virtual pheromone in a physical swarm robot system to coordinate individuals and realise direct/indirect inter-robot interactions, stigmatic behaviour has arisen. Many studies simply consider one pheromone while tackling swarm issues, which isn't true in insects. Pheromones and their interactions lead to various social insect behaviours, sophisticated collective performances, and variable state transitions. However, this communication mechanism restricts the distance between the interacting agents since the CU will be required to follow the sheepdog throughout the task time, which is expensive from the energy consumption perspective. Moreover, it may restrict the flying latitude of the CU and decrease its ability to observe the swarm mobility in real-time.

There are three essential components to be considered in modelling the shepherding task as a digital communication system: the transmitter, transmission channel, and receiver. Each part plays a particular role in signal transmission, as follows [149]:

- The source of the information is the transmitter (Tx) that processes the input signal through modulation or/and coding to produce a transmitted signal suited to the characteristics of the transmission channel.
- The transmission channel (CH) connects the source to the destination electrically. It might be wires, coaxial cables, radio waves, or laser beams. Every channel has transmission loss or attenuation, thus signal strength diminishes with distance.
- The receiver (Rx) prepares the channel output signal for the transducer. Rx operations reverse Tx signal processing by demodulating and/or decoding and amplification to correct transmission failure.

SwarmCom was presented in [244] for mobile ad-hoc networks or mobile robots using e-puck [29], as detailed in Section 2.1. Although channel coding reduces bit error rate at the cost of throughput, with channel coding concepts, SwarmCom's detector adapts to the surroundings and neighbouring robots efficiently. Experiments on up to 30 e-pucks demonstrate that SwarmCom outperforms libIrcom, the existing infrared communication software, in up to 3 times further transmissions ranges, achieves less bit error rate ($\pm[50, 63]$ %), and up to 8 times higher throughput, and uses the less maximum number of communication channels per robot to limit the load per robot in high-density swarms.

In shepherding tasks, the sheepdog's short sight hinders its ability to choose the most effective herding locations. This is because applying forces to the agents demands being in close proximity to the swarm while seeing the positions of all the swarm members in real time necessitates a bird's eye view from a reasonably large distance from the swarm. In real-life scenarios, the shepherd person may have a better view of the sheep from a relatively large distance and guide the sheepdog to the locations that help in guiding the swarm. This is similar to the usage of a sensing system that sends control signals to actuators [20, 21] in autonomous systems. Thanks to communication protocols and technologies, the view of the environment perceived by an agent with visualisation capabilities can be analysed and processed to generate commands for the autonomous sheepdog via a wireless channel.

Finding minimum cost paths using path planning methods including A* [103], Informed Rapidly-exploring Random Tree (IRRT) [245], and IRRT* [246] before starting the swarm guidance task may not be sufficient due to the stochastic property of the problem as the goal location for the guiding agent is changing throughout the task time. The CU observes the locations of all the swarm members and uses these locations to find the actuator's headings and sends them to it in real-time.

The communication link between the CU and actuator (β) is time-varying since the shepherd ought to change its locations throughout the task time as it follows the received headings. The changes in the quality of the communication link affect the success rate of the shepherding task. The mobility of β throughout the task time makes the communication link between itself and the CU subject to different communication models depending on the nature of the communication signal and the environment. Although such data distortion may occur at any point, the standard convention is to lump them entirely on the channel, treating the transmitter and receiver as being ideal.

2.9 Chapter Summary

There are several uses of swarm intelligence in human life. Leader-follower strategy and shepherding technique are the most prevalent strategies for swarm direction. Shepherding is a broader strategy for guiding biological creatures and robots. The autonomous shepherding literature began in the 2000s, influenced by Reynolds' and

Table 2.3: Commonly used RF Technologies in peer-to-peer Communication

Peer to peer Network Technology	**Op. Freq (GHz)**	**Max. B.W.**	**Latency (approx.)**	**Max. Coverage**	**Multi-hop**	**Max # node/net**	**power consumption (Approx.)**	**Application in (FANET)**
IEEE 802.15.4 (ZigBee)	2.4	250 kb/s	10 ms	100 m (LOS)	Yes	1000 router	2050 mW	Bidirectional sensor data
IEEE 802.11a+IEEE 802.11s (WiFi)	5	36 Mb/s	100s of ms	300 m	Yes	32	1.33/0.9 W	Bidirectional sensor data + Control commands

APF rules. Throughout the past several decades, numerous strategies have been presented to enhance current shepherding models and build new ones in order to address challenges in the shepherding tasks, such as the scalability of the problem and the sensing range constraints of the guiding and guided agents.

In the literature, the system architecture for shepherding is based on assumptions that limit its applicability in diverse situations. These assumptions include the diversity of global knowledge within the swarm and the herder's infinite localisation capabilities. These assumptions may hold true in an indoor setting with a perfect camera-to-sheepdog communication system. Nevertheless, communication systems face several obstacles, such as data corruption caused by signal fading and communication channel noise.

Therefore, a cross domain solution is required for shepherding as a dynamic system to ensure the effectiveness of shepherding approaches in a real-world swarm control scenario. The success of the shepherding approach is based not only on the performance of the herding agent in steering the swarm but also on the capacity of the autonomous sheepdog to do this with little communication. This thesis develops and proposes a cross domain approach. The purpose of Chapter 3 is to examine the shepherding of sheep with a restricted sensing range. Chapters 4 and 5 seek to contribute to the current body of research on swarm guiding by assisting in the creation of communication models for shepherding.

Chapter 3

Connectivity-Aware Approaches for Shepherding under Limited Sensing Range Constraints

The work reported in this chapter has been partially published in the following articles:

RE Mohamed, S Elsayed, R Hunjet, H Abbass (2021), *A Graph-based Approach for Shepherding Swarms with Limited Sensing Range.* 2021 IEEE Congress on Evolutionary Computation (CEC).

RE Mohamed, S Elsayed, R Hunjet, H Abbass (2022), *Connectivity-Aware Particle Swarm Optimisation for Swarm Shepherding.* IEEE Transactions on Emerging Topics in Computational Intelligence.

In this chapter, a new approach for herding a flock of agents constrained by a low sensing range is introduced. It models the flock as a graph and seeks to maintain its connectivity. The problem formulation is described, and then the proposed method is discussed, with its performance assessed.

3.1 Introduction

As described in Chapter 2, a swarm can be defined as a group of agents (robots) with limited processing capabilities that can develop advanced behaviours to solve complex tasks through their local interactions [2, 3]. The guidance of a swarm towards a goal in the environment is a non-trivial problem due to the distributed nature of swarm members. Shepherding is a biologically-inspired methodology for swarm guidance whereby single or multiple agents (sheepdog/s) act as pressure point/s to guide the sheep that are agitated from them through some behaviours (i.e., collecting, driving, etc.) to influence swarm members (sheep) towards a goal [11, 12, 13, 80]. One of the primary advantages of shepherding is the reliance on a relatively small number of sheepdogs to guide a significantly large number of sheep. In this chapter Sheepdog and shepherd are used interchangeably to refer to the autonomous sheepdog since it acts as a sheepdog with communication capabilities like the shepherd.

However, as discussed in Chapter 2, in the recent shepherding literature methods in [59], and [17], it was assumed that each swarm member has global knowledge of the locations of all the other members. This is an impractical assumption due to the limited sensing range of agents within the swarm in a real-world setting which leads to a deterioration in performance, even with a small number of sheep. This assumption of a limited view range of the shepherd contradicts the common assumption in the literature of the shepherding methods, as discussed in Chapter 2. This is because the common assumption is that the sheepdog is in close vicinity to its influenced sheep and the shepherd is further away from the flock to ensure complete observability of the former's real-time location.

In this chapter, to guarantee sheep cohesiveness despite the sheep's restricted sensing range, the CU collects data about their locations to represent them as a dynamic network of connected components. Then, two methods for the CU to decide on the near-optimal headings for the shepherd that guarantee flock cohesiveness throughout the herding task are provided. They aim to maintain the connectivity of the sheep's network while the sheepdog influences them towards their home location, as summarised below:

- A swarm optimal herding point (SOHP) approach that employs a unit disc graph (UDG) to create a flock (based on the sheep's sensing range) is developed. In SOHP, a geometric method is used to select a subset of sheep to be impacted to indirectly influence the flock. Then, it combines these geometric principles with particle swarm Optimisation (PSO) assisted by density-based spatial clustering of applications with noise (DBSCAN) to find a near-optimal destination for the shepherd to guide this subset and all the sheep to their home location while preventing flock dispersion in a constrained-time context. The shepherd is also made "connectivity-aware," keeping an adequate distance from the flock to prevent scattering the sheep. This procedure is followed each time the shepherd reaches the estimated herding point.

- A swarm optimal modified centroid push (SOMCP) model is developed. It uses the distributed nature of the system to take into account the topological changes in the sheep's network throughout the shepherd's travel. It improves the shepherd's performance compared with that of the SOHP since the shepherd receives and follows headings from the CU that include near-optimal locations on its path to the near-optimal herding point. The waypoints on this path aid in the flock's cohesion throughout its journey. Furthermore, in the SOMCP, the CU may terminate the shepherd's path before it reaches the final herding point if the sheep's cohesiveness is negatively impacted over a series of time steps.

The proposed methods are tested on multiple scenarios and found capable of achieving a high SR and reducing a mission's completion time (T) compared with those of existing approaches.

3.2 Problem Formulation

In this section, the general shepherding problem and the effect of limited sensing range are described. To be consistent with the literature [59], π refers to a sheep, Π to the set of sheep, β to a shepherd, $R_{\pi\pi}$ to the sheep's sensing range in relation to their peers, R_{π} to the sheep's safe range that prevents it from colliding with its peers, $R_{\pi\beta}$ to the sheep sensing range in relation to the shepherd (β), N to the set of sheep to be herded, and N^t to a subset of sheep yet to reach the home area at time t, where $|N|$ is the cardinality of N. For simplicity, a set's name refers to its size, otherwise, the word 'set'is added before the notation. Moreover, P_i^t is the position of agent i at time t in two dimensions $(x,\, y)$, with $\overrightarrow{pq}$ and $d(p, q)$ representing the unit vector and the Euclidean distance between the two points p and q in a two-dimensional space, respectively.

Definition 3.1. *The global centre of mass (GCM) Γ_π^t is the location of the centre of N^t sheep outside their home area at a particular point in time (t) such that $\Gamma_\pi^t = (\sum P_\pi^{t_x}/|N^t|, \sum P_\pi^{t_y}/|N^t|)$*

Definition 3.2. *The local centre of mass (LCM) γ_π^t is the central location of a set of sheep N_{subset}, where $|N_{subset}| < N$, at time t is calculated for the N_{subset} located in the x-dimension at $p_x = p_{x1}, p_{x2}, .., p_{x|N_{subset}|}$ and similarly for the y-dimension. Therefore,*
$\gamma_\pi^t = (\sum p_x/N_{subset}, \sum p_y/N_{subset})$ [59].

Definition 3.3. *The connected component of the sheep is a set of N_{subset} connected by edges of individual lengths, each less than or equal to the sheep's sensing range ($R_{\pi\pi}$) [247].*

Definition 3.4. *The bridge edge is an edge that when cut, divides a connected component in two [247].*

Definition 3.5. *The intersection graph of circles packed in another circle has a vertex for each circle and an edge of tangential ones [248, 249].*

The sheep form an intersection graph in which each circle's radius represents the sheep's safe range (R_π) which avoids them colliding with their neighbours. Moreover, they also form a connectivity graph (G) at each time step (t), where each sheep is modelled with a circle of radius ($R_{\pi\pi}$) representing its sensing range and connected to every other sheep within it.

Lemma 3.2.1. *To ensure safe UDG connectivity, the ratio between $R_{\pi\pi}$ and R_π must be greater than one.*

Proof. Each sheep agent is modelled with two omni-directional sensors, one for connectivity of its neighbourhood with radius $R_{\pi\pi}$ and the other for collision avoidance with radius R_{π}. If $R_{\pi\pi}$ is less than R_{π}, the sheep will collide before they are able to sense each other. Therefore. to maintain safe separation and connectivity, $R_{\pi\pi}$ must be greater than R_{π}. $\square$

The value of R_{π} and its interaction with other sensor ranges influence dispersion among the sheep. A cohesive flock is formed via the application of different force vectors that represent attraction and repulsion. Each sheep agent is attracted to the LCM of its neighbours within its $R_{\pi\pi}$, which is a location calculated at each time step (t). Also, each sheep is repelled from its neighbours within its safe radius (R_{π}) to avoid collisions and from the shepherd within its agitation range ($R_{\pi\beta}$) ($R_{\pi} < R_{\pi\pi} < R_{\pi\beta}$) to avoid risk.

As discussed in Chapter 2, the sheep agent's motion model proposed in [59] relies on a weighted sum of different forces, that is, the attraction force towards the flock's LCM and the repulsive ones from both neighbouring sheep within a radius of R_{π}, and the repulsive force from the shepherd. The set of sheep within R_{π} of a sheep π is its repulsive set (n_{rep}), while the set of sheep within $R_{\pi\pi}$ of it is its neighbouring set (n). Assuming that there is additive white gaussian noise (AWGN) in the final direction and considering the inertia of the sheep, the resulting force vector applied on each sheep is given in Equation (2.2), as described in [59] where all the agents are assumed to be particles with a unit mass. The total force on each particle ($\pi \in \Pi$) is the superposition of all the forces acting on it, formulated as

Definition 3.6. *A disconnected flock is one divided into more than one connected components, where the minimum distance separating the closest nodes in two different connected components is greater than $R\pi\pi$.*

Each sheep uses its $R_{\pi\pi}$ to find the location of its closest neighbours. It calculates its LCM (γ^t) to which it will be attracted, in order to maintain cohesion when agitated by the sheepdog, if the sheepdog is within $R_{\pi\beta}$, as in Equation (2.2). If

$R_{\pi\pi}$ is not relatively larger than R_{π}, each sheep can view only a limited number of others (N_{subset}), that are noticeably less than N at each time step (t). Accordingly, each sheep has a different γ^t that can be very far from that of another one. In other words, their GCM Γ^t can be used as a reference point for cohesion, if all the sheep are attracted to different $\gamma_i^t \forall i \in \Pi$ that are relatively close to Γ^t.

In that case, they will be attracted to relatively close points which will improve their cohesion over time. Moreover, having all of the sheep relatively close to their GCM makes the sheepdog herd them to home with limited dispersion by influencing their GCM. Nevertheless, if the values of $\gamma_i^t \forall i \in \Pi$ have high deviations, all the sheep will be attracted to relatively distant points which will weaken their cohesion. As, over time, they will be far from their GCM, if the sheepdog influences their GCM during the agitation time, their cohesion will be reduced even further.

Lemma 3.2.2. *If the sheepdog has two disconnected components in two different hemispheres of its $R_{\pi\beta}$, it will disconnect the flock more if it applies the centroid push-based driving rules.*

Proof. According to the definition of a UDG, if the flock is initially connected, a random motion with low probability will unlikely lead to disconnection. However, the intrusion of the sheepdog in the agitation of the sheep in one connected component from a direction that moves it further away from the other will lead to greater sheep dispersion.

In Figure 3.1, a scenario in which there are two connected components, one at the top and the other at the bottom of the flock, is illustrated. These components are in the two green circles, where the green steps represent the graph edges among the sheep, while the two connected components are contained in the flock, as indicated by the blue circle centred around the flock's GCM with a radius equal to the distance to the sheep furthest from their GCM. In this scenario, when the sheepdog has two sheep from the connected component at the top and another from that at the

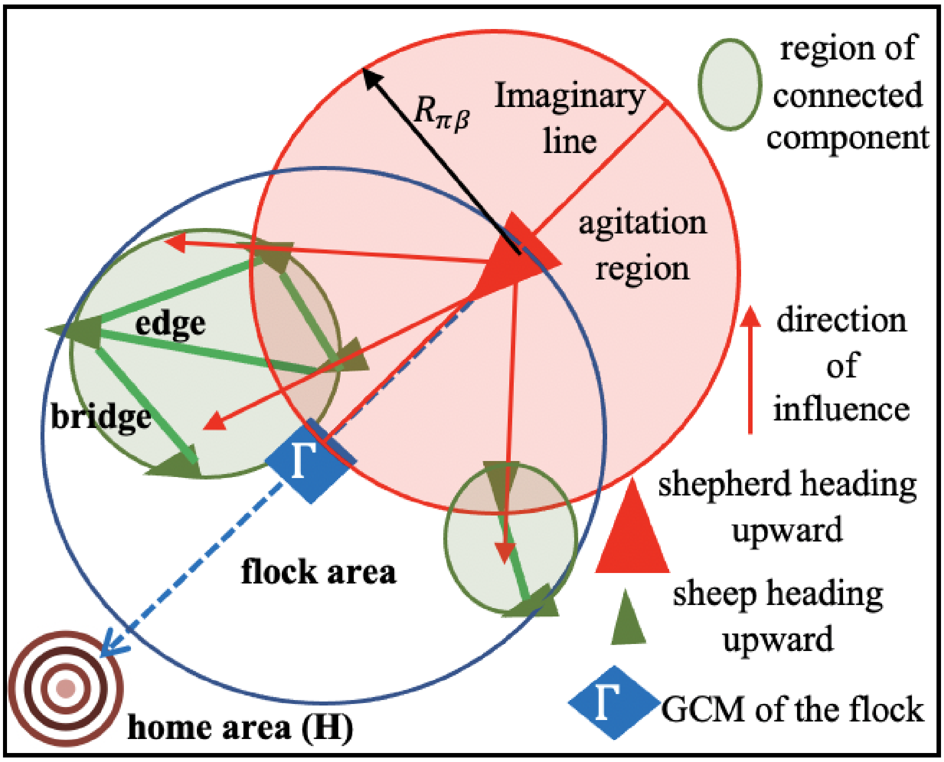

Figure 3.1: Further dispersion in the flock due to influencing the GCM of a flock with two connected components

bottom of the flock in the red circle that represents the sheep's agitation range, it influences the two connected components in opposite directions. This is because the sheepdog applies a centroid push-based method on the sheep that belong to different connected components and guides their GCM towards H by positioning itself on the extension of the blue dashed step connecting their GCM with H, although the two connected components are in two different halves of the agitation circle. This leads to driving these components in two opposite directions indicated by the red arrows which results in greater flock dispersion. □

Remark 2 Using a centroid push by targeting the sheep's GCM causes further dispersion of two or more connected components by influencing them in different directions as the CU considers only the distance between the sheepdog and the GCM.

The effect of competing forces among the sheep combined with agitations from the sheepdog on their movements, as in Equation (2.2), as well as the sheep's lack of awareness regarding their goal location impacts the complexity of the shepherding task. The sheepdog selects suitable goal points to which to guide the sheep agents and avoids flock dispersion all the time. However, traditional path-planning methods might not generate a proper path that satisfies these conditions.

3.3 Model Description and Assumptions

Firstly, in a wireless communication system, transmitted signals may suffer from large-scale fading due to the attenuation of their strengths through free space, known as free-space fading, which depends on the distance between the transmitter and receiver antennas. Also, large-scale fading may occur due to the shadowing effect of environmental obstructions on a signal, including buildings and trees [113]. As initial investigation, the scope of the work presented in this chapter is the shepherding task for sheep with limited viewing ranges. Nonetheless, it is assumed that the communication link between the sheepdog and CU is noiseless, with no delay during the task. In addition, the capacity of the decision-maker to acquire huge amounts of data on the sheep's location and use this information to find near-optimal decisions for the sheepdog are ensured by modelling the controlling elements in the shepherding task as a networked control system (NCS). Enforcing this distributed characteristic also enhances the system's resistance to system failures.

The models used for shepherding in the literature are defined from a distributed system perspective, as depicted in Figure 3.2. They describe the transmission of a/the path from the CU to sheepdog through a communication link instead of centralising the sheepdog's data perception. The model on the top right shows how the sheepdog is assumed to have complete observations and make decisions accordingly and that on the bottom right how the sheepdog receives headings from the CU.

The above assumptions are mapped to the SOHP and SOMCP, as depicted in Figure 3.3 which shows a path's transmission from the CU to shepherd through a

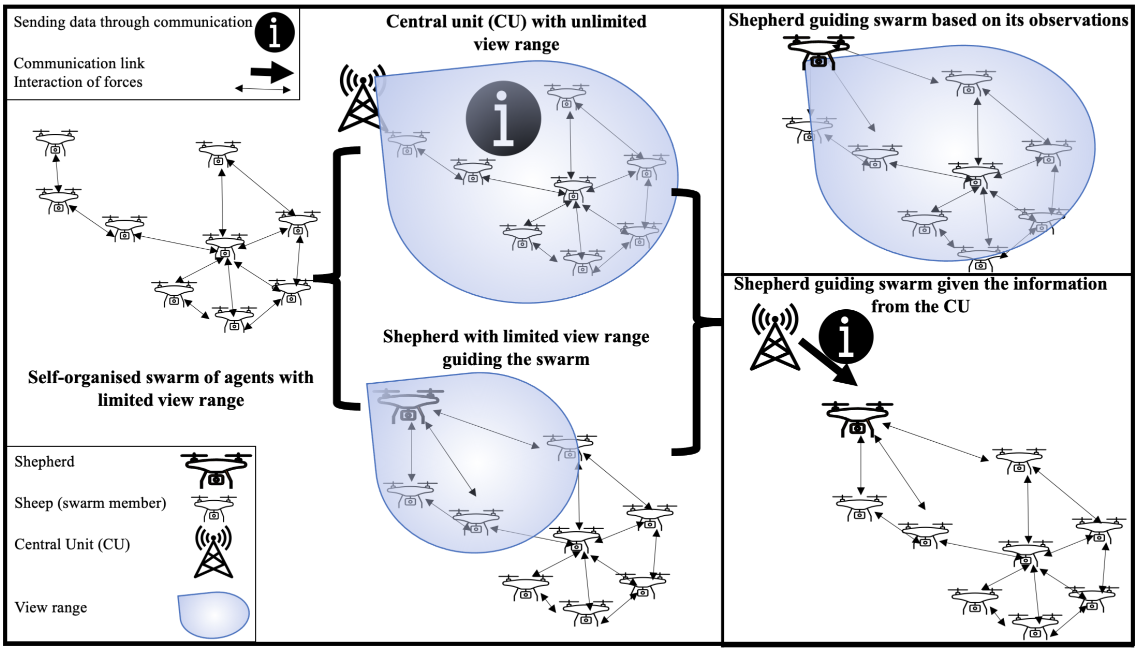

Figure 3.2: Distributed system model between the shepherd and CU in shepherding (a) swarm mobility with control of CU only through a communication or with sheepdog only with limited view range (left), (b) the shepherding system without the CU (top right), and (c) the shepherding system where communications between the CU and the shepherd is modelled (bottom right)

communication link. This centralised model is similar to how a human shepherd (the CU in our model) gives instructions to a sheepdog in a real-life shepherding scenario. The model on the left describes how the shepherd receives headings from the CU as interpreted from how state-of-the-art shepherding models work compared with how it takes place in the SOHP and SOMCP ones on the right. These headings help the shepherd influence the sheep in a direction that ensures a low task time. The SOHP and SOMCP approaches are designed to improve the resilience of shepherding methods to communication issues, such as noise and signal fading [104], by making the transmitting frequency headings lower than those of existing models. This provides a low dependence of the shepherd on the communication link with the CU and ensures low bandwidth usage.

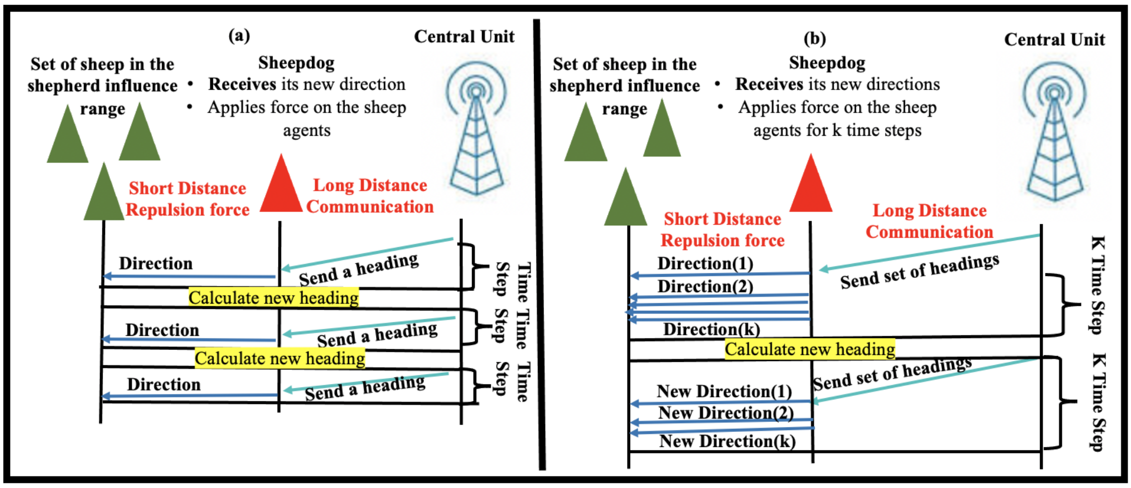

Figure 3.3: Models of communication between shepherd and CU based on (a) methods in literature (top) and (b) SOHP and SOMCP

3.4 Swarm Optimal Herding Point

In this section, the proposed SOHP is described. It can be applied regardless of assumptions about the controlling system. If the system is centralised, perceiving the sheep's location, deciding the near-optimal herding point and heading towards it are the responsibilities of only the shepherd. However, in this chapter, it is assumed that an NCS exists between the CU and shepherd, as stated in Section 3.3 where the communication channel between them is ideal.

3.4.1 Selection of Flock Subset

In the SOHP, the CU's role is to select the set of sheep the shepherd will influence based on observations collected about the environment. The CU models the sheep with limited $R_{\pi\pi}$ as a dynamic UDG that changes every time step (t) due to the sheep's movements. The graph ($G = (N, \mathcal{E})$) considered is undirected and connected, with a node set ($N = 1, 2, ...|N|$) and edge set ($\mathcal{E}$, where its length $\leq R_{\pi\pi}$). The degree of vertex i is $\Delta(i)$ which represents the number of sheep within its sensing range. This set is considered a connected component (cc) when all the sheep in it are connected either directly or indirectly. Any edge that, if removed, causes a cc

to fragment is called a *bridge*.

The aim of this step is to identify the subset of sheep of size $|N_{subset}|$ to be influenced by the shepherd, as shown in Algorithm 1, knowing that $|N_{subset}| < |N^t|$, where $|N^t| = |N|$ at $t = 0$. If needed, the sheep's distances between their home (H) and GCM are arranged in descending order to add the furthest sheep to this subset. Then, the sheep connected to this furthest one are added to N_{subset} until its total size is $|N_{subset}|$. This selection depends on the connectivity metrics of G, as described in Algorithm 1; which indicate if the sheep are not collected (the flock needs to be more cohesive); otherwise, they can be driven to H. This UDG technique represents a sensing-induced graph considering the location of the sheep as well as their sensing range. Note that the number of bridges; node degrees (Δ), and connected components (cc) are needed (measured in steps 1 and 2) to help the sheepdog drive the sheep while maintaining the UDG's connectivity.

Steps 3 to 6 in Algorithm 1 explain the process of selecting the shepherd's proper behaviour which can be either (1) driving the agents to the home destination (H) without considering their connectivity, or (2) collecting the sheep by considering their connectivity while pushing them to H. The driving behaviour is selected if the G satisfies the given three conditions: (i) its Δ_{min} is more than a predefined value Δ_{thresh}; (ii) it has only one n^t_{cc}; and (iii) there are no bridges in the G ($n_{bridges} = 0$).

In that case, the subset of the flock to be influenced is the cc with the sheep that is the furthest from H. In summary, if the swarm is well connected (has one cc with no bridges and a high Δ_{min}), it is assumed to be cohesive and is driven towards H. In contrast, the collecting behaviour is the default in the absence of any of the above-stated conditions. In both cases, the distances from either H or the GCM are collected and stored for each sheep in two lists (D^t_1 and D^t_2, respectively). The index of the sheep, with a maximum D^t_1 (driving behaviour is happening), or a maximum in both D^t_1 and D^t_2 (collecting behaviour is occurring); is stored as 'node'. The selected subset is considered the cc with the 'node'sheep (steps 7 to 14), with a maximum subset $|N_{subset}|_{max}$ (steps 14-18). This enables the shepherd to influence a reasonable/fair number of sheep.

If the conditions for driving behaviour are not met, a subset for the collecting behaviour is selected, with the distance to the GCM used as a criterion. The reason for this is to increase connectivity in the overall sheep graph.

3.4.2 Finding Herding Point

A technique for improving the graph's connectivity while reducing the distance to H is calculated in the CU (see Algorithm 2). This is accomplished by driving the N^t_{subset} of sheep simultaneously towards their LCM ($\gamma_{N_{subset}}$) and H. To do this, a point on the straight line between $\gamma_{N_{subset}}$ and H is selected to be the temporary goal for the sheepdog's influence direction on N^t_{subset}. Then, finding the steering direction is modelled as a single-objective problem (3.1) that is equal to a weighted sum of two average Euclidean distances between each sheep in N^t_{subset} and two points (H and $\gamma_{N_{subset}}$), see Equations (3.2, 3.3), respectively.

$$\min_{x_{i\in N^t_{subset}},y_{i\in N^t_{subset}}} \quad F = w_1 f_1 + w_2 f_2 \tag{3.1}$$

where

$$f_1 = \sum_{i\in N^t_{subset}} \sqrt{(x_i - H_x)^2 + (y_i - H_y)^2} / |N^t_{subset}| \tag{3.2}$$

$$f_2 = \sum_{i\in N^t_{subset}} \sqrt{(x_i - LCM_x)^2 + (y_i - LCM_y)^2} / |N^t_{subset}| \tag{3.3}$$

The problem is modelled as a single-objective optimisation one to find the direction of each sheep in the N^t_{subset} that takes it closer to both $\gamma_{N_{subset}}$ and H. In other words, the objective function F aims to find a target point on the straight-line that connects $\gamma_{N_{subset}}$ and H. This point can be at H for all the sheep in N^t_{subset}; if, and only if, the sheepdog is exhibiting a 'drive'behaviour, where w_1 and w_2 are set to 1 and 0, respectively, as per Equation (3.4). Otherwise, if the sheepdog is undertaking a 'collect'behaviour, the target point can be closer to $\gamma_{N_{subset}}$ than H or vice versa depending on the values of the weights w_1 and w_2 that their total sum 1.

$$w_1 = \begin{cases} e^x/(e^x+1) & 'collect' \\ 1 & 'drive' \end{cases} \tag{3.4}$$

where $x = (\Delta_{avg} - (n_{bridges} + n_{cc})\Delta_{min})/\Delta_{avg}$

$$w_2 = 1 - w_1 \tag{3.5}$$

The problem of finding the direction of each sheep that minimises its Euclidean distances towards both $\gamma_{N_{subset}}$ and H is similar in nature to the well-known population-based optimisation method, PSO; therefore, the sheep in N^t_{subset} are modelled as particles of the initial solutions used in the search for the optimal direction for each swarm. Iterative updates are conducted on the velocity and position of each particle using Equations(3.6, 3.7), respectively. The search range is bounded by each sheep's step size so that feasible solutions are within k_{steps} from each sheep's initial location (Equations (3.8) and (3.9)), such that the inertia of particle w is the sheep's (inertia $W_{\pi v}$), as defined in the motion model in Equation (2.2). Note that Equation (3.10) determines the maximum number of generations used in PSO.

$$v_{i,t+1} = wv_{i,t} + c_1r_1[x_{i,t} - x_{i,t}] + c_2r_2[g_t - x_{i,t}] \tag{3.6}$$

$$x_{i,t+1} = x_{i,t} + v_{i,t+1} \tag{3.7}$$

$$ub = P^t_\pi + k_{steps}\delta_\pi, \tag{3.8}$$

$$lb = P^t_\pi - k_{steps}\delta_\pi \tag{3.9}$$

$$n_{iter} = k_{iter} + k_{iter} * (n_{cc} * (|N| - |N^t_{subset}|)/|N|) \tag{3.10}$$

where w is the inertia coefficient, $0 \leq c_1, c_2 \leq 2$ the acceleration coefficients, r_1 and

r_2 uniform random values $\in]0, 1]$ and k_{iter} the minimum number of generations to be conducted.

The location obtained is used to identify the direction (dir_π) each sheep (π) uses to move from its initial location (P_π^{init}) to the new one (P_π^{fin}), as in Equation (3.11)

$$dir_\pi = \overrightarrow{P_\pi^{init} P_\pi^{fin}} \tag{3.11}$$

These directions are stored in dir_{subset} (step 6 in Algorithm 2). As a result, the final steering direction to which the shepherd guides the N_{subset} is determined according to one of the following conditions (refer to step 7 to 14 in Algorithm 2):

- If all N_{target}^t have the same direction, which is $(0, 0)$, then $\gamma_{N_{subset}}$ is steered towards the GCM (in the case of collecting), or the GCM is steered towards H, if the shepherd is driving.
- If all N_{subset}^t have the same direction, which is not $(0, 0)$, the steering direction of $\gamma_{N_{subset}}$ is the direction on which all the N_{subset}^t agree.
- If the N_{subset}^t of sheep have different directions, these directions are clustered using the DBSCAN method [250]; so that the maximum separation between the points in each cluster is υ and the minimum number of points in a cluster ϵ. Then, the average direction of the highest density cluster is considered the steering one for the LCM of the N_{subset}^t.

The next herding point H^t is D_{herd} units away from $\gamma_{N_{subset}}$ and in the opposite direction to dir^* (step 15 in Algorithm 2). The sheepdog's final destination is then set to H^t for the following time steps until the path is terminated; (Algorithm 2, step 15). According to the system model in Fig 3.3, the CU sends the H^t obtained to the shepherd which then the shepherd goes in a straight line towards it. This temporary path terminates when the shepherd reaches H^t which leads to the generation of a new path at/by the CU through the sequences of Algorithms 1 and 2.

Selections of the sheep subset and near-optimal herding point by SOHP is inspired by the centroid push-based shepherding model [59], i.e., the centroid of the flock moves progressively closer to the goal while maintaining the flock cohesion. The SOHP is designed to improve the performance of a single shepherd in overcoming the sheep's dispersion due to their limited sensing range and comprises two main steps applied sequentially. The approach assumes the availability of real-time complete information about the sheep's location, for finding the near-optimal herding point for the sheepdog. The first step is selecting the shepherd's behaviour and a subset of sheep to influence, and the second one is estimating the near-optimal herding point.

In the following section, the other components that are added to the SOHP to form the swarm optimal modified centroid push (SOMCP) method are described. The first one finds the near-optimal path for the shepherd from its current location to the near-optimal herding point by using an estimation model for the sheep's mobility. The second uses the sheep's real-time locations to decide on the termination of the sheep-dog's path if it is leading to further dispersion in the sheep's sensing-induced graph.

3.5 Swarm Optimisation-based Modified Centroid Push-based Shepherding Model

In this section, the proposed method SOMCP, which aims to improve shepherding given the sheep's limited sensing range is described. The algorithms are designed for the purpose of replacing a farmer and the sheepdogs with robotic/AI counterparts.

The proposed method consists of the four main steps in Algorithm 3 applied sequentially. It starts with the collection of observations by the CU that is assumed to have a complete view of the environment in real-time. The data collected about the sheep's locations help the CU to generate a UDG of the sheep and select both the shepherd's behaviour and the subset that it should influence (Algorithm 1). This behaviour can be either driving the sheep to home regardless of their respective positions or collecting them to improve their cohesion. The next step is to find a

Algorithm 1: SubsetSelection($P_{i \in N^t}$)

Input : $P_{i \in N^t}$
Output : N^t_{subset},Behaviour

Create G^t
Find n_{cc}, Δ_{min}, and $n_{bridges}$ at time t
if $n_{cc} > 1 || n_{bridges} > 1 || \Delta_{min} < \Delta_{thresh}$ **then**
  Behaviour $\leftarrow$ 'Collect'
else
  Behaviour $\leftarrow$ 'Drive'
Measure D^t_1
if *Behaviour=='Collect'* **then**
  Measure D^t_2
  node $\leftarrow$ index of sheep at $max(D^t_1, D^t_2)$
else
  sort D^t_1 Descending
  node $\leftarrow$ id of sheep at max(D^t_1)
$cc^* \leftarrow$ the connected component of node
if $|cc^*| > |N_{subset}|_{max}$ **then**
  $N^t_{subset} \leftarrow$ nearest $|N_{subset}|_{max}$ to node
else
  $N^t_{subset} \leftarrow cc^*$

Algorithm 2: HerdingPointEstimation(N^t_{subset},Behaviour)

Input : N^t_{subset},Behaviour
Output : H^t

if *Behaviour='Drive'* **then**
  $w_1 = 1$ and $w_2 = 0$
else
  set w_1, w_2, see (3.4) and (3.5)
Generate initial solutions as the recent directions for each $\pi \in N^t_{subset}$
$dir_{subset} \leftarrow$ the best direction for each $\pi \in N^t_{subset}$, optimise (3.1) by PSO
if dir_{subset} *not converged* **then**
  Set υ_{DBSCAN},d_{DBSCAN}(DBSCAN parameters)
  DBSCAN(dir_{subset})
  $dir^* \leftarrow mean(cluster_{max})$ (average of cluster with maximum size)
else
  $dir^* \leftarrow any(dir_{subset})$
if $dir^* == (0, 0)$ **then**
  $dir^* \leftarrow \overrightarrow{\gamma_{N_{subset}} H}$
$H^t \leftarrow \gamma_{N_{subset}} - dir^* \times D_{herd}$

Algorithm 3: Swarm Optimisation-based Modified Centroid Push Shepherding Model (SOMCP)

$t = 0$
$terminate = False$
while $t < T$ **do**
    `/* the CU actions */`
    if $terminate == True$ **then**
        1. Find N^t_{subset} of sheep and the adopted behaviour by β ; `// Section 3.4.1`
        2. Find location for β to herd N^t_{subset} (H^t) ; `// Section 3.4.2`
        3. Find path to H^t ; `// Section 3.5.1`
    else
        `/* the Shepherd actions */`
        follow the new path
        if $min(d(P_{i \in N^t}, \beta)) < d_{stop}$ **then**
            - Stop
    `/* the CU actions */`
    4. CU monitors the effect of β on G^t ; `// Section 3.5.2)`
    5. $terminate \leftarrow$ PathTerminationCriteria$(P_{i \in N^t}, P^t_{\beta})$; `// see Section 3.5.2`
    increment(t)

near-optimal herding point to which the shepherd should navigate to, as described in Algorithm 2. Next, the CU finds a path that the shepherd should follow to reach this herding point (Algorithm 4). Finally, it decides to terminate this path by sending the shepherd a new one whenever the requirements for preserving connectivity are violated (Algorithm 5).

