Enhanced Broadcast Control for Coverage Tasks of Multi-agent Systems

Shalini Darmaraju
B.Eng. (Hons) (Mechanical-Aeronautics)

A thesis submitted for the degree of Doctor of Philosophy at
Monash University in 2022
School of Engineering

COPYRIGHT NOTICE

ABSTRACT

Multi-agent system technologies have earned substantial research interest over the past two decades due to its potential for a wide variety of engineering and industrial applications. A promising application is to realise autonomous multi-vehicle and multi-robot systems. Traditionally, multi-agent control research focuses on the design of centralised, decentralised or distributed systems. However, an upcoming and different perspective of control technique termed the Broadcast Control (BC) has been proposed for multi-agent systems. BC is a stochastic gradient-based method and advantageous because it greatly decreases the communication volume of the whole system. A variant of BC, Pseudo-perturbation-based Broadcast Control (PBC) showcases higher performance compared to the BC scheme. This thesis proposes the further enhancement of existing BC schemes and explores the scheme's possibilities for applications for physical agents in a coverage task. The first part of the thesis develops a broadcast control scheme named *Multi-step Broadcast Control (MBC)* that incorporates virtual multi-steps along a horizon. This scheme differs from PBC in the literature, which incorporates virtual multi-steps from the current location. The second part of the thesis aims to test the adaptability of the MBC scheme and its suitability for physical agents. The MBC scheme has been developed to include agent path constraints to achieve this. Finally, a novel predictive BC scheme named *Receding Horizon based Broadcast Control (RHBC)* that samples multiple horizons in the virtual layer in a cyber-physical model is developed. The proposed enhancements stand as the best performing BC scheme to date for the coverage task of multi-agent systems. The enhancements are also adaptable for coverage applications involving physical agents.

DECLARATION

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature : .....

Print Name : .....

Date : .....

LIST OF PUBLICATIONS

Published

S. Darmaraju, M.A.S. Kamal, M. Shanmugavel, and C.P. Tan, "Coverage control of mobile agents using multi-step broadcast control," Robotica, pp. 1–16, 2022.

S. Darmaraju, M.A.S. Kamal, M. Shanmugavel and C.P. Tan, "Area coverage by a group of UAVs using the broadcast control framework," IFAC-PapersOnLine, 52(12), pp. 370-375, 2019.

S. Darmaraju, M.A.S. Kamal, M. Shanmugavel and C.P. Tan, "Coverage Control of a Mobile Multi-Agent Serving System in Dynamical Environment," in 2018 Joint 7th International Conference on Informatics, Electronics and Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision and Pattern Recognition (icIVPR), Kitakyushu, Japan, pp. 508-513, 2018.

Under preparation for submission

S. Darmaraju, M. Shanmugavel, M.A.S. Kamal, and C.P. Tan, "Multi-step broadcast control for area coverage by multiple unmanned aerial vehicles".

S. Darmaraju, M.A.S. Kamal, and C.P. Tan, "Receding horizon based broadcast control for coverage tasks of multi-agent systems".

ACKNOWLEDGEMENTS

Completing my PhD was indeed a marathon and arduous event, and I would not have been able to complete this journey without the aid and support of the following individuals:

First and foremost a sincere appreciation and gratitude to all my supervisors. Professor Ir. Edwin Tan Chee Pin has been an exceptional mentor. His patience, guidance and constant feedback have taught me the mechanics of research and got me to the end despite various challenges. His commitment to research and excellence was a great inspiration. I am indebted to Associate Professor Md. Abdus Samad Kamal for introducing me to the field of Broadcast Control. His advice on the topic, his sharp mind, and patience have been very helpful throughout my PhD journey. I owe my gratitude to Associate Professor Madhavan Shanmugavel, who accepted me as a graduate student and for the co-supervision.

I would also like to acknowledge the Malaysian Ministry of Higher Education for financial support through their Fundamental Research Grant Scheme. Sincere appreciation also goes to Monash University. Ancora Imparo; yes, I am always learning. The journey has taught me that knowledge is astronomical, and we truly stand on the shoulders of giants. I want to convey my warm thank you to all my fellow postgraduate friends who have journeyed along the way. I am honoured to dedicate this thesis to my parents and brother, who have given me their all. Their love, support and blessings empower me. This thesis is my humble offering at the Divine Lotus Feet of God.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction and Motivation

Multi-agent systems encompass multiple entities known as agents that collaborate to achieve predefined individual or collective objectives in a shared environment [1]. The term agent has been traditionally used in the field of software engineering and computer science to mean *"an encapsulated computer system that is situated in some environment and is capable of flexible, autonomous action in that environment in order to meet its design objectives"* [2]. Examples of multi-agent systems are prevalent in nature and can be seen in flocks of birds flying in migration, schools of fish swimming and in groups of ants transporting food.