Each point on the path sent by the CU aims to help the sheepdog improve the sheep's connectivity and reduces their distance to home during its traversal to the path's final point. The path the sheepdog follows is penalised by the CU whenever it disturbs the connectivity among the sheep or pushes them far from home. If the number of penalties exceeds the limit of a specific path, this path is prematurely terminated, with a new one to a new goal point is sent from the CU to the shepherd. The shepherd follows this path while using its limited sensing range to maintain a certain distance from the sheep and stops if it gets closer than a threshold distance (d_{stop}) to any sheep. The components added to the SOHP to make the SOMCP are

Algorithm 4: FindingHerdingPath($H^t, \delta_\beta, P^t_\beta, d_{exp}, n_{exp}, m_{solutions}$)

Input : $H^t, \delta_\beta, P^t_\beta, d_{exp}, n_{exp}, m_{solutions}$
Output : σ^t_{opt}

Calculate $d(P^t_\beta, H^t)$
set $d_{exp} = m_{exp} \times \delta_\beta$
Calculate $n_{points} = integer(d(P^t_\beta, H^t)/d_{exp})$
$\sigma^t_{temp} \leftarrow [P^t_\beta, H^t]$
if $d(P^t_\beta, H^t) \leq d_{exp} || \forall \pi P^t_\beta, H^t \notin R_{\pi\beta} || m_{exp}(n_{points} + 1) < \frac{R_{\pi\pi} - R_\pi}{2\delta_\pi}$ **then**
    $\sigma^t_{opt} = \sigma^t_{temp}$
    $n_{points} = 0$
else
    Cut σ^t_{temp} into equidistant $(n_{points} + 1)$ straight steps
    Generate m random $p_{n,m}$ points within d_{exp} from each n point in the n_{points}
    $\sigma^t_{opt} \leftarrow P^t_\beta$
    for $n = 1 : n_{points}$ **do**
        Initialise PSO parameters
        Initial PSO solutions $\leftarrow p_{n,m}$
        Evaluate F_2 using (3.16)
        Sort Solutions to get $p^*_{n,m}$, as per (3.17), (3.9)
        $\sigma^t_{opt} \leftarrow p^*_n$
    $\sigma^t_{opt} \leftarrow H^t$
Send σ^t_{opt} to the shepherd

described in the following subsections.

3.5.1 Path To Herding Point

The shepherd's step size per time step is limited, and its arrival from its current location (P^t_β) to the obtained herding point (H^t) takes more than one time step. If the shepherd is initially within the agitation range of any sheep, the time taken for its traversal to H^t within this range of any sheep will noticeably change the $G^{t_{new}}$ at $t_{new} > t$. This is because this unpredictable change in the graph during the shepherd's traversal may make this H^t obtained not suitable for either influencing the sheep towards H if the shepherd is driving, or improving their cohesion while guiding them to H if the shepherd is collecting. Since $D_{herd} < R_\beta$, then H^t may be within R_β of more than one sheep, so the shepherd's traversal to H^t will lead to a high probability of change in G^t. To minimise the effect of this on the sheep's connectivity, an algorithm that finds a path for the shepherd from P_β^t to the obtained best herding point obtained (H^t) is proposed. In this algorithm, the CU searches for a feasible path from the current location of the shepherd (P_β^t) to the estimated herding point (H^t) calculated at the time $(t-1)$, whilst minimising the sheep's dispersion during the shepherd's traversal.

This path is obtained by initially creating a set of straight-lines (σ_{temp}) by dividing the line between the sheepdog's location and the herding point into $n_{points}+1$ segments of equal lengths called the 'expand distance'($d_{exp} = m_{exp}\delta_\beta$). This expanded distance represents the distance the shepherd's extends its effect through its influence on the sheep. In Algorithm 4, steps 1 to 2, the expand distance is m_{exp} multiples of the shepherd's maximum step size (δ_β). Dividing the σ_{temp} into the d_{exp} enables studying the effect of the shepherd on G^t as it reaches each end of the d_{exp} on σ_{temp}, since its travel time is proportional to the number of d_{exp} per σ_{temp} denoted as n_{points} time steps and can be measured by $n_{points} = \sigma_{temp}/d_{exp}$, as calculated in step 3, Algorithm 4. If the shepherd moves at a constant velocity v, the maximum number of time steps required to travel from P_β^t to H^t is $m_{exp}(1+n_{points})$, as the shepherd follows σ_{temp}. Thus, changes in the graph can be measured at n_{points} times throughout the shepherd's traversal to evaluate the cumulative change in G^t that impacts the task time and, consequently, the energy consumed by the shepherd to finish the task as modelled in (3.12)

$$E_{tot} = \sum_{t=0}^{T} P(t) \tag{3.12}$$

where T is the total task time, P the total power consumed, and E_{tot} the total energy consumed by the shepherd during the task time. Therefore, minimising the task time decreases the energy consumed by the shepherd which is a dynamic agent constrained by its battery size. If the sheep become less connected due to a decrease in the average node degree (Δ_{avg}) or an increase in $d(P_\pi^t, H)_{avg}$, by influencing the sheep in a direction further from home, as the shepherd travels on the σ_{temp}, the overall task time increases. However, generating an alternative path (σ^t_{opt}) that may be longer than the σ^t_{temp} may improve the Δ_{avg} without increasing the $d(P^t_\pi, H)_{avg}$.

Lemma 3.5.1. *The shepherd may follow the σ^t_{temp}, if the n_{points} are outside the $R_{\pi\beta}$ of all the sheep or $n_{points} = 0$.*

Proof. According to the real-life observations reported in [59], all the sheep are homogeneously grazing by moving randomly with a probability of 5% when the sheepdog is outside the $R_{\pi\beta}$ of all of them. Freely roaming with limited velocity makes the probability of an edge cut in a limited time ($m_{exp}(n_{points} + 1)$ time steps) tends to zero. Thus, the shepherd can follow the shortest path to reach H^t without searching for another one. □

Lemma 3.5.2. *For a connected component, the number of time steps that lead to an edge cut is upper-bounded by*

$$max(0, (R_{\pi\pi} - R_\pi)/2\delta_\pi) \tag{3.13}$$

Proof. For two neighbouring sheep moving at constant velocities (δ_π), and separated by at least $R_{\pi\pi} - R_\pi$ units to avoid a collision, the longest time needed to cut their linking edge is when they move with $delta_\pi$ in two opposite directions to cover this maximum edge length ($R_{\pi\pi} - R_\pi$). □

Lemma 3.5.3. *For every σ_{temp} estimating the intermediate points can be skipped if, and only if, the equation* (3.14) *is satisfied, that is,*

$$m_{exp}(n_{points} + 1) < \frac{R_{\pi\pi} - R_\pi}{2\delta_\pi} \tag{3.14}$$

Proof. If the time taken by the shepherd to reach H^t exceeds $min(0, (\frac{R_{\pi\pi} - R_\pi}{2\delta_\pi})$ time steps for a bridge edge in the flock, then following the σ_{temp} while guiding the sheep in the wrong direction leads to cutting the flock into two connected components. Thus, estimating the n_{points} for finding the σ^t_{opt} is required to avoid the negative

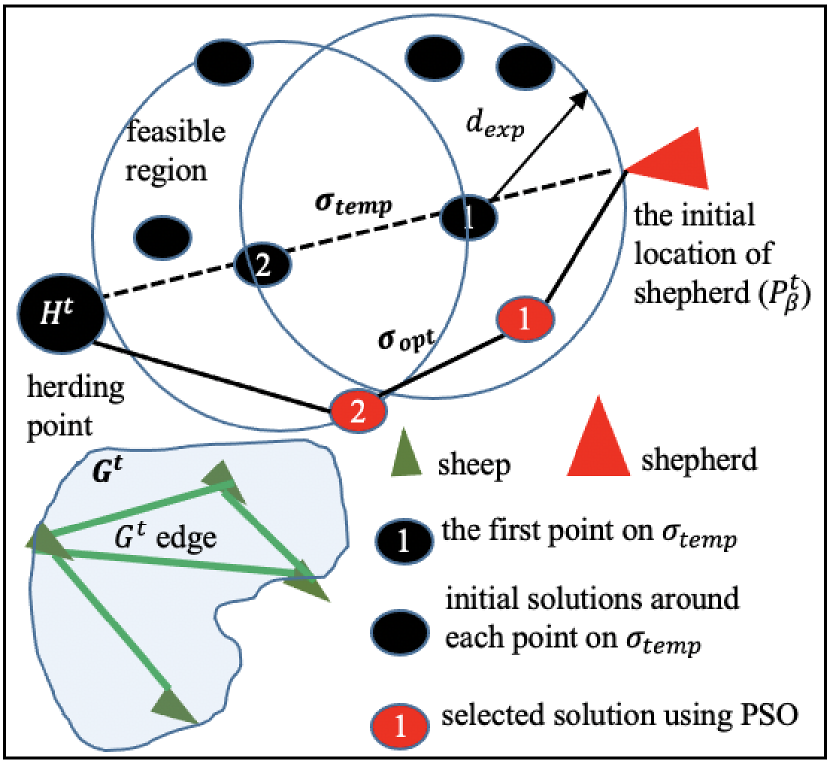

Figure 3.4: Shepherd moving from top right corner towards H^t by following the solid black step, σ_{opt} path with $n_{points} = 2$

impact of incorrectly positioning the shepherd within the influence area of G^t. This can be demonstrated with a simple numeric example: given $R_{\pi\pi} = 40$, $R_\pi = 2$, and $\delta_\pi = 2$ then the minimum time to cut an edge is $\frac{40-2}{/}2 * 2 \approx 10$ time steps; then, for the σ_{temp} with $n_{points} = 2$ and $m_{exp} = 2$ means that taking $2 \times 3 = 6$ time steps in the wrong direction will not be enough to cut this edge. Accordingly, $\sigma^t_{opt} = \sigma^t_{temp}$ to save the CU's processing power and the shepherd's energy by following the shortest path (σ^t_{temp}) (Algorithm 4, steps 4 to 7). Otherwise, estimating the effect of positioning the shepherd at each of the n_{points} on G^t is required to conserve its connectivity throughout its traversal. □

To find the near-optimal path (σ_{opt}) from the σ_{temp} one given that it has $n_{points} \geq 1$, random solutions ($n_{solutions}$) are generated within a d_{exp} distance from each point on the σ_{temp} path and the shepherd is positioned at each point in the $n_{solutions}$. A perception model is used to estimate the new sheep graph ($G^{t'}_{new}$) metrics including Δ_{avg} and $d(P_\pi^t, H)_{avg}$ at time step t_{new} given the shepherd's new location, the previous state of the sheep graph ($\prime G^{t_{new}-1}$), and the interaction of forces modelled in Equation (2.2). Then, the solution that leads to the maximum Δ_{avg} without an increase in the $d(P_\pi^t, H)_{avg}$ is chosen to replace each intermediate point on the σ_{temp} path.

The problem of finding the best replacement for each of the n_{points} as a series of single-objective constrained optimisation (CO) problems is modelled. Each of them is a sum of the objective function (F) representing the Δ_{avg} and penalty function (g) that shows the change in the $d(P_\pi^t, H)_{avg}$ from the previous time step as in (3.15). Evaluations of the new n_{points} are conducted sequentially starting from the closest point to the shepherd on the σ_{temp} path towards the H^t, which enables the CU to choose the near-optimal positions for the shepherd towards H^t based on the most recently estimated changes in G^t. The selected solution replaces its corresponding point on the σ_{temp} to form the σ_{opt}. It is worth highlighting that the algorithm prioritises the points on the σ_{temp} because they belong to the shortest path for the shepherd (Algorithm 4, steps 8 to 19). In Figure 3.4, how the CU creates the σ_{opt} for the shepherd with two intermediate points between the P_β^t and the H^t is illustrated.

$$F_2 = F + g \tag{3.15}$$

In [251], the application of PSO for determining a flock's velocity was studied to improve it according to the flocking rules proposed in [252]. Despite the previous approaches for solving the single agent and multi-agent path planning problem [61, 62, 90], there are no similar methods in the literature in which some agents are controlled by another, as in the shepherding scenario with constraints on the sheep's sensing range.

Motivated by its popularity and encouraging results for solving different types of optimisation problems [253], PSO is used to search for the best solution around each point on the σ_{temp}. In Algorithm 4, the problem is modelled so that each solution is the particle that is moving within the d_{exp} around the point in the σ_{temp} with the velocities and positions stated in Equations(3.6, 3.7), respectively, while satisfying the boundaries in Equation (3.18). Although the problem definition and the swarm indications are different from the ones used in Algorithm 2, the search methods used are the same. In Algorithm 4, steps 12 to 18, the CU sequentially searches for the (near) optimal solutions that form the σ_{opt}. Then, it adds H^t to it and sends it to the shepherd, as in Algorithm 4, steps 19 and 20, respectively.

$$\min_{x_\beta, y_\beta} \quad F_2 = -\Delta_{avg} + g \tag{3.16}$$

$$\text{s.t.} \quad g = \sum d(P_\pi^t, H)/N^t - \sum d(P_\pi^{t-1}, H)/N^{t-1} \tag{3.17}$$

$$ub = P_\pi^t + m_{exp}\delta_\beta, lb = P_\pi^t - m_{exp}\delta_\beta \tag{3.18}$$

Nevertheless, this approximation of the sheep's reactions to each position of the shepherd on the n_{points} and their generated solutions can not accurately estimate the new G^t due to the jittering in the sheep's influencing forces (see Equation (2.2)). Accordingly, cumulative changes in their average distance to home ($d(P_\pi^t, H)_{avg}$) and average node degree (Δ_{avg}) and the shepherd's distance to H^t are compared with the estimated values, which shows that their gradients are correlated. This motivated further study on the effect of the shepherd's traversal on the sheep graph with a high degree of confidence in the estimated $d(P_\pi^t, H)_{avg}$, Δ_{avg}, and $d(P_\beta^t, H^t)$.

3.5.2 Premature Termination of Path

As the shepherd receives the path, it starts following it while considering its distance to it's closest sheep using the range of its omni-directional field-of-sensing range with radius $R_{\pi\beta}$. If this distance between the shepherd and the closest sheep is less than a threshold distance (d_{stop}), the sheepdog stops and only continues following the path as the sheep move away from the sheepdog. Meanwhile, the CU monitors the changes in the locations of the sheep and shepherd (Algorithm 5, steps 1 to 5). It uses simple rules to ensure that the values estimated in creating the path match the actual measures of the sheep's node degree and distance to home. Therefore, the progress of the shepherd towards task completion involves following the estimated path to the herding point. If this preservation of node connectivity and minimisation of the sheep's distance to H are not achieved for a number of times equal to n_{stop} for each herding path, the path is terminated. Subsequently, a new herding point is generated (Algorithm 2) and a path to be sent to the shepherd is constructed (Algorithm 4). The counter for the number of violations in each path is initialised to zero as the new herding path is created, and it is incremented if any of the following conditions occur (Algorithm 5, steps 6 and 7):

1. the sheep's average node degree (Δ_{avg}) decreases from its value in the previous time step;
2. the sheep's average distance to home increases from its value in the previous time step;
3. the sheep's average distance to the GCM increases from its value in the previous time step; or
4. the shepherd stops due to being closer than d_{stop} to any sheep.

When the counter n_{stop} reaches the number of points in the σ^t_{opt} (n_{point}), the CU terminates the path for the shepherd by first creating a new one σ^{t+a}_{opt} and second sending it to the shepherd, where $t+a > t : a \in \mathbf{N}$ (Algorithm 5, step 8 and 9). This monitoring algorithm ensures that the displacement between the points on the newly generated path and the sheep is sufficiently large to avoid further dispersion because the new graph state is considered for the generation of the σ^{t+a}_{opt}. Moreover, it ensures that all the generated points match the estimations of their desired objectives which influence the sheep towards home and improve flock cohesion. However, this leads to a higher frequency of path estimations and, thus, transmissions, if the generated paths are frequently violating the aforementioned conditions for flock cohesion or distance to home.

Algorithm 5: PathTerminationCriteria($P_{i \in N^t}, P^t_\beta$)

Input : $P_{i \in N^t}$,P^t_β
Output : *terminate*

Observe the locations of the sheep and the shepherd $P_\pi \forall \pi \in \Pi$ and P_β
Calculate $d(P_{i \in N^(t)}, H)$, $d((P_{i \in N^(t}, \Gamma^t)$, $d(P_{i \in N^(t)}, \beta)$ and $d(P^t_\beta, H^t)$
Save $max(d(P_{i \in N^(t)}, H))$, $max(d((P_{i \in N^(t}, \Gamma^t))$ and $min(d(P_{i \in N^(t)}, \beta))$
Generate the sheep connectivity graph G
Calculate Δ_{avg} of G and save it
if $(d(P_{i \in N^(t+\tau)}, H)) > (d(P_{i \in N^(t)}, H)||(d(P_{i \in N^(t+\tau)}, \Gamma^{t+\tau})) > (d((P_{i \in N^(t}, \Gamma^t)||\Delta^{(}_{avg} t + \tau) < \Delta^t_{avg}||min(d(P^{(}_\beta t + \tau), P_{i \in N^(t)})) > d_{stop}$ **then**
    increment(n_{stop})
if $n_{stop} == n_{thresh}||d(P^t_\beta, H^t) < \delta_\beta$ **then**
    $terminate \leftarrow True$

In Figure 3.4, the general workflow of the SOMCP, which shows the role of the shepherd in taking the decision to stop in order to avoid further dispersion of the sheep, is shown. Moreover, the role of the CU is to use its observations of the locations of the flock and the shepherd, and, consequently, changes in the flock graph to decide the termination of the generated path when n_{stop} reaches its threshold value. Then, it calculates a new path by first considering the new graph's connectivity state combined with geometric rules, PSO and data clustering (DBSCAN) methods to find the new herding point which it sends to the shepherd. Following this step, the CU creates a path between the shepherd's current location relative to this herding point by estimating the effect of its locations during its traversal to the herding point on the sheep graph. This allows the CU to create a set of paths that enables the shepherd to reach its herding point while it improves the sheep's connectivity and avoids influencing them away from their home location (H).

3.6 Results

In this section, the experimental setup and results achieved by the SOMCP are discussed and compared with those of other algorithms.

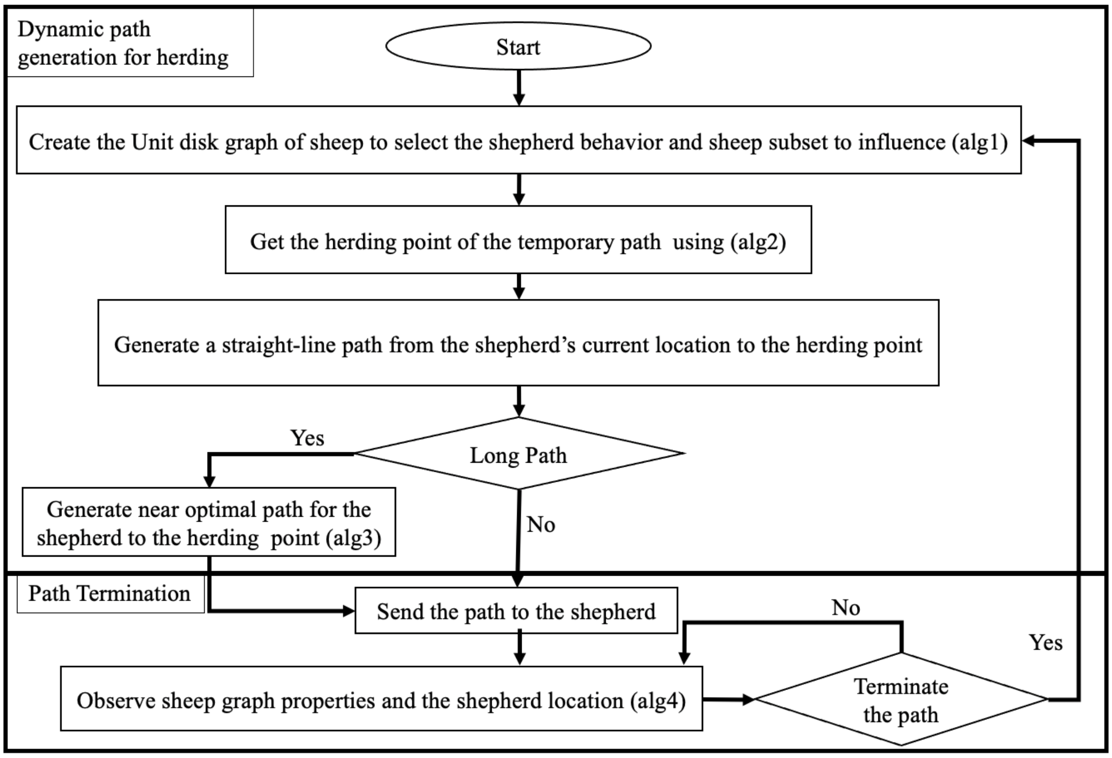

Figure 3.5: SOMCP workflow

3.6.1 Experimental Setup

In this section, the simulation setup used to assess and compare the proposed method, its predecessor SOHP and state-of-the-art methods Strombom's model [59] and CADSHEEP [17] is explained. Also, the effects of using PSO to search for the points to find the near-optimal path (σ), the errors in the functions used to estimate the objective function (Equation (3.16)), and the generation of σ are analysed. Finally, the results are presented.

Similar to the testing process conducted for the SOHP, multiple scenarios are generated to test the SOMCP. Firstly, the sheep are initialised in a square of length l, as formulated in Equation (3.19)

$$l = kR_{\pi\pi}\sqrt{N} \tag{3.19}$$

where k is a constant. Subsequently, the initial sheep density (ρ) is

$$\rho = N/l^2 = N/(kR_{\pi\pi}\sqrt{N})^2 = 1/(kR_{\pi\pi})^2 \tag{3.20}$$

The maximum time allowed to complete the shepherding task is mathematically derived in the SOHP by mapping the task time in [59], described in Equation (3.21), to the product of a function in the sheep's initial density and sensing range ($R_{\pi\pi}$). The density factor (k) that forms the first term which is added to an estimation of the time taken by the shepherd and the sheep furthest from H to reach H, as described in Equations(3.22, 3.23), respectively. This yields the final maximum task time formula in Equation (3.24)

$$T_{max} = 20N + 630 \tag{3.21}$$

$$20N \rightarrow kR_{\pi\pi}N \tag{3.22}$$

$$630 \rightarrow k_1(d(H^0, H)/\delta_\pi + d(H^0, P^0_\beta)/\delta_{\beta)} \tag{3.23}$$

$$T_{max} = k_1(d(H(0), H)/\delta_\pi + d(H(0), P^0_\beta)/\delta_\beta) + kR_{\pi\pi}N \tag{3.24}$$

where P^0_β and H^0 are the initial locations of the shepherd and herding point, respectively, while δ_β and δ_π are the maximum step size for the shepherd and the sheep, respectively. The flock is randomly initialised at five different density factors ($0 < k < 1$) using the simulation parameters described in Table 3.1. Note that the bold font highlights the maximum SR, Δ_{avg} and % of sheep at H at the end of the task time obtained from Equation (3.24), as well as the minimum T, and n_{cc} since these values are the most desirable for the shepherding task.

Table 3.1: System Parameters

Parameter	Value
Number of sheep	$[50, 75, 100, 200]$
Maximum size of N_{subset} ($\|N_{subset}\|_{max}$)	$N/2$
Environment length L	300
Shepherd initial location P_{β}^{0}	(L, L)
Distance tolerance constant k_1	2
Home location (H)	(0,0)
Home radius R_H	50
Minimum initial sheep distance to H	$L/4$
Density factor (k)	[1/4,1/3,1/2,2/3,3/4]
Shepherd maximum step size δ_{β}	5
Sheep maximum step size δ_{π}	2
Sheep step size in grazing	0.05
Sheep sensing radius for shepherd $R_{\pi\beta}$	70
PSO in Algorithm 2: c1,c2,w,k_{iter}	1.5,2,0.3,21
Number of steps bounding PSO search k_{steps}	7
DBSCAN: υ,ϵ	$N^t/10, 0.5$
Agitation weight $W_{\pi\beta}$	1.9
Sheep collision avoidance radius R_{π}	3
Collision weight $W_{\pi\pi}$	1.5
Sheep sensing radius $R_{\pi\pi}$	15
Cohesion weight $W_{\pi\Lambda}$	1
Threshold node degree Δ_{thresh}	$N^t/2$
Herding distance D_{herd}	$R_{\pi\beta}/2$
Jittering weight $W_{e\pi}$	0.3
Shepherd‘s stopping distance d_{stop}	$3R_{\pi}$
Inertia weight $W_{\pi\upsilon}$	0.5
Agent mass m	1 kg
Packet length b	0.5 MB
Energy dissipation in electronics E_{elec}	50 nj/b
Time duration for sensing transmitted packet T_{sense}	0.5 ms
PSO in Algorithm 4: $c1_2$,$c2_2$,w_2,$k2_{iter}$,$n_{solutions}$	1.5,2,0.1,4,5
Flash reading current for 1 B data (I_{read})	6.2 mA
Time duration for flash reading T_{read}	565 μs
Supply voltage V_{sup}	2.7 V

3.6.2 The Effect of Removing Components From SOMCP

The herding path is generated by first creating a straight-line one composed of an equidistant set of points between the shepherd's current location and the near-optimal herding point obtained by Algorithm 2. The distance between the points forming that path is $2 \times d_{exp} = 2 \times m_{exp} \times \delta_{\beta}$. Then, for each point on the path, the aim is to find a point around it on the straight-step path that maximises the sheep's node degree without increasing its distance to home. Therefore, the new path is generated by consecutively selecting the best points in a set of points as solutions around each of the equidistant sets of points. Each set of solutions initially includes a point on the straight-step path and four randomly generated ones within a distance of $d_{exp} = m_{exp} \times \delta_{\beta}$ from it.

Since PSO is used in path creation by searching for the points around the initial solutions of each σ_{temp}. If PSO is not used, one of the five initial solutions will be chosen if it leads to the maximum node degree and doesn't increase the sheep's average distance to home when the sheepdog is positioned in it. An evaluation of the objective function in Equation (3.16) is performed for each point in the solution set, with the shepherd replacing that point on the straight-line with the best one. These steps are repeated consecutively for each point on the straight-step path. When using PSO, a further search within d_{exp} is conducted to find each point on the path consecutively. In this subsection, the efficiency of the SOMCP without PSO at different lengths of the d_{exp} is analysed to highlight the effect of using PSO in path creation rather than arbitrarily searching five generated solutions. The SOMCP is implemented without PSO as a path planner for different values of m_{exp} of $1, 2$ and 3.

In these experiments, one sheepdog herds $N = 50$ sheep initialised with a density factor of $k = 2/3$ for 25 runs to measure its success rate SR, the percentage (%) of sheep at H during T_{max}, the task time (T), the average node degree (Δ_{avg}) and the average number of connected components of the sheep (n_{cc}) presented in Table 3.2. These results show that increases in the m_{exp} when PSO is not used

Table 3.2: Numerical Results For 50 Sheep

Metric	Mean ± Standard Deviation				Best			
m_{exp}	1	2	3	1*	1	2	3	1*
SR	**95**	68	18	64	100	100	100	100
task time	555.5 ± 101	679.2 ± 101	787.6 ± 27	**553.0** ± 135	**367**	529	707	452
node deg	14.3 ± 1.9	15.6 ± 2.3	**19.0**± 2.9	14.4 ± 6.1	18	22	**26**	7
n_{cc}	2.8 ± 0.4	2.7 ± 0.4	2.9 ± 0.5	**2.3** ± 0.3	2.3	**1.9**	2.0	2.0
% at H	**97.6** ± 11	85.3 ± 28	30.2 ± 41	96 ± 9	100	100	100	100

degrade the performance due to the inaccuracy of the estimations in Equation (3.16). Accordingly, $m_{exp} = 3$ has the longest T, minimum SR and lowest percentage of sheep at home (% of sheep at H) compared with lower m_{exp}; in other words, the smaller the search distance, the higher the SR, mainly because of the limited number of solutions, that is, only five. Also, the smaller the m_{exp}, the more points in the path that minimise errors in the estimations in Equation (3.16). Consequently, finding the best solution within a very small distance provides better estimation accuracy and, therefore, a relatively high probability of path improvement.

The errors in the estimations of the sheep's Δ_{avg} and distance to home for each new point selected are shown in Figure 3.6. The dotted step represents the average for 25 runs of the estimated values and the dot-dashed one is the average for 25 runs of the actual ones of both the Δ_{avg} and $d(H, \pi)_{avg}$. These results indicate slight differences between the actual and estimated Δ_{avg} and distances to home, respectively, in all cases both with and without PSO. This is due to jittering in the sheep's movements being magnified as the distances between the points forming the straight-step path increase as the m_{exp} increases when PSO is not used. This is because when the m_{exp} is relatively large, as the number of estimations for the whole path decreases and the displacement between each two estimations increases, the effect of jittering is more noticeable. However, the errors in estimations of the Δ_{avg} and $d(H, \pi)_{avg}$ in Figure 3.6 have relatively little effects on the accuracy of estimations of the objective function in Equation (3.16) when PSO is used, despite the value of $m_{exp} = 3$. This is because, the more solutions generated during the PSO iterations, the higher the probability of accurately choosing better points in terms of the shepherding performance, as highlighted by comparing the SR, % of sheep at H, and T in Table 3.2 for $k = 2/3$.

It is worth highlighting that DBSCAN is used to find the cluster of the near-optimal directions for N_{subset} sheep, with the mean direction of the highest density cluster considered as the solutions of directions. According to Table 3.2, the effect of DBSCAN is validated when PSO is used for path planning in the case of the best estimation accuracy, where $m_{exp} = 1$. In this variant of the SOMCP, using DBSCAN to select the cluster with the highest density is replaced with the mean direction of the near-optimal directions of the N_{subset} sheep. This step allows the outlier sheep in the N_{subset} to influence the selected direction, which leads to further dispersion of the flock, despite the use of PSO in path planning. This variant of SOMCP is refereed to as '1^* '. The results in Table 3.2 show that removing the DBSCAN component reduces the SR of the task compared with those of the relatively high-accuracy estimation cases ($m_{exp} = 1, 2$), despite the use of PSO for path planning. However, the overall performance of the SOMCP without DBSCAN is relatively better than that when PSO is not used in path planning when $m_{exp} = 3$; this is due to the limited number of outliers in N_{subset} that is limited to a relatively small number of sheep ($|N_{subset}| \leq 50/2$).

3.6.3 Effect of Estimation Model

In this subsection, the same settings as in the previous one are used to show the effect of the proposed model for estimating the reactions of the sheep graph to the shepherd's locations. An example of point selection by the shepherd when it herds 50 sheep at time step 300 is depicted in Figure A.I. The points on the shepherd's path are selected from a set of points within $m_{exp} = 3$ multiples of its maximum step size (δ_β) around its initial location denoted by '0 ', and the given numbers showing their sequences in that path. The subfigures in Figure A.I show the values of the three decision metrics used to select the points on the herding path: (a) the Δ_{avg}; (b) the $d(H, \pi)_{avg}$; and (c) the distance between the shepherd and final point on the path, respectively. These points are presented in all the subfigures to show how the new point achieves a relatively high Δ_{avg} within the d_{exp} around the shepherd's prior location, and the lowest possible $d(H, \pi)_{avg}$ when the shepherd approaches its herding point. It's worth noting that the negative of the Δ_{avg} is minimised to align with the second objective in Equation (3.16). The results demonstrate that using PSO with a limited number of iterations to search for the next point satisfies the objectives of the problem by considering the three ruling metrics for the task's success. The points on the path improve the Δ_{avg}, minimise the $d(H, \pi)_{avg}$, and make the shepherd move closer to its herding point than all the points around it, thereby maintaining a cohesive flock during traversal to the herding point estimated in Algorithm 2.

3.6.4 Comparative Analysis

In this subsection, the performance of the proposed technique is compared with those of Strombom's model [59], CADSHEEP [17], and SOHP. The metrics used in the comparison are the average (+/−) one standard deviation of the following metrics: (a) SR; (b) the task time (T); (c) % sheep at H at $t = T_{max}$; (d) Δ_{avg}, as defined in Section 3.2; (e) the number of cc (n_{cc}), as explained in Section 3.2; (f) the total energy consumed by the shepherd when receiving the path or heading point from the CU, sensing the influence region around it and moving from one point to another, as formulated in Equation (3.25) which shows the effect of the task time on a dynamic system restricted by a specific battery life (Equation (3.12)); (g) the average of the absolute differences between the estimated LCM γ^t of all the sheep and their actual GCM Γ^t, which shows the effect of small $R_{\pi\pi}$ on the flock cohesion, as discussed in Section 3.2; and (h) the number of transmissions of either the herding point or path n_{Tx} for which the Strombom [59] and CADSHEEP [17] models assume that the shepherd receives a new heading every time step while, in the SOHP and SOMCP, the CU sends a set of headings every set of time steps in an adaptive fashion, as described in Figure 3.3. The results presented in Appendix 6.3 in Tables (1, 3, 5, and 7) show the efficiency of the proposed approach with swarm

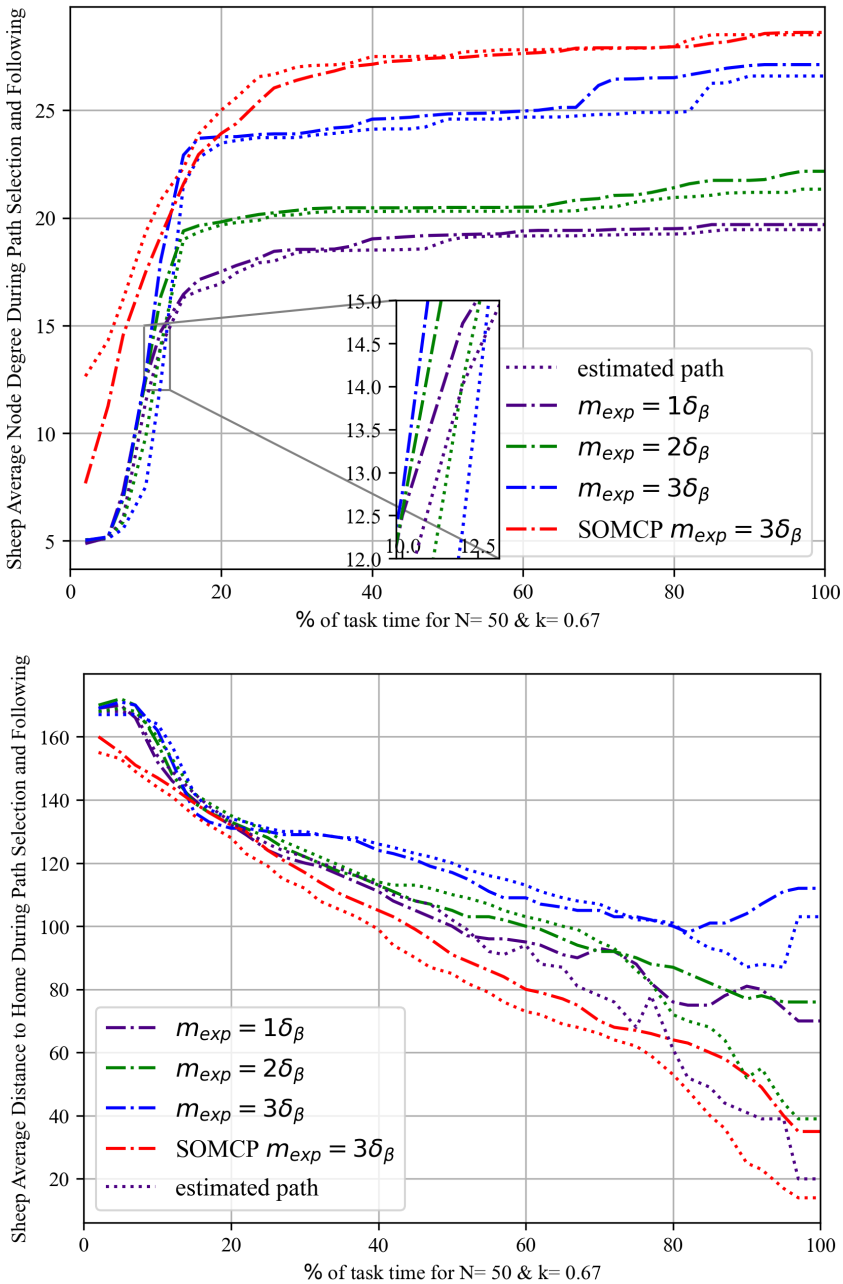

Figure 3.6: The difference between the actual (dot-dashed steps) and estimated (dotted steps) values for Δ_{avg} (left) and $d(H, \pi)_{avg}$ through the percentage of task time for $N = 50, k = 2/3$, where the search distance $k_{steps} = 3$ in PSO

sizes of 50, 75, 100, and 200 sheep, respectively.

$$E_{tot} = E_{motion} + E_{sense} + E_{rx} \tag{3.25}$$

For simplicity, the kinetic energy is formulated as $E_{motion} = (1/2)mv^2$, m is a unit mass, and v is the velocity, while the energy consumed in receiving control packets from the CU is formulated as $E_r = bE_{elec}$ where b is the number of bits, E_{elec} is an electronic constant, and the energy consumed in reading the packets as $E_{sense} = bV_{sup}I_{sens}T_{sens}$, where V_{sup} is the supply voltage, I_{sense} is the total current required for sensing activity and T_{sense} is the time duration of sensing.

The results in Tables (1,3,5,and 7) in Appendix 6.3 demonstrate that the SOHP enhances the SR for 50 and 75 sheep in the case of low initialisation densities with $k = 2/3, 3/4$, with improvements at these densities and $k = 1/2$ for 100 and 200 sheep, respectively. The overall degradation in high-density cases is below 15% and is offset by the improvements in the lower 2 and 3 densities for 50 and 100 sheep, respectively, of up to 50%. The task time in the high-density initialisation cases for 50 sheep is longer for SOHP by an average of 15%, once again offset by an average 30% decrease in the lower density cases. Also shown is that the SOHP is capable of driving a large percentage of the sheep to the goal even if the overall goal is not achieved.

The results in Tables (1,3,5,and 7) in Appendix 6.3 demonstrate that the SOHP enhances the SR for 50 sheep by up to 100%, with the T offset by an average 100% reduction in the lower-density scenarios. It is also apparent that SOMCP is capable of driving a large percentage of the sheep to the H. Based on the Δ_{avg} and n_{cc} values obtained (Tables(1,3,5 and 7), in Appendix 6.3), the CADSHEEP and Strombom models maintain relatively high node degrees and low numbers of connected components in most of these initialisation cases. However, compared with the SOHP and SOMCP, in the majority of the cases, they could not attain a lower T or higher SR.

This demonstrates that using graph metrics effectively in the proposed approaches (the SOHP and the SOMCP) achieves the main objective of improving the efficiency of the shepherding task in terms of success rate and task time without being intensively concerned about enhancing them. The bold fonts in the tables highlight the maximum SR, Δ_{avg} and percentage of sheep at home at the end of the task time obtained from Equation 3.24 as well as the minimum task time, and the number of connected components (n_{cc}), the values of which are the most desirable for the shepherding problem.

This observation is also confirmed in Figure 3.7 which shows changes in the Δ_{avg} and rates of reaching home for 100 sheep. In Figure A.III, it is clear that the SOMCP improves the Δ_{avg} avg more quickly than the other three models. Moreover, the curve is stable as the Δ_{avg} reaches a reasonably high value in low-density initialisation scenarios. Although for the high-density initialisation scenarios, these low Δ_{avg} values lead to a longer T than those of the other models, this is acceptable. This can be observed in Figs.(A.II and A.IV), where the SOMCP at $k = 1/4$ has approximately 15%, 7%, and 50% slower task time for 75 , 100 and 200 sheep, respectively, than CADSHEEP. However, as k increases, it achieves a higher percentage of sheep at H than the other models/algorithms. This allows us to conclude that the SOMCP is more suitable for dealing with dispersed sheep without prioritising their connectivity over the task completion time.