Multi-agent systems play an essential role in both defence and civilian sectors, potentially affecting areas such as search and rescue, transportation, and surveillance. Individual agents (e.g., unmanned aerial/ground/underwater vehicles) in those systems have restricted capabilities owing to short sensing and communication ranges and small processing capacity. However, compared to a single intelligent agent, their collective behaviour has substantial benefits, such as high scalability, large-scale spatial distribution, robustness, and low cost [3]. The deployment of large-scale multi-agent systems with restricted costs and smaller sizes

can accomplish tasks that would otherwise be impossible to do by a single agent. Multi-agent systems are helpful in a variety of applications, including search and rescue [4], surveillance [5, 6], space exploration [7], tracking/classification [8–10], and radiation shielding and site cleaning [11].

Multi-agent systems have also been studied and used in fields such as cooperative mobile robotics [12], wireless sensor networks [13], distributed artificial intelligence and computing [14,15], social studies [16], biology [17], smart grids [18], traffic management [19, 20], and supply-chain management [21]. Coverage control is an appealing coordination strategy for many of the aforementioned applications as it allows a group of mobile robots to spatially distribute themselves based on the relative importance of different areas within a domain, which is classically defined by spatial fields and denoted in the literature as density functions.

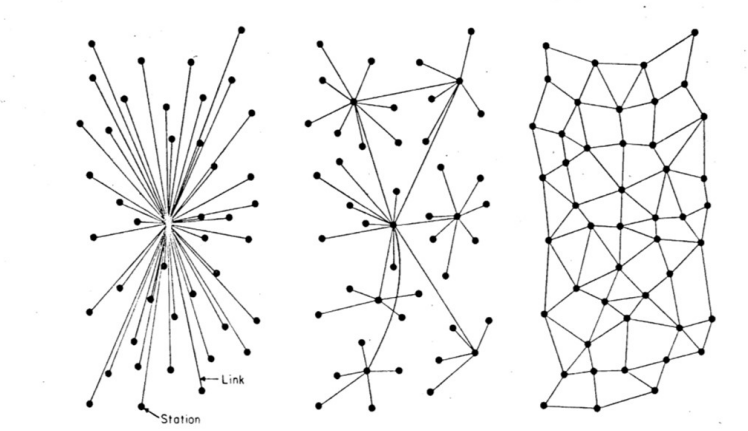


Figure 1.1: Types of communication network (a) centralized, (b) decentralised and (c) distributed by Baran [22].

In the literature, three types of prevalent control techniques for multi-agent systems are: (a) centralised [23], (b) decentralised [24], and (c) distributed multi-agent control [3]. The communication network of these techniques can be visualised as in Fig. 1.1. The centralised control approach assumes global knowledge

of the multi-agent system. It attempts to achieve some control target while considering all agents' states, which necessarily suffers from scalability issues. The decentralised control scheme computes control actions based only on an agent's local information. In contrast, the distributed control scheme computes control actions based on both the agent's knowledge and the surrounding agents' information.

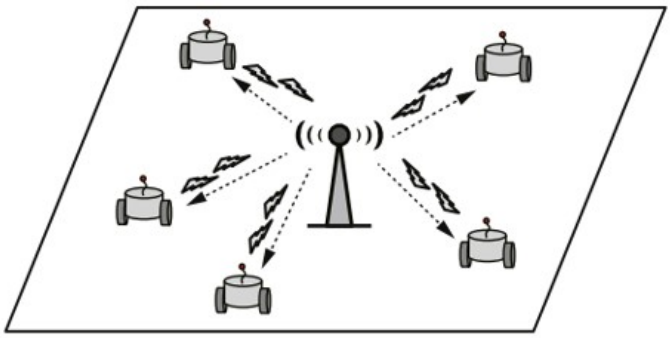


Figure 1.2: Multi-agent system based on broadcast proposed by Azuma et al. [25]

Both decentralised and distributed control algorithms provide scalable solutions and may be applied with little connectivity. All centralised, decentralised, and distributed systems focus on agent-to-agent communication; however, a new approach or outlook has been recently proposed for multi-agent systems, known as Broadcast Control (BC) (Fig. 1.2). BC aims to solve the multi-agent problem with a one-to-all broadcast system where the group performance is observed, and the same signal is sent to all agents indiscriminately. It is relatively new in the research field, and hence there is much unexplored in BC schemes for multi-agent systems. Therefore the study of the BC schemes is the primary interest and focus of this thesis.

This thesis proposes enhancements for the BC scheme. The proposed approaches are twofold as follows:

- Firstly, individual agents take virtual multi-steps along a horizon before a physical step is taken. This scheme is named Multi-step Broadcast Control (MBC). Through this, the agents are capable of a foresighted target of the environment, and it is useful considering uncertainties in a complex environment.

- Secondly, a complete BC scheme is run multiple times in a receding horizon manner within a cloud framework (cyber layer) before positional data is transmitted to physical agents in the physical layer. This scheme is named Receding Horizon based Broadcast Control (RHBC).

In RHBC the virtual agents in the clouds are connected to their respective physical world agents in the environment via wireless transmission. Specifically, the physical agents are considered to be seamlessly integrated into the cloud network. They can be controlled remotely and act as points of access to the physical world from the virtual world. This setup places the intelligent processing workload (control scheme) in cloud architecture.

The choice between MBC or RHBC scheme lies in the type and scale of the problem, the number of agents and the resources available to the users. As BC schemes in the literature do not include agent motion constraints, this aspect is explored in this thesis. A decision-making scheme incorporated into the BC scheme considering agent path constraints will produce less erratic, continuous and realistic manoeuvres for the agents.

1.2 Thesis Structure

This thesis is organised as follows. Chapter 2 presents a survey of the literature that underlies the thesis. First, the multi-agent coverage coordination task is introduced. Next, the origins of broadcast control are explored. The integration of BC and the receding horizon control (RHC) method are also explored. Finally, the possibility of combining BC and RHC in a cyber-physical system (CPS) framework is reviewed.

In Chapter 3, the proposed MBC scheme is introduced. The necessary background material on existing BC schemes is presented to assist the interpretation of the proposed MBC. The performance of the proposed MBC scheme is tested against existing BC schemes in a coverage problem with varying density. The work in this chapter has been published; its details are as follows: S. Darmaraju, M. A. S. Kamal, M. Shanmugavel, and C.P. Tan, "Coverage control of mobile agents using multi-step broadcast control," Robotica, pp. 1–16, 2022 and S. Darmaraju, M.A.S. Kamal, M. Shanmugavel and C.P. Tan, "Coverage Control of a Mobile Multi-Agent Serving System in Dynamical Environment," in 2018 Joint 7th International Conference on Informatics, Electronics and Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision and Pattern Recognition (icIVPR), Kitakyushu, Japan, pp. 508-513, 2018.

The decision-making process to incorporate agent path constraints into the MBC scheme is presented in Chapter 4. The agent path is designed as a subset of Dubin's model, and its effect on the travel path is studied in the coverage problem using varying Gaussian density functions. Furthermore, the scheme's adaptability

is tested by varying the number of agents in the MBC scheme. The work in this chapter has been published; its details are as follows: S. Darmaraju, M.A.S. Kamal, M. Shanmugavel and C.P. Tan, "Area coverage by a group of UAVs using the broadcast control framework," IFAC-PapersOnLine, 52(12), pp. 370-375, 2019.

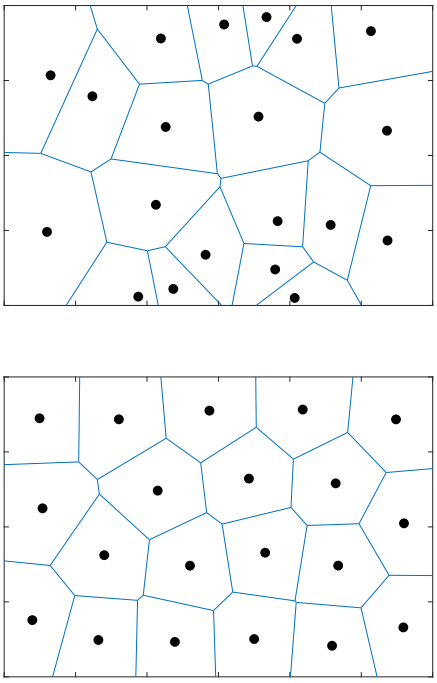
In Chapter 5, a different approach to predictive BC is developed as the RHBC scheme. In this version, the BC scheme is entirely run in the cyber layer of a CPS infrastructure and results from a coverage control study are presented.

Finally, Chapter 6 summarises the contributions of the thesis and provides suggestions for future research directions in this area.

2 | Literature Review

2.1 Introduction

The deployment of a large group of agents for various applications using multi-robot and multi-vehicle systems is becoming increasingly popular. They are inherently robust to failures of single-agent systems or communication links. Coverage coordination is one of the fundamental tasks used to test such multi-agent systems' efficiency. Coverage coordination tasks deal with how agents are deployed over a domain of interest and are capable of various challenging applications such as search and rescue, surveillance and environmental monitoring. The details of the coverage task and control approaches utilised for agent deployment will be reviewed in this chapter. In literature, a recent stochastic optimisation technique, the broadcast control (BC) scheme, and its variant has been introduced as a low communication-based paradigm for multi-agent systems and were shown to be applicable to coverage tasks. The origins of BC and its extensions are discussed as part of the literature survey for the thesis's work.