The total energy consumed by the shepherd is proportional to the task time. Thus, the E_{total} is the least for the SOMCP compared to its peers in all cases except at $k = 1/4$ for 75 and 100 sheep. However, the average absolute error of the estimated centre of mass of the sheep is always the least for the SOMCP which means that it maintains sheep cohesion relatively better than its peers. Similarly, the number of transmissions (n_{Tx}) it requires during the T is noticeably less than those of its peers in all cases regardless of the T because the shepherd receives a whole path and continues to follow it for a large number of time steps which exceeds the time taken.

This number of time steps exceeds the time taken by the shepherd to follow a

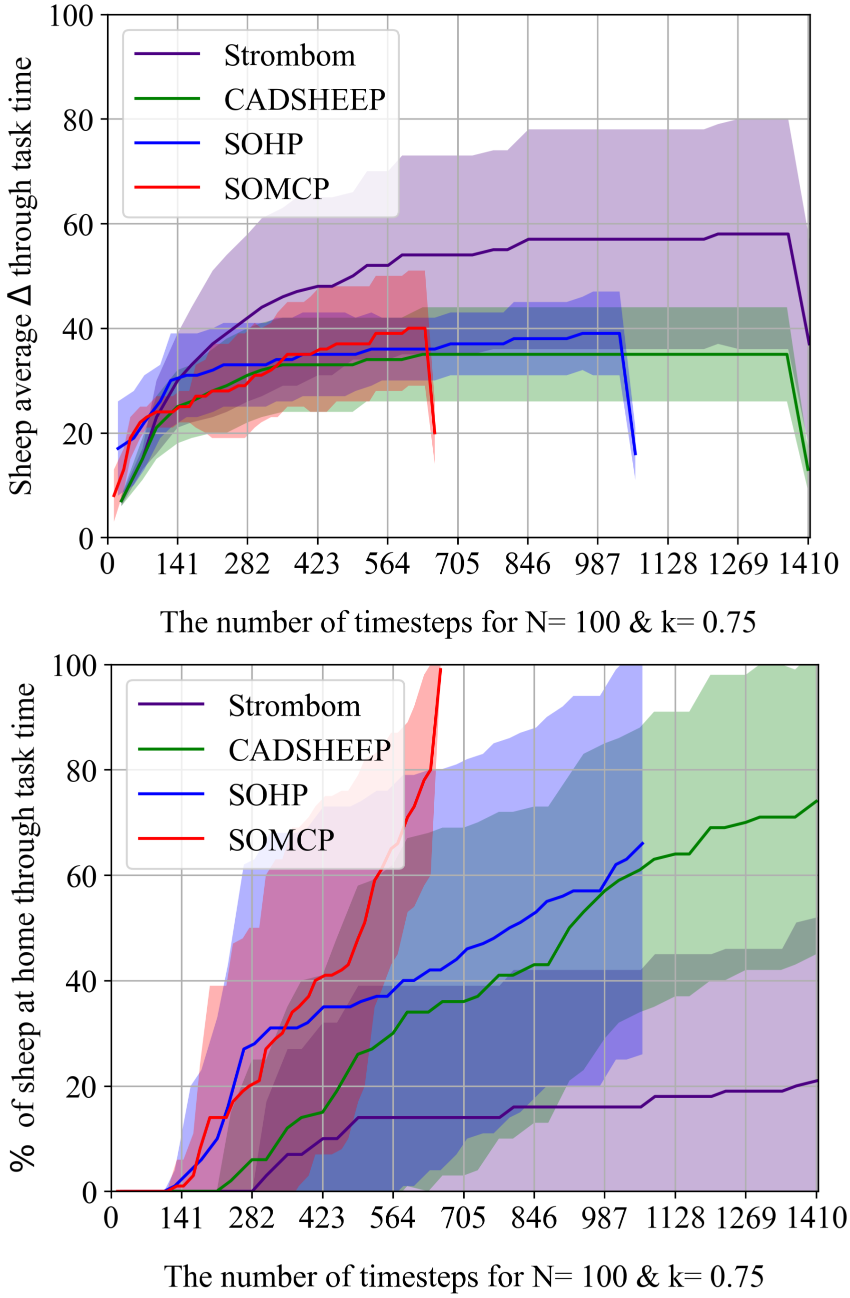

Figure 3.7: The change in the average node degree (left) and the average percentage of sheep reaching home (+/−) one standard deviation for N=100 initialised with the lowest density, k=0.75

straight-line from its initial location to the near-optimal herding point, as in the SOHP. This very low number of transmissions adds a level of independence for the shepherd from its controlling unit CU which makes the SOMCP more resilient to communication problems that may occur during the task time on the communication link between the shepherd and its CU during the task time.

3.7 Chapter Summary

In this chapter, a decentralised model similar to how a human shepherd (the CU in our model) instructs sheepdogs in real-life shepherding scenarios was presented. It aligns with the assumption that the CU, as the decision-maker, can obtain the real-life locations of all the sheep in real time to model them as dynamic network components given their limited sensing range. Two algorithms based on the UDG of the sheep were introduced.

The high tendency of sheep to disperse as they are herded due to their limited sensing range was addressed by quantifying the properties of the sheep graph to be optimised. Two approaches (the SOHP and SOMCP) were proposed. In the SOHP, PSO is employed to search for a near-optimal herding point for the sheepdog that minimises the sheep's distance to home while ensuring cohesion. In the SOMCP, PSO is used to find the path the shepherd could take to reach that herding point while improving the sheep graph's connectivity and avoiding influencing the sheep away from their home location. Finally, in the SOMCP, a set of graph metrics-based rules are used to penalise the shepherd by terminating the path it is following if it adversely affects the flock's cohesion or distance to home several times during its traversal.

The algorithms were tested on multiple scenarios with different settings. The results indicated that cohesion among the sheep was achieved if, and only if, they could locate enough of their neighbours. Furthermore, the SOMCP outperformed existing shepherding approaches for herding up to 200 sheep initialised at different densities considering their limited sensing range. Although the proposed algorithms demonstrated promising results, they assumed ideal communication between the CU and the shepherd. Since shepherding may be undertaken in remote areas without access to CU locations with large transmitter antennas and power sources, the problem of noisy communication end-end systems and channels is addressed in the next chapter.

The algorithms were tested on multiple scenarios with different settings. The results indicated that using SOHP and SOMCP, the cohesion among sheep was achieved if, and only if, they could locate enough of their neighbours. Furthermore, the SOMCP technique outperformed the existing shepherding approaches in herding up to 200 sheep initialised at different densities considering their limited sensing range.

The proposed algorithms explained in this chapter showed promising results, but they assumed ideal communication between the CU and the shepherd. Since shepherding may be done in remote areas without access to central unit locations with large transmitter antennas and power sources, the problem of noisy communication end systems and channels will be addressed in the next chapter.

Chapter 4

Cooperative Learning for Shepherding with Time-varying and Noisy Communication Channels

The work, reported in this chapter, has been partially published in the following article:

RE Mohamed, R Hunjet, S Elsayed, H Abbass, *Deep Learning For Noisy Communication System*. 2021 31st International Telecommunication Networks and Applications Conference (ITNAC), 40-47

The aim of this chapter is to improve shepherding in the presence of time-varying, noisy communication channels. The system model and problem formulation are discussed after the introduction. Then, the proposed cooperative learning solution to the problem is explained followed by a modified version of the approach that improves bandwidth utilisation in the training phase. Finally, the results and analysis of the proposed technique are articulated.

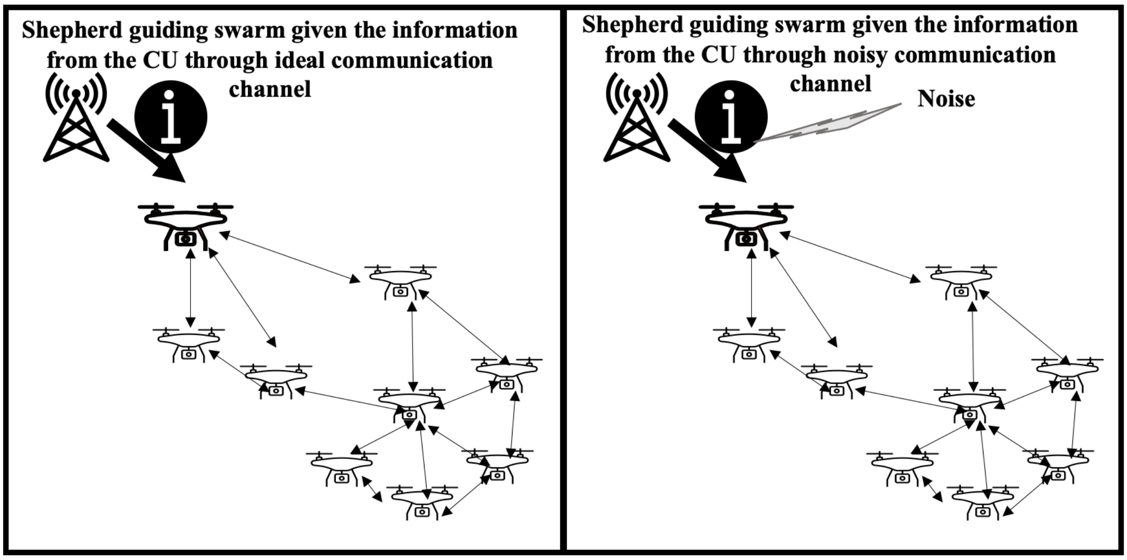

Figure 4.1: The effect of channel noise on the shepherding task

4.1 Introduction

In accordance with the shepherding paradigm outlined in Chapter 3, the shepherd receives instructions from a central unit (CU) that possesses real-time, comprehensive knowledge about the locations of the sheep. It is assumed that the shepherd (as a receiver (Rx)) and CU (as a transmitter (Tx)) are working under assumptions of an ideal communication system, as shown on the left-hand side in Figure 4.1. However, since the work of shepherding takes place in distant locations, unforeseen channel fluctuations may occur during the task, as shown on the right-hand side in Figure 4.1. This exposes the system to the unpredictability of the communication channel, which may result in actuator failures or sheep misguidance due to erroneous information received from the CU.

As described in Chapter 2, in communication systems, noise may be defined as any undesired signal that interferes with the communication, measurement or processing of an information-bearing signal [254]. To address the problem of a noisy communication channel in the shepherding tasks in this chapter,learning-based solutions for a common communication system suffering from high levels of noise are proposed. They are designed and validated through training and testing the models for a large number of transmissions regardless of the transmitted data. They are then combined with the shepherding methods proposed in Chapter 3 to improve the efficiency of the swarm optimal herding point (SOHP) model/method under the effect of a noisy communication channel.

Despite the commercial success of wireless communication systems in recent years, the radio channels in mobile radio systems are very noisy and time-variant due to the effects of different noise sources [151]. Therefore, recovering data after a high level of corruption has been an overarching problem in the last decade. As discussed in Chapter 2, numerous modulation schemes have been devised to balance the spectral efficiency and noise resistance of transmitted data. Quadrature amplitude modulation (QAM), which encodes the digital information in a signal's amplitude and phase, outperforms all hard-coded modulation schemes at very low signal-to-noise ratios (SNRs) [255]. For lower-order QAMs, e.g., 4-QAM, constellations with fewer bits per symbol have lower spectral efficiency and greater distortion tolerance [256, 257]. Nonetheless, existing modulation models rely on their channel models being used efficiently.

Due to inadequate environmental information in real-time events, modelling a channel's time-varying behaviour and predicting noise levels in dynamic communication system scenarios are not viable [258]. Modelling a stochastic communication channel as part of a neural network (NN) requires knowing the gradient of the instantaneous channel transfer function which is not applicable to a wireless channel in a dynamic environment [104]. Moreover, existing modulation models are not designed to improve the resistance of a signal to ranges of high levels of noise compared with its power which has low SNRs. As discussed in Chapter 2, researchers proposed different machine learning (ML) methods for removing noise from transmitted data (i.e., NNs), deep learning (DL), deep reinforcement learning (RL) and autoencoders (AEs) [1, 223, 259, 260, 261]. A de-noising AE has been shown to perform better than the basic one on noise reduction tasks.

In this chapter, a study of the effects of different noise levels in end-devices, namely internal noise, and the impact of channel noise ($\mathcal{N}$) on the performance of our proposed learning framework applied to a common communication system model is presented. Firstly, an independent pre-training collaborative learning (IPCL) framework that reduces the effect of the internal noise in each end-system individually is introduced. In it, de-noising AEs are designed for the Tx and Rx to remove internal noise. Then, collaborative learning by other AEs at the Tx and Rx that are initialised with the same weights as in the AEs for internal noise reduction is employed. This reduces the effect of channel noise on the received data in a short training time.

Eliminating the feedback channel (F-CH) in the assumption during the collaborative learning phase of IPCL is studied in one IPCL version, namely, IPCL-no-FCH. Considering the efficiency of 4-QAM in improving a signal's resistance to high noise levels, the design of IPCL-no-FCH is improved using constellation mapping of 4-QAM through limited bandwidth during training. The same design as that of the Tx in IPCL is used, with modulations of the symbols represented as I/Q signals. However, the Rx is redesigned so that it has one or two AEs, each of which is trained on each dimension of the mapped symbols separately.

The contributions of this chapter are summarised as follows:

- In a general communication system, similar AEs at the Tx and Rx are designed to make them learn to overcome noise from internal devices and channels during data transmission. This is a two-phase training method developed to enable the communicating end-systems to obtain low levels of error after a short training time in challenging communication scenarios.

- In a general communication system with modulated data, the proposed DL-based method is improved to allow the Tx and Rx to cope with challenging SNR levels with minimal use of the communication channel in the training phase.

- For a shepherding problem with noisy communication channels, a hybrid de-

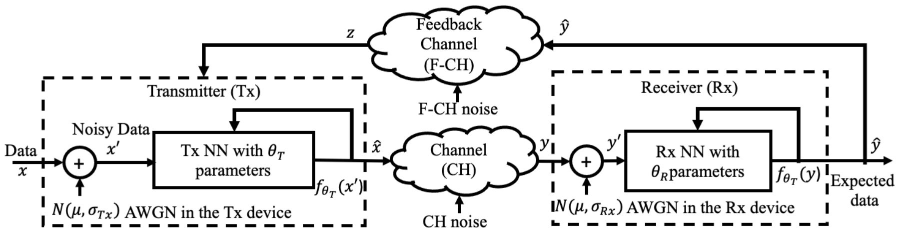

Figure 4.2: The Communication Model

modulation method at the Rx, where 4-QAM is used with a NN at the Rx to improve the success rate of transmissions, is proposed. Then, it is validated on a shepherding scenario as an example of a dynamic system's communication problem, where SNR changes due to the mobility of the Rx.

4.2 Channel Model For Shepherding

In this chapter, a communication system with one Tx and one Rx, which both experience internal noise distributed as additive white gaussian noise (AWGN), with a zero mean ($\mu = 0$) and standard deviations (σ_{Tx} and σ_{Rx} for the Tx and Rx, respectively)is presented. According to the proposed framework, each end-system (Tx and Rx) has a NN denoted by NNT and NNR, respectively. They communicate through a stochastic forward CH and an F-CH described as a random function with a probability $y = P(.|x)$ for each input (x). The sequence of the data exchange scenario in this system depicted in Figure 4.2 is described in the following:

1. The Tx transmits a message (x) which, due to its internal noise, becomes $x' = x + \mathcal{N}(0, \sigma_{Tx})$. Then, the NNT transmits its estimated value of $\hat{x}$ through CH.
2. The CH transforms $\hat{x}$ into y; thus, $y = P(.|\hat{x})$.
3. The Rx receives y which, due to its internal noise, it becomes $y' = y + \mathcal{N}(0, \sigma_{Rx})$, then, the NNR interprets it as $\hat{y}$.

4. The F-CH transforms the $\hat{y}$ into z; thus, $z = P(.|\hat{y})$, which means that the Tx becomes aware that the transmitted data (x) is interpreted at the Rx as z.

Within the shepherding context, the CU acts as the decision-maker that observes the sheep's dynamics and sends their herding locations to the shepherd, which acts as an actuator steering the sheep to their home location, as modelled in Chapter 3. In a shepherding task, the effect of a noisy communication system on the efficiency of the sheepdog (Rx) that receives the herding points from the CU (Tx) to guide the sheep is illustrated in Figure 4.1.

According to the SOHP, in/at every set of time steps, the CU sends the herding point to the shepherd over an AWGN communication channel. Then, in the following set/time step, it may send an error-checking message via an ideal communication channel to enable the shepherd to verify the data and follow only the herding points received accurately. If the previous locations received by the shepherd are inaccurate due to the channel noise, it stops.

According to the SOHP, the CU sends the herding points to the shepherd at a variable rate, whereby transmissions occur whenever the shepherd reaches the recently sent herding point. This low bandwidth utilisation due to sending only one location to the shepherd in 2 or 3 dimensions as well as a low transmission rate makes the SOHP resistant to channel noise. This is because an inaccurate transmission can be followed by a large number of retransmissions which may increase the probability of successful transmissions unless the SNR is extremely low. The time differences between subsequent herding points ($t_{i,j}$) depend on the amount of time taken by the shepherd to move from one herding point (i) to the next one ($i + 1$). The longer it is, the higher the probability of successful retransmission.

The problem of a noisy communication system is described in the next section to show how the data exchange between the Tx (which may represent the CU) and Rx (which may represent the sheepdog) is affected by channel noise. This general model is applicable to any dynamic communication system through which the Tx or Rx move.

4.3 Problem Formulation

To minimise the error between a message (x) and its value at the Rx output ($\hat{y}$), the mean squared error (MSE) should be minimised. Losses in the MSEs on the Tx and Rx sides are minimised by finding the optimised NN parameters (θ_{Tx} and θ_{Rx}) as formulated in Equations (4.1, 4.2), respectively. These minimisations take place during both the individual and collaborative training phases to reduce the internal and channel noises, respectively. Since the actual speed of training depends on the local hardware available to each agent, the number of epochs is an indication of it. An epoch is the time taken to pass one message between both end-systems and receive feedback when it's available. As, during training, learning occurs, both the θ_{Tx} and θ_{Rx} are updated. However, during testing, this does not occur.

$$\min_{\theta_{Tx}} \quad L(\theta_{Tx}) = \frac{1}{Ep_{indT}} \sum_{t=0}^{t=Ep_{indT}} (\hat{x}_t - x_t)^2 \tag{4.1}$$

$$\min_{\theta_{Rx}} \quad L(\theta_{Rx}) = \frac{1}{Ep_{indT}} \sum_{t=0}^{t=Ep_{indT}} (\hat{x_r} - x_t)^2 \tag{4.2}$$

where Ep_{indT} is the number of epochs during the independent training phase, all the messages are vectors of size B, x_t and y_t are the actual generated messages at different random seeds at the Tx and the Rx, respectively, and $\hat{x}_t$ and $\hat{y}_t$ are the estimated messages at the NNT and NNR, respectively.

Despite having two different entities in the system (the Tx and the Rx) that have to optimise their internal parameters (θ_{Tx} and θ_{Rx}, respectively), the Tx can send a modified version of x that the Rx can interpret as x with a minimum error. The problem of de-noising the data on the Rx side is defined as minimising the MSE between the originally transmitted data and the final output at the Rx side. The minimisation process of the MSE in the whole communication system, formulated in Equation (4.3), needs to be fast because of the stochastic time-varying nature of the wireless communication channel with different effects of signal fading and

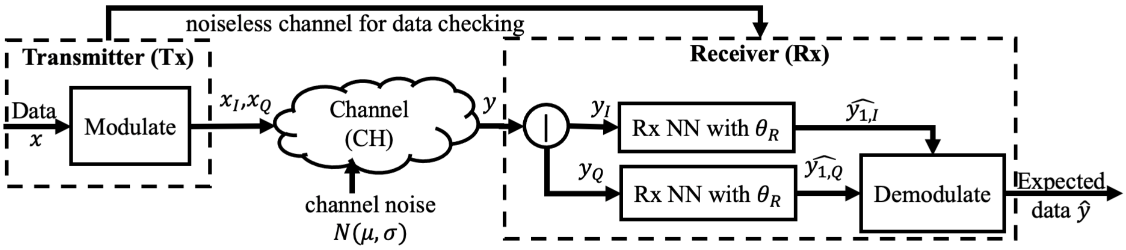

Figure 4.3: The communication model

shadowing.

$$\min_{(\theta_{Tx}, \theta_{Rx})} \quad L(\theta_{Tx}, \theta_{Rx}) = \frac{1}{Ep_{indT}} \sum_{t=0}^{t=Ep_{indT}} (\hat{y}_t - x_t)^2 \tag{4.3}$$

Definition 4.1. *An external epoch (Ep_{ext}) is the transmission of a message of any size through the communication channel to allow the Rx channel to optimise its weights to remove the effect of noise from a noisy signal.*

An external epoch is used for data transmission at the beginning of the communication process with data saved at the Tx and Rx without using the error-checking channel. The shorter its length , the higher the success rate of transmission. After each external epoch, the NN needs to train the transmitted data multiple times without the need for new transmissions. This training phase, which is conducted at the Rx independently without the need for any bandwidth, is defined as an independent training phase.

Definition 4.2. *An internal epoch (Ep_{ind}) is defined as the use of a transmitted message of any size to optimise the weights of the NN in the communication system.*

An internal epoch helps to remove the effect of the channel noise a transmitted signal experienced during the most recent transmission.

4.3.1 Modulated Data

In this subsection, a data exchange where the data are modulated in a scenario is depicted in Figure 4.3. This scenario is designed to address the noisy communication channel regardless of the internal noise that has been considered in Section 4.3. Its sequence is described as follows:

1. The Tx transmits a message x that is modulated by 4-QAM to be represented as a complex number with x_I and x_Q values on the imaginary real axis, respectively.

2. The channel (CH) leads to the scattering of the x_I and x_Q around their original values, so the data is received as complex numbers $y = P(.|x)$.

3. The Rx receives the transmitted data with symmetrically added noise in the real and imaginary parts $y = x + \mathcal{N}(\mu, \sigma)$, and splits it into y_I and y_Q so that one Rx each AE maps y_I and y_Q into x_I and x_Q, respectively in IPCL-NF-2AE; or only one AE maps the imaginary and real parts of the received complex numbers in IPCL-NF-1AE.

4. The Tx uses the noiseless channel to send the error-checking data to the Rx so that the Rx can identify the corrupted messages.

4.3.2 The Probability Of Error

All the constellation points of x are represented as either x_I and x_Q due to the symmetric AWGN and I/Q signal and the communication channel may impose. This noise is denoted as $\mathcal{N}(\mu, \sigma)$, where the variance is half the noise power $\sigma^2 = N_o/2, \mu = 0$.

In 4-QAM, the symbols are represented as $X = 0, 1, 2, 3$. These values are encoded to colors, where 0 is red for the values $x_I + jx_Q = \frac{E_s}{2}(-1-j)$, 1 is green for the values $x_I + jx_Q = \frac{E_s}{2}(1+j)$, blue is 2 for the values $x_I + jx_Q = \frac{E_s}{2}(-1-j)$, and 3 is black for the values $x_I + jx_Q = \frac{E_s}{2}(-1+j)$, as shown Figure 4.4. When the transmitted constellation points are affected by AWGN from the channel, the received ones are scattered away from their original locations on the I/Q signal. The effect of this scattering is proportional to the relative power of the added noise as a function of σ. As depicted on the left-hand side in σ. As depicted on the left-hand side in Figure 4.4, the power of the noise in the top sub-figure is less than that in the bottom one ($\sigma_1 < \sigma_2$)), that is the constellation points received are more scattered in the case of (σ_2). This high level of scattering of these constellation points at σ_2 leads to an error in interpreting a large portion of the transmitted symbols at the Rx, where many 0 ones in red are considered 1 (green) or 2 (blue).

Definition 4.3. *The symbol error rate (SER) is the probability of sending a symbol (x), with at least one of its bits decoded incorrectly.*

The probability of incorrect interpretations of the symbols at the Rx is represented by the shaded part of the probability distribution of the constellation points received (the right-hand side in Figure 4.4). Deviations of the constellation points of one symbol in the region of another will cause misinterpretation at the Rx. This increases when the SNR decreases and leads to an increase in the SER.

To minimise the SER, the weights of the Rx AEs (θ_{Rx}) are optimised. Similar to IPCL, the problem of de-noising the data on the Rx side is defined as minimising the MSE between the originally transmitted data (x) and final output on the Rx side ($\hat{y}$)) by restoring the values in both dimensions of the IQ space. This minimisation process in the whole communication system, formulated in Equations(4.4, 4.5), should be fast because of the stochastic time-varying nature of the wireless communication channel and the different effects of signal fading and shadowing.

$$\min_{(\theta_{Rx},I)} \quad L(\theta_{Rx,I}) = \frac{1}{Ep_{ext}} \sum_{t=0}^{t=Ep_{indT}} (\hat{y_{I,t}} - x_{I,t})^2 \tag{4.4}$$

$$\min_{(\theta_{Rx},Q)} \quad L(\theta_{Rx,Q}) = \frac{1}{Ep_{ext}} \sum_{t=0}^{t=Ep_{indT}} (\hat{y_{Q,t}} - x_{Q,t})^2 \tag{4.5}$$

It is assumed that the constellation points are equally likely. The scaling factor for normalising the average energy of the transmitted symbols to 1 is $\sqrt{\frac{E_s}{M}}$, where M is the number of bits per symbol. For the 4-QAM's $M = 2$ *bits/symbol*, then the scaling factor is $\sqrt{\frac{E_s}{2}}$, where $E_s = A^2$. Since $\sigma^2 = N_o/2, \mu = 0$, it is assumed that the additive noise $\mathcal{N}(\mu, \sigma)$ follows the gaussian probability distribution function given that the probability of a random value is calculated as follows $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-(x-\mu)^2}{2\sigma^2}) = \frac{1}{\sqrt{\pi N_o}} \exp(\frac{-x^2}{N_o})$

The probability that the symbol s is decoded correctly only if y falls in the area in the none overlapping region is $p(correct|s) = p(y_Q > 0|s) \times p(y_I > 0|s) = (1 - \frac{1}{\sqrt{\pi N_o}} \int\limits_{-\infty}^{0} e^{-(\frac{y_Q - (\frac{\sqrt{E_S}}{2})^2}{N_o})} dy_Q \times (1 - \frac{1}{\sqrt{\pi N_o}} \int\limits_{-\infty}^{0} e^{-(\frac{y_I - (\frac{\sqrt{E_S}}{2})^2}{N_o})} dy_I = (1 - \frac{erfc(\sqrt{E_s/2N_o})}{2})^2$

The symbol is in error if at least one of its bits is decoded incorrectly. The probability of symbol error is $1 - p(correct|s) = 1 - (1 - \frac{erfc(\sqrt{E_s/2N_o})}{2})^2 = erfc(\sqrt{E_s/2N_o}) - \frac{1}{4} erfc^2(\sqrt{E_s/2N_o})$

This scaling factor is also the minimum distance between every two symbols in a 4-QAM system with a quadrature modulator, channel and quadrature demodulator. Then, using the union bound of the probability of a symbol error, the SER at the Rx under the effect of AWGN is modelled as $SER_{4-QAM} = erfc(\sqrt{E_s/2N_o}) - \frac{1}{4} erfc^2(\sqrt{E_s/2N_o})$, where $erfc(x) = \frac{2}{\sqrt{\pi}} \times \int\limits_{-\infty}^{0} e^{-(x)^2} dx$

The weights of the Rx AEs (θ_{Rx}) are optimised to minimise the SER. Similar to IPCL, the problem of de-noising the data at the Rx side is defined as minimising the MSE between the originally transmitted data (x) and the final output at the Rx side ($\hat{y}$) by restoring the values in both dimensions so that the imaginary and real I/Q signals are reconstructed. This minimisation process in the entire communication system for each of the I and Q components is modelled as in Equations (4.4, 4.5). The learning process should be fast to match the stochastic time-varying nature of the wireless communication channel with different effects of signal fading and shadowing.

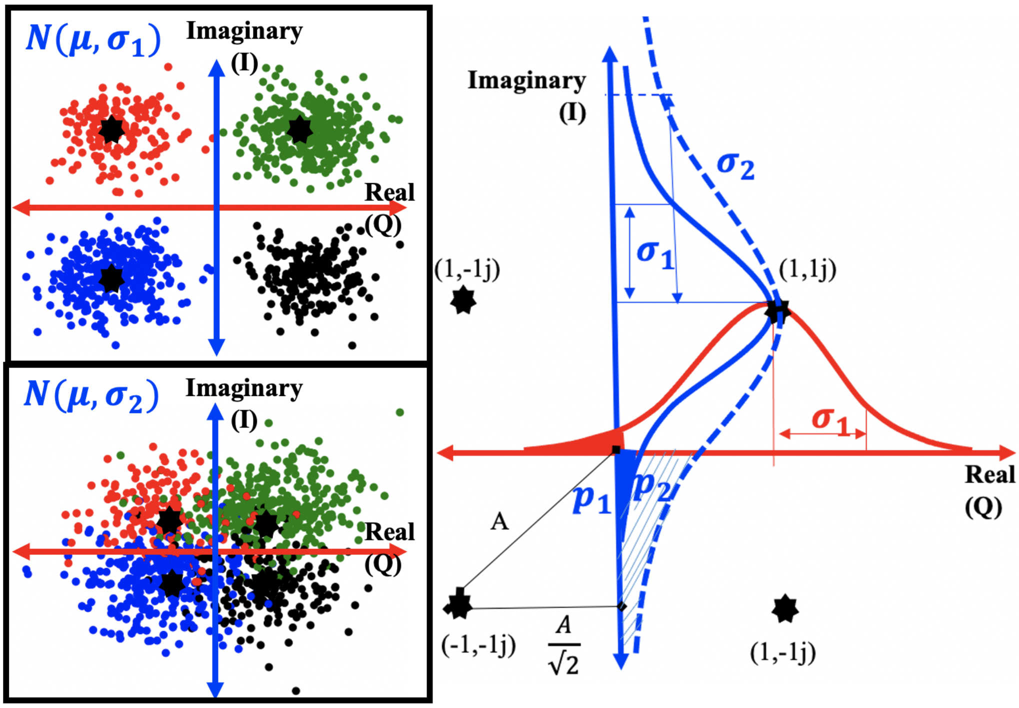

Figure 4.4: The effect of noise levels on the data modulated using 4-QAM

4.4 Cooperative Learning Method

In this section, the IPCL framework described includes (a) the design of the NNs at the end-systems which is based on the success of AEs for noise reduction; and (b) a two-phase sequential learning method inspired by the deployment of DL methods in the communication process [1] is carried out at the NNs without adding assumptions about the communication channels.

4.4.1 The Independent Learning of End-systems

Based on the observations of the policy-based method in [1], the learning process takes a long time to stabilise, especially when the internal noise is high, regardless of the CH noise level. Combining this observation and the recent improvements in deploying NNs for noise reduction [262], a pre-training phase performed by the Tx and Rx individually in a completely independent fashion is proposed. This is because, as the Tx and Rx are in different places, allowing them to overcome their internal noise individually facilitates their capability to initialise their communication processes and resolve the noise introduced by their communication channels in a short time.

In this phase, the input for the Tx is a set of randomly generated independent and identically distributed (i.i.d.) data messages (x), each of which is a vector of size B, and the output is the data to be sent through the channel ($\hat{x}$). The aim is to find the NN parameters that minimise the MSE between x and $\hat{x}$, which is defined in Equation (4.1). The same process is performed independently at the Rx side, where the loss function is Equation (4.2). The procedure in Algorithm 6 enables the Tx and Rx to independently map the noisy inputs to the actual ones by finding the optimal NN parameters (θ^*_{Tx} and θ^*_{Rx}) during the training time (Ep_{indT}).

This learning phase facilitates cooperative learning between the Tx and the Rx resulting in fast convergence, which minimises the time required in the initialisation of the communication process. AAEs are proposed for the Tx and Rx because of their

Algorithm 6: IndependentTraining($P_{i \in N^t}$)

Input : data
Output : θ
$t = 1$
while $t \in 1, 2, .., Ep_{indT}$ **do**
    Generate Batch size of i.i.d. data
    Calculate $L(\theta)$
    Optimise the θ
    increment(t)
end

efficient noise reduction in different domains [262], as shown in the left sub-figure in Figure 4.5. A simple AE is designed at both ends of the communication system, with an Adam optimiser [263, 264] used to search for the optimal NN parameters that lead to the minimum loss during the limited independent training time (Ep_{indT}). This optimiser has an adaptive step size that follows the update rule described in Equation (4.6) [263]

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{\upsilon}_t} + \epsilon}\hat{m}_t \tag{4.6}$$

where θ_t is the vector of the NN parameters at the previous time step, such that $\theta \in \mathbb{R}^{\mathrm{d}}$, η is the learning rate, $\epsilon = 10^{-8}$ a smoothing term that avoids division by zero and $\hat{m}_t$ and $\hat{\upsilon}_t$ are the bias-corrected first and second moment estimates and are calculated by Equations (4.7, 4.8), respectively.

$$\hat{m}_t = \frac{\beta_1 m_{t-1} + (1-\beta_1)g_t}{1-\beta_1} \tag{4.7}$$

$$\hat{\upsilon}_t = \frac{\beta_2 \upsilon_{t-1} + (1-\beta_2)g_t^2}{1-\beta_2} \tag{4.8}$$

where g_t is the gradient of the objective function with respect to θ, while m_t and υ_t are estimates of the first moment (the mean) and the second moment (the uncentred variance) of the gradients, respectively, and $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are the decay rates. For simplicity, the AEs of the Tx and Rx are identically structured. Each has a relatively low depth and a small number of neurons in each layer, with the number of neurons a function of the number of channels used, as in the NN design proposed in [1]]. This number is set to four which establishes the number of neurons in the NN model, as per the policy-based method in [1]. The differences between the proposed AEs and the NN design used in [1] are illustrated on the left- and right-hand sides in Figure 4.5, respectively, with the times taken by both methods for training shown in the performance analyses in Section 4.7.

4.4.2 End-systems Collaborative Learning

The second learning phase is the collaborative learning phase which enables the Tx and the Rx to overcome the CH noise. Note that initially trained end-systems (Tx and Rx) contribute effectively to the communication process reducing the time needed for the collaborative learning phase to result in a relatively low level of error.

This phase includes the transmission of data along the communication channel

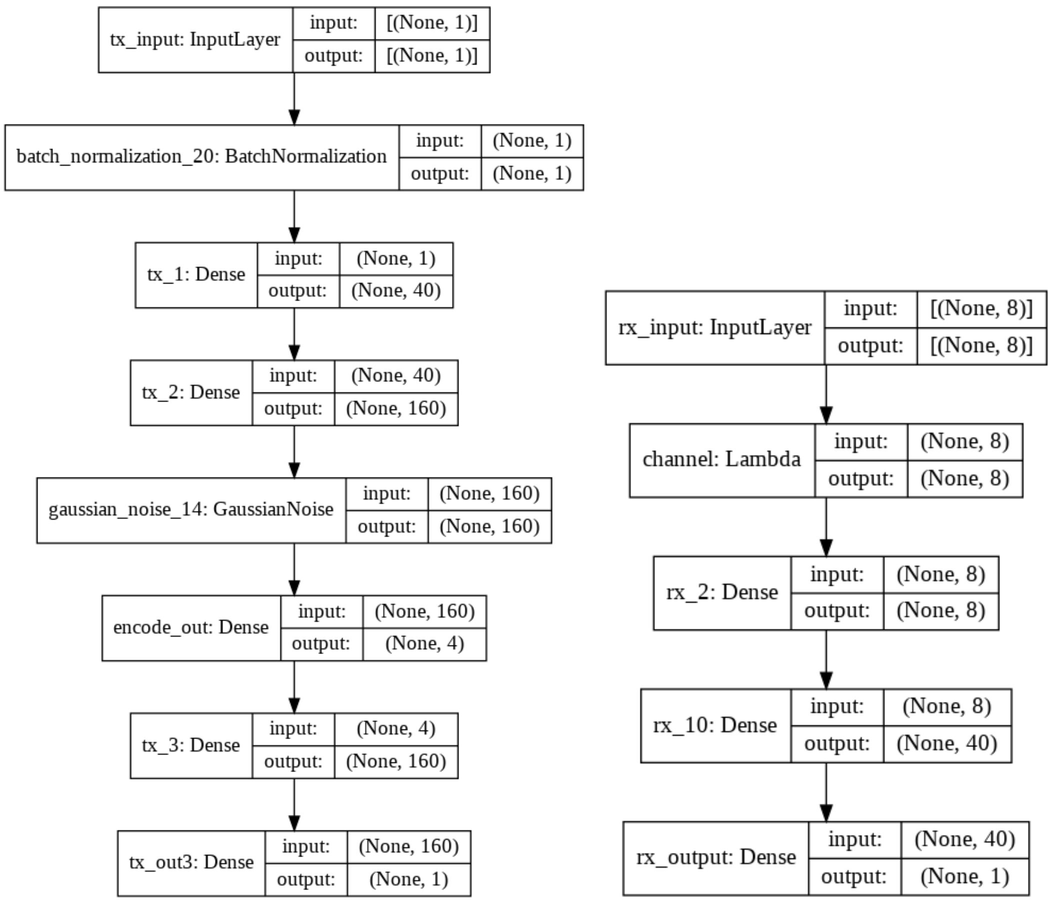

Figure 4.5: The proposed Autoencoder model for the Tx which is similar to the one for the Rx (left) vs the NN design in [1] for the Rx

where the seeds of the data generators are the same in both ends. This assumption was first introduced in [1] to make sure that the training of Tx and Rx is independent of the CH function, and it was justified by the ability to use the same random number generators at both end-systems (Algorithm 7, step 3). In that case, both the Tx and Rx have the same numbers and start communicating them, so the Tx updates its θ_{Tx} as it receives the estimated data at the Rx (z) and the Rx updates its θ_{Rx} as it transmits the data x generated by the Tx (Algorithm 7, steps 4 to 9) as described in Figure 4.2. This means that in the presence of F-CH, the loss functions at the Tx and Rx at each time step during the collaborative training time ($t \forall t \in \mathbb{N} = [1, 2, .., Ep_{collT}]$) are modelled as in Equations (4.9, 4.10), respectively.

$$\min_{\theta_{Tx}} \quad L(\theta_{Tx}) = \frac{1}{Ep_{collT}} [t = 0]t = Ep_{indT} \sum (z_t - x_t)^2 \tag{4.9}$$

$$\min_{\theta_{Rx}} \quad L(\theta_{Rx}) = \frac{1}{Ep_{collT}} [t = 0]t = Ep_{indT} \sum (\hat{y}_t - x_t)^2 \tag{4.10}$$

In the proposed collaborative training algorithm, relying on the assumption that Rx is able to generate the same random numbers during that phase is eliminated, which means that the Rx will not contribute to the training process. Thus, the loss function is only modified at the Tx, as in Equation (4.9), while the Rx will have its parameters modified as performed in the previous phase (the independent learning phase), as in Equation (4.2). The removal of this assumption makes our algorithm suited also to the difficult situation where Rx is unable to cooperate. This variant of IPCL is called IPCL-no-Rx to highlight that the Rx is not contributing in the collaborative training phase.