(a) (b)

Figure 2.1: The coverage environment partitioned to Voronoi cells for a set of 20 agents. (a) Non-optimal coverage (b) Optimal coverage.

2.2 Multi-agent Coverage Coordination Task

Coverage is a coordination task that deals with how to deploy agents in order to optimise surveillance of a domain of interest. The domain of interest is typically represented in Voronoi diagrams, and each agent will be in charge of a partitioned Voronoi cell. For the given finite set of points $x_1, ..x_n$, each point, x_k, has a corresponding Voronoi cell, R_k, that consists of every point in the plane whose distance to x_k are less than or equal to its distance to any other point from the set. It makes sense to prescribe an equal division among the agents in the scenario of homogeneous teams when the mass of each region is the same for all. Figure 2.1 shows the conditions of non-optimal and optimal coverage of agents for a

given domain. Gage [26] introduced the notion of coverage as one of the ensemble motion behaviour models of multi-robot systems. Gage defines three basic types of coverage:

- blanket coverage, where the goal is to achieve a static arrangement of nodes that maximises the total detection area;
- barrier coverage, where the goal is to minimise the probability of undetected penetration through the barrier;
- sweep coverage, which is more-or-less equivalent to a moving barrier.

According to taxonomy by Gage [26], the deployment task discussed in the thesis is a blanket coverage problem. An example of an application using blanket coverage is a communications relay task. Similarly, distributing submerged robots equally across a coral reef to monitor coral health, deployment of wheeled robots with cameras to cover a room for surveillance, groups of robots deployed to perform clean-up over an oil spill or demining robots positioned to service a minefield are examples of coverage applications for multi-robots [27].

On the other hand, the coverage problem is similar to the conventional art gallery problem in computational geometry [28]. The art gallery issue aims to identify the smallest number of cameras that may be put in a polygonal setting to observe the complete gallery environment. While there are several techniques for solving the art gallery problem, they all presume previous solid knowledge of the domain.

The objective of the coverage task is to guide networked mobile agents to the final configuration by optimising a cost function. The coverage cost (objective)

function is frequently used in coverage tasks to quantify how successfully agent networks cover a mission field [27, 29–31]. Various coverage cost functions have been considered in the literature; the total sensing performance of a sensor network [32–36], the joint event detection probability [37–39], and as well as the reaction time from a sensor network to any location in a mission field [40–42].

Cortés et al. [32] proposed a controller for multi-robot sensor coverage that drives the robots continuously toward the centroids of Voronoi cells. This fundamentally geometric technique is a seminal work and has been widely used. It has been extended to robots with a restricted sensing radius [40], heterogeneous groups of robots and non-convex settings [43, 44], and to integrate learning in unfamiliar situations [34]. Bullo et al. [45] and Martinez et al. [46] provide much of the literature on distributed coverage systems in a unified manner, while Schwager et al. [47, 48] successfully built coverage controllers for robotic systems. Recently, Inoue et al. [49] have proposed a generalised Voronoi-tesselation method in an optimal transport framework, which was shown to achieve better control than the basic Voronoi-tesselation method [32].

Li and Cassandras [37], for example, offer a method for placing robots to optimise the probability of detecting an event in the environment. Another common technique is for robots to drive away from or toward one another by following the negative gradient of artificial potential fields. These have been utilised for sensor coverage [50]. Probability-based control methods for coverage were put forward by Izumi et al. [51] and Inoue et al. [52]. Though geometric, potential field, and probabilistic approaches to multi-agent/robot coverage have different models and purposes, they are all based on optimisation and controlled by controllers that solve this optimisation through the evolution of a dynamical system [27].

The work by Sunan et al. [53], is the latest review for coverage tasks with a particular focus on unmanned aerial vehicles. In their work, they have categorised previous work into ten classes; search space-based, sensing function, potential function, spherical model, Voronoi partition, sensed space partition, detection probability, and Markov decision-based methods. In the work of Sunan et al. [53], there was no obvious category stated for broadcast control. The application of broadcast control is minimal for coverage control and has been only shown in the works of Azuma et al. [25] and Ito et al. [54]. The BC method is relatively new in multi-agent systems as it provides a new outlook away from most literature. However, the BC control framework as a technique viable for task coordination of multi-agent systems has been acknowledged in review works of [55, 56] and in [49, 57].