Similarly, the presence of a feedback channel is not guaranteed; the elimination of assumption for feedback channel is addressed by modifying Tx's loss function to be similar to its independent training process in Equation (4.1) while the Rx will keep using its collaborative training function in Equation (4.10). This variant of IPCL is called IPCL-no-FCH. Moreover, a highly challenging scenario was also considered in our design, where neither the Rx updates its parameters nor does the

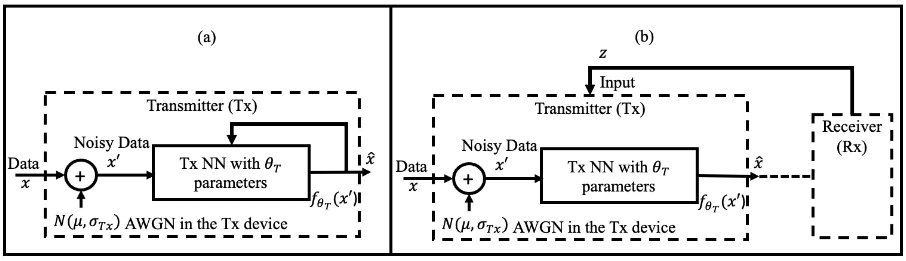

Figure 4.6: Tx training process (a) independently, and (b) collaboratively

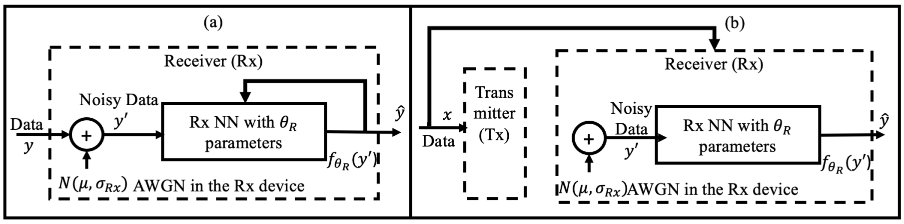

Figure 4.7: Rx training process (a) independently, and (b) collaboratively

Tx receive data from the Rx through F-CH. This version of IPCL is referred to as IPCL-no-Rx-FCH. This case represents the absence of collaborative training where the time taken in the collaborative training Ep_{collT} represents an extension to the independent training because the training functions for the Tx and Rx as modelled as Equations (4.1 and 4.2), respectively.

The updates in the NNT and NNR in the first and second training phases of IPCL are shown in Figures (4.6 and 4.7), respectively, where sub-figure (a) shows the individual training and sub-figure (b) shows the collaborative one. The main differences between these two phases is the ability of each NNT to adapt to what is interpreted at the output of the Rx (see Figure 4.6) and the ability of NNR to be trained on the same data generated at the Tx (see Figure 4.7).

4.5 Learning The Constellation

This section describes the proposed approach to allow the receiver to independently learn the changes that the channel adds to the modulated symbols x_I and x_Q in a short training time. The AEs at the Rx for the two-dimensional data x_I and x_Q

Algorithm 7: CollaborativeTraining($x, Epochs_{collTrain}$)

Input : x, Ep_{colT}
Output : $\theta^*_{Tx}, \theta^*_{Rx}$
$t = 1$
while $t \in 1, 2, .., Ep_{colT}$ **do**
  generate x_t then $x' = x_t + \mathcal{N}(\mu, \sigma_{Tx})$ ▷ **At the Tx**
  $x_t + \mathcal{N}(\mu, \sigma_{Tx}) \xrightarrow{NNT} \hat{x}_t$
  update θ_{Tx}
  $\hat{x}_t \xrightarrow{CH} y_t$
  ▷ **At the Rx**
  if *Rx can use the seed in Tx* **then**
    generate y_t from the same seed
  else
    generate y_t from a different seed
  $y' = y_t + \mathcal{N}(\mu, \sigma_{Rx}) \xrightarrow{NNR} \hat{y}_t$
  $\hat{y}_t \xrightarrow{F-CH} z_t$
  update θ_{Rx}
  ▷ **At the Tx**
  if *F-CH exists* **then**
    Train the NNT to map the output of the Rx to the actual input (input:z_t output:x_t)
  else
    Train the NNT to map its output to the actual input (input:$\hat{x}_t$ output:x_t)
  increment(t)
end
Calculate MSE using Equation 4.3 where $Ep_{indT} \leftarrow Ep_{colT}$
$\theta^*_{Tx} \leftarrow \theta_{Tx}$ and $\theta^*_{Rx} \leftarrow \theta_{Rx}$

are designed as the AE described in IPCL . However, the use of AEs at the Rx to learn constellation points of I/Q signals by learning the signal in each dimension independently to improve the efficiency of the communication system and minimise the SER after a short training time.

4.5.1 Learning Phase

The proposed IPCL-NF-2AE and IPCL-NF-1AE have two AEs and one AE, respectively. In IPCL-NF-2AE, each AE is trained on filtering out noise from one dimension of the complex numbers that result from modulating the symbols using 4-QAM at the Tx. While in IPCL-NF-1AE, the AE is being trained on both dimensions of this data to remove the effect of the channel noise from it. Training AE/s at the Rx to remove the effect of channel noise from the results of the modulated data has a noticeable effect on removing noise in a short training time. The steps for the Rx training can be summarised in Algorithm 8.

Thanks to the use of AE in the Rx that has high efficiency in noise reduction in different domains [262] where the Adam optimiser [263, 264] is used to search for the optimal parameters that lead to the minimum loss during a limited training time. The time spent on training on the same message is referred to as the number of internal epochs (Ep_{ind}). In this training process, the same transmitted message is used to train the AE with Adam optimiser that has an adaptive step size following the update rule in [263]. It states that $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + v}}\hat{m}_t$, where θ_t is the vector of the AE parameters at the previous time step; such that $\theta \in \mathbb{R}^{\mathrm{d}}$, η is the learning rate, $\epsilon = 10^{-8}$ is a smoothing term that avoids division by zero, and $\hat{m}_t$ and $\hat{v}t$ are the bias-corrected first and second moment estimates that are calculated by Equations (4.7 and 4.8), respectively.

Algorithm 8: TrainingRx($P_{i \in N^t}$)

Input : data
Output : θ

$t = 1$
for $t \in 1, 2, .., Ep_{ext}$ **do**
  Generate Batch size (B) of i.i.d. data
  modulate the data at the Tx
  send the modulated data on the channel as symbols
  $t_{in} = 1$
  for $t_{in} \in 1, 2, .., Ep_{ind}$ **do**
    Calculate $L(\theta, I)$ for x_I and the received y_I and $L(\theta, Q)$ for x_Q and the received y_Q
    Optimise θ_I, and θ_Q
    Demodulate the data at the Rx $\hat{y}_I$ and $\hat{y}_Q$
    increment(t_{in})
  end
end

In the training phase, the input for the Tx is a set of randomly generated independent and identically distributed (iid) messages of data x of B symbols that are sent as one message. As 4-QAM modulation is used, these symbols are mapped to their corresponding complex numbers of x_I imaginary parts and x_Q real parts. The set of x_I and x_Q are sent through the channel that is corrupted with AWGN symmetrically on both dimensions.

The seeds of the data generators are the same at the Tx and Rx, as assumed in [1]. This allows the AE or AEs in the Rx in IPCL-NF-1AE or IPCL-NF-2AE to be trained on data that Rx initially has. Thus, an error-checking method is not required in the training phase as it will be trained on a corrupted version of this data by the noisy communication channel from the Rx. Thus, the two AEs in the IPCL-NF-2AE have the x_I and x_Q and their corrupted versions from the channel y_I and y_Q throughout the external training epochs, as in Algorithm 8 (steps 2 to 5). If only one AE is used in the Rx, it will be changing its parameters for x_I and x_Q sequentially.

The aim of this training phase is to find the AE's parameters $\theta_{Rx,I}$ and $\theta_{Rx,Q}$ that minimise the mean squared error (MSE) between $\hat{y}_I$ and x_I and $\hat{y}_Q$ and x_Q, respectively as in Equation (4.3). This happens during the Ep_{ext} as in steps 8 and 9 in Algorithm 8. However, there is no strict need to achieve $MSE = 0$ because $\hat{y}_I$ and $\hat{y}_Q$ will be demodulated to generate the symbols $\hat{y}$, see Algorithm 8 step 10. These symbols can be recovered correctly if the MSE is within the range of the decision boundary between symbols ($MSE < \sqrt{E_s}/2$).

4.6 Hybrid Receiver Model For Shepherding

In Chapter 3, shepherding methods for solving the problem of a sheep's low sensing range within the flock. For the shepherding system as a dynamic system, a communication link between the sheepdog and the CU as the decision-maker is proposed. The SOHP and swarm optimal modified centroid push (SOMCP) introduced in Chapter 3 work under assumptions of an ideal communication channel. In the

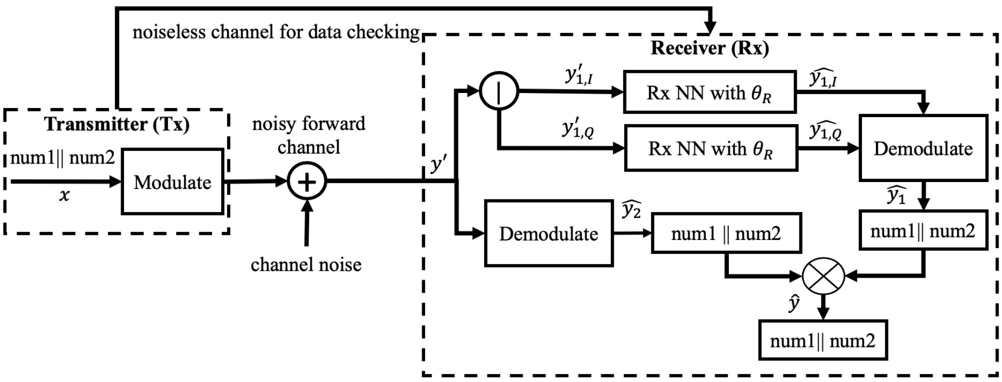

Figure 4.8: The proposed receiver hybrid model

SOHP, the CU sends the new headings to the sheepdog while, in the SOMCP, it sends a path for the sheepdog to follow. Despite sending short data without requiring high data rates, SOHP is not designed to consider the effects of channel fading and AWGN on the transmitted data. To improve the performance of the sheepdog in interpreting the herding points received under the effects of channel fading and noise that lead to a time-varying SNR, a hybrid demodulation model that combines the use of a 4-QAM demodulator and the trained Rx in IPCL-NF-2AE to decrease the SER in a larger range of SNRs is proposed.

The performance of the IPCL-NF-2AE based on the model in Figure 4.2 is improved at the Rx to suit the scenarios with time-varying SNR levels. Since 4-QAM performance is plausibly in the $SNR > 5$ range, training the Rx at lower SNR levels for a short time ensures a low SER for a larger SNR range when combined with 4-QAM demodulation. Therefore, combining the trained IPCL-NF-2AE with 4-QAM demodulator increases the range of the SNR when the SER is small. This hybrid model, denoted as IPCL-NF+4QAM, is illustrated in Figure 4.8.

An error-checking message transmitted over an ideal communication channel from the Tx may enable the Rx to adapt to the time-varying channel model in a dynamic problem like autonomous shepherding. The capability of the Rx to guarantee the accuracy of its data is critical in dynamic system applications to ensure the physical safety of the actuator. Moreover, the AE in the Rx can be trained online over the channel model using the data from successful transmissions. This improves the levels of efficiency of the IPCL-NF-2AE+4QAM and IPCL-NF-1AE+4QAM when working on channels with low SNRs that change throughout the task time.

In the proposed hybrid model, the 4-QAM demodulator generates a set of symbols ($\hat{y_1}$ and $\hat{y_2}$) to be checked for errors at the Rx. Then, if either $\hat{y_1}$ or $\hat{y_2}$ will be considered by the Rx if any of them is error-free, this data is used to retrain the Rx's AE. However, if both $\hat{y_1}$ and $\hat{y_2}$ are not error-free, this leads to an error that can not be recovered and requires re-transmission of the data. Therefore, if both $\hat{y_1}$ and $\hat{y_2}$ are incorrectly interpreted, the shepherd (Rx) stops. Since the CU can oversee the environment, it observes the stationary condition of the shepherd and re-transmits the herding point so that the shepherding task can be completed.

4.7 Results

In this section, an analysis of the IPCL framework, descriptions of the testing scenarios and the results obtained from evaluating the proposed approach are discussed.

This section describes the testing scenarios and the results obtained on evaluating the proposed approach.

4.7.1 Training and Testing Performances

At different SNRs, the bit error rates (BERs) and SERs of the policy method in [1] are compared with those of the proposed trained Rx with two AEs, each of which is trained and tested in one dimension. The proposed method uses the same AE in training and testing in both the I and Q dimensions of the constellation as well as 4-QAM [256]. All the methods that require training are trained on batches (messages) of 1000 i.i.d. randomly generated numbers converted to symbols through quantisation for $Ep_{ext} = 10$ with $Ep_{ind} = 100$. Then, they are tested using 10^5 i.i.d symbols and corrupted with AWGN.

,

4.7.2 Experimental Setup

The work in [1], denoted as a "policy" method, is divided into seven methods, that is, three of its variations, IPCL and three of IPCL's variations. These allow both the methods in the literature and our proposed framework with different system assumptions to be tested to investigate the effect of each of their components on their capability to minimise the error in their estimations of the message at the Rx. The methods used in our comparisons as well as the components changed in each one, accompanied by justifications of the changes, are summarised as follows:

- The "policy" method is an RL-based one proposed in [1] with a perturbation variance equals to 10^{-4}.
- The "policy-trained" method is the policy method [1] with independent training phases on both end-systems (the Tx and Rx) in order to investigate the effects of the independent training phase on the current design.
- The "policy-AE-Tx" method is the "policy-trained" one but uses our proposed AE as the Tx NN in order to investigate its effect on an asymmetric NN design at the end-systems in the absence of an independent training phase.
- The "policy-AE-trained-Tx" method is the "policy-AE-Tx" one but the independent training phase is deployed on both the Tx and Rx in order to investigate the effect of the independent training phase on asymmetric end-systems.
- The proposed "IPCL" framework with both training phases and the assumptions in [1] includes the capability of the Tx to receive feedback from the Rx through the F-CH and the Rx to contribute to the collaborative training phase, with the gaussian noise variance in AE is 0.1.
- The "IPCL-no-FCH" method is the IPCL but without the availability of an F-CH during the collaborative learning phase. This leads to the Tx being unable

to receive data from the Rx and the effect of an F-CH to be investigated during the collaborative training phase.

- The "IPCL-no-FCH-Rx" method is the "IPCL-no-FCH" one but without the assumption that the Rx is capable of generating the same data as the Tx during the collaborative learning phase or contributing to the collaborative training process. This helps the investigation of the worst possible scenario in which the Tx cannot adapt to the Rx while the Rx is unable to contribute to the collaborative training phase which leads to further training for only the Tx.

- The "IPCL-no-Rx" method is a version of the IPCL one that includes the assumption that an F-CH exists but with no updates in the Rx in order to investigate the effect of the Rx's training during the collaborative training phase but without the capability of the NN to update its weights according to the data received at the Rx.

All the methods are tested on the same scenario for 5000 epochs, where a message of data (x)) is sent from the Tx to Rx in one epoch and, if an F-CH exists, a feedback message may pass from the Rx to Tx during the same epoch. Each message is a set of i.i.d. a randomly generated batch (B) of numbers ranging from zero to one. For simplicity, the Tx and Rx are assumed to be symmetric as are the forward and feedback channels. This setting allows us to analyse the effects of the NN design and training framework proposed for minimising the MSE at the Rx end in the presence of noise at the end-systems and channels. The system parameters are presented in Table 4.1.

To analyse the IPCL framework in terms of the number of epochs required for collaborative training sufficient to minimise the MSE (Equation (4.3)), testing scenarios with different numbers of epochs in the collaborative training phase are run. Note that the collaborative training time is the initialisation one for the Tx and Rx which represents the overhead in the communication process. The shorter that time,

Table 4.1: Communication System Assumptions

Parameter	Value
Epochs in independent training	100
Epochs in collaborative training	$[10, 100, 1000]$
Epochs in testing	5000
Batch size (B)	1024
SNR at the end-system	$SNR = [-4, -2, 0, 2, 4, 10^6]$
SNR at the channels	$SNR = [4, 10, 40, 10^6]$

the more adaptive the end-systems are to channel variations and the more efficient communication systems.

4.7.3 Comparative Analysis

Three different lengths of the collaborative training phase (10, 100 and 1000 epochs) are considered in order to investigate their effects on the capability of the Rx to reduce the effect of noise on the data transmitted by the Tx. The level of noise is defined by the SNR in decibel (dB) units, as defined in [265], and formulated in Equation (4.11). Both the communication CH/channel and end-systems are assumed to add AWGN to the transmitted data, as described in Section 4.3 and illustrated in Figure 4.2.

$$SNR(dB) = 10 \log_{10} \frac{P_{signal}}{P_{noise}} \tag{4.11}$$

where the signal and noise powers are P_{signal} and P_{noise}, respectively.

The analysis starts with tests of all the models when noises at the end-systems and channel are negligible ($SNR = 10^6$). The results show that the proposed AE converges to a small level of error at least five times faster than the NN model in [1] during the independent training phase, as shown in the top right sub-figure in Figure 4.9. Moreover, in the first 100 collaborative training epochs, the initially trained Tx variant of the policy method [1] (policy-trained) exhibits worse progress while its asymmetric variant (policy-AE-trained-Tx) seems to converge faster when the Tx is a pre-trained AE but no progress when independent training is not performed (policy-AE-Tx). However, through the 1000 epochs of the collaborative training time, policy-AE-trained-Tx degrades after 100 epochs and policy-AE-Tx displays no improvement until the end of 1000 epochs. It is worth to note that the behaviour of IPCL is identical to IPCL-no-FCH in that case.

However, over the 1000 epochs of the collaborative training time, the policy-trained method degrades after 100, presumably because it uses the loss to train the Tx during that phase which has very small values (in the case of a noiseless channel) compared with that of the actual data involved in the independent training phase. Policy-AE-Tx shows no improvement until the end of the 1000 epochs because, as the Tx and Rx have different designs, they face an instability problem in training due to their use of very small error values in learning and asymmetric designs of their NNs.

For IPCL-no-FCH, as the MSE is almost zero from the beginning to the end of the training phase because the Tx is not able to obtain any information from the Rx while the NNR is already trained to provide a small error independently, the collaborative training leads to neither an improvement nor degradation at either end. The other variations of IPCL show relatively fast convergence rates compared with those based on the policy method within 10 and 100 epochs. However, by the end of the long training time (1000 epochs), only policy-trained and policy-AE-Tx ones have high MSEs, as shown in Figure 4.9, subfigure (c).

The MSE of the communication system with noiseless channels is tested for 5000 epochs when the numbers of epochs in the collaborative training phase are 10, 100 and 1000. The results in Figure 4.10 show the average MSE$\pm$1standard deviation. When the end-systems are noiseless (Figure 4.10(a)), the MSE decreases and almost vanishes due to the long collaborative training phase (1000 epochs) for all the tested methods except the modified versions of the policy one [1]. The shortest training time (10 epochs) is sufficient for only the IPCL, IPCL-no-FCH and policy-AE-trained-Tx methods to achieve unnoticeable MSEs of 0.002, 0.002, and 0.007, respectively. The non-contribution at the Rx side means that the IPCL-no-Rx one has the worst performances in 10 and 100 epochs compared with those of all its peers. Moreover, the asymmetry of the policy-AE-trained-Tx leads to its instability which results in a higher MSE despite the increased training time. All the methods show improvement as the training time increases from 10 to 100 epochs with policy one showing the best level.

In Figure 4.10 (b) and (c), the MSEs are shown at different noise levels at the end-systems of $SNR = [-4, 4]$when the collaborative training time is 10 and 100 epochs, respectively. By comparing sub-figures (a) and (b), a noticeable improvement can be observed for all the variations of the policy method [1] as the training time increases. However, the policy-AE-trained-Tx exhibits unexpected behaviour by degrading as the training time increases and the noise level decreases to $SNR = [0, 2, 4]$. It is worth noting that IPCL and its versions show little improvement as the training time increases despite the IPCL and IPCL-no-FCH techniques having the lowest MSEs which suggests they converge quickly. The IPCL method generally performs better than the policy one [1]. This is somewhat expected as the latter was predominantly designed to overcome noise at the communication channel, not the end-systems.

Therefore, to fairly compare these methods, a testing scenario in which only the channels are noisy ($SNR = 40, 10, 4$) while the end-systems are noiseless to fairly compare the policy method [1] to IPCL, as in Figure 4.11. The policy method [1] outperforms all the others only when the collaborative training time is long (1000 epochs) in all cases of the channel noise reaching an 0.0005 error. However, after a shorter learning time (100 epochs), it performs better than all its versions, except the policy-trained one that is better for a high SNR ($SNR = 40$). This could be because the design objective of the policy method [1] was to solve low SNR cases. This proves that the performance of the policy method [1] is not guaranteed for all SNRs, especially in limited training times. In contrast, IPCL and IPCL-no-FCH perform almost the same in the three SNR channels, especially when the SNR is high, but both take a shorter time to converge to lower MSEs as the SNR decreases despite the IPCL's superior learning rate. Nevertheless, IPCL and all its versions cannot reach the very low MSE achieved by the policy method [1] when the SNR is relatively low (10 and 4). Moreover, IPCL-no-Rx seems to take a relatively longer time than all the IPCL variations to converge to a considerably low MSE. This is due to the uncooperative behaviour at the Rx which makes the Tx adapt to its own noise, channel noise and Rx's noise individually while IPCL-no-FCH-no-Rx seems to struggle to learn as the SNR decreases.

To compare the performances of all the methods in scenarios of a noisy channel as well as end-systems, all the models are tested on a channel noise of $SNR = 4$ and different end-system noise levels of $SNR = [-4, 4]$ during collaborative phases for 10 and 100 epochs. Figure 4.12 shows the training performances over the collaborative training time when the SNR at the end-systems is -4 (sub-figure (a)) and while the MSEs of each method tested on each SNR value at the end-systems when the training times are 10 and 100 epochs (sub-figures (b) and (c), respectively). The progress of the policy-AE-trained-Tx fluctuates over time while all the other variations of the policy method [1] converge relatively quickly, with policy-AE-Tx showing the highest convergence rate. In contrast, IPCL shows almost the same small error level throughout all the training times while IPCL-no-FCH-no-RX degrades.

The training performance when $SNR = -4$ at the end-systems (Figure 4.12(a)) affects the results of all the methods when tested after 10 epochs (Figure 4.12(b)) or after 100 epoch (Figure 4.12(c)). The IPCL and its versions show greater than 50% reductions in their MSEs compared with those of the policy method and all its versions, except the policy-AE-trained-Tx that improves noticeably as the SNRs at the end-systems increase (see Figure 4.12(b)). The improvements in the policy-AE-trained-Tx and policy-AE methods for 10 epochs as the SNRs at the end-systems increase is because of the capability of the AEs to overcome internal noise in a short time when the SNR is relatively high. IPCL and all its variations converge to relatively low MSEs in a very short time due to the use of AEs in the NN design and individual training phase. The long collaborative training phase (100 epochs) allows all the methods with symmetric NN designs at the end-systems to converge to relatively low MSEs (see Figure 4.12(c)) while the policy-AE-Tx and policy-AE-trained-Tx methods achieve better performances as the SNRs at the end-systems

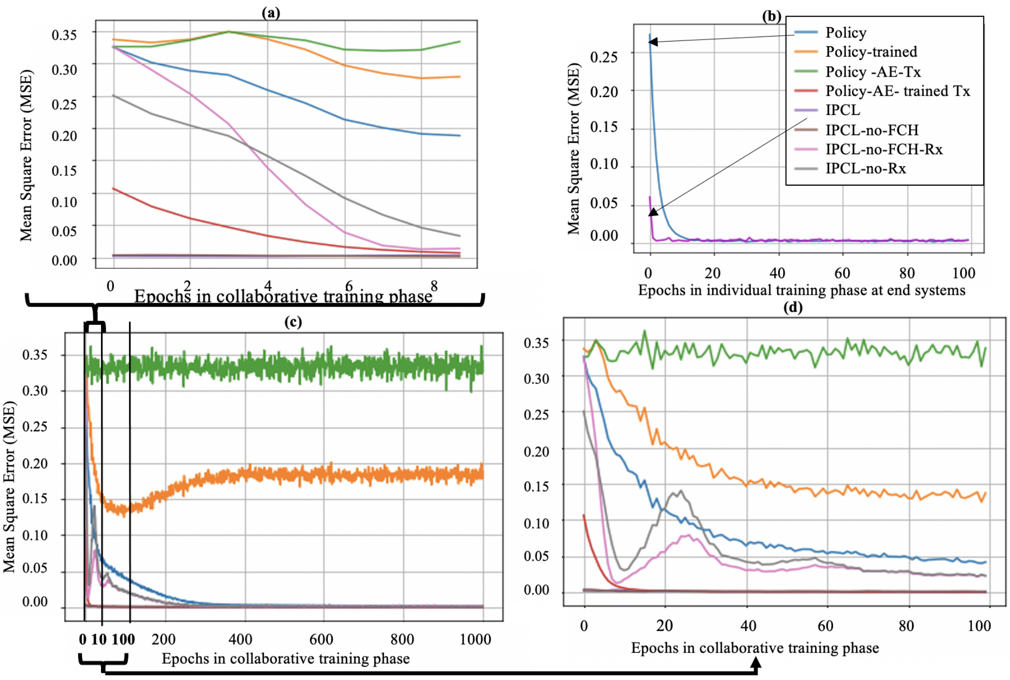

Figure 4.9: The MSE of noiseless end-systems over noiseless channels with collaborative training phase $Ep_{colT} = 1000$ in sub-figure (c), the progress in the first 10 and 100 epochs are shown in sub-figures (a) and (d), respectively, while sub-figure (b) shows the progress of the independent training at the end-systems using the NN model designed in [1] vs the proposed AE

increase.

4.7.4 Performance Analysis of Modulated Data

The progress of the training phase in terms of improving the variances of the symbols received at $SNR = 1$ throughout the training time for up to $Ep_{ext} = 100$ is depicted in Figure 4.13. It is clear that the 4-QAM has a high SER since a large portion of the symbols representing '0' are found in areas of the other symbols which occurs for all the other symbols. Moreover, the time taken by the policy method [1] and proposed Rx when 2 AEs are used is relatively too large to have a clear border between the different symbols in 2 dimensions. It can be noted that the proposed Rx model achieves an acceptable performance after the first 10 external epochs ($Ep_{ext} = 10$) and a superior one after $Ep_{ext} = 100$.

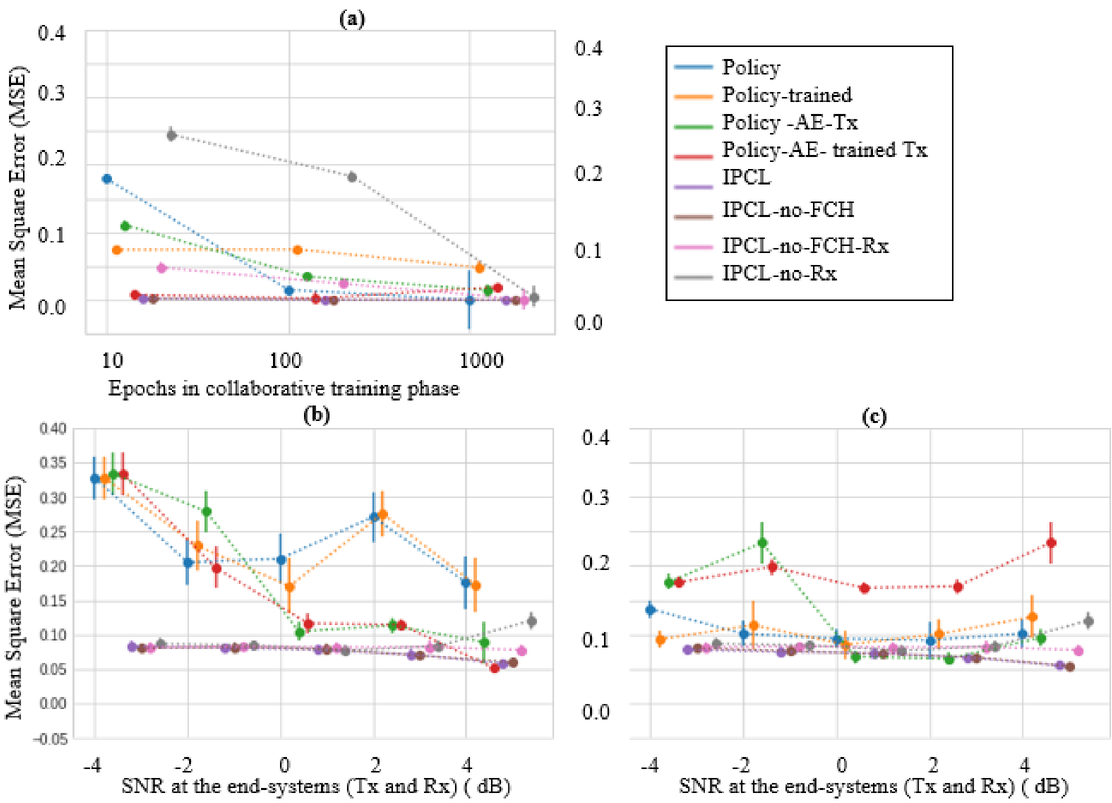

Figure 4.10: The MSE for noiseless channels and (a) noiseless end-systems trained at $Ep_{colT} = 10, 100, 1000$ and end-systems with different SNRs for (b)$Ep_{colT} = 10$ and (c)$Ep_{colT} = 100$

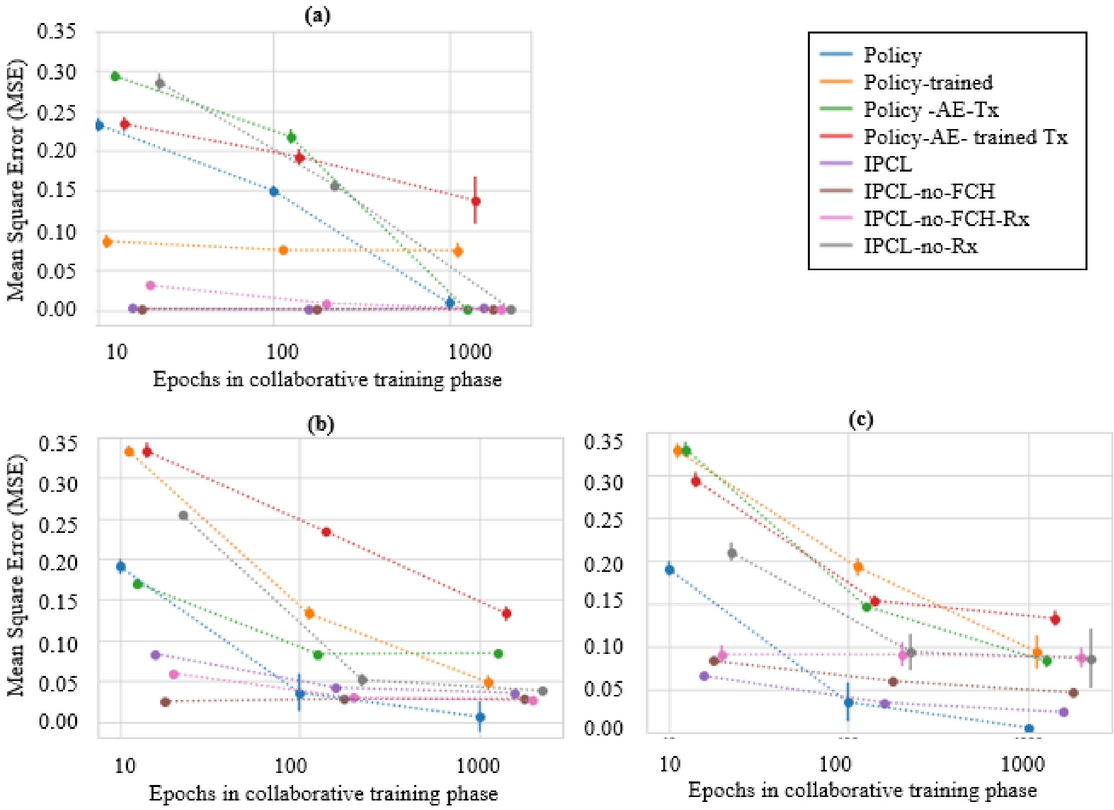

Figure 4.11: The MSE for noiseless end-systems trained collaboratively for $Ep_{colT} = [10, 100, 1000]$ and the channel noises are $SNR = [4, 10, 40]$

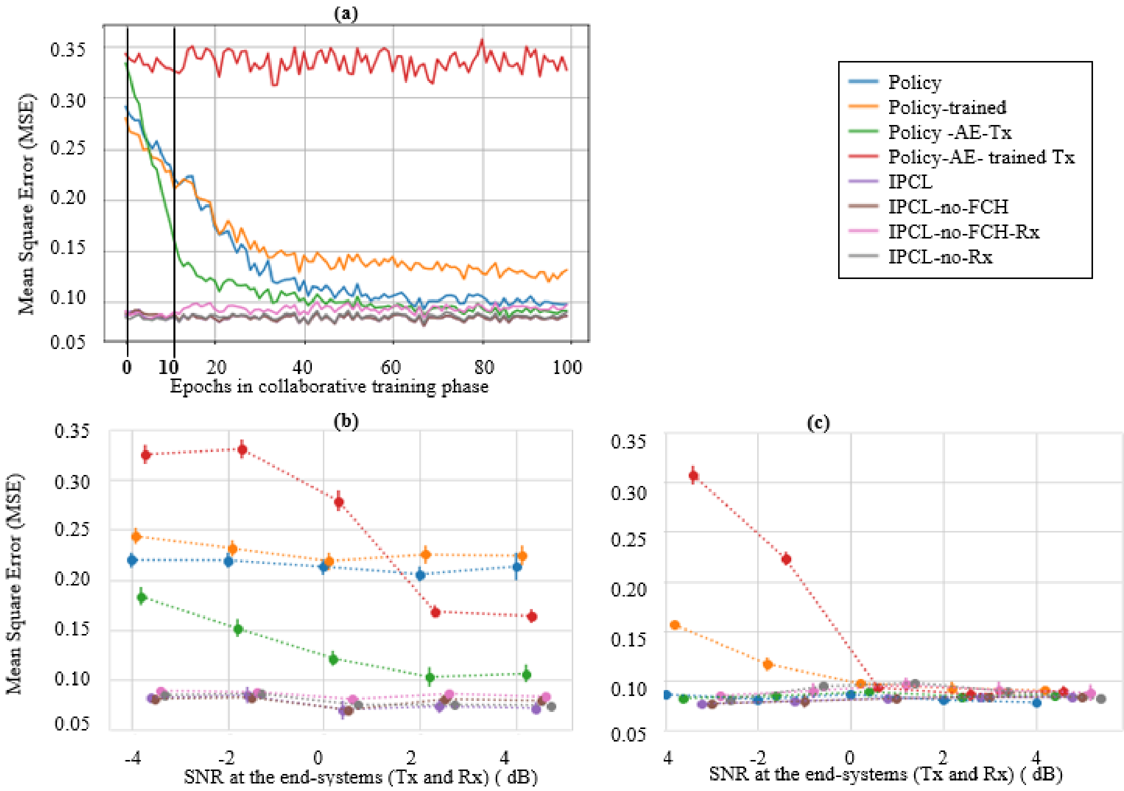

Figure 4.12: The MSE for channels with $SNR = 4$ (a) through collaborative training phase of 100 epochs where the horizontal line shows the MSE after the first 10 epochs, and for different SNRs at the end-systems after (b) 10 epochs and (c) 100 epochs of the collaborative training phase

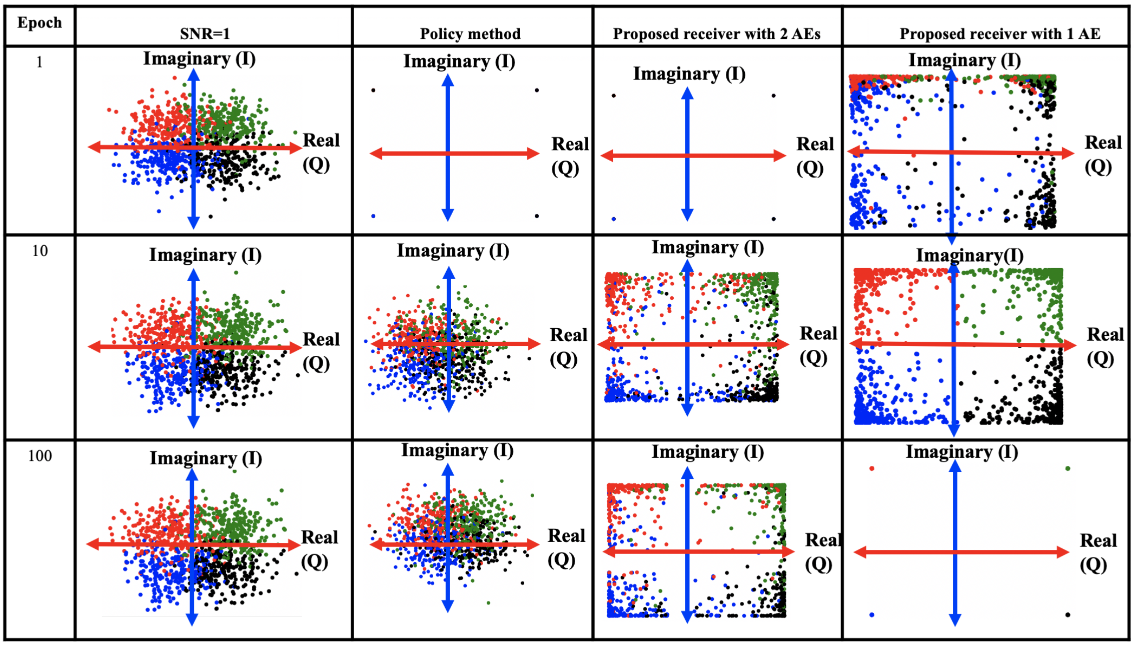

Figure 4.13: Training progress after 1,10, and 100 external epochs for the policy method [1], IPCL-NF-2AE, and IPCL-NF-1AE

The changes in the SERs of all the methods at $SNR = 1$ are tested. They are trained for $Ep_{ext} = 10$ of 1000 symbols after $Ep_{ext} = [10, 100]$ of training with increments of 10 epochs. This testing scenario is common for investigating the efficiency of NNs. It is designed as follows: (a) a message of size 1000 is generated using the same normal distribution of the data used in learning but at a different seed that is not repeated in either the training or testing epochs; (b) as the number of external epochs for testing is $Ep_{ext} = 100$, 10^5 symbols are used for testing; and (c) all the Rx are tested on the same data at the same time.