2.3 Broadcast Control

The concept of broadcast control was first conceived by Ueda et al. [58], which showed that their broadcast feedback approach was inspired by biological muscle control. Specifically, the control architecture was designed for skeletal muscles comprising a vast number of tiny functional units, called sarcomeres. The actuator system's output is an aggregate effect of many cellular units, each with a bistable ON-OFF state. The error between the aggregate output and a reference input is "broadcast" by a central controller. Instead of commanding individual units to perform specific actions, the central controller uniformly broadcast the total error signal to all cellular units. Each cellular unit then takes a stochastic decision with a state transition probability that is influenced by the broadcasted error.

Results of [58] show that even though no individual commands are sent to the individual cells when their state transition probabilities are adjusted in proportion to the broadcasted error signal, the ensemble of cells can track the desired trajectory. No addressing method is required for broadcast control because the information is given to all cells rather than a specific cell. As a result, the approach was highly scalable to a wide range of cellular control systems. It is to be noted that the cells were designed as a two-state discrete-time, non-homogeneous Markov process. Julius et al. [59] applied a similar idea of [58] for the lactose regulation system of the E.coli bacteria. Here the bacteria were also modelled as a two-state Markov chain model. The feedback information is read from the bacterial colony as a global quantity, which was considered the control system's output.

Unlike in [58, 59], where broadcast control was proposed for Markov models, Das and Ghose [60, 61] proposed for multi-agent systems. However, the proposed scheme was specialised only for the positional consensus problem. An excellent example of consensus problems is when the agents want to converge to a point. The work of Das and Ghose was inspired by Bretl [62], where a second-order cone programming method technique [63–65] was used to show consensus for two agents. When the number of agents was increased, the agents could only come close within a certain distance to each other. Das and Ghose formulated their strategy using a linear programming technique based on the basic Bretl's model with an additional randomisation feature [66] that aids in dislodging the solution from local minima. Their approach allowed many agents to achieve positional consensus or point convergence on repeated algorithm applications without jeopardising the broadcast constraint on the control command.

Azuma et al. [25] introduced a stochastic gradient-based broadcast control (BC) framework for multi-agent coordination. The scheme's goal was to design the information to be broadcasted and the local actions of the agents to complete a specified motion-coordination task. It is demonstrated that the local actions must incorporate randomness to realise the broadcast control. The task achievement degree is broadcasted to achieve this approach, and the agents alternate between random and deterministic walks. The BC has been shown to asymptotically accomplish the goal with probability 1. In their work, Azuma et al. conducted both numerical simulation and experimentation to demonstrate the proposed scheme's effectiveness. Unlike in [61], BC offers a general framework that can handle any multi-agent coordination tasks such as coverage, rendezvous and assignment. BC only requires observing the group performance (the achievement degree and highly compressed information). In [61] complete observation of agents is required, and the framework was specified as a consensus problem. Other examples, such as Berman et al. [67] and Mesquita et al. [68] developed stochastic control techniques for multi-agent systems and mentioned the use of broadcast controllers. Nevertheless, these works are different compared to the work of [25] and the component regarded as broadcast controllers are not feedback controllers which utilise data based on agents' states.

Several studies have extended the BC framework for: quantised environments [69], instability studies [70], Markovian environment [71] and even consensus problem with agent-to-agent communication – broadcast mixed environment [72]. An advanced BC theory was proposed by Ito et al. [54] as Pseudo-perturbation-based Broadcast Control (PBC). The PBC showed improved control performance compared to the BC law. As the BC scheme is based on stochastic optimisation tech-

niques, the agents take numerous random actions. Such random actions degrade control performance for coordinating tasks and lead to unsafe circumstances.

To overcome these shortcomings, the PBC is designed to employ multiple virtual random actions rather than the single physical action of the BC scheme. Like BC, the PBC also realises coordination tasks with probability 1. Altogether, in PBC, unavailing actions are minimised, and agents' states converge at least twice as rapidly as under the BC scheme. Additionally, increasing the number of multiple virtual actions enhances control performance even further since averaging several actions decreases randomness. The PBC has also been successfully applied as a coordination method for vehicles on merging roads to realise smooth traffic merging [73, 74]. Specific metrics had to be designed to solve the merging task using PBC. Mainly a suitable time-invariant objective function which is also locally convex to be globally minimised. The BC methods have been combined with other elements such as the receding horizon technique and implemented in a cyber-physical framework for better performance. These works will be reviewed in the following subsections.

2.3.1 Combined Broadcast Control and Receding Horizon Technique

In this section, literature that combines broadcast control for multi-agent systems and the concept of receding horizon control is discussed. Kumar and Kothare [75] proposed a broadcast stochastic receding horizon control scheme for multi-agent systems. Like the general broadcast idea for closed-loop control systems, the suggested strategy's primary vision is to generate and broadcast the optimal

control input to all the agents in a swarm utilising aggregate agent behaviour as the sole available feedback information. Their main addition was integrating the broadcast concept with current probabilistic tools and the theory of optimum control policy based on a finite receding horizon. In this work, the dynamical behaviour of individual agents is represented as a discrete-time finite-state Markov chain model. The controller employs the measured aggregate system error as the solely available feedback information and designs the optimal control inputs in a predictive framework. The computed control inputs are the agents' state transition probabilities, subsequently broadcast to the entire swarm to accomplish the desired system behaviour.

Using the same broadcast and receding horizon control framework, Kumar and Kothare [76] have also shown the system can be executed for trappings of Brownian ensemble. Their framework creates a control input independent of the number of particles and broadcasts it to all particles in the ensemble using measurements from a single particle as the only available feedback information. Using the suggested control action, they demonstrate the existence of a minimal zone in which all particles may be pushed and confined indefinitely.

As BC schemes are implemented through wireless communication, vehicular networking, based on wireless communications between vehicles and with other infrastructures, has been an area highly sought after for applications of BC theory. Accordingly, literature that combines BC with receding horizon control techniques for vehicles is explored next. Kianfar et al. [77] presented a distributed receding horizon approach adopted for active steering control of a cooperative vehicle platoon in the lateral direction. Here, the vehicle computes its control action locally and broadcasts its intention to its follower. Any deviation from the projected

states from each vehicle's intention is penalised and constrained in the optimisation problem, which is addressed locally by the vehicles. The proposed system was applied to the double lane change scenario.

In the case of cooperative vehicle safety systems, which rely on the periodic broadcast of each vehicle state information to track neighbours' positions and hence to predict likely collisions, Zhang et al. [78] have utilised a predictive model to predict in a dynamic and receding-horizon fashion, the ideal information dissemination rate by considering the dynamics of vehicle density. The model predicted the state of information dissemination rate in a short-term manner and minimised the prediction errors.

A work by Kamal et al. [79] provided a unique adaptive traffic signal management strategy for a mixed manual-automated traffic situation in a typically isolated intersection. The traffic lights are optimised in a receding horizon control framework that minimises all vehicles' overall crossing time while accounting for their dynamical states. By keeping the core elements of standard signal management systems, the control method allows the comfortable crossing of manually operated vehicles. The ideal signal change timings are broadcast one cycle ahead, allowing autonomous vehicles to adjust their speed to cross the junction with the least stop delay.

2.3.2 Combined Broadcast Control and Receding Horizon Technique in a Cyber-physical Framework

In a recent work, Kamal et al. [80] implemented broadcast and receding horizon control for traffic control in a cyber-physical framework to facilitate real-time implementation of the traffic control scheme. The receding horizon control (RHC) approach performed robustly against uncertain traffic factors. The application in this work to determine the crossing time of vehicles can be taken as a one-dimensional coverage problem as a Gaussian density shaped distribution was used as a desirability function of green light for the traffic signals.

Multi-agent systems and cyber-physical systems (CPS) have a lot in common. They may help each other accomplish complexity management, decentralisation, intelligence, modularity, flexibility, resilience, adaptability, and responsiveness [81]. CPS blends two worlds: 1) embedded systems, which display real-time and tightly deterministic behaviour, and 2) cloud systems, which exhibit highly probabilistic and optimal behaviour with no rigid time restrictions [82]. The work of Fortino et al. [83, 84] proposes smart objects-oriented Internet of Things (IoT) systems based on cloud and agents. On the other hand, Bradley and Atkins [85] cover research on co-optimisation and co-regulation methods using both cyber and physical resources, with applications to mobile robotic and vehicle systems.

2.4 Summary

In this chapter, the literature on multi-agent coverage tasks and broadcast control as a control technique for multi-agent systems were reviewed. Studies on coverage tasks show that it is a fundamental coordination task employed to test the effectiveness of multi-agent control algorithms. Correspondingly, the coverage task is applicable for many real-life coordination tasks for multi-robot and multi-vehicle systems. Though many coordination algorithms for multi-agent systems focus on a distributed fashion, a new paradigm has been recently proposed in the literature, termed Broadcast Control (BC).

The BC framework is based on a stochastic optimisation method that aims to reduce the communication volume of the multi-agent system without any agent-to-agent based communication. The work in this thesis is inspired by this control scheme applicable to multi-agent systems. As BC has only been tested very sparingly for coverage tasks of multi-agent systems, this thesis aims to address this gap by proposing enhancements to the existing BC schemes using a few predictive approaches to improve the coverage coordination tasks. The enhancements are inspired by work in the literature that focused on combining broadcast control with receding horizon techniques in a CPS framework for improved performance.