The SER values depicted in Figure 4.14 show that the IPCL-NF-1AE is capable of minimising the effect of noise after only $Ep_{ext} = 10$ in training to reach $SER = 0.056$ which is the smallest SER achieved by all the methods in that short training time. This is because the IPCL-NF-1AE uses one AE to train both the I and Q components which means that the AE is trained for 20 external epochs and is efficient only because the distributions of the data and signal noise are identical on both the I and Q components. That is why the IPCL-NF-2AE behaves similarly to the IPCL-NF-2AE but at a slower pace. On the other hand, as the policy method was designed

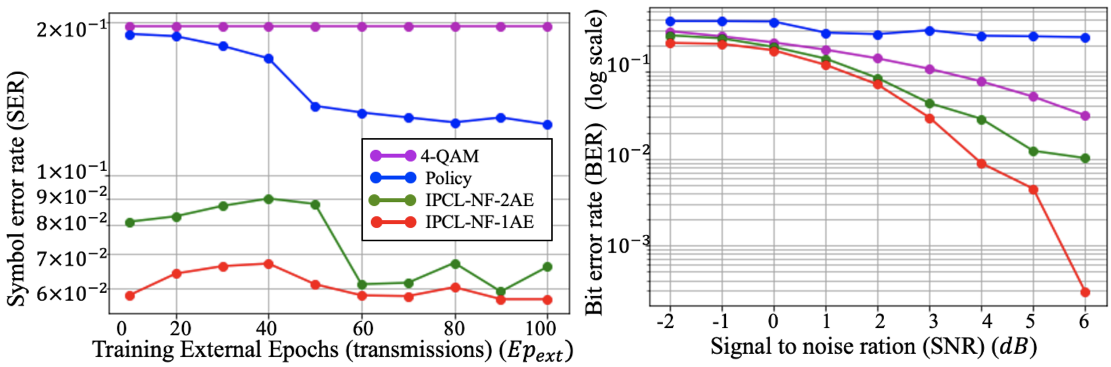

Figure 4.14: SER at SNR=1 over a range of training external epochs (left), and BER at different SNRs (right)

to be efficient after a long training time, it converges steadily while the IPCL-NF1AE and IPCL-NF-2AE do so quickly but take about 60 more training external epochs after the $10^{th}$to reach stability.

The BERs of the IPCL-NF-1AE and IPCL-NF-2AE are measured over a short training time with 4-QAM using the same testing scenario at challenging levels of noise ($SNR = [-2, 6]$). The results in Figure 4.14 show that our learning-based models perform better than 4-QAM in this range since they are trained on the same data distribution and the effect of noise for $Ep_{ext} = 10$ is $SNR = 1$ beforehand. It is clear that 4-QAM performs better than the policy method due to its short training time and the inability of the policy method [1] to generalise over a wide range of SNRs. However, the IPCL-NF-1AE reaches $BER = 10^{-3}$ when $SNR \geq 5$, which proves its capability to generalise after a short training time.

4.7.5 Performance Analysis Of Shepherding

A testing scenario that matches the message content in shepherding, where two integer values representing the shepherd's new locations in the x and y dimensions, respectively, are generated randomly and sent from the CU to the shepherd, is designed. Each of these values is represented by 5 symbols since the size of the shepherding area is shaped as a square with side lengths of $L = 300$ in the x and y dimensions. Then, to minimise the effect of the SER in time-varying SNR scenarios,

Message: (90,109)

x- location 90 → [0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0]

Interpreted by 4-QAM: [0 2 1 1 0 0 3 1 1 0 1 3 1 1 0 1 1 2 1 0 1 2 1 1 0 1 3 1 1 0 0 3 1 1 0 0 3 1 1 0 1 3 1 1 1 2 3 1 1 0]

Final Symbols using 4-QAM → [1, 3, 1, 1, 0] incorrect

Interpreted by IPCL-NF-1AE: [0 3 1 1 0 0 1 1 1 0 0 3 1 1 1 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0 0 3 3 1 0 0 3 1 1 0 0 3 1 1 0 0 3 1 1 0]

Final Symbols using IPCL-NF-1AE → [0, 3, 1, 1, 0] correct

Correct repetition in IPCL-NF-1AE symbols : 8/10, 10/10, 10/10, 8/10, 10/10

y- location 109 → [1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0 1 3 2 1 0]

Interpreted by 4-QAM: [1 3 1 1 0 1 3 1 2 0 1 3 2 1 0 1 3 2 1 0 2 3 2 1 0 1 1 2 1 0 1 2 2 1 2 1 3 2 2 0 1 3 2 1 0 1 3 2 1 0]

Final Symbols using 4-QAM → [1, 3, 2, 1, 0] correct

Interpreted by IPCL-NF-1AE: [0 1 1 1 0 1 3 1 0 0 1 3 2 1 0 1 3 2 1 0 2 2 2 1 0 1 1 2 1 0 1 2 2 0 0 1 3 2 2 0 0 3 2 1 0 1 3 2 1 0]

Final Symbols using IPCL-NF-1AE → [1, 3, 2, 1, 0] correct

Correct repetition in IPCL-NF-1AE : 6/10, 6/10, 8/10, 6/10, 10/10

Figure 4.15: An example for a message at SNR=1

making some symbols redundant/removing some redundant symbols is considered. Each value is repeated 10 times sequentially and, consequently, are its corresponding symbols. At the Rx, the symbols repeated for each number are counted and that with the highest number of repetitions is considered the correct one.

Transmissions of the values of 90 for the x-axis and 109 for the y-axis at $SNR = 1$ are illustrated in Figure 4.15, where the values are converted to five symbols ($[1, 0, 2, 1, 0]$ and $[1, 3, 2, 1, 0]$, respectively) which are repeated 10 times sequentially. Then, for each number, the IPCL-NF-1AE receives the symbols to remove their noise and determines the probability of each being repeated. The output of the IPCL-NF-1AE yields the original transmitted number since, for the x-axis location, the first and fourth symbols are repeated 8 times correctly and the rest 10 times correctly. It can also retrieve the values for the y-axis $[1, 3, 2, 1, 0]$ despite the lower repetition rate of the correct symbols. Nevertheless, the Rx with 4-QAM cannot recover the data for the x-axis correctly under the effect of this challenging SNR level but succeeds in finding the correct symbols for the values of the y-axis.

For the shepherding scenario, the shepherd moves to influence the sheep which affects the strength of the communication signal transmitted by the CU. Therefore, the SNR decreases as the shepherd moves further away from the CU. For simplicity regarding the communication model, a free-space one that is a function of the distance between the Tx and Rx to map the effect of the signal strength on the SNR is used, as defined in Friis' formula [128], $SNR(t) = \frac{E_s}{N_o} \times \frac{G_t G_r \lambda^2}{(4\pi d^2)}$, where $SNR(t)$is the SNR at the transmission time, d the distance between the shepherd and CU at that time, G_t and $G_r$the gains of the Tx and Rx, respectively, and λ the wavelength of the transmitted signal. This dynamic scenario may be effective for testing the performance of our proposed method on time-varying SNR levels since the shepherding task, a time-constrained dynamic one, demonstrates the applicability of the proposed method for dynamic system applications. In this scenario, the assumption of an/error-checking message is used so that the shepherd does not head in the incorrect direction unnecessarily. Therefore, using 4-QAM against that/instead of our hybrid Rx method (IPCL-NF-1E+4QAM) is tested to investigate the capability of the AE to learn online from successful transmissions. The shepherding scenarios involve $N = 50$ sheep initialised at three different densities characterised by k which is inversely proportional to the sheep's initial densities ($k = [1/2, 2/3, 3/4]$]). Each scenario is repeated for 25 sequential episodes, with the SOHP [used for shepherding with the same parameter in Table 4.1.

The home location is at $(0, 0)$ and the minimum initial sheep's distance to the home is one-fourth of the dimension of the shepherding area which is a square of $L =$ 300units. The sheep forces follow the motion model in Equation (2.2) with values for the different weights ($[W_{\pi\pi}, W_{\pi\Lambda}, W_{e\pi}, W_{\pi\epsilon}, W_{\pi\beta}] = [1.5, 1, 0.3, 0.5, 1.9]$)and ranges of sheep sensing ($[R_{\pi\pi}, R_{\pi\beta}, R_{\pi}] = [15, 70, 3]$). The sheep's velocities at grazing and during agitation are 0.05 and 2, respectively, while the maximum velocity of the shepherd is a 5-unit distance/unit time. The maximum task time is $T_{max} =$ 1200 steps where, at most, one step is taken by each agent in the environment and one transmission of the herding point may be sent by the CU over the corrupted communication channel followed by a shorter error-checking message sent over an ideal communication channel.

The example in Figure 4.15 shows that one location can be recovered correctly but the other cannot. This incorrect x-axis location may drastically affect the shepherd-

Table 4.2: Numerical Results For 50 Sheep

	Metric	Best Mean			Mean		
Metric	Model	4-QAM	IPCL-NF-1AE+4QAM	SOHP	4-QAM	IPCL-NF-1AE+4QAM	SOHP
SR	1/2	100	100	100	90	90	70
	2/3	100	100	100	40	65	83
	3/4	100	100	100	25	50	80
T	1/2	117	105	118	425 $\pm$ 369	533 $\pm$ 416	340 $\pm$ 182.7
	2/3	334	512	649	1022 $\pm$ 272	935 $\pm$ 244	459 $\pm$ 152
	3/4	539	563	370	1058 $\pm$ 255	1115 $\pm$ 165	548 $\pm$ 151
$s_T(\%)$	1/2	60	52	100	44 $\pm$ 22	46 $\pm$ 25	100 $\pm$ 0
	2/3	46	64	100	39 $\pm$ 19	60 $\pm$ 14	100 $\pm$ 0
	3/4	40	53	100	36 $\pm$ 18	52 $\pm$ 2	100 $\pm$ 0

ing task as the shepherd goes to the wrong location which may mislead the sheep and degrade the success rate (SR) of the shepherding task. However, the shepherd may stop when it receives incorrect locations if an error-checking message is sent through an ideal channel. In both cases, the shepherding task mandates that the shepherd receive an accurate location in both the x and y dimensions, whereby a transmission is considered successful. The following metrics are measured to prove the effectiveness of the IPCL-NF1E+4QAM in improving the shepherding task compared to the 4-QAM under the effect of a time-varying SNR and assumption of an ideal communication channel when the SOHP method is used for shepherding: (a) the success rate of the shepherding task (SR) is the percentage of successful episodes; (b) the task time (T), which is the average time required to finish each episode; and (c) the average percentage of successful transmissions S_T refers to the average ratio of successful to the total number of transmissions.

The assumption of ideal communication, referred to as SOHP, is the baseline for comparison with the effect of channel noise when using only the 4-QAM and when combining it with IPCL-NF-1AE as IPCL-NF-1AE+4QAM. In it, the AE in the Rx is trained for only $Ep_{ext} = 10$ on B symbols, each of which includes $Ep_{ind} = 100$ at $SNR = 3$. The range of the shepherd's mobility is directly proportional to the value of k due to the level of scattering in the flock's cohesion during the task time. Therefore, when $k = 3/4$, the average SNR is the lowest ($0.5\ dB$) due to the relatively large distances that the shepherd moves away from the CU. For $k = 2/3$, the average $SNR = 1.5\ dB$ and, for $k = 1/2$ the average ($SNR = 2.5\ dB$). For all cases, the range of SNR is $SNR = [-5, 20]\ dB$.

It is depicted in table4.2 that, overall, the IPCL-NF-1AE+4QAM method achieves higher success rates for transmissions (S_T) and, thus, a higher for the shepherding task (SR) than 4-QAM. When the density of sheep is low, the shepherd needs to go far away from the CU during its task which makes the SNR relatively low and decreases the S_T and SR. Using IPCL-NF-1AE+4QAM, the SR achieved is greater than that of the 4-QAM alone, being at least 65% of the SR achieved for shepherding under the assumption of an ideal communication CH/channel while 4-QAM leads to only half that ratio.

4.8 Chapter Summary

In dynamic systems where the communication CH/channel varies over time, adaptive Tx and Rx should be trained in the smallest possible number of transmissions to optimise their NN parameters and achieve low MSEs before exchanging messages. IPCL was introduced to address the problem of long initialisation times at the Tx and Rx in order to overcome noise at both end-systems and the communication channel. The noise model discussed in this chapter was the AWGN one that is common in both electronic devices and communication channels. The purpose of the IPCL framework for end-systems was to reduce the effect of noise on the data interpreted at the Rx measured as the MSE.

IPCL was tested in different communication system scenarios, including the absence of an F-CH, the inability of the Rx to cooperate in the collaborative learning phase and in both cases combined. It performed better than its peers without the need to model the channel in a very short learning time. The analyses of different communication scenarios showed that the IPCL converges to low error levels under the effect of an SNR as low as 4 in only 10% of the training time compared with state-of-the-art methods.

Then, a modified IPCL-NF-CH learning-based method for modulated data in an I/Qsignal was proposed and compared with 4-QAM. Despite the high resistance of 4-QAM to noise levels without prior training, the fast adaptation of the proposed method displayed superior performance in a relatively wide range of SNRs. The fast adaptation of the Rx in the absence of F-CHs and the presence of rapidly changing channel noise was studied.

Through a simulation using Python 3, in the case of symmetric AWGN, the proposed IPCL-NF-1AE using the same AE to remove the effect of channel noise from both dimensions reduced the BERs to the range of $[30\%, 1\%]$ of its values with $4-$QAM at $SNR = [-2, 6]$, respectively. Moreover, the BER achieved with IPCL-NF-1AE at $SNR = 0$ was comparable to that in recent work using a NN to tackle the noisy communication CH/channel problem at $SNR = 5$ [260]. When $SNR = 1$, the SER was less than 20% of its value with $4-$QAM regardless of the training time.

Finally, a combined hybrid model, in which the Rx was trained on only one SNR level and combined with 4-QAM to allow it to remove noise from a larger range of SNR levels, was proposed. However, this method was only valid when an error-checking message was sent to the Rx on the ideal channel. Its efficiency for shepherding tasks, whereby a CU sent locations to a shepherd every few time steps to effectively guide a swarm, was demonstrated. Even in the presence of a large range of SNRs, due to the large distances between the Tx and Rx, the proposed approach (IPCL-NF-2AE) enabled the shepherding performance to be nearly equivalent to that achieved under the assumption of an ideal communication channel.

Chapter 5

Shepherding under dynamic communications systems

The work reported in this chapter has been partially published in the following article:

Reem E Mohamed, Saber Elsayed, Robert Hunjet, Hussein Abbass (2022), *Reinforcement Learning for Solving Communication Problems in Shepherding.* IEEE Symposium Series On Computational Intelligence.

In this chapter, the aim is to improve shepherding in the presence of a fading communication channel with added noise by introducing a learning-based mobility model for the central unit (CU). The mobility of the CU helps to improve the success rate of transmissions to the shepherd by minimising the distance between the CU and the sheepdog during the transmission time. The system model and trade-off between the energy consumption of movements and the probability of success are discussed after the introduction. Then, the proposed mobility model is explained. Finally, the results and analysis of the technique are articulated.

5.1 Introduction

In accordance with the shepherding paradigm outlined in Chapter 3, the shepherd receives instructions from the CU, which possesses real-time, comprehensive knowledge about the sheep' locations. In Chapter 4, it is assumed that this NCS of

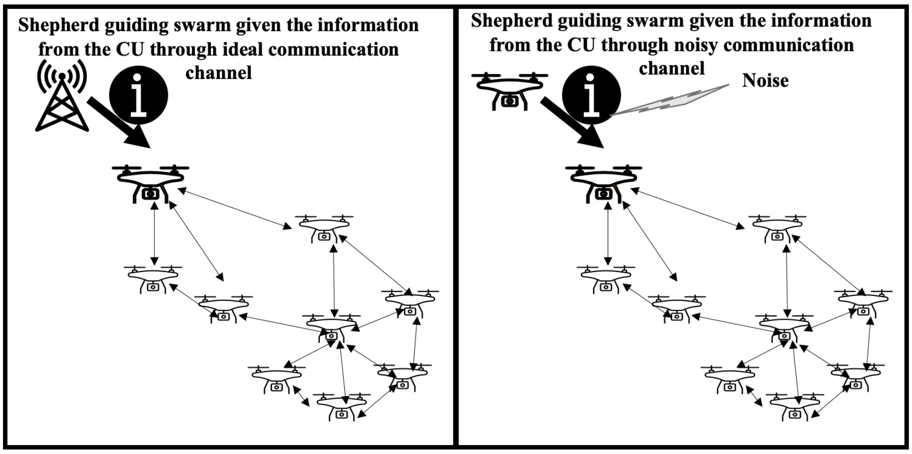

Figure 5.1: Comparison between the system model in Chapter 3 (left) and Chapter 5 (right)

controllers has only one mobile component to perform the herding task, that is, the shepherd, as shown in the left sub-figure in Figure 5.1. However, since shepherding may take place in distant locations, the strength of a transmitted signal may degrade over the duration of the task. This results in inaccurate interpretations of the headings the shepherd receives and exposes the system to unpredictability which may result in actuator failures or sheep misguidance. The mobility of the CU is introduced in this chapter to limit the effect of signal fading on the SR of transmitted data, as shown in the right sub-figure in Figure 5.1.

However, while most system models in the literature for shepherding methods [17, 18, 66] are shepherd-centric, the notion that the shepherd can simultaneously observe the sheep and approach them to exert influence may not be feasible. This is because influencing the sheep requires maintaining close proximity to them whereas obtaining a comprehensive observation demands a long-distance, unobstructed perspective [56]. The system modelled in Chapter 3 shows a communication system in which the CU monitors environmental changes and subsequently transmits the near-optimal herding locations obtained to the shepherd. However, the communication scenario is simplified by assuming a noiseless, low-latency link between the CU and herder to focus on shepherding.

These communication assumptions seem plausible because the CU is assumed to be stationary with a huge antenna and sufficient energy resources to withstand any distortion of wireless communications. However, depending on a huge transmitter antenna at the CU to ensure the efficacy of the communication link limits applying shepherding in distant places that may lack sufficient power supplies. Therefore, restricting the size of the CU to that of a dynamic system, such as a drone, makes the system more dynamic despite being more susceptible to the influence of the noise of wireless communication channels. The effect of free-space fading on a signal with added noise on the SR of the SOHP is studied as the shepherd guides the sheep through an obstacle-free environment.

A change in distance between the CU and the sheepdog (β) leads to a low transmission success rate of the herding points which cascades into degrading the SR of the shepherding task. This stimulates the incorporation of mobility into the CU in order to counteract the influence of noise on a weakened signal coming from the CU to a far β.

To build a distributed system model for a swarm guidance system working under realistic channel assumptions, the energy consumption and communication link between the shepherd (actuator) and CU (decision-maker) are studied. The right sub-figure in Figure 5.1 suggests that the swarm-guiding task may have three types of agents that can be described as follows.

- A CU, a mini-UAV that can fly to a medium altitude, as studied in [266]. This altitude allows a wider view of the surroundings than other agents in the system and less chance of occlusion.
- The actuator (shepherd) and a mini-UAV or -UGV with a lower cost, speed, size and weight than the CU to guide the swarm in restricted terrains. If it is above the ground, it flies at a low altitude, as studied in [266], which is effective for influencing the swarm by applying forces within its restricted sensing range and providing precise observations in crowded environments.
- A swarm of guided agents, UGVs/UAVs that are on, or close to, the ground

with small batteries and omni-directional sensors in the x and y dimensions with limited ranges and another sensor in the positive z dimension with a higher range for sensing a flying shepherd.

To limit the time of the shepherding task for a robot receiving guiding commands from a CU, the maximum is modelled as a function of $|\pi|$ and the initial density of the sheep, as in Chapter 3. Due to the energy restrictions of these agents, which may be modelled as $E = \int_{t=0}^{T} P(t)dt$, where $T \in \mathbb{N}$, a limited task time is critical for replacing the shepherd with a robot herding additional swarms of robots or sheep. Reducing the duration of activity reduces the total energy required to complete the task.

Maintaining a high received power at the β requires a mobile CU to track it throughout the task. The constrained proximity of the Tx to Rx may reduce signal fading, ensuring a high SNR and a higher S_T. Keeping the Tx and Rx on the same trajectory to enhance SNR hinders decentralisation and increases energy consumption at the CU. Due to its stochastic behaviour, the CU's mobility choice is limited by a trade-off between the S_T and the Tx-Rx distance threshold. The problem is expressed as an MDP [267, 268] by defining states and actions for the CU to handle this trade-off while recognising the unpredictability of the AWGN channel.

Reinforcement learning teaches an agent to do actions that provide the best rewards by adapting to its environment. Off-policy reinforcement learning techniques converge quicker than on-policy methods in communication resource allocation problems despite their instability [269]. Thus, Q-learning is used since it is a typical off-policy learning method with a dynamic nature that matches with the aim of letting the CU learn velocities in real-time with a minimal number of transmissions and without a training time. Q-learning allows the CU to modify its velocity to provide an adequate S_T at a low SNR without a training phase, unlike model-free methods [1] that need time and bandwidth for training.

A modified Q-learning technique for saving energy at the CU while maintaining the S_T over a variety of demanding SNR levels without the need for a training phase is presented. Then, an analysis is carried out on the ϵ-greedy approach for exploration in the Q-learning technique, in which the CU investigates random actions when a random number produced is smaller than ϵ or, otherwise, exploits the action based on the values in the Q-table. The contributions of this work are summarised as follows.

- Modelling the challenge of dynamic agents interacting through a stochastic channel as an MDP which results in using an incremental search (IS) method as a means of enhancing transmission success.
- Proposing two distinct reward functions for a ϵ-greedy Q-learning approach to ensure the success of swarm steering despite the random behaviour of the communication channel with low energy consumption and then using the IS method for Q-table initialisation.

5.2 Problem Formulation

In the first subsection in this section, the challenges of the forces involved in shepherding and their notations are discussed. In the second part, the trade-off between the requirements of the communication channel and energy use is elaborated in detail.

5.2.1 The Shepherding Problem

Shepherding tasks are a sub-class of swarm guidance ones because both are performed by a single agent (or many) serving as a sheepdog to guide a swarm of agents that resembles a flock of sheep being herded to its home location [59]. Enforcing the guiding agent to preserve swarm cohesion may minimise the task time. This is significant because the swarm is impacted by the sheepdog's repulsion force, with each individual sheep avoiding contact with the shepherd when it is inside its agitation range ($R_{\pi\beta}$). The swarm maintains collective motion by avoiding internal collisions, with each member repelling the others within its collision avoidance range or safe radius (R_π). Moreover, each swarm member is drawn to the local centres of mass (LCMs) of its neighbours within its range of view ($R_{\pi\pi}$) during the herding time, where $R_\pi < R_{\pi\pi} < R_{\pi\beta}$. Thus, the short $R_{\pi\pi}$ of each swarm member diminishes its capacity to maintain cohesiveness among the whole swarm, since various swarm members may estimate different LCMs from those of their neighbours when they flee from the sheepdog which causes them to move in different/opposite directions. The projected end direction of each swarm member at/in each time step is influenced by its inertia and jittering which may be modelled as AWGN. The resultant force vector of each sheep, while the shepherd is in its $R_{\pi\beta}$, is the weighted sum of the forces in Equation (2.2), as described in [59], where all the agents are considered particles with a unit mass.

In this chapter, T_{max} is used to refer to the maximum task time as described in Chapter 3, and $T \in \mathbb{N}$ to all the time steps performed inside the task time $[t] \triangleq 0, ..., t-1$. Thus, the CU transmits herding points to the shepherd at a predetermined number of time steps ($t_H = t_{h(i)}, t_{h(i+1)}, .., t_{h(f)} \Rightarrow t_{h(f)} \leq t-1 \Rightarrow t_H \subset T$) and the difference between each pair of subsequent herding point broadcasts ($t_{i,j}$) is dependent on the amount of time it takes the shepherd to walk from herding point i to the next on $(i+1)$. Therefore, $t_H = t_{h(i)}, t_{h(i+1)}, .., t_{h(f)} \Rightarrow t_{h(f)} \leq t-1 \Rightarrow t_H \subset T$ regardless of the number of herding points in the task. $t_{i,i+1} = alpha$ is stated as a function of the shepherd's maximum velocity (δ_β), as in Equation (5.1)

$$\alpha = \frac{d(P_\beta(t), P_\beta(t+\alpha))}{\delta_\beta} \tag{5.1}$$

The shepherd behaves as an actuator by directing the sheep to their home location as they move towards the herding locations assessed and transmitted by the CU as soon as they are received. In one time step, the herding point is sent by the CU and acknowledged by the shepherd over an ideal feedback channel. Therefore, the shepherd obtains the headings at successive time steps in t_H which are $t_{h(1)} + 1, t_{h(2)} + 1, .., t_{h(f)} + 1$; nevertheless, for convenience, t_H represents both transmission and receiving times.

5.2.2 Trade-off between Communication Efficiency and Energy Consumption

The communication link is presented as a time-varying one since the shepherd's position varies as it follows the new herding point supplied by the CU over the duration of the task. Changes in the quality of the communication connection influence the transmission success rate S_T and, by extension, the shepherding task's SR. This connection is applicable/susceptible to multiple communication models based on the nature of the surrounding environment, such as whether it is congested or free of obstructions. In an obstruction-free one, the transmitted herding point at time $t_h(i)$ from the CU to the shepherd experiences a free-space path loss proportionate to the distance between the Tx and Rx ($d_{CU-\beta}(t_{h(i)})$), as defined by the Friis formula in Equation (2.3) [128]

This is the simplest version of a communication model in which the distance between the transmitting and receiving stations is the sole element impacting the quality of their communication connection. As the distance between the Tx and Rx varies, this quality influences the ratio of the received to actual power sent. According to Equation (2.3), as the distance between the Tx and Rx increases, a lower proportion of the transmitted power is received. Consequently, the influence of noise at the receiver side becomes more pronounced than that on the transmitted power. This is because, as a form of a linear time-invariant (LTI) system, communication channels are susceptible to various types of noise. In this study, for the sake of simplicity, AWGN is used with a zero mean and variance to represent its power as added to the received power. If the received power is insufficient, the extra noise dramatically changes the sent signal. Accordingly, the shepherd obtains an altered herding point at $t_{h(i)}$from the CU. In other words, the ratio of the received to added noise power ($\gamma(t_{h(i)})$) in Equation (5.2) should be reasonably high to ensure the correctness of the sent data; otherwise, the data will be affected by the added noise.

$$\gamma(t_h(i)) = \frac{P_r(d_{CU-\beta}(t_{h(i)}))}{N_0(t_{h(i)})} \tag{5.2}$$

By substituting with Equations(5.2, (2.3)), it is observed that the increase in $d_{CU-\beta}(t_{h(i)})$ or ratio of the signal-to-noise power at the transmitter $\frac{P_t}{N_0(t_{h(i)})}$ increases the probability of error in the transmission, as in Equation (5.3)

$$\gamma(t_h(i)) = \frac{P_t}{N_0(t_h(i))} \times \frac{G_t G_r \lambda^2}{(4\pi d_{CU-\beta}(t_{h(i)}))^2} \tag{5.3}$$

Despite the fact that the CU transmits the herding points with a high SNR on the transmitter side ($\gamma_T = 10 \log \frac{P_t}{N_0}$), the SNR received may be low if $d_{CU-\beta}(t_{h(i)})$ is large. Considering the SNR at the Rx, in order to examine the influence of mobility on the accuracy of transmissions separately, it is assumed that γ_T is fixed.

As modelled by [1]], an error in transmission is caused by additive noise introduced to a transmitted signal as a random signal with a variance denoting the strength of the noise and a zero mean. As the distance between the Tx and Rx increases, the level of this noise rises relative to the signal power received, as estimated in Equation (5.3). The distance between the position of the real herding point ($h(i)$) and the received location ($h'(i)$) is denoted as $d(h(i), h'(i))$, where $d(.,.)$ is a function that calculates the Euclidean distance between its two inputs taking their dimensions into account. If $d(h(i), h'(i)) > err_{thresh}$, the CU re-transmits $h(i)$ to the shepherd in the next timestep which indicates that the shepherd has already made one step towards an erroneous herding point previously altered owing to a poor-quality link. Moreover, the error-free nature of this retransmission depends on the network's quality at the time of the retransmission. Thus, the shepherd may take further steps in the incorrect direction until one of the retransmissions is successful. It is important to remember that the longer the time spent in retransmitting, the longer the shepherd requires to finish the shepherding task effectively. Therefore, the SR declines dramatically when a high proportion of transmissions are received with considerable deviations from the herding point larger than the err_{thresh}.

As the CU transmits its herding cues, it may stay a short distance from the shepherd to prevent transmission errors. According to the free-space communication model, the process of following the shepherd may ensure the shepherd's capacity to receive the new herding point with no/little mistake and, hence, preserve the SR of the shepherding algorithm used. However, the energy required by the CU in motion is significantly more than that of the shepherd as it is a more powerful dynamic agent with a greater processing capability, is larger and has a greater mass. For simplicity, the energy consumptions of the shepherd and CU are modelled in two-dimensional space for a vehicle in Equations(5.4, 5.5) respectively

$$E_{\beta,tot} = \frac{m_\beta v_\beta^2}{2} + E_r \tag{5.4}$$

$$E_{CU,tot} = \frac{m_{CU} v_{CU}^2}{2} + E_t \tag{5.5}$$

where m_β and m_{CU} are the masses, and v_β and v_{CU} are the velocities for the shepherd and the CU, respectively, E_r and E_t the amounts of energy consumed by the shepherd and CU, respectively when receiving and transmitting the control packets. However, the energy consumed in communication modelled in [270] is negligible compared with that in moving particles with considerable masses.

5.3 The Mobility Problem As Markov Decision Process

Since $m_{CU} > m_\beta$, the CU expends a disproportionately large amount of energy by following the shepherd, despite the fact that the S_T may be attained. The CU's approach to the herding location may be controlled to prevent this excessive energy consumption. However, this technique may not be adequate to improve the quality of the communication link with the shepherd. In other words, if this distance remains constant at a particular value, the validity of this approach in varied situations cannot be guaranteed. Therefore, the distance the CU has to travel to reach the shepherd's herding location must vary according to the requirements of the shepherding task and the unknown quantity of added noise. This is because repeated retransmissions due to successive transmission failures raise the task time to an unexpected limit that may surpass the maximum task time specified, as in Equation (5.6).

$$T_{max} = k_1(d(H(0), H)/\delta_\pi + d(H(0), P_\beta^0)/\delta_\beta) + kR_{\pi\pi}N \tag{5.6}$$

where $d(H(0), P_\beta^0)$ represents the initial displacement between the shepherd's home position and the initial herding point, and $d(H(0), H)$ represents the initial displacement between the home location and the first herding point. And $k_1 > 1$ is a constant reflecting a tolerance in the shepherd's travel time at the maximum velocity (δ_β) and k a constant representing the initial density of the sheep, with its value increasing as this initial density decreases. In other words, the value of k is directly proportional to the complexity of the work and, therefore, the maximum task duration. The sensing range of a sheep relative to its peers is $R_{\pi\pi}$, where N is the total number of sheep.

For simplicity, the energy used by the CU to reach the current herding point during the transmission of a new one ($H(t+\alpha)$) is a function of its velocity Equation (5.5)which is the change in the CU's position over time (α) as in Equation (5.7)

$$v_{CU} = d(P_{CU}(t+\alpha), P_{CU}(t))/\alpha \tag{5.7}$$

If the CU is a flying drone, its total consumption of moving energy is equal to the sum of its kinetic and potential energies, with the latter constant at a constant height (h), gravitational force (g), and agent's mass (m). Therefore, its kinetic energy is the determining element in overall energy consumption, as described in Equation (5.8). For the sake of simplicity, it is assumed that the CU is a flying drone with a comprehensive view of its surroundings in real-time as it flies at a constant altitude.

Consequently, only the kinetic energy is used (Equation (5.9))

$$E_{total} = E_{KE} + mgh \tag{5.8}$$

$$E_{KE} = \frac{mv^2}{2} = \frac{m}{2}\frac{d(P_{CU}(t+\alpha), P_{CU}(t))^2}{\alpha^2} \tag{5.9}$$

This indicates that the CU should keep $v_{CU} = v_\beta$ during the duration of the task in order to maintain the shortest distance feasible throughout transmissions of the herding points. However, the quantity of additional noise is unpredictable which implies that the proximity of the CU to the shepherd is not required for transmission success.

The unknown likelihood of transmission success is addressed to model the problem as an MDP, with both transmissions and re-transmissions having unknown probabilities of success ($P(s, a)$). This is the uncertain output of the MDP and the energy required to move the CU is the known component. However, the distance between the Tx and Rx at the time of each transmission is a known determining factor in the chance of success (Equation (5.3)), in addition to being the primary factor in the CU's energy consumption (Equation (5.5)). Therefore, this distance is referred to as the state (s) of each transmission operation. To make these states discrete, natural values are used to describe the $d_{Cu-\beta}$ in each transmission attempt.

Then, a set of feasible destinations the CU may target within α time steps is proposed, such that each one represents a specific action the CU selects. The number of actions that can be conducted for each transmission is set to five $[a_0, a_1, a_2, a_3, a_4]$. As these actions are ordered ascendingly based on their distance from P_{CU}, a_0 corresponds to the nearest destination to the CU as it has minimal energy expenditure but a low probability of transmission success. In contrast, a_4 corresponds to a movement towards the shepherd's position which incurs the most energy by the CU but ensures a better likelihood of transmission success than all the other activities. The success or failure of subsequent transmissions influences the chance of completing the shepherding task. In other words, the rate of successful transmission procedures must be sufficient to guarantee the task's accomplishment within the allotted time frame.

5.4 Reinforcement Learning For Velocity Adaptation

The current herding point is evaluated only when the shepherd reaches the previously predicted one, according to the SOHP, since it was not intended to solve any communication issues. Since the shepherding problem is time-varying and the availability of training data is expensive due to the physical damage and power consumption of the controlling agents during their interaction with biological sheep or other robots, real-time learning can be an efficient method. The proposed method is integrated in SOHP to ensure that the suggested technique is applicable to any swarm-guiding task in which a transmitter provides headings to a receiver as an actuator in real-time. While the transmitter's mobility cost is quite high, the task must be accomplished promptly.

As noted, the proposed framework seeks to reduce the expenditure of energy when moving the CU and ensure a high (S_T), that retains the shepherding method's efficiency despite channel noise. This is achieved by (a) searching for the optimum action to be performed by the CU to guarantee a high SR in the shepherding task; (b) using Q-learning to enhance the search for the CU's near-optimal velocities; and (c) updating Q-learning to incorporate an IS to discover the CU's near-optimal velocities. All these stages are described in detail below.

5.4.1 Incremental Search

Due to the mobility of communicating end-systems in free space and the impact of noise on a broadcast signal, the wireless communication model for the dynamic system is time-varying. As a transmission failure indicates that the transmission's efficiency at the transmission time is inadequate to ensure its success, the CU must increase its velocity to move closer to the herding point. This guarantees the lowest possible energy usage and transmission error rate in the communication channel's worst-case scenario which produces a high SR for the shepherding task. This problem is modelled as an MDP to find the near-optimal CU speed that provides a high rate of successful transmissions while using the least amount of energy.

In an MDP model, the starting distance between the CU and shepherd during the first transmission attempt is indicated by the state variable (s) that represents the distance between the CU and β. The CU specifies this state after determining the shepherd's herding point. The straight line from the CU to the shepherd's new herding location ($P_\beta(t+\alpha)$) is divided into five equidistant points representing destinations for the CU. Each of the set of five CU actions corresponds to a heading to one of the destinations. The larger the action ($a = a_0, ..a_4$), the closer this destination point is to the herding one which requires greater velocity. In other words, a_0 corresponds to the lowest velocity of the CU and a_5 relates to the most significant velocity among all potential actions in this state. Because of the short distance to β, there is a high probability of transmission success if the CU selects a5 which may result in completing the shepherding task. On the other hand, a movement consumes more energy since the CU spends the same time performing any of the chosen actions. The following are simplified definitions of the activities in each state:

- a_0 indicates that the CU should proceed to the first point on the straight line connecting the CU to the shepherd's new herding point ($P_\beta(t+\alpha)$) within α time steps;
- a_1 indicates that the CU should proceed to the second point within $alpha/2$ time steps;
- a_2 specifies that the CU must arrive at the third point on the straight line connecting the CU to the shepherd's new herding point ($P_\beta(t+\alpha)$) within $alpha/3$ time steps;

- $a3$ specifies that the CU must arrive at the fourth point on the straight line within $\alpha/4$ time steps; and

- a_4 means that the CU should travel to the shepherd's location ($P_\beta(t)$) within $\alpha/5$ time steps.

These actions are chosen progressively in terms of energy expenditure to sustain the CU's energy as it attempts alternative actions for each state. In other words, at each step, the CU selects a value that expends the least amount of energy in motion by travelling at the slowest feasible speed. If the transmission fails after α time steps, the CU picks up a_0 and retransmits the same message to β every time step, increasing its velocity so that it reaches a_1 in $\alpha/2$ time steps. Each of the five actions in Algorithm 9 is repeated until the transmission succeeds.

To maintain the energy of the CU as it tries different actions for each state, the actions are selected incrementally in terms of energy consumption. In other words, at each stage, the CU chooses a_0, so that it consumes the least amount of energy by moving at the lowest possible velocity. When the transmission fails after α time steps, the CU chooses a_1 and keeps re-transmitting the same message to the β every time step as it increases its velocity to reach the corresponding destination to a_1 in $\alpha/2$ time steps. This procedure is carried out for all five actions until the transmission succeeds (Algorithm 9).

5.4.2 Q-learning For Mobility

According to the problem description in Section 5.2, due to the unpredictability of the communication channel, it is not possible to provide an evaluation function for each state. Therefore, off-policy reinforcement learning is more successful than policy-based reinforcement learning for CU mobility problems regarding action selection. Solving the MDP via tabular model-free reinforcement learning, that is, Q-learning may be more effective than IS for selecting an action in each state, given that the state representing the distance between the CU and shepherd when trans-

Algorithm 9: Incremental Search(P_{CU})

Input : sheep information, method for finding herding point, $P_{cu}(t)$,$P_{\beta}(t)$

Output : the velocity of the CU for transmitting the new herding point to the shepherd ($v_{cu}(t, t+\alpha)$)

$t = 0$
while *task not ended* **do**
    $i = 0$, $retx = 0$
    $H(t+\alpha) \leftarrow$ method(sheep information)
    given $P_{\beta}(t), H(t+\alpha)$, calculate the time α Equation (5.1)
    while $i \leq length(a) \& (retx == 0 || err > err_{thresh})$ **do**
        select a_i
        set $P_{cu}(t+\alpha)$ to a point within $d(P_{CU}, H(t+\alpha))/length(a)$ from the new herding point $H(t+\alpha)$ on the straight line linking P_{CU} and $H(t+\alpha)$
        Calculate $v_{cu}(t, t+\alpha)$ for the selected action
        if $d(P_{CU}, H(t+\alpha)) \leq d(P_{CU}, H(t+\alpha))/length(a)$ **then**
            send $H(t+\alpha)$ to β
            Receive acknowledgement from β
            Calculate the difference between the received herding point and the actual one $d(h(i), h'(i))$
            if $d(h(i), h'(i)) > err_{thresh}$ **then**
                increment(retx)
                if $i < length(a)$ **then**
                    increment(i)
    end
end

mitting may have been visited earlier in that episode or a previous one. This facilitates selecting an action corresponding to altering the CU's velocity to a value that enables it to be positioned at a location that is highly likely to achieve transmission success.

In Q-learning, each action in each stage is assigned a Q-value. The shepherd evaluates a transmission's performance by executing an action in each state and modifying these actions according to the rewards received, with the Q-value indicating the potential success of the future activity. A Q-table with actions as columns and states as rows is populated with the Q-values learnt during episodes of the shepherding scenario via transmissions. This table enhances the learning experience for transmissions repeated in the same condition. It also allows the CU to provide values for assessing the activity depending on the energy needed to carry it out, assuming a successful transmission. In the ϵ-greedy variation of Q-learning, the selection of an action begins with the generation of a random value, if it is less than ϵ, a random action is chosen for the given state regardless of the Q-values. Also, the action with the highest Q-value is calculated based on the previously earned reward and chosen according to Equation (5.10)

$$Q_t(s,a) = (1 - lr)Q_{t-1}(s,a) + lr(R(s,a) + \gamma^{max}Q(\prime s, \prime a)) \tag{5.10}$$

where Q_t and Q_{t-1} represent the current and previous Q-values, respectively, and lr and γ represent the learning rate and discount reward, respectively, with their range of values $[0, 1]$. The discount reward is the proportional weight of a future benefit in comparison with the current reward. That for performing action a in state s is $R(s, a)$, whereas the highest Q-value projected after discounting is $\gamma^{max}Q(s', a')$. The optimal action-value function may be expressed as a function of the agent's rewards (r) for choosing a certain action (a) in each state (s), as in Equation (5.11),

$$Q_*(a) = \mathbb{E}[R_t|A_t = a]\forall a \in a_0, a_1, a_2, a_3, a_4 = \sum_R P(r|a)r \tag{5.11}$$

The reward ($R(s,a)$) indicates the action's effectiveness based on its capability to transmit effectively while conserving the CU's energy. Since transmission success is a binary enumeration, the $R(s,a)$ for the Tx may be zero if the transmission fails ($Tx = 0$) and greater than zero if it succeeds ($Tx = 1$). A reward function in which the success of transmission is proposed. A reward function in which transmission success is proposed is divided by the square of the distance travelled by the CU prior to this transmission (Equation (5.12)) and referred to as Q1. Another reward function, in which the value of a transmission error is multiplied by the number of transmission errors as a function of the Euclidean distance between the actual herding point and the received one is proposed. It represents the effect of noise in successful transmissions, as shown in Equation (5.13), and is called Q2.

$$R(s,a) = \frac{Tx}{v_{CU}(t+\alpha)^2} \tag{5.12}$$

$$R(s,a) = \frac{Tx}{v_{CU}(t+\alpha)^2 + (d(h,h') + 1)} \tag{5.13}$$

Despite the adaptive nature of the ϵ-greedy technique, in the first episode, as the CU has no experience picking all the potential actions in every state, it selects actions randomly to assign Q-values after using the rewards function. This may result in a poor learning rate, particularly if the number of broadcasts in each episode is limited. Motivated by the IS approach presented in Algorithm 9, a Q-table with values that emphasise the activities with the lowest energy usage is presented. Action a_0 has the greatest Q-value compared with those of all the others (steps 8 and 9 in Algorithm 10) and the CU is more likely to choose activities with the lowest energy consumption. The notation for the initial Q-values/value per action is $q_{init,i}$, where i is the action number and, consequently, $a_i \forall i = [0,4], i \in N$, as explained in steps 8 to 18. The reward is then determined depending on the approach used, that is, Q1 or Q2. The Q-values of the chosen activities are updated based on the success of the transmissions and the energy expended by the CU, as detailed in steps 15 to 19 in Algorithm 10. The Q-table is then preserved for use in future episodes, as detailed in steps 22 and 23 in Algorithm 10.

Algorithm 10: Q-learning for velocity adaptation

Input : sheep information, method for finding herding point, $P_{cu}(t)$,$P_{\beta}(t)$, maximum number of episodes *episodes*

Output : $v_{cu}(t, t+\alpha)$

```
1  t = 0
2  episode = 1
3  while episode ≤ episodes do
4      while episode not ended do
5          retx = 0
6          while new H(t) ||retx > 0 do
7              generate random number p
8              if episode == 1 then
9                  set initial Q-values with q_{init,i} ∀i ∈ a
10             if p < ϵ then
11                 a ← random action
12             else
13                 a ← argmax(Q(s))
14             v_cu(t, t + α), d(h, h'), retx ←Alg. 9 (step 8:17)
15             if  method Q1 is used then
16                 calculate the reward using Equation (5.12)
17             else
18                 calculate the reward using Equation (5.13)
19             update Q-value for the selected action as in Equation (5.10)
20         end
21         save the Q-table
22         increment episode
23     end
24 end
```

5.5 Results

In this section, the influence of stochastic channel noise on the SOHP for shepherding tasks is analysed by evaluating the mean ±1 standard deviation and best mean over 25 episodes for various metrics in Tables 5.2,5.3, respectively.

The SNR range at the receiver is $\gamma(d) = [-15, 36]$ dB and, when the CU is stationary, is increased to $\gamma(d) = [5, 36]$ dB by moving the CU to maintain a limited distance from the β, as modelled in Equation (5.3). These problematic SNR levels result from using high-frequency waves in transmission which suffer drastically in an open environment as the distance between the Tx and Rx increases during the shepherding task. For this task, the flock is randomly initialised at three distinct densities, as indicated by the density factors (k) and simulation parameters shown in Table 5.1.

Under the assumptions of the presence of channel noise ($\mathcal{N}$) and one transmission per time step, ten techniques introduced in the SOHP are compared. The following is a summary of these methods: (a) the SOHP refers to the absence of any technique for CU mobility, the stationary condition of the CU and the noiselessness of the channel; (b) SOHP-N is the designation for the SOHP in the presence of noise; (c) IS' refers to the IS approach developed in Algorithm 9; (d) Q1 refers to moving the CU using Q-learning, where the reward function is Equation (5.12), (e) Q1-IS1, Q1-IS2, and Q1-IS3 are Q1 with starting values $q(a)_1$, $q(a)_2$, and $q(a)_3$, respectively, for the Q-table; (f) Q2 refers to moving the CU using Q-learning, where the reward function is Equation (5.13);(g) Q2-IS1, Q2-IS2, and Q2-IS3 refer to the usage of Q2 with starting values of $q(a)_1$, $q(a)_2$, and $q(a)_3$, respectively, for the Q-table.

The influence of two reward functions in Equations(5.12, 5.13) is investigated by analysing the capability of each design to determine the trajectory of the CU's velocities modelled as actions that maximises the received rewards, as in Equation (5.14)

$$\arg\max_a Q_*(a) \tag{5.14}$$

The shepherding task's SR, which is the proportion/ratio of successful to total number of episodes, task time (T)and percentage of sheep reaching home at the end of the task time ($N_{\%}$),is assessed. Then, the metrics related to transmission performances, such as the average value of rewards (R) obtained by the CU using the suggested reward functions in Equations(5.12, 5.13) are analysed. These functions describe the reward value (R) as a function of the CU's velocity (v_{CU})which is proportional to its energy consumption. Its average velocity is measured because if it is the same as the shepherd's, transmission success is assured owing to the lack of any free-space path loss. The lower the average CU velocity relative to the shepherd's maximum speed the better. The percentage of successful transmissions (S_T) and cumulative changes in reward values over all 25 episodes of the task are evaluated to illustrate how transmission success is assured and improved despite the CU's low velocity, as in Figure 5.2.

The influence of all the strategies on the shepherding task's SR success rate are compared. In Table 5.2, the minimum of the average T and the maximum of the transmission S_T are evaluated. Due to the CU's mobility , the range of the SNR at the Rx achieves an average of $\gamma = [5, 36]$ for the various suggested approaches, thereby improving the S_T. Nonetheless, the high error threshold ($err_{thresh} = 35$) results in the re-transmissions of certain faulty transmissions being disregarded. However, these erroneous herding sites have a modest impact on the overall SR of the shepherding task compared with those of the ideal communication channel scenario in Table 5.2 for the situations examined in this study. Furthermore, the average velocity of the shepherd is between 4 and $5m/s$, which is more than the one for the CU (v_{CU}), as shown in Table 5.2.

In Figure 5.2, from left to right, the capability of the learning approach to improve throughout the course of transmission attempts for three distinct initial sheep densities of $k = 1/2, 2/3, 3/4$ is demonstrated. The reward functions in the SOHP-N and IS (Equation (5.13)) are used to assess the success of each attempt, whether it is a new one or a retransmission of a failed one, in order to compare the various suggested approaches. This enables the use of a single metric to compare Q1 and Q2 and their variations for the second reward as Q2 and its versions usually perform better in low-density initialisation scenarios. In all circumstances, the CU in the SOHP does not acquire knowledge over time since its/the line expands during all transmissions. Due to the numerous failures of herding point broadcasts, the maximum number of transmissions is completed in these 25 episodes. This is shown by the shortest slopes of all the presented approaches in all circumstances. While Q1-IS1 may be regarded as the most progressive learner throughout the tasks, it requires numerous retransmissions to perform successfully which necessitates a greater number of transmissions throughout the 25 episodes than the IS, other versions of Q1 and all those of Q2.

As shown in Table 5.2, comparing the approaches for the shepherding task for $k = 1/2$, Q1-IS3 is the most efficient, yielding $SR = 88\%$. It is superior to Q2 which is followed by Q2-IS3. According to Figure 5.2, Q1-IS3 is better than Q1-IS2 because it achieves its peak sooner as does Q2-IS3 which makes it far superior to Q2-IS2. Similarly, for $k = 2/3$, Q1-IS3 has the highest SR of $SR = 72\%$, followed by Q1-IS2, with Q2-IS3 having the highest of the Q2 variations of $SR = 68\%$, as shown in Table 5.2. According to Figure 5.2, Q1-IS3 is superior to Q1-IS2 because it reaches its peak sooner which indicates a more progressive learning process given the slope of each line. Q2-IS3 is the most progressive of the Q2 learning methods, with Q2-IS1 superior to Q2-IS2 since it achieves its peak a little sooner. Nonetheless, Q2-IS2 performs better than the other Q2 variations during the first 100 transmissions of the challenge. Therefore, the IS model performs somewhat better than the other Q2 ones.

In contrast, for $k = 3/4$, Q2-IS3 has the highest overall SR, as shown in Table 5.2, although Q1-IS2 is the best of the Q1 variants. According to Figure 5.2,, Q1-IS3 is superior to Q1 because of its more progressive learning process which enables it to achieve its peak sooner as does Q2-IS1 which makes it superior to Q2 and Q2-IS2. However, in the first 100 transmissions, Q1-IS2 and Q2-IS3 are equally progressive, although Q1-IS2 is the most progressive learner. Q2-IS3 is the best of its versions, followed by Q2, Q2-IS2 and Q2-IS1. Q1-IS3 and Q1-IS2 are the optimal variations of Q1 for high- and low- density initialisation conditions for the sheep, respectively.

The best T values in Table 5.3.also support these outcomes but are not good indicators of the best versions of Q2 and Q2-IS1 have the best task times for these conditions, respectively while Table 5.2 indicates that Q2 and Q2-IS3 are often the best variations of Q2, there are exceptions. Due to the instability of the Q-learning approach, even Q1-IS3, Q1-IS2, Q2 and Q2-IS3 are characterised by significant standard deviations in T and $N_{\%}$.

Moreover, according to Figure 5.2, the IS provides the largest commutative reward compared to all the Q2 versions, mainly when the task duration is quite long. This is because of the high retransmission rate which increases the likelihood of obtaining rewards at the lowest energy cost. It indicates that the commutative reward value is insufficient to determine the performances of the techniques on the shepherding task. However, in addition to the slopes, the total number of transmissions may be a reliable measure of the shepherding task's duration and SR. When $k = 3/4$, the IS has a higher SR than all the Q versions.

5.6 Chapter Summary

In this chapter, the shepherding problem was studied from a communication perspective in which a CU (as the transmitter) continuously communicates with a moving shepherd (as the receiver) that guides a flock of sheep to their home location. Its main contribution was to maximise the probability of success of transmission in the presence of channel noise by finding a near-optimal velocity for the CU subject to the constraint on its energy consumption (due to constraints on its mobility and task time).

To this end, the problem of maximising the SR of transmissions at the minimum possible velocity was modelled as an MDP. To solve it, the IS method for selecting the CU's velocity for each state action was introduced. It was used to initialise the Q-table for two distinct Q-learning-based techniques, namely, Q1 and Q2, for each of which a different reward function was defined. From the results, the following conclusions can be drawn: (a) if the Q-values of the actions are large enough and the difference between them significantly high, the performances of both the Q1 and Q2 versions will be satisfactory; and (b) the limited number of transmissions in the shepherding task shows the instability of the Q-learning method.

Table 5.1: System Assumptions

Parameter	Value
Signal: γ_T, λ, G_t, G_r	80 dB,60 mm,15,1
modulation: quadrature amplitude modulation	$QAM4$ [256]
Shepherding task: Number of sheep N	[50]
Environment: length L, Density factor k	300, [1/2,2/3,3/4]
Shepherd: initial location P^0_β, maximum velocity δ_β	(L, L), 5
Home: Location$H$, radiusR_H	(0,0),50
Minimum initial sheep distance to H	$L/4$
Sheep: velocity in grazing, maximum velocity δ_π	0.05,2
Sheep ranges: sensing $R_{\pi\pi}$, agitation$R_{\pi\beta}$, collision avoidanceR_π	15,70,3
weights: Collision $W_{\pi\pi}$, Cohesion $W_{\pi\Lambda}$, Jittering $W_{e\pi}$, Inertia $W_{\pi\epsilon}$, Agitation $W_{\pi\beta}$	1.5,1,0.3, 0.5, 1.9
CU:initial location, initial v, mass, maximum v	(L, L), 3 m/s,5 kg,20 m/s
PSO:c1,c2,w,k_{iter}	1.5,2,0.3,21
DBSCAN: υ_{DB},ϵ_{DB}	$N^t/10, 0.5$
SOHP: Threshold node degree Δ_{thresh}, Maximum size of the subset of sheep $\lvert N_{subset}\rvert_{max}$, maximum search distance k_{steps}, Herding distance $D_{herding}$	$N^t/2$, $N/2$, 7,$R_{\pi\beta}/2$
Q-learning:Learning rate lr , random decision variable ϵ	0.9,0.1
Q-table: initial values $q(a)_1$,$q(a)_2$,$q(a)_3$	[0.1,0.09,0.08,0.06, 0.05], [0.5,0.4,0.3,0.2,0.1], [1,0.9,0.8,0.6,0.5]

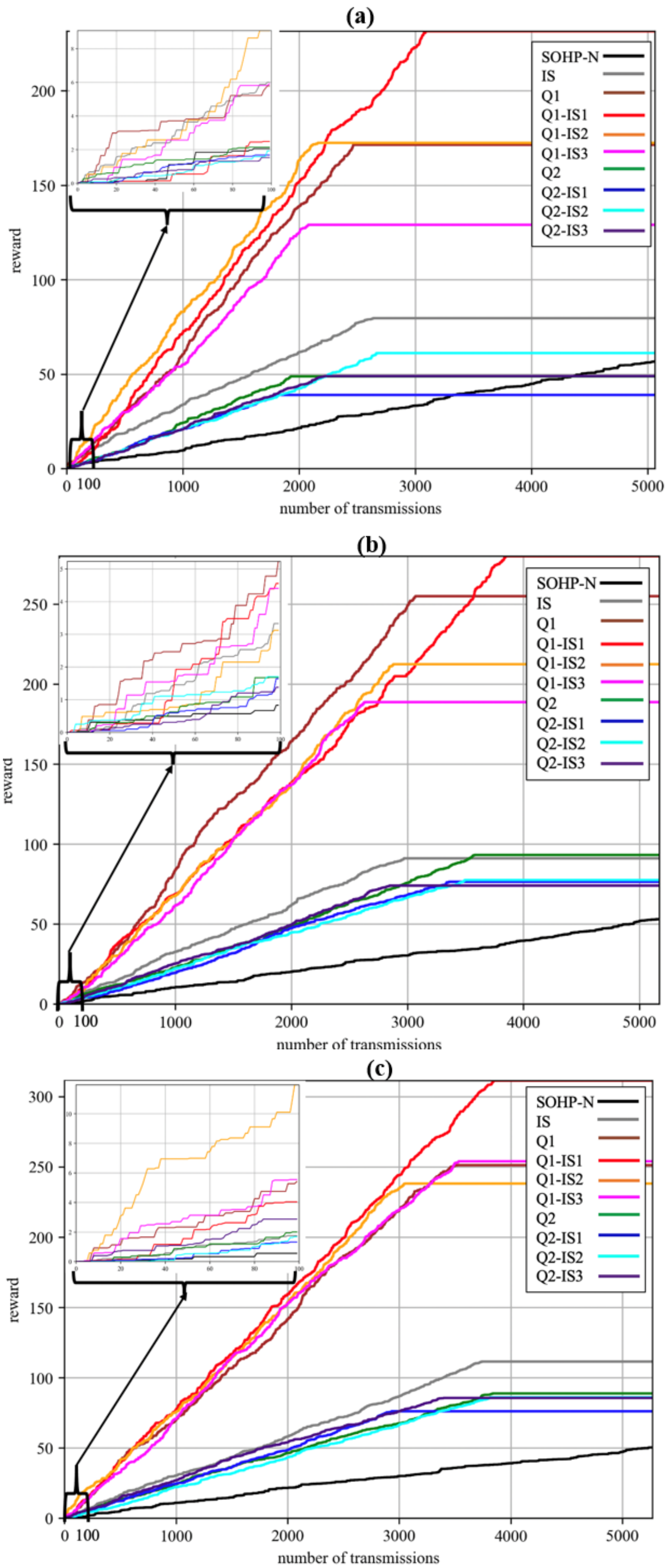

Figure 5.2: Cumulative reward throughout all transmissions for two different reward functions Equation (5.12) for Q1, Q1-IS1, Q1-IS2, Q1-IS3, and Equation (5.13) for SOHP-N, IS, Q2, Q2-IS1, Q2-IS2, Q2-IS3 tested on 50 sheep initialised at three different densities (a) $k = 1/2$, (b) $k = 2/3$, and (c) $k = 3/4$

Table 5.2: Task best measurements in 25 episodes

Metric		Best										
	k	SOHP	SOHP-N	IS	Q1	Q1-IS1	Q1-IS2	Q1-IS3	Q2	Q2-IS1	Q2-IS2	Q2-IS3
T	1/2	118	313	216	259	286	213	114	120	188	240	236
	2/3	374	457	312	338	354	310	287	384	331	336	388
	3/4	370	473	446	349	584	300	390	523	343	649	368
$R * 10^{-3}$	1/2	0	2	4	22	11.7	13.2	4.3	5.1	3.1	3.2	3.3
	2/3	0	1.2	6	11	4.3	9	9	2.4	3.6	2.3	3.2
	3/4	0	1	3	8	7	11	12.5	2	3.6	1.7	4.4
$v_{CU} m/s$	1/2	0	0	1.2	1.1	1	1.1	1.1	1.1	1.1	1.2	1.1
	2/3	0	0	1.2	1	1.2	1.1	1.1	1.1	1.1	1.1	1.1
	3/4	0	0	1.2	1.1	1	1.1	1	1.1	1.1	1.1	1
$S_T\%$	1/2	100	21	43	71	53	40	83	68	80	39	58
	2/3	100	15	50	37	33	50	43	46	36	29	55
	3/4	100	15	49	42	40	49	57	44	41	40	49

Table 5.3: Mean measurements plus or minus one standard deviation for 25 episodes

Metric		Mean±st.dev										
	k	$SOHP$	$SOHP-N$	IS	Q1	Q1-IS1	Q1-IS2	Q1-IS3	Q2	Q2-IS1	Q2-IS2	Q2-IS3
SR	$\frac{1}{2}$	70.8	44	64	68	52	76	88	88	88	60	80
	2/3	83.3	48	72	52	20	72	72	36	32	32	68
	3/4	79.2	36	32	48	8	56	56	36	56	32	64
T	1/2	340.30 ±182.7	520.8 ±106.8	441 ±158.6	440.3 ±130.3	579.5 ±170.2	373.1 ±160.4	321 ±155.6	314 ±161.6	340 ±121.1	477.8 ±141.7	392.9 ±141
	2/3	459.4 ±152	660.3 ±106	568 ±162	612 ±153	705 ±107	546 ±175	503 ±190	685 ±108	668 ±121	620 ±139	590 ±129
	3/4	547.8 ±151.3	745.9 ±115	749 ±113	676 ±175	806 ±47	613 ±200	654 ±175	745 ±107	624 ±191	782 ±59	679 ±171
$N_{\%}$	1/2	84 ±0.37	66 ±41.6	88.4 ±23.4	82.4 ±21.5	83.6 ±21.9	92.3 ±14.7	98.5 ±5.1	95.8 ±11.9	98.3 ±7.6	85.4 ±25.6	95.5 ±9.7
	2/3	91.9 ±23.9	85 ±25	86 ±26	75 ±34	57 ±40	93 ±15	92 ±20	65 ±40	52 ±44	83 ±25	91 ±17.5
	3/4	79.4 ±41.00	69 ±34	66 ±32	76 ±32	18.5 ±33	85 ±25	80 ±30	63 ±40	78 ±34	47 ±46	77.5 ±37
$v_{CU}m/s$	1/2	0±0	0±0	1.4±1.9	1.2±0.1	1.2±0.1	1.2±0.1	1.3±0.2	1.2±0.1	1.3±0.1	1.3±0.1	1.2±0.1
	2/3	0±0	0±0	1.4±0.2	1.2±0.1	1.3±0.1	1.2±0.1	1.2±0.1	1.2±0.1	1.2±0.1	1.2±0.1	1.2±0.1
	3/4	0±0	0±0	1.3±0.2	1.2±0.1	1.2±0.1	1.2±0.1	1.1±0.1	1.2±0.1	1.2±0.1	1.2±0.1	1.1±0.1
$R*10^{-3}$	1/2	0±0	0.8±0.4	2±0.9	6±4	5±1.3	4.7±2.3	5.8±3	6±3	2.3±1.3	1.4±0.6	1.7±0.7
	2/3	0±0	0.5±0.2	1.8±1.1	5±2	3±0.7	5±2	5±2	1.2±0.4	1.2±0.7	1.1±0.5	1.6±0.6
	3/4	0±0	0.5±0.2	1.3±0.5	4±1.6	3.4±1	4.6±2	4.3±2.4	1.2±0.4	1.5±0.7	1±0.3	1.4±0.9
$S_T\%$	1/2	100±0	14±10	34±28	33±23	34±27	37±31	35±23	36±29	36±26	35±31	37±32
	2/3	100±0	11±8	34±27	35±26	21±50	36±33	37±34	35±27	35±32	32±32	37±26
	3/4	100±0	11±8	33±23	34±31	35±32	39±38	32±27	33±18	37±30	32±16	37±30

Chapter 6

Conclusion and Future Directions

This chapter serves as a conclusion to the thesis. First, a summary of the significant contributions made by this thesis is presented. Then, conclusions are discussed. Finally, the chapter ends by highlighting several future research directions.

6.1 Summary of Contributions

As discussed in Chapters 1, and 2, swarm intelligence (SI) is a sub-field of artificial intelligence where several agents collaborate to complete a task. SI has shown success in many real-world applications. Shepherding is a swarm control mechanism that has shown success over the last two decades. It is a biologically inspired swarm-guiding strategy where single or multiple agents (sheepdogs) function as pressure points to guide swarm members (sheep) to a destination.

However, the shepherding literature overlooked the limited sensing range of the controlled agents, which affects the swarm cohesion and, in turn, the mission's success rate (SR). Also, existing work focused on centralised shepherding models, which may affect the SR. On the other hand, a distributed shepherding model, where a central unit (CU) passes perceived information about the information to the shepherd to herd the swarm, brings other challenges in herding the swarm. This is due to noisy and dynamic communication channels.

Therefore, this thesis aimed to design a multi-agent learning system for effective shepherding control in a partially observable environment under communication constraints. To achieve that goal, different contributions were introduced, as outlined below.

- In Chapter 3, a swarm optimum herding point (SOHP) proposal was introduced to find the shepherd's herding point considering the sheep's limited sensing range and the cohesion of the swarm. In it, the graph metrics were used in a geometric method for finding the most effective set of sheep to be guided is proposed. Then, particle swarm optimisation (PSO) and density-based spatial clustering of applications with noise (DBSCAN) were used to estimate the sheepdog's near-optimal herding location while considering flock connection.
- Also, in Chapter 3, an improved version of SOHP was proposed, named swarm optimum modified centroid pushing (SOMCP). In it, the CU used the observed graph metrics to estimate the sheepdog's intermediate waypoints to the herding points obtained by SOHP by simulating the effect of the shepherd's location on graph metrics. Then, PSO was used to find a near-optimal path for the shepherd. It is worth mentioning that while SOHP was only considering the interaction of the swarm of sheep with the sheepdog regardless of its initial location, SOMCP added optimisation in path planning for the sheepdog task. Thus, SOHP may be opening a new direction in robot mobility optimisation that considers herding task constraints.
- In Chapter 4, the shepherding task was modelled as a distributed system with a stationary CU, where the CU acts as the transmitter (Tx) to pass information to the sheepdog (the receiver (RX)) to complete the shepherding task. The chapter aimed to tackle the noisy communication channel between the CU and the shepherd. Firstly, an independent pre-training collaborative learning (IPCL) framework was proposed. In IPCL, the Tx and Rx used NNs that learn collaboratively to overcome the channel noise without modelling the commu-

nication channel. Then, three other variants (IPCL-NF-1AE, IPCL-NF-2AE, and IPCL-NF-QAM-1AE) were proposed to solve the time-varying signal-to-noise ratio (SNR) in the shepherding task. In these methods, a feedback channel was not assumed.

- In Chapter 5, the shepherding task was modelled as a distributed system with mobile CU to improve shepherding in dynamic communication systems. Different from Chapter 4, where a stationary CU was assumed, Chapter 5 assumed the CU could move closer to the sheepdog during the shepherding process. The introduction of the mobility of the CU was to limit the effect of signal fading on the SR of the transmitted data. Therefore, a mobility mechanism based on Q-learning was introduced to improve the SR of shepherding as a dynamic communication system for learning the CU's velocity that leads to a high probability of success in transmission while avoiding large energy consumption in the CU's mobility

6.2 Conclusions

The proposed algorithms were capable of increasing the success rate of the shepherding process. Below, I elaborate on the detailed conclusion of each chapter.

6.2.1 Shepherding swarm With limited sensing range

As mentioned earlier, in Chapter 3, the sheep sensing induced graph was modelled in real-time as a dynamic network to represent the sheep's sensing range. Then SOHP and SOMCP were proposed to find the near-optimal locations for the sheepdog to herd a flock of sheep with a limited sensing range. The performance of the methods was tested for 50 to 200 sheep initialised at different initial densities, where initial high density is the easiest task, and initial low density is the most difficult one. The efficiency of the methods was measured, assuming that the CU and the shepherd communicate over the ideal communication channel. The following findings could be derived.

- The sheepdog could efficiently guide the swarm without constantly communicating with the CU to preserve SOHP and SOMCP resilience.

- SOHP and SOMCP used an average of 10% and 5%, respectively, of the number of transmissions from the CU to the sheepdog compared to the methods in the literature where the sheepdog receives a new location every time step.

- Estimating the shepherd's near-optimal path in SOMCP showed the effectiveness of using PSO for finding the shepherd's locations that meet the problem's objectives (minimising the distance between the sheep and their home location, maximising the sheep node degrees, and minimising the shepherd's path length)

- The use of PSO to choose each point in the route that is within a certain distance of the preceding point enables the shepherd to go closer to the herding point while keeping a cohesive flock

- Using PSO improved SOMCP's SR by 30% more than using DBSCAN for 50 sheep initialised at $k = 2/3$. This is because the accuracy of DBSCAN in selecting the herding point could be adversely affected if the points on the shepherd path were chosen incorrectly.

- SOMCP outperformed SOHP and traditional shepherding methods by half mission time and tripling SR with varying initial densities

- At a high density of sheep initialisation, SOMCP showed at least 15%, 7%, and 50% slower task time for 75, 100 and 200 sheep, respectively, compared to a well-known existing algorithm.

6.2.2 Cooperative learning for shepherding with time-varying and noisy communication channels

As indicated in Chapter 4, the problem of noisy communication channels was addressed, where a deep learning-based method was implemented in the transmitter and receiver agents to de-noise data collaboratively without requiring a predefined channel model. IPCL and its versions were tested on 10^5 transmissions of i.i.d. messages randomly generated at the Tx over a noisy communication channel with $SNR = 40, 10, 4$ and end-systems noise levels $SNR = [-4, 4]$ for up to 1000 transmissions (epochs). From the experimental analysis conducted, the following conclusions can be derived.

- Through IPCL, the MSE of the received data at Rx was decreased to its half for communication channels with low SNR after only 10% of the training time used in an NN-based method in the literature.
- Removing the assumption of feedback channel in IPCL-no-FCH showed a close performance to IPCL with the assumption of a feedback channel in the three SNR channels $SNR = 40, 10, 4$, and they both take a shorter time to converge quickly to lower MSE as SNR decreases.
- IPCL and all its versions couldn't reach the MSE achieved by the policy method [1] when SNR was relatively low (10 and 4) after 1000 training epochs.
- IPCL-no-Rx took a relatively long time compared to all IPCL's versions to converge to a considerably low MSE due to adapting only the Tx adapt to the channel noise and the Rx's noise, while IPCL-no-FCH-no-Rx struggled in learning as SNR decreases.
- IPCL showed the same small error level as policy-auto-encoder-Tx but with a higher convergence rate throughout the training time, while IPCL-no-FCH-no-RX degrades over time.
- At $SNR = [-2, 6]$, the following was observed:

- After $Ep_{ext} = 10$, IPCL-NF-1AE achieved the smallest SER achieved by all the methods in that short training time $SER = 0.056$.
- IPCL-NF-2AE behaved similarly to the IPCL-NF-1AE but at a slower pace.
- At least $Ep_{ext} = 70$ external epochs were required for SER stability in IPCL-NF-1AE and IPCL-NF-2AE.
- IPCL-NF-1AE reached $BER = 10^{-3}$ when $SNR \geq 5$, which proved its ability to generalise after a short training time.
- The BER achieved by IPCL-NF-1AE at $SNR = 0$ was comparable to the one achieved in the recent work in the use of NN to tackle the noisy communication channel problem at $SNR = 5$ [260].
- For IPCL-NF-1AE at $SNR = 0$, SER is less than 20% of its value with 4-QAM, regardless of the training time.

• The SR of the shepherding task achieved by IPCL-NF1E+4QAM was greater than that by the 4-QAM alone by 25% when $k = 2/3, 3/4$. Also, 4-QAM required at least 10% longer task time than IPCL-NF1E+4QAM.

6.2.3 Shepherding under dynamic communications systems

Finally, the work introduced in Chapter 5 to handle shepherding under dynamic communication systems led to the following finding.

• With SOHP, the proposed mobility method at the CU led to doubling the SR of the shepherding task compared to the use of SOHP when the CU is stationary under the effect of the same SNR.

• In the easiest testing scenario ($k = 1/2$, the mobility of the CU led to a higher SR than the one obtained under the assumption of an ideal communication channel due to adding some random behaviour to the shepherd on receiving inaccurate herding points.

- It was observed that a big gap between the initial Q-values could lead to a higher SR, as shown in Q1-IS3 and Q2-IS3.

6.3 Future Research Directions

This section introduces possible future directions in each chapter.

Possible extensions to the work presented in Chapter 3 are as follows.

- Studying the effect of cluttered environments on the flock's graph topology.
- Investigating the effect of different variants of PSO and their parameters in the performance of SOMCP and SOMCP.
- Analysing the challenges that may hinder the complete real-time perception of the environment and flock connectivity by the CU, along with their effect on herding point selection and herding path.
- Designing a realistic system model for swarm guidance applicable to tasks other than shepherding.
- Investigating the problem and proposed solution in SOMCP as a multi-objective optimisation problem.
- Modelling the shepherding task in a three-dimensional environment to provide better insight into the suitability of the proposed method for different swarm guidance applications and their associated challenges.
- Deploying SOMCP in the real-life shepherding scenario for further validation and improvements.

For the work presented in Chapter 4, below are possible future research directions.

- Validating the performance of the IPCL approaches on different dynamic tasks with different channel models.

- Extending the work to time-varying communication channel models that suits mobile physical systems in different environments.
- Analysing the performance of IPCL-NF-1AE to improve the stability of its performance when the initial density of sheep is low ($k = 3/4$).
- Allowing online learning throughout the shepherding tasks without offline training.
- Extending the work to multiple receivers to model the cooperation between multiple actuator acting as shepherds in performing the shepherding task.
- Extending the proposed methods to a two-way communication scenario and asymmetric AWGN where the variance of the normal distribution is different in each direction.

The findings of the proposed method introduced in Chapter 5suggest the following research directions:

- Investigating the use of deep Q-learning for online learning to improve the stability of the CU learning process.
- Analysing the task performance when the same Q-table is updated for different shepherding tasks in different environments.
- Updating the reward function to penalise transmissions so that the task SR and the cumulative reward functions become correlated.
- Extending the work to scenarios that include changes in the channel model that may be used effectively to test the adaptability of the proposed model.

Bibliography

[1] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep Reinforcement Learning Autoencoder with Noisy Feedback," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. Avignon, France: IEEE, 2020, pp. 1–6.

[2] A. Hussein and H. Abbass, "Mixed initiative systems for human-swarm interaction: Opportunities and challenges," in *2nd Annual Systems Modelling Conference (SMC)*, 2018, pp. 1–8.

[3] a. Giusti, J. Nagi, L. Gambardella, and G. a. Di Caro, "Distributed consensus for interaction between humans and mobile robot swarms," *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Demonstration track*, pp. 1503–1504, 2012.

[4] P. G. Faria Dias, M. C. Silva, G. P. Rocha Filho, P. A. Vargas, L. P. Cota, and G. Pessin, "Swarm robotics: A perspective on the latest reviewed concepts and applications," *Sensors*, vol. 21, no. 6, pp. 1–30, 2021.

[5] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm Robotic Behaviors and Current Applications," *Frontiers in Robotics and AI*, vol. 7, no. April, 2020.

[6] D. Tarapore, R. Groß, and K. P. Zauner, "Sparse Robot Swarms: Moving Swarms to Real-World Applications," *Frontiers in Robotics and AI*, vol. 7, no. July, pp. 1–9, 2020.

[7] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2012.

[8] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, "Swarm Distribution and Deployment for Cooperative Surveillance by Micro-Aerial Vehicles," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 84, no. 1-4, pp. 469–492, 2016.

[9] G. Frorq, D. W. Krph, and D. Q. G. Deurdg, "Analysis of military applicatio of UAV swarm technology," in *3rd International Conference on Unmanned Systems (ICUS)*, 2020, pp. 1200–1204.

[10] industryarc, "SI sectors," https://www.industryarc.com/Research/Swarm-Intelligence-Market-Research-505414, 2020, accessed: 2022-09-30.

[11] D. Strömbom and A. J. King, "Robot Collection and Transport of Objects: A Biomimetic Process," *Frontiers in Robotics and AI*, vol. 5, no. May, pp. 1–7, 2018.

[12] J. M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodríguez, and N. M. Amato, "Shepherding Behaviors," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 4, pp. 4159–4164, 2004.

[13] H. A. Abbass and R. A. Hunjet, Eds., *Shepherding UxVs for Human-Swarm Teaming*, 1st ed. Springer, Cham, 2021.

[14] N. K. Long, K. Sammut, D. Sgarioto, M. Garratt, and H. A. Abbass, "A Comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 523–537, 2020.

[15] J. M. Lien, S. Rodríguez, J. P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. April, pp. 3402–3407, 2005.

[16] H. A. Abbass and D. J. R. Editors, *Foundations of Trusted Autonomy.* Springer Cham, 2018.

[17] S. Elsayed, H. Singh, E. Debie, A. Perry, B. Campbell, R. Hunjel, and H. Abbass, "Path planning for shepherding a swarm in a cluttered environment using differential evolution," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 2020, pp. 2194–2201.

[18] H. El-Fiqi, B. Campbell, S. Elsayed, A. Perry, H. K. Singh, R. Hunjet, and H. A. Abbass, "The Limits of Reactive Shepherding Approaches for Swarm Guidance," *IEEE Access*, vol. 8, pp. 214 658–214 671, 2020.

[19] R. Maeda, T. Endo, and F. Matsuno, "Decentralized Navigation for Heterogeneous Swarm Robots with Limited Field of View," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 904–911, 2017.

[20] B. Espiau, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[21] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[22] G. Asaamoning, P. Mendes, D. Rosário, and E. Cerqueira, "Drone swarms as networked control systems by integration of networking and computing," *Sensors*, vol. 21, no. 8, pp. 1–23, 2021.

[23] A. Nayyar, B. L. Nguyen, and N. G. Nguyen, *The Internet of Drone Things (IoDT): Future Envision of Smart Drones.* Springer Singapore, 2020, vol. 1045. [Online]. Available: http://dx.doi.org/10.1007/978-981-15-0029-9{_}45

[24] M. S. Innocente and P. Grasso, "Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems," *Journal of Computational Science*, vol. 34, pp. 80–101, 2019. [Online]. Available: https://doi.org/10.1016/j.jocs.2019.04.009

[25] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J.Sel. A. Commun.*, vol. 37, no. 11, p. 2503–2516, nov 2019. [Online]. Available: https://doi.org/10.1109/JSAC.2019.2933891

[26] A. Hussein, S. Elsawah, and H. A. Abbass, "Swarm collective wisdom: A fuzzy-based consensus approach for evaluating agents confidence in global states," *IEEE International Conference on Fuzzy Systems*, vol. 2020-July, 2020.

[27] A. A. Khuwaja, Y. Chen, N. Zhao, M. S. Alouini, and P. Dobbins, "A survey of channel modeling for uav communications," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.

[28] Y. Liu and K. M. Passino, "Swarm intelligence: Literature overview," *Department of electrical engineering, the Ohio State University*, 2000.

[29] C. M. Cianci, X. Raemy, J. Pugh, and A. Martinoli, "Communication in a swarm of miniature robots: The e-Puck as an educational tool for swarm robotics," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4433 LNCS, pp. 103–115, 2007.

[30] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3293–3298, 2012.

[31] A. D. Marchese, C. D. Onal, and D. Rus, *Towards a Self-contained Soft Robotic Fish: On-Board Pressure Generation and Embedded Electro-permanent Magnet Valves.* Heidelberg: Springer International Publishing, 2013, pp. 41–54. [Online]. Available: https://doi.org/10.1007/978-3-319-00065-7_4

[32] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science Robotics*, vol. 1, no. 1, pp. 1–12, 2016.

[33] G. M. Whitesides, "Soft Robotics," *Angewandte Chemie - International Edition*, vol. 57, no. 16, pp. 4258–4273, 2018.

[34] A. Spielberg, A. Amini, L. Chin, W. Matusik, and D. Rus, "Co-learning of task and sensor placement for soft robotics," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1208–1215, 2021.