3 Development of Multi-step Broadcast Control (MBC) Scheme

3.1 Introduction

The previous chapter (Chapter 2) reviewed the literature on coverage tasks of multi-agent systems, Broadcast Control (BC) schemes and its extensions using the receding horizon technique and cyber-physical framework. This chapter introduces the development of the Multi-step Broadcast Control (MBC) scheme for multi-agent systems. In order to design a broadcast communication based stochastic control scheme that is able to anticipate the target environment, two main principles are used. First, the theory of simultaneous perturbation stochastic approximation (SPSA) represents the randomness inherent in the agent's movements. Secondly, the Broadcast Control (BC) scheme demonstrates how the convergence of the collective dynamics of the multi-agent system can converge using SPSA.

Section 3.2 begins by providing the theory of SPSA, which are fundamental to developing the proposed MBC scheme. Section 3.3 describes the coverage problem formulation applicable to the MBC scheme. Section 3.4 reviews the existing BC

schemes. The development of the proposed MBC and the convergence properties is outlined in Section 3.5. The numerical simulations using the MBC scheme and its analysis are presented and discussed in Section 3.6. Finally, Section 3.7 summarizes the entire chapter with its findings and limitations.

3.2 Simultaneous Perturbation Stochastic Approximation (SPSA)

Stochastic approximation (SA) is an established recursive procedure for finding the roots of equations in the presence of noisy measurements. SA can be implemented to obtain the extremas of functions in stochastic optimisation problems such as in this thesis.

The standard SA procedure depends on p-dimensional algorithms based on standard finite-difference gradient approximations. On the other hand, the simultaneous perturbation stochastic approximation (SPSA) utilises the "simultaneous perturbation" gradient approximation which uses $1/p$th of the amount of data needed as in the standard approach to reach the same level of estimation accuracy.

Considering the problem of finding the root, θ^* of the gradient function,

$$g(\theta) = \frac{\partial L(\theta)}{\partial \theta} = 0 \tag{3.1}$$

for some differential loss function $L =: R^p \rightarrow R^1$. When L and g are observed

directly, methods such as steepest descent, Newton-Raphson and scoring can be used to find θ^*. Where L is observed in the presence of noise, an SA algorithm of the generic Kiefer-Wolfowitz/Blum type is suitable. In contrast to SA algorithms based on finite difference methods, which necessitate $2p$ (noisy) measurements of L at each iteration, the simultaneous perturbation algorithm only requires $2q$, $q \geq 1$, measurements of L at each iteration, where for large p we typically have q. Therefore through SPSA, there is potential for a significant improvement in efficiency provided that the number of iterations does not increase to negate the reduced amount of data per iteration.

For standard form for SPSA, the estimate for θ at the kth iteration is given as

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \tag{3.2}$$

where the gain sequence a satisfies certain well-known conditions. The difference being that in the method of steepest descent $g(\cdot)$ replaces $\hat{g}_k(\cdot)$. The term "simultaneous perturbation" as applied to Eq. (3.2) arises from the fact that all elements of the $\hat{\theta}_k$ vector are being varied simultaneously.

Though the stochastic approximation is not a solution for multi-agent problems but for static optimisation problems in a centralised manner, the Broadcast Control (BC) schemes discussed in Section 3.4 have successfully implemented this for stochastic multi-agent systems.

3.3 Coverage Problem Formulation

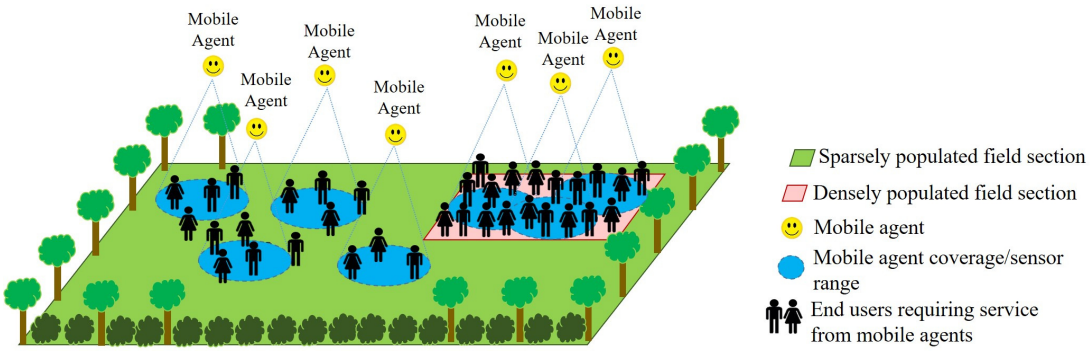


Figure 3.1: Mobile agents provide coverage service over a bounded area (field). The bounded area consists of densely and sparsely populated sections.