[35] MIT, "Soft Robotic Fish Swims Alongside Real Ones Coral Reefs," https://news.mit.edu/2018/soft-robotic-fish-swims-alongside-real-ones-coral-reefs-0321, 2018, accessed: 2022-10-30.

[36] NASA, "Swarmies," https://www.nasa.gov/content/meet-the-swarmies-robotics-answer-to-bugs, 2020, accessed: 2022-10-30.

[37] N. Testing, "NASA Field Test," https://www.nasa.gov/content/swarmies-shuffle-through-field-tests, 2020, accessed: 2022-10-30.

[38] L. Kleinrock, "Distributed systems," *Commun. ACM*, vol. 28, no. 11, p. 1200–1213, nov 1985. [Online]. Available: https://doi.org/10.1145/4547.4552

[39] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, vol. 21, no. July, pp. 25–34, 1987.

[40] C. Hartman and B. Beneš, "Autonomous boids," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 199–206, 2006.

[41] A. W. Woolley, I. Aggarwal, and T. W. Malone, "Collective Intelligence and Group Performance," *Current Directions in Psychological Science*, vol. 24, no. 6, pp. 420–424, 2015.

[42] A. C. Schultz, J. J. Grefenstette, and W. Adams, "Robo-shepherd: Learning complex robotic behaviors," in *the International Symposium on Robotics and Automation*, no. TSI Press, ASME, 1996, pp. 763–768.

[43] H. Çelikkanat and E. Şahin, "Steering self-organized robot flocks through externally guided individuals," *Neural Computing and Applications*, vol. 19, no. 6, pp. 849–865, 2010.

[44] H. Zhao, H. Liu, Y. W. Leung, and X. Chu, "Self-Adaptive Collective Motion of Swarm Robots," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1533–1545, 2018.

[45] X. Dong and G. Hu, "Time-varying formation control for general linear multi-agent systems with switching directed topologies," *Automatica*, vol. 73, pp. 47–55, 2016.

[46] ——, "Time-Varying Formation Tracking for Linear Multiagent Systems with Multiple Leaders," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3658–3664, 2017.

[47] L. Briñón-Arranz, A. Seuret, and C. Canudas-de Wit, "Cooperative control design for time-varying formations of multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2283–2288, 2014.

[48] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 109–117, 2000.

[49] K. Yamagishi and T. SUZUKI, "Collective movement control method for swarm robots based on thermodynamic model," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 11, pp. 513–519, 2016.

[50] J.-P. Hansen and L. Verlet, "Phase transitions of the lennard-jones system," *Phys. Rev.*, vol. 184, pp. 151–161, Aug 1969. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.184.151

[51] M. S. Talamali, T. Bose, M. Haire, X. Xu, J. A. Marshall, and A. Reina,

"Sophisticated collective foraging with minimalist agents: a swarm robotics test," *Swarm Intelligence*, vol. 14, no. 1, pp. 25–56, 2020.

[52] N. R. Clayton and H. Abbass, "Machine teaching in hierarchical genetic reinforcement learning: Curriculum design of reward functions for swarm shepherding," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 1259–1266.

[53] D. Helbing, A. Johansson, and H. Z. Al-Abideen, "Dynamics of crowd disasters: An empirical study," *Physical review E*, vol. 75, no. 4, p. 046109, 2007.

[54] A. Fujioka, M. Ogura, and N. Wakamiya, "Shepherding algorithm based on variant agent detection for heterogeneous flock," in *2022 61st Annual Conference of the Society of Instrument and Control Engineers (SICE)*. IEEE, 2022, pp. 87–91.

[55] N. M. Baldock and R. M. Sibly, "Effects of handling and transportation on the heart rate and behaviour of sheep," *Applied Animal Behaviour Science*, vol. 28, no. 1-2, pp. 15–39, 1990.

[56] K. J. Yaxley, K. F. Joiner, and H. Abbass, "Drone approach parameters leading to lower stress sheep flocking and movement: sky shepherding," *Scientific Reports*, vol. 11, no. 1, pp. 1–9, 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-87453-y

[57] L. Skrba, S. Dobbyn, R. Mcdonnell, and C. O. Sullivan, "Animating Dolly : Real-time Herding and Rendering of Sheep," in *Seventh Irish Workshop on Computer Graphics*. The Eurographics Association, 2006.

[58] Dylan Trotsek, *Robot Intelligence An Advanced Knowledge Processing Approach*. Springer London, 2017, vol. 110, no. 9.

[59] D. Strombom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. T. Sumpter, and A. J. King, "Solving the shepherding problem: heuristics for

herding autonomous, interacting agents," *Journal of The Royal Society Interface*, vol. 11, no. 100, pp. 20 140 719–20 140 719, 2014.

[60] A. Procaccini, A. Orlandi, A. Cavagna, I. Giardina, F. Zoratto, D. Santucci, F. Chiarotti, C. K. Hemelrijk, E. Alleva, G. Parisi, and C. Carere, "Propagating waves in starling, Sturnus vulgaris, flocks under predation," *Animal Behaviour*, vol. 82, no. 4, pp. 759–765, 2011.

[61] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *Proceedings of the IEEE Conference on Decision and Control*, vol. 54, no. 2, pp. 2937–2942, 2009.

[62] W. Yuan, N. Ganganath, C.-t. Cheng, S. Valaee, Q. Guo, and F. C. M. Lau, "Energy-Efficient Semi-Flocking Control of Mobile Sensor Networks on Rough Terrains," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 4, pp. 622–626, 2019.

[63] A. Ghosh and S. Boyd, "Growing Well-connected Graphs," in *45th IEEE Conference on Decision & Control*, 2006, pp. 6605–6611.

[64] T. Anderson, C. Y. Chang, and S. Martinez, "Maximizing Algebraic Connectivity of Constrained Graphs in Adversarial Environments," in *2018 European Control Conference (ECC)*, Limassol, 2018, pp. 125–130.

[65] D. Strömbom, T. Hassan, W. Hunter Greis, and A. Antia, "Asynchrony induces polarization in attraction-based models of collective motion," *Royal Society Open Science*, vol. 6, no. 4, 2019.

[66] A. Hussein, E. Petraki, and H. A. Abbass, "Autonomous Swarm Shepherding Using Curriculum-Based Reinforcement Autonomous Swarm Shepherding Using Curriculum-Based Reinforcement Learning," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, no. May, 2022.

[67] H. T. Nguyen, M. Garratt, L. T. Bui, and H. Abbass, "Disturbances in influence of a shepherding agent is more impactful than sensorial noise during

swarm guidance," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 2257–2264.

[68] Z. Kowalczuk, W. Jedruch, and K. Szymanski, "The Use of an Autoencoder in the Problem of Shepherding," in *23rd International Conference on Methods and Models in Automation and Robotics, MMAR 2018*, 2018, pp. 947–952.

[69] A. Fischer and C. Igel, "An Introduction to Restricted Boltzmann Machines," in *17th Iberoamerican Congress, CIARP*, vol. 3665 LNCS, 2012, pp. 14–36.

[70] E. Bingham and H. Mannila, "Random projection in dimensionality reduction : Applications to image and text data," in *the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 5–10.

[71] A. Pierson and M. Schwager, "Controlling Noncooperative Herds with Robotic Herders," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 517–525, 2018.

[72] S. Gade, A. A. Paranjape, and S. J. Chung, "Herding a flock of birds approaching an airport using an unmanned aerial vehicle," in *AIAA Guidance, Navigation, and Control Conference*, 2015.

[73] R. Himo, M. Ogura, and N. Wakamiya, "Iterative shepherding control for agents with heterogeneous responsivity," *Mathematical Biosciences and Engineering*, vol. 19, no. 4, pp. 3509–3525, 2022.

[74] A. Fujioka, M. Ogura, and N. Wakamiya, "Shepherding algorithm for heterogeneous flock with model-based discrimination," *Advanced Robotics*, pp. 1–16, 2022.

[75] A. Li, M. Ogura, and N. Wakamiya, "Communication-free shepherding navigation with multiple steering agents," *Arxiv*, 2022.

[76] J. Hu, A. E. Turgut, T. Krajník, B. Lennox, and F. Arvin, "Occlusion-based coordination protocol design for autonomous robotic shepherding tasks," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[77] F. Arvin, A. E. Turgut, T. Krajník, S. Rahimi, I. E. Okay, S. Yue, S. Watson, and B. Lennox, "ϕ clust: Pheromone-based aggregation for robotic swarms," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4288–4294.

[78] F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, and B. Lennox, "Mona: an affordable open-source mobile robot for education and research," *Journal of Intelligent & Robotic Systems*, vol. 94, pp. 761–775, 2019.

[79] X. Li, H. Huang, A. V. Savkin, and J. Zhang, "Robotic Herding of Farm Animals Using a Network of Barking Aerial Drones," *Drones*, vol. 6, no. 2, 2022.

[80] A. A. Paranjape, S. J. Chung, K. Kim, and D. H. Shim, "Robotic Herding of a Flock of Birds Using an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 901–915, 2018.

[81] W. Lee and D. Kim, "Autonomous shepherding behaviors of multiple target steering robots," *Sensors (Switzerland)*, vol. 17, no. 12, 2017.

[82] J. Zhi and J.-M. Lien, "Learning to herd agents amongst obstacles: Training robust shepherding behaviors using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4163–4168, 2021.

[83] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[84] V. D. Blondel and J. N. Tsitsiklis, "Survey of computational complexity results in systems and control," *Automatica*, vol. 36, no. 9, pp. 1249–1274, 2000.

[85] F. Sahin and J. S. Bay, "Structural bayesian network learning in a biological decision-theoretic intelligent agent and its application to a herding problem in the context of distributed multi-agent systems," *Proceedings of the IEEE*

International Conference on Systems, Man and Cybernetics, vol. 3, pp. 1606–1611, 2001.

[86] J. F. Harrison, C. Vo, and J. M. Lien, "Scalable and robust shepherding via deformable shapes," in *Third International Conference on Motion in Games*, vol. 6459 LNCS. Springer International Publishing, 2010, pp. 218–229.

[87] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multi-robot herding," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 1843–1849, 2015.

[88] J. Harvey, "The blessing and curse of emergence in swarm intelligence systems," in *Foundations of trusted autonomy.* Springer, Cham, 2018, pp. 117–124.

[89] H. Hoshi, I. Iimura, S. Nakayama, Y. Moriyama, and K. Ishibashi, "Robustness of Herding Algorithm with a Single Shepherd Regarding Agents' Moving Speeds," *Journal of Signal Processing*, vol. 22, no. 6, pp. 327–335, 2018.

[90] J. Amiryan and M. Jamzad, "Adaptive motion planning with artificial potential fields using a prior path," in *International Conference on Robotics and Mechatronics, ICROM 2015*, 2015, pp. 731–736.

[91] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[92] G. Asaamoning, P. Mendes, D. Rosário, and E. Cerqueira, "Drone swarms as networked control systems by integration of networking and computing," *Sensors*, vol. 21, no. 8, p. 2642, 2021.

[93] M. Baumann, "Learning shepherding behavior," Ph.D. dissertation, Faculty of Computer Science, Electrical Engineering and Mathematics University of Paderborn, 2015. [Online]. Available: https://d-nb.info/1082284319/34

[94] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[95] T. Hailing, S. Zongtan, Y. Yuan, and X. Yuan, "Analysis and Insights from the MIT Technology Review "Top 10 Breakthrough Technologies" in the Past Six Years," *Chinese Journal of Engineering Science*, vol. 19, no. 5, p. 85, 2017.

[96] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3671–3678.

[97] D. Coleman, I. A. Şucan, M. Moll, K. Okada, and N. Correll, "Experience-based planning with sparse roadmap spanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 900–905.

[98] M. Zucker, J. Kuffner, and J. A. Bagnell, "Adaptive workspace biasing for sampling-based planners," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3757–3762.

[99] A. H. Qureshi and M. C. Yip, "Deeply Informed Neural Sampling for Robot Motion Planning," *IEEE International Conference on Intelligent Robots and Systems*, pp. 6582–6588, 2018.

[100] C. K. Go, B. Lao, J. Yoshimoto, and K. Ikeda, "A Reinforcement Learning Approach to the Shepherding Task Using SARSA," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2016-Octob, pp. 3833–3836, 2016.

[101] J. Hu, A. E. Turgut, T. Krajnik, B. Lennox, and F. Arvin, "Occlusion-Based Coordination Protocol Design for Autonomous Robotic Shepherding Tasks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 1, pp. 126–135, 2022.

[102] H. Abbass, J. Harvey, and K. Yaxley, "Lifelong Testing of Smart Autonomous

Systems by Shepherding a Swarm of Watchdog Artificial Intelligence Agents," *arXiv*, 2018. [Online]. Available: http://arxiv.org/abs/1812.08960

[103] P. E. Hart, N. J. Nilsson, and B. Raphael, "Formal Basis for the Heuristic Determination eijj ,," *Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[104] D. Tse and P. Viswanath, "The wireless channel," in *Fundamentals of Wireless Communication*, 2012, pp. 10–48.

[105] B. Jones, "Electrical noise as a measure of quality and reliability in electronic devices," in *Advances in electronics and electron physics.* Elsevier, 1993, vol. 87, pp. 201–257.

[106] R. G. Carvajal, J. Galan, and A. Torralba, "Electrical noise as a reliability indicator in electronic devices and components," *IEE Proceedings - Circuits, Devices and Systems*, vol. 149, no. 1, pp. 13–22, 2002.

[107] C.-H. Chen, "Thermal Noise Measurement and Characterization for Modern Semiconductor Devices," *IEEE Instrumentation & Measurement Magazine*, vol. 272, no. 5648, p. 108, 2021.

[108] H. Dong, Z. Wang, and H. Gao, "Robust h_{\infty} filtering for a class of nonlinear networked systems with multiple stochastic communication delays and packet dropouts," *IEEE Transactions on signal processing*, vol. 58, no. 4, pp. 1957–1966, 2009.

[109] S. G. Chappell, J. C. Jalbert, P. Pietryka, and J. Duchesney, "Acoustic communication between two autonomous underwater vehicles," in *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology (AUV'94).* IEEE, 1994, pp. 462–469.

[110] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," *IEEE journal of oceanic engineering*, vol. 25, no. 1, pp. 72–83, 2000.

[111] D. K. Misra, *Radio-Frequency and Microwave Communication Circuits.* A WILEY-INTERSCIENCE PUBLICATION, 2004, vol. 8.

[112] S. Scone, "Opportunistic communication schemes for unmanned vehicles in urban search and rescue," Ph.D. dissertation, Loughborough University, 2010.

[113] W. Khawaja, I. Guvenc, D. W. Matolak, U. C. Fiebig, and N. Schneckenburger, "A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2361–2391, 2019.

[114] A. A. Khuwaja, Y. Chen, N. Zhao, M. S. Alouini, and P. Dobbins, "A survey of channel modeling for uav communications," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.

[115] N. Goddemeier and C. Wietfeld, "Investigation of air-to-air channel characteristics and a UAV specific extension to the rice model," *2015 IEEE Globecom Workshops, GC Wkshps 2015 - Proceedings*, pp. 2–6, 2015.

[116] İlker Bekmezci, O. K. Sahingoz, and Şamil Temel, "Flying ad-hoc networks (fanets): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870512002193

[117] T. Andre, K. A. Hummel, A. P. Schoellig, E. Yanmaz, M. Asadpour, C. Bettstetter, P. Grippa, H. Hellwagner, S. Sand, and S. Zhang, "Application-driven design of aerial communication networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 129–137, 2014.

[118] L. Afonso, N. Souto, P. Sebastião, M. Ribeiro, T. Tavares, and R. Marinheiro, "Cellular for the skies: Exploiting mobile network infrastructure for low altitude air-to-ground communications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 8, pp. 4–11, 2016.

[119] E. Biglieri, "Coding and Modulation for a Horrible Channel," *IEEE Communications Magazine*, vol. 41, no. 5, pp. 92–98, 2003.

[120] M. Zimmermann and K. Dostert, "Analysis and modeling of impulsive noise in broad-band powerline communications," *IEEE Transactions on Electromagnetic Compatibility*, vol. 44, no. 1, pp. 249–258, 2002.

[121] J. Haring, "Coding for impulsive noise channels," *Proc. ISPLC 2001, Malmo, Sweden*, 2001.

[122] I. Bárány and V. Vu, "Central limit theorems for gaussian polytopes," *The Annals of Probability*, vol. 35, no. 4, pp. 1593–1621, 2007. [Online]. Available: http://www.jstor.org/stable/25450023

[123] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[124] J. G. Proakis, "Digital communications, 1983."

[125] M. Schwartz, W. R. Bennett, and S. Stein, *Communication systems and techniques*. John Wiley & Sons, 1995.

[126] S. Stein, "Fading channel issues in system engineering," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 68–89, 1987.

[127] B. Sklar, " Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 136–146, 1997.

[128] H. T. Friis, "A Note on a Simple Transmission Formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[129] J. D. Mahony, "Omnidirectional pattern directivity in the absence of minor lobes: revisited," *IEEE Antennas and Propagation Magazine*, vol. 40, pp. 76–81, 1998.

[130] A. Bhowal, R. S. Kshetrimayum, A. Bhowal, and R. S. Kshetrimayum, "Channel model," *Advanced Spatial Modulation Systems*, pp. 25–42, 2021.

[131] M. Mahbub, "UAV Assisted 5G Het-Net: A Highly Supportive Technology for 5G NR Network Enhancement," *EAI Endorsed Transactions on Internet of Things*, vol. 6, no. 22, p. 166003, 2020.

[132] F. Giannetti, M. Luise, and R. Reggiannini, "Mobile and personal communications in the 60 GHz band: A survey," *Wireless Personal Communications*, vol. 10, no. 2, pp. 207–243, 1999.

[133] W. M. Carey, "Lloyd's mirror-image interference effects," *Acoust. Today*, vol. 5, no. 2, pp. 14–20, 2009.

[134] D. Cox, R. Murray, and A. Norris, "Sod-mhz attenuation measured in and around suburban houses," *Bell Syst. Tech*, vol. 1, 1984.

[135] Y. Xing, T. S. Rappaport, and A. Ghosh, "Millimeter Wave and Sub-THz Indoor Radio Propagation Channel Measurements, Models, and Comparisons in an Office Environment," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3151–3155, 2021.

[136] D. C. Cox, R. R. Murray, and A. W. Norris, "800-mhz attenuation measured in and around suburban houses," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 6, pp. 921–954, 1984.

[137] S. Seidel, T. Rappaport, S. Jain, M. Lord, and R. Singh, "Path loss, scattering and multipath delay statistics in four european cities for digital cellular and microcellular radiotelephone," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 4, pp. 721–730, 1991.

[138] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path loss prediction based on machine learning: Principle, method, and data expansion," *Applied Sciences*, vol. 9, no. 9, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/9/1908

[139] I. SEN, "5 ghz channel characterization for airport surface areas and vehicle-vehicle communication systems," Ph.D. dissertation, Russ College of Engineering and Technology of Ohio University, 2007. [Online]. Available: https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=ohiou1184346908&disposition=inline

[140] T. S. Rappaport *et al.*, *Wireless communications: principles and practice.* prentice hall PTR New Jersey, 1996, vol. 2.

[141] Y. Okumura, "Field strength and its variability in vhf and uhf land-mobile radio service," *Rev. Electr. Commun. Lab.*, vol. 16, pp. 825–873, 1968.

[142] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, 1980.

[143] Z. Nadir and M. I. Ahmad, "Pathloss Determination Using Okumura-Hata Model And Cubic Regression For Missing Data For Oman," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, no. March, pp. 17–19, 2010.

[144] M. Farhoud, A. El-Keyi, and A. Sultan, "Empirical correction of the okumura-hata model for the 900 mhz band in egypt," in *2013 Third International Conference on Communications and Information Technology (ICCIT)*, 2013, pp. 386–390.

[145] P. E. Mogensen and J. Wigard, "Cost action 231: Digital mobile radio towards future generation system, final report." in *Section 5.2: On antenna and frequency diversity in GSM. Section 5.3: Capacity study of frequency hopping GSM network*, 1999.

[146] Y. Singh, "Comparison of Okumura, Hata and COST-231 Models on the Basis of Path Loss and Signal Strength," *International Journal of Computer Applications*, vol. 59, no. 11, pp. 37–41, 2012.

[147] M. M. Rahman, T. K. Godder, B. C. Singh, and M. T. Parvin, "Performance Comparison of AWGN, Flat Fading and Frequency Selective Fading Channel for Wireless Communication System using 4QPSK," *Online*, vol. 01, no. 02, pp. 82–90, 2011.

[148] W. Hayward and D. DeMaw, *Solid State Design for the Radio Amateur, Chapter 5 "Receiver Design Basics"*, 11th ed. American Radio Relay League, 1977.

[149] B. A. Carlson and P. B. Crilly, *COMMUNICATION SYSTEMS An Introduction to Signals and Noise in Electrical Communication*, 5th ed. McGraw-Hill Education, 1937.

[150] D. Van Meter and D. Middleton, "Modern statistical approaches to reception in communication theory," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 119–145, 1954.

[151] C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[152] H. Meyr, "Digital communication receivers: synchronization," *Channel Estimation and Signal Processing*, 1998.

[153] U. Mengali, *Synchronization techniques for digital receivers.* Springer Science & Business Media, 2013.

[154] Y.-C. Wu and E. Serpedin, "Design and analysis of feedforward symbol timing estimators based on the conditional maximum likelihood principle," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1908–1918, 2005.

[155] J. G. Proakis, "Coded modulation for digital communications over rayleigh fading channels," *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 66–73, 1991.

[156] V. Lottici and M. Luise, "Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 661–669, 2004.

[157] C. Herzet, N. Noels, V. Lottici, H. Wymeersch, M. Luise, M. Moeneclaey, and L. Vandendorpe, "Code-aided turbo synchronization," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1255–1271, 2007.

[158] M. K. Simon and M.-S. Alouini, "A unified approach to the performance analysis of digital communication over generalized fading channels," *Proceedings of the IEEE*, vol. 86, no. 9, pp. 1860–1877, 1998.

[159] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital communication*. Springer Science & Business Media, 2012.

[160] J. Hayes, "Adaptive feedback communications," *IEEE Transactions on Communication Technology*, vol. 16, no. 1, pp. 29–34, 1968.

[161] J. Cavers, "Variable-rate transmission for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 20, no. 1, pp. 15–22, 1972.

[162] S. Otsuki, S. Sampei, and N. Morinaga, "Square-qam adaptive modulation/tdma/tdd systems using modulation level estimation with walsh function," *Electronics Letters*, vol. 31, no. 3, pp. 169–171, 1995.

[163] W. Webb and R. Steele, "Variable rate qam for mobile radio," *IEEE Transactions on Communications*, vol. 43, no. 7, pp. 2223–2230, 1995.

[164] Y. Kamio, S. Sampei, H. Sasaoka, and N. Morinaga, "Performance of modulation-level-controlled adaptive-modulation under limited transmission delay time for land mobile communications," in *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, vol. 1. IEEE, 1995, pp. 221–225.

[165] S. M. Alamouti and S. Kallel, "Adaptive trellis-coded multiple-phase-shift keying for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 42, no. 6, pp. 2305–2314, 1994.

[166] T. Ue, S. Sampei, and N. Morinaga, "Symbol rate and modulation level controlled adaptive modulation/tdma/tdd for personal communication systems,"

in *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, vol. 1. IEEE, 1995, pp. 306–310.

[167] A. J. Goldsmith and S.-G. Chua, "Variable-rate variable-power mqam for fading channels," *IEEE transactions on communications*, vol. 45, no. 10, pp. 1218–1230, 1997.

[168] J. A. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Communications magazine*, vol. 28, no. 5, pp. 5–14, 1990.

[169] P. S. Chow, J. M. Cioffi, and J. A. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *IEEE Transactions on communications*, vol. 43, no. 2/3/4, pp. 773–775, 1995.

[170] T. Inoue, S. Nakajima, T. Masamura, T. Kaitsuka, and K. Satoh, "30/20 ghz band scpc satellite communication using small earth stations," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 4, pp. 600–608, 1983.

[171] R. Cox, J. Hagenauer, N. Seshadri, and C.-E. Sundberg, "Subband speech coding and matched convolutional channel coding for mobile radio channels," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1717–1731, 1991.

[172] A. Goldsmith, *Wireless communications.* Cambridge university press, 2005.

[173] A. Samukic, "Umts universal mobile telecommunications system: development of standards for the third generation," *IEEE transactions on vehicular technology*, vol. 47, no. 4, pp. 1099–1104, 1998.

[174] H. Holma, A. Toskala, K. Ranta-aho, and J. Pirskanen, "High-speed packet access evolution in 3gpp release 7 [topics in radio communications]," *IEEE Communications Magazine*, vol. 45, no. 12, pp. 29–35, 2007.

[175] L. Calderin, S. Ramakrishnan, A. Puglielli, E. Alon, B. Nikolić, and A. M. Niknejad, "Analysis and design of integrated active cancellation transceiver

for frequency division duplex systems," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 8, pp. 2038–2054, 2017.

[176] J. She, J. Mei, J. Ho, P.-H. Ho, and H. Ji, "Layered adaptive modulation and coding for 4g wireless networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, 2010, pp. 1–6.

[177] Y. Wang, W. Liu, and L. Fang, "Adaptive modulation and coding technology in 5g system," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 159–164.

[178] F. Peng, J. Zhang, and W. E. Ryan, "Adaptive modulation and coding for ieee 802.11n," in *Proceedings of the 2007 IEEE Wireless Communications and Networking Conference*. USA: IEEE Computer Society, 2007, p. 656–661. [Online]. Available: https://doi.org/10.1109/WCNC.2007.126

[179] C. S. Park and S. Park, "Implementation of a Fast Link Rate Adaptation Algorithm for WLAN Systems," *Electronics*, 2021.

[180] B. Vucetic, "An adaptive coding scheme for time-varying channels," *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 653–663, 1991.

[181] A. Aghamohammadi, H. Meyr, and G. Ascheid, "Adaptive synchronization and channel parameter estimation using an extended kalman filter," *IEEE Transactions on Communications*, vol. 37, no. 11, pp. 1212–1219, 1989.

[182] P. K. Singya, P. Shaik, N. Kumar, V. Bhatia, and M.-S. Alouini, "A survey on higher-order qam constellations: Technical challenges, recent advances, and future trends," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 617–655, 2021.

[183] M. L. Moher and J. H. Lodge, "Tcmp-a modulation and coding strategy for rician fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 9, pp. 1347–1355, 1989.

[184] S. Sampei and T. Sunaga, "Rayleigh fading compensation for qam in land mobile radio communications," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 2, pp. 137–147, 1993.

[185] I. Abou-Fayçal, M. Médard, and U. Madhow, "Binary adaptive coded pilot symbol assisted modulation over rayleigh fading channels without feedback," *IEEE Transactions on Communications*, vol. 53, no. 6, pp. 1036–1046, 2005.

[186] J. K. Cavers, "Performance of tone calibration with frequency offset and imperfect pilot filter (mobile radio)," *IEEE transactions on vehicular technology*, vol. 40, no. 2, pp. 426–434, 1991.

[187] X. Cai and G. B. Giannakis, "Error probability minimizing pilots for OFDM with M-PSK modulation over Rayleigh-fading channels," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 1, pp. 146–155, 2004.

[188] H. Tang, K. Y. Lau, and R. W. Brodersen, "Interpolation-based maximum likelihood channel estimation using ofdm pilot symbols," in *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, vol. 2. IEEE, 2002, pp. 1860–1864.

[189] O. Simeone, Y. Bar-Ness, and U. Spagnolini, "Pilot-based channel estimation for ofdm systems by tracking the delay-subspace," *IEEE transactions on wireless communications*, vol. 3, no. 1, pp. 315–325, 2004.

[190] X. Dong, W.-S. Lu, and A. C. Soong, "Linear interpolation in pilot symbol assisted channel estimation for ofdm," *IEEE transactions on wireless communications*, vol. 6, no. 5, pp. 1910–1920, 2007.

[191] R. Negi and J. Cioffi, "Pilot tone selection for channel estimation in a mobile ofdm system," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 3, pp. 1122–1128, 1998.

[192] S. Song and A. Singer, "Pilot-aided ofdm channel estimation in the presence

of the guard band," *IEEE Transactions on Communications*, vol. 55, no. 8, pp. 1459–1465, 2007.

[193] R. J. Baxley, J. E. Kleider, and G. T. Zhou, "Pilot design for ofdm with null edge subcarriers," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 396–405, 2009.

[194] H. Steendam and S. Member, "On the Pilot Carrier Placement in Multicarrier-Based Systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1812–1821, 2014.

[195] R. L. Stratonovich, "On the theory of optimal non-linear filtering of random functions," *Theory of Probability and its Applications*, vol. 4, pp. 223–225, 1959.

[196] ——, "Application of the markov processes theory to optimal filtering," *Radio Engineering and Electronic Physics*, vol. 5, pp. 1–19, 1960.

[197] ——, "Optimum nonlinear systems which bring about a separation of a signal with constant parameters from noise," *Radiofizika*, vol. 2, no. 6, pp. 892–901, 1959.

[198] G. Rajender, T. Anilkumar, and K. S. Rao, "Estimation and performance — kalman based approach: A review," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1145–1151.

[199] K. J. Kim, J. Yue, R. A. Iltis, and J. D. Gibson, "A qrd-m/kalman filter-based detection and channel estimation algorithm for mimo-ofdm systems," *IEEE transactions on wireless communications*, vol. 4, no. 2, pp. 710–721, 2005.

[200] M. Siyau, T. Li, J. Prieto, J. Corchado, and J. Bajo, "A novel pilot expansion approach for mimo channel estimation and tracking," in *2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*. IEEE, 2015, pp. 1–5.

[201] P. Banelli, R. C. Cannizzaro, and L. Rugini, "Data-aided kalman tracking for channel estimation in doppler-affected ofdm systems," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 3. IEEE, 2007, pp. III–133.

[202] A. Bateman, "Feedforward transparent tone-in-band: Its implementations and applications," *IEEE Transactions on Vehicular Technology*, vol. 39, no. 3, pp. 235–243, 1990.

[203] M. Huang, X. Chen, L. Xiao, S. Zhou, and J. Wang, "Kalman-filter-based channel estimation for orthogonal frequency-division multiplexing systems in time-varying channels," *IET communications*, vol. 1, no. 4, pp. 795–801, 2007.

[204] D. Angelosante, E. Biglieri, and M. Lops, "Sequential estimation of multipath mimo-ofdm channels," *IEEE transactions on signal processing*, vol. 57, no. 8, pp. 3167–3181, 2009.

[205] J. Wang, O. Wen, H. Chen, and S. Li, "Power allocation between pilot and data symbols for mimo systems with mmse detection under mmse channel estimation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, pp. 1–9, 2011.

[206] F. Rottenberg, F. Horlin, E. Kofidis, and J. Louveaux, "Generalized optimal pilot allocation for channel estimation in multicarrier systems," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2016, pp. 1–5.

[207] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*. Springer, 2003, pp. 81–100.

[208] X. P. Zhang, "Thresholding neural network for adaptive noise reduction," *IEEE Transactions on Neural Networks*, vol. 12, no. 3, pp. 567–584, 2001.

[209] K. Zhang, Z. Yang, and T. Ba, "Networked Multi-Agent Reinforcement Learning in Continuous Spaces," in *IEEE Conference on Decision and Control (CDC)*, no. Cdc, Miami Beach, FL,, 2018, pp. 2771–2776.

[210] K. Tsagkaris, A. Katidiotis, and P. Demestichas, "Neural network-based learning schemes for cognitive radio systems," *Computer communications*, vol. 31, no. 14, pp. 3394–3404, 2008.

[211] M. Shafique, T. Theocharides, C. S. Bouganis, M. A. Hanif, F. Khalid, R. Hafiz, and S. Rehman, "An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era," *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, vol. 2018-Janua, pp. 827–832, 2018.

[212] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. July, pp. 504–507, 2006.

[213] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent Neural Networks for Noise Reduction in Robust ASR Andrew," in *Interspeech*, 2012, pp. 22–25.

[214] N. Faruk, S. I. Popoola, N. T. Surajudeen-Bakinde, A. A. Oloyede, A. Abdulkarim, L. A. Olawoyin, M. Ali, C. T. Calafate, and A. A. Atayero, "Path loss predictions in the vhf and uhf bands within urban environments: experimental investigation of empirical, heuristics and geospatial models," *IEEE access*, vol. 7, pp. 77 293–77 307, 2019.

[215] R. He, Y. Gong, W. Bai, Y. Li, and X. Wang, "Random forests based path loss prediction in mobile communication systems," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, 2020, pp. 1246–1250.

[216] C. Nguyen and A. A. Cheema, "A deep neural network-based multi-frequency

path loss prediction model from 0.8 ghz to 70 ghz," *Sensors*, vol. 21, no. 15, p. 5100, 2021.

[217] A. M. Al-Samman, T. A. Rahman, M. H. Azmi, and S. A. Al-Gailani, "Millimeter-wave propagation measurements and models at 28 ghz and 38 ghz in a dining room for 5g wireless networks," *Measurement*, vol. 130, pp. 71–81, 2018.

[218] L. Pometcu and R. D'Errico, "An indoor channel model for high data-rate communications in d-band," *IEEE Access*, vol. 8, pp. 9420–9433, 2020.

[219] M. K. Elmezughi, T. J. Afullo, and N. O. Oyie, "Investigating the impact of antenna heights on path loss models in an indoor corridor environment," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2020, pp. 1–7.

[220] Y. Shen, Y. Shao, L. Xi, H. Zhang, and J. Zhang, "Millimeter-wave propagation measurement and modeling in indoor corridor and stairwell at 26 and 38 ghz," *IEEE Access*, vol. 9, pp. 87 792–87 805, 2021.

[221] M. K. Elmezughi, O. Salih, T. J. Afullo, and K. J. Duffy, "Comparative Analysis of Major Machine-Learning-Based Path Loss Models for Enclosed Indoor Channels," *Sensors*, vol. 22, no. 13, pp. 1–25, 2022.

[222] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[223] S. Park, O. Simeone, and J. Kang, "Meta-Learning to Communicate: Fast End-to-End Training for Fading Channels," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May, no. 2017, 2020, pp. 5075–5079.

[224] B. Crow, I. Widjaja, J. Kim, and P. Sakai, "Ieee 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[225] bluetooth SIG, "bluetooth SIG," https://www.bluetooth.com/, 1990, accessed: 2022-10-30.

[226] H. Nawaz, H. M. Ali, and A. A. Laghari, "UAV Communication Networks Issues: A Review," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1349–1369, 2021. [Online]. Available: https://doi.org/10.1007/s11831-020-09418-0

[227] Wikipedia, "ZigBee Alliance," https://csa-iot.org/0, 1990, accessed: 2022-10-30.

[228] J. C. Liberti and T. S. Rappaport, "Geometrically based model for line-of-sight multipath radio channels," *IEEE Vehicular Technology Conference*, vol. 2, no. 2, pp. 844–848, 1996.

[229] C. Seker, M. T. Güneser, and T. Ozturk, "A review of millimeter wave communication for 5g," in *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2018, pp. 1–5.

[230] Engineering and T. H. Wiki, "Milestones: WiFi Precursor, 1987," https://ethw.org/Milestones:WiFi_Precursor,_1987, 1987, accessed: 2022-10-30.

[231] S. C. Ergen, "Zigbee/ieee 802.15. 4 summary," *UC Berkeley, September*, vol. 10, no. 17, p. 11, 2004.

[232] W. Guo, W. M. Healy, and M. Zhou, "Impacts of 2.4-GHz ISM band interference on IEEE 802.15.4 wireless sensor network reliability in buildings," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 9, pp. 2533–2544, 2012.

[233] J.-s. Lee and Y.-c. Huang, "ITRI ZBnode : A ZigBee / IEEE 802 . 15 . 4 Platform for Wireless Sensor Networks," in *International Conference on Systems, Man, and Cybernet*. IEEE, 2006, pp. 1462–1467.

[234] P. Benavidez, K. Nagothu, A. K. Ray, T. Shaneyfelt, S. Kota, L. Behera, and M. Jamshidi, “Multi-domain robotic swarm communication system,” in *IEEE International Conference on System of Systems Engineering, SoSE 2008*, 2008.

[235] T. Andre, K. A. Hummel, and A. P. Schoellig, “Application-Driven Design of Aerial Communication Networks,” *IEEE Communications Magazine*, no. May, pp. 129–137, 2014.

[236] R. K. Megalingam, D. Nagalla, P. R. Kiran, R. T. Geesala, and K. Nigam, “Swarm based autonomous landmine detecting robots,” in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017, pp. 608–612.

[237] B. Balagopal, S. Kerrigan, H. Kim, M.-Y. Chow, M. Bourham, and X. Jiang, “A smart sensor prototype for vibration sensing in nuclear power plants,” in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 1127–1132.

[238] P. Sikka, C. Crossman, P. Corke, D. Swain, P. Valencia, and G. Bishop-Hurley, “Wireless adhoc sensor and actuator networks on the farm,” *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks, IPSN ’06*, vol. 2006, pp. 492–499, 2006.

[239] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello, “Mixed-initiative planning and execution for multiple drones in search and rescue missions,” in *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, vol. 2015-Janua, 2015, pp. 315–323.

[240] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, “LSAR: Multi-UAV Collaboration for Search and Rescue Missions,” *IEEE Access*, vol. 7, pp. 55 817–55 832, 2019.

[241] A. Mostefaoui, M. Raynal, C. Travers, S. Patterson, D. Agrawal, and A. E. Abbadi, “From static distributed systems to dynamic systems,” in *24th IEEE*

Symposium on Reliable Distributed Systems (SRDS'05). IEEE, 2005, pp. 109–118.

[242] S. J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A Survey on Aerial Swarm Robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[243] T. Liu, X. Sun, C. Hu, Q. Fu, and S. Yue, "A multiple pheromone communication system for swarm intelligence," *IEEE Access*, vol. 9, pp. 148 721–148 737, 2021.

[244] S. M. Trenkwalder, I. Esnaola, Y. Kaszubowski Lopes, A. Kolling, and R. Groß, "SwarmCom: an infra-red-based mobile ad-hoc network for severely constrained robots," *Autonomous Robots*, vol. 44, no. 1, pp. 93–114, 2020. [Online]. Available: https://doi.org/10.1007/s10514-019-09873-0

[245] S. Kiesel, E. Burns, and W. Ruml, "Abstraction-guided sampling for motion planning," *Proceedings of the 5th Annual Symposium on Combinatorial Search, SoCS 2012*, pp. 162–163, 2012.

[246] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT star: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2997–3004, 2014.

[247] B. Bollobas, *Modern Graph Theory*, 1st ed. Springer, New York, NY, 1998, vol. 23.

[248] P. Erdös, A. W. Goodman, and L. Pósa, "The Representation of a Graph by Set Intersections," *Canadian Journal of Mathematics*, vol. 18, pp. 106–112, 1966.

[249] M. Schaefer, "Complexity of some geometric and topological problems," pp. 334–344, 2010.

[250] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining.* AAAI Press, 1996, pp. 226–231.

[251] S. Etemadi, R. Vatankhah, A. Alasty, G. R. Vossoughi, and M. Boroushaki, "Leader connectivity management and flocking velocity optimization using the particle swarm optimization method," *Scientia Iranica*, vol. 19, no. 5, pp. 1251–1257, 2012.

[252] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[253] K. E. Parsopoulos, M. N. Vrahatis *et al.*, "Particle swarm optimization method for constrained optimization problems," *Intelligent technologies–theory and application: New trends in intelligent technologies*, vol. 76, no. 1, pp. 214–220, 2002.

[254] S. V. Vaseghi, "Noise and Distortion," *Advanced Digital Signal Processing and Noise Reduction*, vol. 9, pp. 29–43, 2001.

[255] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise," *IEEE Transactions on Communications*, vol. 22, no. 1, pp. 28–38, 1974.

[256] M. S. De Alencar, *Modulation Theory.* River Publishers, 2018, vol. 1.

[257] P. K. Singya, P. Shaik, N. Kumar, V. Bhatia, and M. S. Alouini, "A Survey on Higher-Order QAM Constellations: Technical Challenges, Recent Advances, and Future Trends," *IEEE Open Journal of the Communications Society*, vol. 2, no. February, pp. 617–655, 2021.

[258] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[259] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.

[260] F. Ait Aoudia and J. Hoydis, "End-to-end learning for ofdm: From neural receivers to pilotless communication," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 1049–1063, 2022.

[261] P. Gallinari, Y. Lecun, S. Thiria, and F. Fogelman Soulie, "Memoires associatives distribuees: Une comparaison (distributed associative memories: A comparison)," in *Proceedings of COGNITIVA 87, Paris, La Villette, May 1987*. Cesta-Afcet, 1987.

[262] L. Yasenko, Y. Klyatchenko, and O. Tarasenko-Klyatchenko, "Image noise reduction by denoising autoencoder," in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. IEEE, 2020, pp. 351–355.

[263] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[264] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv*, pp. 1–14, 2016. [Online]. Available: http://arxiv.org/abs/1609.04747

[265] Y. Lai, "Taking the Mystery out of the Infamous Formula," SNR= 6.02 N+ 1.76 dB," and Why You Should Care," *Imid 2009*, pp. 1069–1072, 2009.

[266] C. C. Crouch, "Integration of Mini-UAVss at the Tactical Operations Level: Implications of Operations, Implementation, and Information Sharing," *Naval Postgraduate School*, p. 200, 2005.

[267] R. Bellman, “Markovian decision processes,” *Mathematics in Science and Engineering*, vol. 130, no. C, pp. 172–187, 1957.

[268] S. Feyzabadi and S. Carpin, “Risk-aware path planning using hirerachical constrained Markov Decision Processes,” in *IEEE International Conference on Automation Science and Engineering*, vol. 2014-Janua, no. February, 2014, pp. 297–303.

[269] M. Wiering and M. van Otterlo, *Conclusions, future directions and outlook.* Springer, New York, NY, 2012, vol. 12.

[270] R. E. Mohemed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, “Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks,” *Computer Networks*, vol. 114, pp. 51–66, 2017.

Appendix

Table 1: Mean ±1 Standard Deviation Results, For 50 Sheep, Obtained by the Proposed approach and Existing Algorithms

Metric		Mean±st.dev			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
SR	1/4	100	100	100	100
	1/3	100	100	100	100
	1/2	32	84	70.83	**100**
	2/3	36	40	83	**100**
	3/4	40	40	79.17	**96**
T	1/4	89.7±0.7	90.0±0.9	107.3±1.9	**67.3**±7.4
	1/3	93.1±0.8	92.8±0.8	110.9±4.3	**68.0**±6.8
	1/2	520.6±27.0	437.4±149.8	340.3±182.7	**147.3**±70.7
	2/3	648.9±19.3	688.7±121.0	459.4±152.3	**304.4**±162.3
	3/4	706.0±20.9	777.0±133.8	547.8±151.3	**281.7**±89.1
Δ_{avg}	1/4	**34.1**±0.8	34.0±1.0	32.4± 8.0	32.4± 2.6
	1/3	31.4± 0.8	**31.7**± 1.1	30.4± 0.8	28.8±1.9
	1/2	**27.0** ±18.5	16.4± 8.9	21.1 ±4.1	21.2 ± 4.9
	2/3	**19.3**±7.1	10.5± 3.63	14.9±3.6	17.4±2.1
	3/4	**20.9**± 6.6	9.4±4	15.9± 2.6	14.7±2.9
n_{cc}	1/4	**1.0**±0.0	**1.0**±0.0	**1.0**±0.0	**1.0**±0.0
	1/3	**1.0**±0.0	**1.0**±0.0	**1.0**±0.0	**1.0**±0.0
	1/2	1.0±0.5	**1.0**±0.3	**1.0** ±0.3	**1.0** ±0.2
	2/3	2.6±0.4	**2.2**±0.2	**2.2**±0.4	1.9± 0.2
	3/4	2.8 ±0.5	**2.6** ±0.5	2.6 ±0.4	2.8 ±0.4
% at H	1/4	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	1/3	100 ± 0	100 ± 0	100 ±0	100 ± 0
	1/2	40 ± 44	89 ± 25	77 ± 40	**100** ± 0
	2/3	45 ± 44	73 ± 26	91 ± 23	**100** ± 0
	3/4	47 ± 47	74 ± 27	79 ± 41	**99** ± 3
E_{total}	1/4	845 ± 3	852 ± 1	999 ± 14	559 ± 71
	1/3	833 ± 7.4	839 ± 1.2	1003 ± 37	564 ± 57
	1/2	2845 ± 671	4187 ± 903	2057 ± 1044	1464 ± 429
	2/3	3420 ± 1416	5210 ± 602	3720.6 ± 1118	2253 ± 836
	3/4	6691 ± 757	6124 ± 632	3240 ± 437	2447 ± 336
$\lvert\Gamma^t - \gamma^t\rvert$	1/4	143 ± 1	142 ± 2	110 ± 1	104 ± 6
	1/3	152 ± 3	152 ± 2	118 ± 1	108 ± 5
	1/2	305 ± 7	187 ± 6	121 ± 13	117 ± 10
	2/3	287 ± 119	227 ± 23	131 ± 14	124 ± 9
	3/4	330 ± 12	247 ± 33	137 ± 3	127 ± 12
n_{Tx}	1/4	90 ± 0	90 ± 1	26 ± 2	10 ± 0
	1/3	93 ± 1	93 ± 1	28 ± 3	11 ± 2
	1/2	603 ± 19	540 ± 130	96 ± 70	17 ± 3
	2/3	774 ± 14	716 ± 100	235 ± 100	51 ± 47
	3/4	832 ± 14	850 ± 16	206 ± 33	35 ± 3

Table 2: Numerical Results For 50 Sheep Showing The Best Mean

Metric		Best Mean			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
T	1/4	89	89	104	**60**
	1/3	**91**	**92**	105	**68**
	1/2	100	101	118	**78**
	2/3	340	363	256	**99**
	3/4	378	424	370	**170**
Δ_{avg}	1/4	35.7	**36.1**	34.4	33.0
	1/3	29.3	**29.5**	28.8	29.0
	1/2	**41.3**	22.1	32.1	26.0
	2/3	24.5	18.8	17.9	**25.2**
	3/4	**38.3**	17.3	20.1	16.6
n_{cc}	1/4	**1.0**	**1.0**	**1.0**	**1.0**
	1/3	**1.0**	**1.0**	**1.0**	**1.0**
	1/2	1.8	1.6	**1.0**	**1.0**
	2/3	**1.5**	2.2	2.1	1.9
	3/4	2.5	2.1	1.8	1.8
% at H	1/4	100	100	100	100
	1/3	100	100	100	100
	1/2	100	100	100	100
	2/3	100	100	100	100
	3/4	100	100	100	100
E_{total}	1/4	841	850	990	501
	1/3	828	838	966	513
	1/2	2315	3145	1182	1027
	2/3	2510	4517	2635	1627
	3/4	5855	5548	2963	2081
$\lvert\Gamma^t - \gamma^t\rvert$	1/4	142	139	109	97
	1/3	150	149	116	102
	1/2	300	180	107	108
	2/3	205	207	114	117
	3/4	317	213	135	115
n_{Tx}	1/4	90	89	24	10
	1/3	92	92	25	9
	1/2	592	390	40	14
	2/3	758	601	147	18
	3/4	819	832	180	33

Table 3: Numerical Results For 75 Sheep

Metric		Mean±st.dev			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
SR	1/4	100	100	100	100
	1/3	56	88	83	**92**
	1/2	40	60	70	**88**
	2/3	32	36	62	**88**
	3/4	12	16	33	**88**
T	1/4	**94.6** ± 0.6	94.72 ± 0.6	122.2 ± 2.2	105.2 ± 36.9
	1/3	156.8 ± 160.0	167.1±154.2	**132.4**±16.3	133.0±15.2
	1/2	573.3±217.4	666.7±208.5	319.8±54.7	**288.9**±67.3
	2/3	953.1±209.6	783.0±283.1	542.5±210.0	**424.9**±58.7
	3/4	1132.8±105.3	917.6±310.0	763.7±179.5	**431.3**±48.3
Δ_{avg}	1/4	56.2 ± 1.0	**56.2** ± 1.3	48.7 ± 1.5	37.7 ± 9.0
	1/3	**49.9** ± 18.3	39.2 ± 8.1	38.9 ± 8.8	21.3 ± 9.1
	1/2	21.2± 6.1	**30.2**± 9.8	29.6± 4.8	30.1± 5.6
	2/3	12.2± 6.6	**27.0**± 5.8	17.0± 8.0	22.7± 3.3
	3/4	11.9± 4.8	**31.2**± 12.5	18.2± 9.5	24.4± 2.8
n_{cc}	1/4	**1.0** ±0.0	**1.0** ± 0.0	**1.0** ±0.0	1.1 ±0.2
	1/3	1.3 ± 0.3	1.5 ± 0.4	1.2 ± 0.3	**1.2** ±0.2
	1/2	2.0±0.3	2.2±0.4	1.7±0.3	**1.5**±0.4
	2/3	2.7±0.6	2.7±0.3	**1.1**±1.2	2.2±0.4
	3/4	3.6±0.9	2.8±0.3	**1.5**±1.2	2.2±0.4
% at H	1/4	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	1/3	94± 19	**100**± 0	**100** ± 0	**100** ± 0
	1/2	83±29	50±44	**100** ± 0	**100** ± 0
	2/3	68±24	61±43	78± 38	**100**± 0
	3/4	66±31	58±45	88±29	**100** ± 0
E_{total}	1/4	831±5	835±4	805±8	794±5
	1/3	1152±567	1276±986	1113±907	749±616
	1/2	4243±1444	4545±1904	2004±1021	1644±972
	2/3	6898±979	5097±2086	2303.4±1445	1778±1111
	3/4	8540±908	6200±2258	3891±1533	2998±1098
$\|\Gamma^t - \gamma^t\|$	1/4	150±2	151±2	149±2	149±2
	1/3	175±34	172±19.0	169±30	160±27
	1/2	231±40	315±61	175±40	167±27
	2/3	267±32	323±58	173±33	178±41
	3/4	270±24	311±27	200±34	187±10
n_{Tx}	1/4	92±1	93±1	20±0	20±0
	1/3	157±160	167±154	23±5	15±8
	1/2	573±217	666.7±209	38±6	19±2
	2/3	953±210	783±283	163±175	43±4
	3/4	1133±105	918±310	139±187	100±8

Table 4: Numerical Results For 75 Sheep Showing The Best Mean

Metric		Best			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
T	1/4	94	93	118	**75**
	1/3	**95**	**95**	120	116
	1/2	285	315	**177**	184
	2/3	348	355	**200**	342
	3/4	779	395	**270**	328
Δ_{avg}	1/4	57.8	**58.3**	51.2	47.3
	1/3	**77.5**	48.7	50.9	36.4
	1/2	32.4	**56.7**	38.4	39.0
	2/3	34.5	**36.0**	28.9	29.3
	3/4	22.8	**50.5**	29.5	28.6
n_{cc}	1/4	1.0	1.0	1.0	1.0
	1/3	1.0	1.0	1.0	1.0
	1/2	1.5	1.7	1.0	1.0
	2/3	1.8	2.3	**1.0**	1.5
	3/4	2.2	2.1	**1.0**	1.7
% at H	1/4	100	100	100	100
	1/3	100	100	100	100
	1/2	100	100	100	100
	2/3	100	100	100	100
	3/4	100	100	100	100
E_{total}	1/4	820	826	792	785
	1/3	810	812	722	610
	1/2	2303	1776	1592	1283
	2/3	3011	2328	2192	1529
	3/4	5579	3198	2994	2600
$\lvert\Gamma^t - \gamma^t\rvert$	1/4	147	148	146	146
	1/3	156	160	150	148
	1/2	184	221	174	163
	2/3	219	231	170	166
	3/4	235	247	185	172
n_{Tx}	1/4	91	92	20	9
	1/3	95	95	20	13
	1/2	285	315	25	22
	2/3	348	355	45	39
	3/4	779	395	63	59

Table 5: Numerical Results For 100 Sheep Showing The Mean ±1 Standard Deviation

Metric		Mean±st.dev			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
SR	1/4	100	100	100	100
	1/3	100	100	100	100
	1/2	44	68	68	**100**
	2/3	28	32	60	**88**
	3/4	20	20	40	**88**
T	1/4	92 ±0	92 ± 1	122 ± 2	**67** ± 20
	1/3	143 ± 107	132 ± 108	251 ± 228	**92** ± 24
	1/2	619 ± 237	591 ± 211	726 ± 207	**84** ± 53
	2/3	924 ± 205	958 ± 163	1019 ± 254	**445** ± 294
	3/4	1034 ± 247	1113 ± 128	1062 ± 500	**499** ± 288
Δ	1/4	45.7 ± 1.0	45.9 ± 1.1	**49.0** ± 1.5	36.4 ± 3.8
	1/3	**41.0** ± 6.4	36.9 ± 6.6	38.6 ± 9.1	30.4 ± 6.0
	1/2	31.0 ± 12.5	20.3 ± 7.6	27.3 ± 5.3	**33.6** ± 4.1
	2/3	**27.5** ± 10.7	11.7 ± 3.6	25.7 ± 4.3	17.7 ± 5.9
	3/4	**34.4** ± 14.1	11.1 ± 3.2	16.2 ± 5.8	19.3 ± 31.3
n_{cc}	1/4	1.0 ±0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	1/3	1.0 ± 0.1	1.0 ± 0.1	1.2 ± 0.3	**1.0** ± 0.1
	1/2	2.2 ± 0.4	2.0 ± 0.3	2.5 ± 0.3	**1.1** ± 0.2
	2/3	2.7 ± 0.2	2.7 ± 0.7	3.0 ± 0.4	**2.4** ± 0.5
	3/4	**2.8** ± 0.4	3.3 ± 0.9	3.7 ± 1.3	2.9 ± 1.0
% at H	1/4	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	1/3	**100** ± 0	**100** ± 0	96 ± 9.6	**100**± 0
	1/2	53 ± 44	87 ± 21	83 ± 32	**100** ± 0
	2/3	55 ± 39	76± 22	71 ± 40	**94** ± 18
	3/4	27 ± 40	70 ± 25	67 ± 40	**91** ± 27
E_{total}	1/4	817± 5	821± 7	1092± 33	1110± 368
	1/3	1559± 1143	2242± 1401	1883± 1437	2267± 1146
	1/2	4777± 2377	5526± 1883	4944± 1268	4192± 2171
	2/3	7046± 3033	9299± 1846	6768± 1610	6136.4± 2679
	3/4	8809± 3343	10821± 1584	6937± 3187	6630± 3012
$\|\Gamma^t - \gamma^t\|$	1/4	157.6± 1.6	158.2± 1.8	117.7± 0.9	105.8± 8.5
	1/3	225.6± 83.5	201.2± 36.4	130.2± 27.4	117.3± 15.9
	1/2	296.7± 44.8	255.5± 35.5	164.1± 37.8	168.6± 40.6
	2/3	322.2± 50.6	292.7± 17.6	183.7± 29.7	188.8± 46.4
	3/4	347.2± 55.2	288.4 ± 16.4	217.2± 63.1	176.7± 24.2
n_{Tx}	1/4	94± 1	94± 1	34± 2	14± 10
	1/3	421± 298	316± 215	100± 114	54± 91
	1/2	796± 298	772± 262	332± 107	138± 126
	2/3	1055± 385	1236± 198	482± 127	137± 141
	3/4	1413± 229	1410± 163	515± 251	147± 110

Table 6: Numerical Results For 100 Sheep Showing The Best Mean

Metric		Best Mean			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
T	1/4	91	91	118	**71**
	1/3	95	95	125	**67**
	1/2	183	211	396	**72**
	2/3	455	163	461	**144**
	3/4	381	**128**	248	225
Δ	1/4	47.9	48.2	**51.2**	43.2
	1/3	**60.8**	42.3	51.0	38.3
	1/2	**61.8**	33.4	39.5	40.0
	2/3	**53.8**	17.9	33.8	29.9
	3/4	**60.9**	17.5	27.8	31.3
n_{cc}	1/4	1.0	1.0	1.0	1.0
	1/3	1.0	1.0	1.0	1.0
	1/2	1.4	1.5	1.9	**1.0**
	2/3	2.1	1.9	2.3	**1.4**
	3/4	2.0	2.3	2.3	**2.0**
% at H	1/4	100	100	100	100
	1/3	100	100	100	100
	1/2	100	100	100	100
	2/3	100	100	100	100
	3/4	100	100	100	100
E_{total}	1/4	807	803	1017	664
	1/3	802	800	1067	1090
	1/2	1747	2086	2810	1364
	2/3	2252	5426	3267	2833
	3/4	2884	6581	1777	3367
$\lvert\Gamma^t - \gamma^t\rvert$	1/4	155.0	154.4	116.3	85
	1/3	170.5	170.1	93.5	90.9
	1/2	214.2	206.7	132.6	121.4
	2/3	242.1	245.5	134.6	152.0
	3/4	262.4	256.6	115.7	142.0
n_{Tx}	1/4	94	93	30	8
	1/3	99	99	38	9
	1/2	307	226	172	16
	2/3	374	668	201	44
	3/4	394	931	106	36

Table 7: Numerical Results For 200 Sheep Showing The Mean ±1 Standard Deviation

Metric		Mean±st.dev			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
SR	1/4	100	100	100	100
	1/3	55	64	**100**	**100**
	1/2	44	12	92	**100**
	2/3	40	8	72	**80**
	3/4	0	0	40	**64**
T	1/4	164.0 ±189.3	**142.4** ± 79.4	272.5 ± 49.1	480.8 ± 251.0
	1/3	1438.0 ± 493.0	918.0 ± 373.0	596.2 ± 137.4	**476.0** ± 24.3
	1/2	2172.0 ± 681.0	1755.0 ± 261.0	1140.0 ± 260.0	**684.7** ± 453.1
	2/3	1914.0 ± 649.0	2343.0 ± 231.0	1797.0 ± 352.6.0	**1266.0** ± 333.2
	3/4	2267.0 ± 970.0	2684.0 ± 420.0	2192.0 ± 442.5	**1908.0** ± 409.0
Δ	1/4	**82.3** ± 18.6	81.2 ± 17.8	15.8 ± 1.5	15.5 ± 23.8
	1/3	**87.9** ± 42.4	39.9 ± 17.2	8.6 ± 4.4	30.5 ± 6.1
	1/2	**77.8** ± 37.8	22.3 ± 7.4	5.8 ± 13.4	33.6 ± 4.1
	2/3	**81.6** ± 38.7	22.9 ± 6.1	25.1 ± 7.3	17.0 ± 7.9
	3/4	**37.9** ± 18.4	24.9 ± 3.8	16.0 ± 8.5	16.5 ± 8.0
n_{cc}	1/4	2.8 ±0.3	2.8 ± 0.3	**1.7** ± 0.2	1.8 ± 0.2
	1/3	2.4 ± 0.6	2.3 ± 0.3	2.3 ± 0.3	**2.2** ± 0.1
	1/2	**3.0** ± 0.5	3.1 ± 0.7	**3.0** ± 0.3	3.1 ± 1.2
	2/3	**2.6** ± 0.6	3.8 ± 0.8	3.5 ± 0.5	2.8 ± 0.5
	3/4	**3.3** ± 0.4	4.3 ± 0.6	4.4 ± 1.9	4.2 ± 1.2
% at H	1/4	98 ± 10	**100** ± 0	**100** ± 0	**100** ± 0
	1/3	28 ± 25	32 ± 25	**100** ± 0	**100**± 0
	1/2	22 ± 25	6 ± 62	93 ± 22	**100** ± 0
	2/3	0 ± 0	35 ± 39	88 ± 31	**93** ± 22
	3/4	0 ± 0	24 ± 42	43 ± 48	**80** ± 38
E_{total}	1/4	1179± 1339	1008± 435	2004± 350	1110± 369
	1/3	7108 ± 3769	6474 ± 2634	1884± 1438	2267± 1146
	1/2	12617± 5001	13039 ± 2253	10945± 1269	11192± 2171
	2/3	10892 ± 6477	18162 ± 3195	13207± 2592	11305± 2450
	3/4	22420 ± 2331	21618± 1643	16111 ± 3252	14024 ± 3006
$\|\Gamma^t - \gamma^t\|$	1/4	190± 33	180± 2	110± 9	101± 8.9
	1/3	334 ± 61	254 ± 42	130± 27	117± 16
	1/2	387± 40	296± 16	154± 31	128± 21
	2/3	360± 55	310± 17	112 ± 50	240± 22
	3/4	214 ± 33	341 ± 22	127± 63	242± 23
n_{Tx}	1/4	164 ±189	142 ± 79	65± 3	20± 18
	1/3	1438 ± 493	918 ± 373	191 ± 17	49 ± 14
	1/2	2172 ± 681	1755 ± 261	140 ± 29	85 ± 53
	2/3	1914 ± 649	2343 ± 231	410 ± 364	305 ± 183
	3/4	1267 ± 97	2684 ± 20	659 ± 505	508 ± 495

Table 8: Numerical Best Results For 200 Sheep Showing The Best Mean

Metric		Best Mean			
Metric	Model	Strombom [59]	CADSHEEP [17]	SOHP	SOMCP
T	1/4	**100.0**	**100.0**	211.0	210.0
	1/3	476.0	**333.0**	377.0	373.0
	1/2	786.0	895.0	833.0	**672.0**
	2/3	1894.5	1961.0	**1272.0**	1306.0
	3/4	**801.0**	2654.0	1485.0	1436.0
Δ	1/4	**96.9**	96.2	51.2	46.5
	1/3	**147.7**	70.4	42.0	38.4
	1/2	**157.0**	39.6	45.8	40.0
	2/3	**137.0**	33.5	17.0	40.0
	3/4	**119.7**	32.7	46.0	40.0
n_{cc}	1/4	2.0	**1.0**	**1.0**	**1.0**
	1/3	2.0	2.0	2.0	2.0
	1/2	1.4	1.5	1.9	**1.3**
	2/3	2.5	**2.2**	2.3	2.7
	3/4	**3.0**	3.1	3.5	3.2
% at H	1/4	100	100	100	100
	1/3	100	100	100	100
	1/2	100	100	100	100
	2/3	0	**100**	**100**	**100**
	3/4	0	98	**100**	**100**
E_{total}	1/4	737	737	1017	665
	1/3	803	800	1068	1090
	1/2	10329	3770	3099	3086
	2/3	3099	8833	11287	9544
	3/4	12320	18902	15545	12001
$\|\Gamma^t - \gamma^t\|$	1/4	173	175	101	97
	1/3	250	203	197	91
	1/2	324	316	142	101
	2/3	54	297	105	202
	3/4	197	316	117	201
n_{Tx}	1/4	94	93	30	8
	1/3	476	333	175	38
	1/2	786	895	95	72
	2/3	661	1961	401	264
	3/4	801	2654	548	444

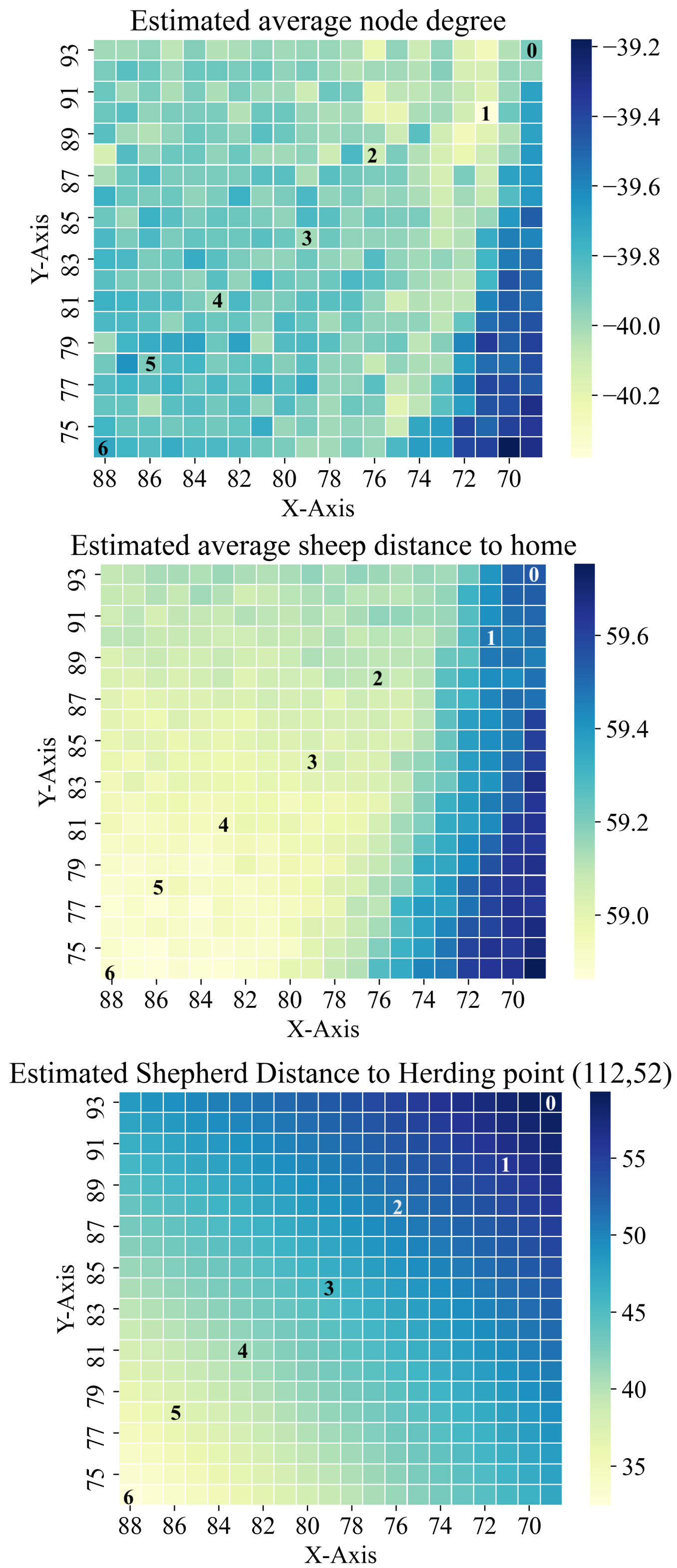

Figure A.I: The estimated (a) Δ_{avg}, (b) average sheep distance to home,and (c) the distance between the shepherd and the last herding point in the herding path (from left to right), the numbers represent the points selected for σ_{opt}

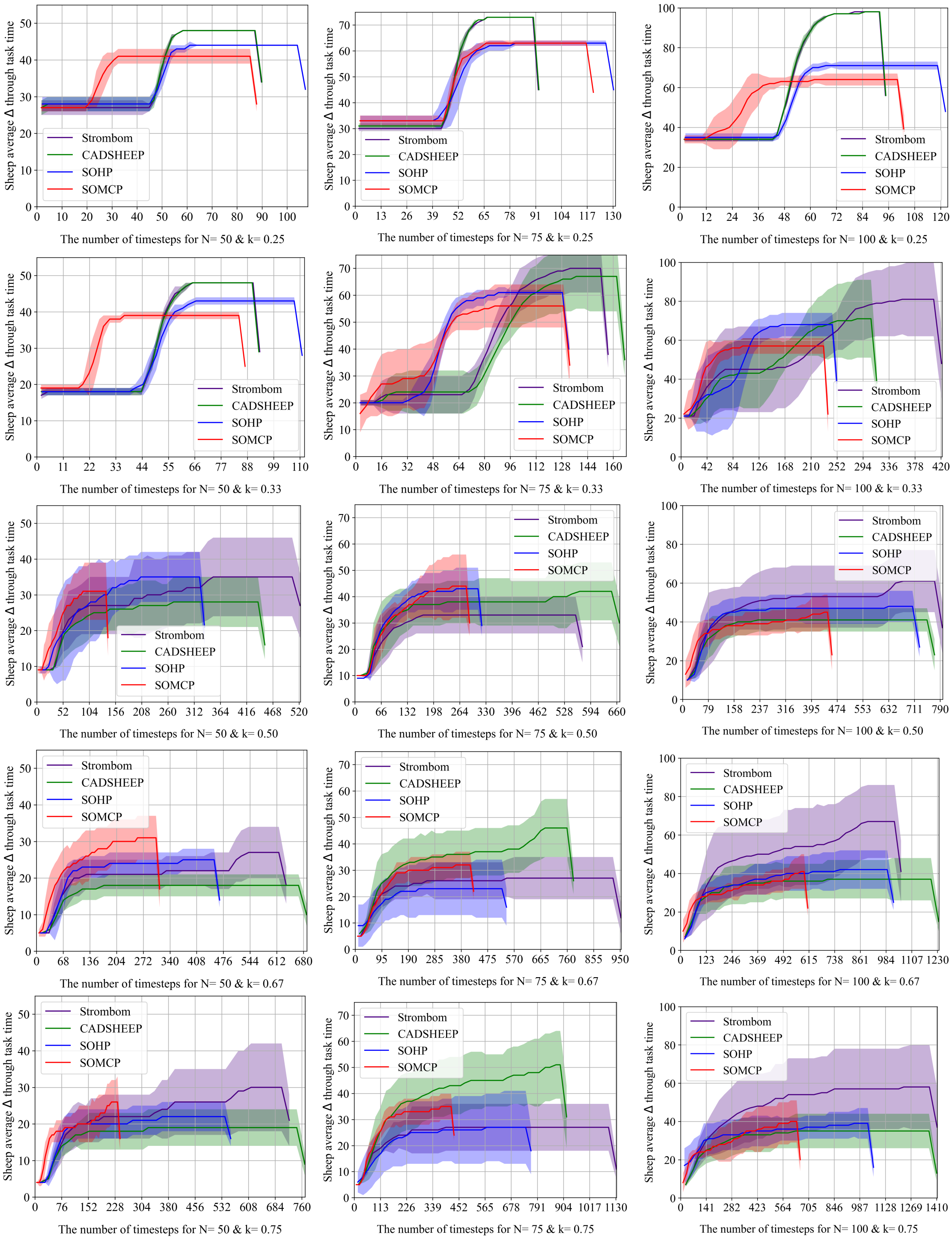

Figure A.II: The change in the average node degree (+/−) one standard deviation for N=50 (left), N=75, and N=100 throughout the task time measured at different k=1/4,1/3,1/2,2/3,3/4 from top to bottom

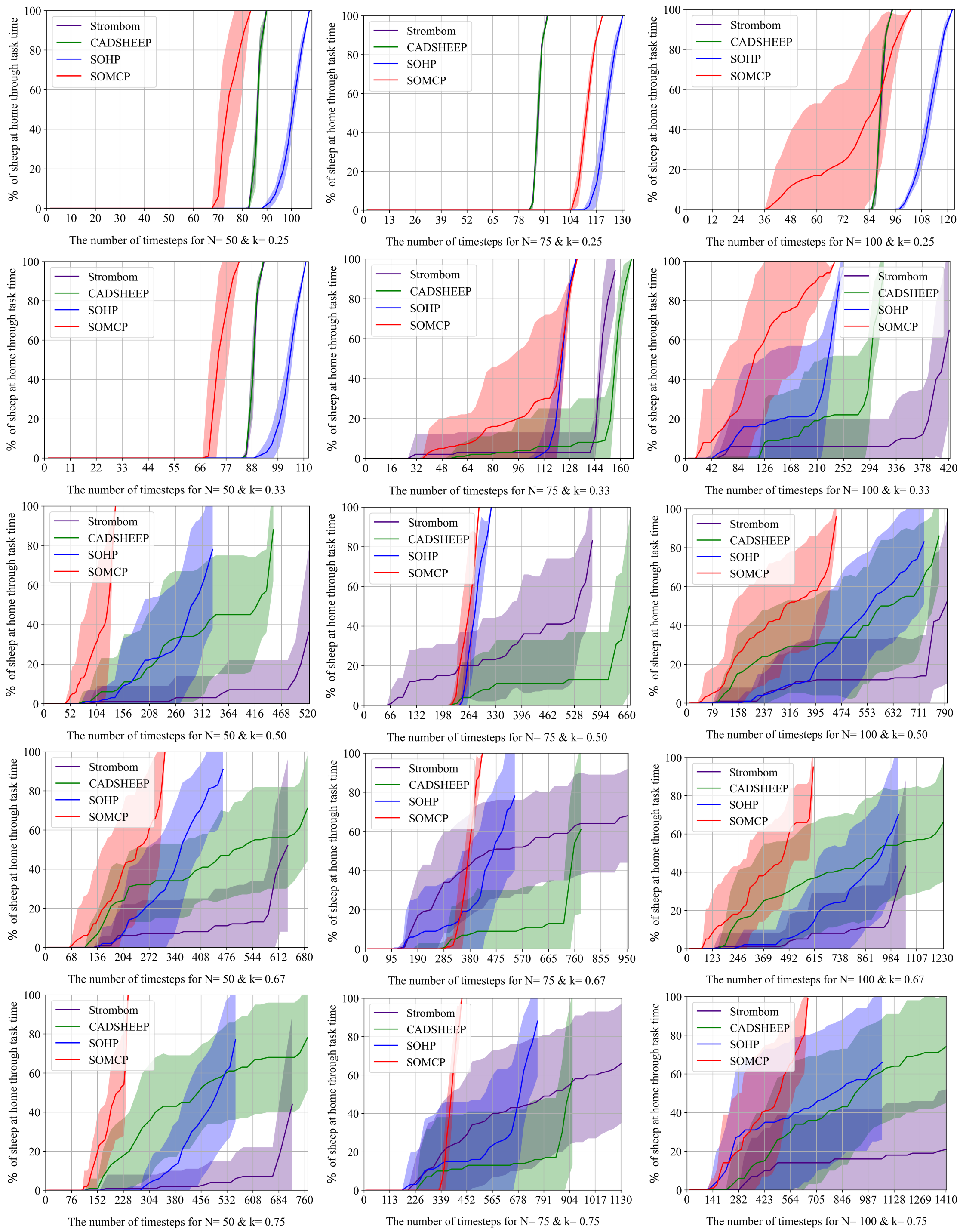

Figure A.III: The change in the average percentage of sheep reaching home (+/−) one standard deviation for N=50 (left), N=75 and N=100 (right) throughout the task time measured at different k=1/4,1/3,1/2,2/3,3/4 from top to bottom

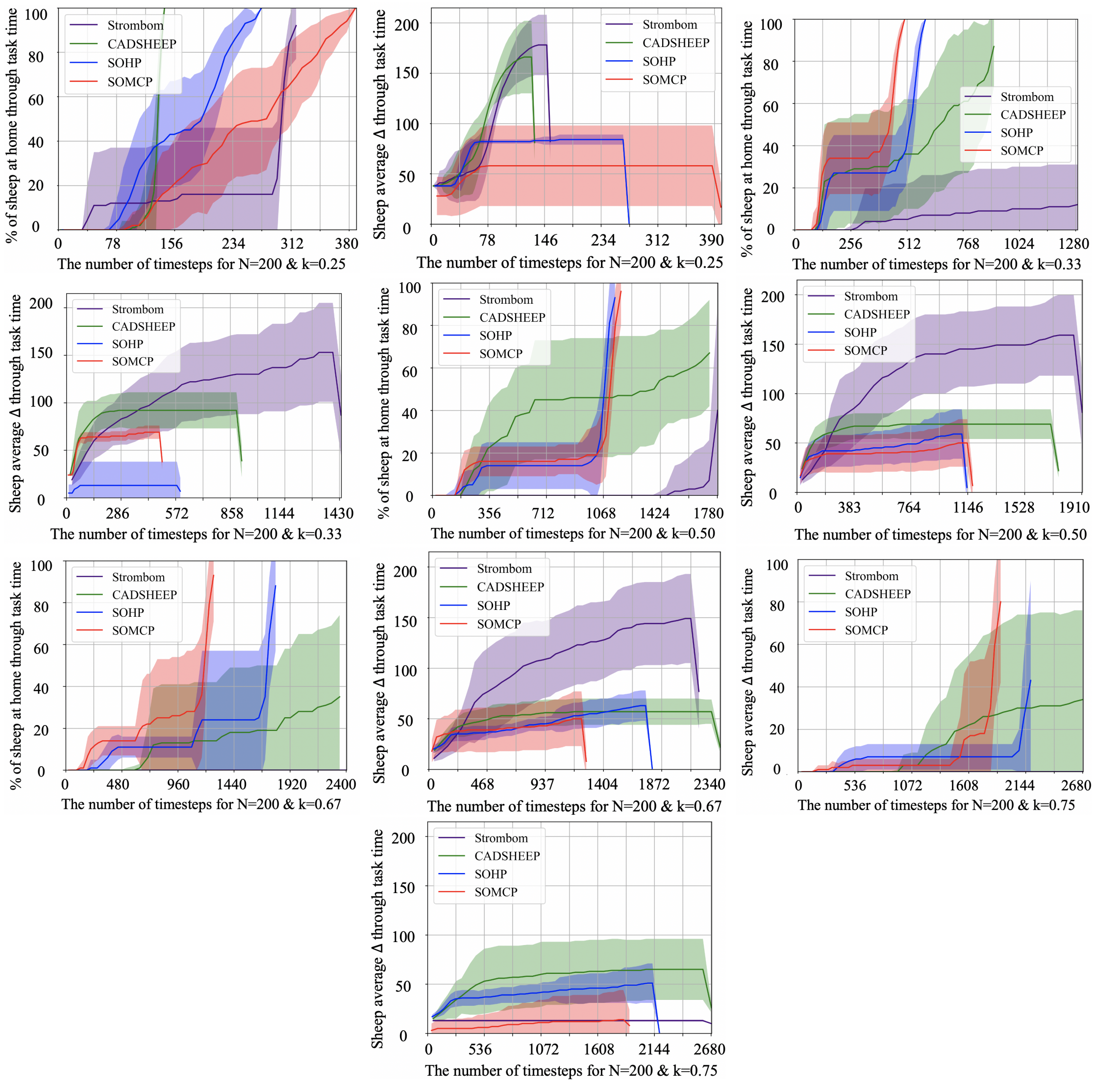

Figure A.IV: The change in the average percentage of sheep reaching home (left) and the average node degree of sheep (+/−) one standard deviation for N=200 measured at different k=1/4,1/3,1/2,2/3,3/4 from top to bottom