Consider an area with sections of varying density, with some being densely populated and some sparsely populated. To fairly and equitably serve the population within the bounded area, multiple mobile agents need to be deployed optimally depending on the number of available agents and population density pattern over the area (Fig. 3.1). By allocating a portion of the area to each agent, the entire area should be covered fully by all the agents. Such a deployment of multiple agents is known as the *coverage task*.

Traditionally, coverage problems involve deploying a set of agents to provide equal coverage distribution over the bounded area, where agents are distributed at an equal distance from each other. However, in this paper, the final distribution of agents varies according to population density; i.e. not all agents are placed at an equal distance to each other. For example, more agents need to be deployed to dense sections than to sparsely populated sections.

The objective function of the coverage task is [32]

$$J_{obj}(x) = \int_Q \min_{i \in 1,2,..N} f(|| \ q - x_i \ ||)\phi(q)(dq) \tag{3.3}$$

where f is the coverage performance function. The term q represents the uniformly distributed points in the environment, Q and ϕ represents a weighting function that determines the relative importance of points in Q. Also, x_i is the position of the i-th agent in the domain where $x_i = [x_1, x_2, ...x_n]^T \in \mathbb{R}^n$. The total number of agents is N. The objective function, $J(x(t))$ aims to reduce J, which is indicative of the global coverage performance of the multi-agent system, given as

$$J(x(t)) = \min_{x \in \mathbb{R}^{n\mathcal{N}}} J(x) \tag{3.4}$$

The minimum of J is achieved when N agents are placed optimally in the space. The Voronoi tessellation method [86], which partitions the entire area into sub-areas, is used to compute and visualize the coverage achievement. Created by points $(p_1, ...p_n)$, the optimal partition of Q follows the Voronoi partition $V(P) = \{V_1, ...V_n\}$ that is given by

$$V_i = \{q \in Q | \ \| \ q - x_i \ \| \leq \| \ q - x_j \ \|, \forall j \neq i\} \tag{3.5}$$

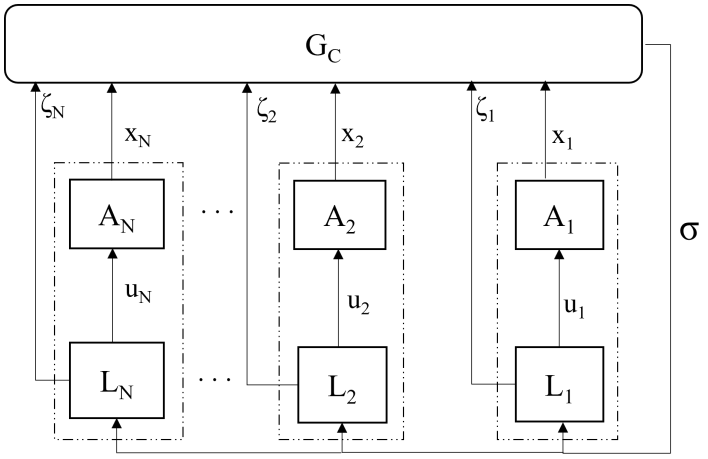


Figure 3.2: Framework of BC/PBC/MBC of a multi-agent system. The broadcast system consists of N agents A_i, local controllers L_i, and a global controller G_C. In the BC scheme, no local controller L_i has to send its state ζ_i to the global controller, G_C.

3.4 Broadcast Control Schemes

3.4.1 Broadcast Control (BC)

In this section, an overview of the BC [25] control scheme is given. The BC multi-agent scheme consists of a global controller, G_C, with N agents, each denoted as $A_i, i = 1, 2, ...N$, and the respective local controllers, L_i. The connectivity between the agents and the controllers is shown in Fig. 3.2.

Here, *step* is defined as the physical move at time, t; the next physical move is at $t + 1$ and so forth. In BC, at each consecutive step, agents alternate between taking a randomly generated move and a deterministic move (Fig. 3.3(a)). The global controller G_C computes the collective performance of the whole system at each step by evaluating the objective function, $J(x)$. Using the difference between the objective function of random and deterministic moves, G_C calculates and

broadcasts this value as a scalar control signal, σ_B to all agents. Using σ_B, the local controller L_i calculates and determines the control action $u_i(t)$. The state equation of an agent is represented as follows:

$$A_i : \ x_i(t+1) = x_i(t) + u_i(t), \ i = 1, 2, \ldots, N \tag{3.6}$$

where $x_i(t) \in \mathbb{R}^n$ is the position in the n-dimensional space and $u_i(t) \in \mathbb{R}^n$ is the control input.

The local controller L_i of agent A_i produces the control signal based on the information received from G_C, as follows:

$$L_i : \begin{cases} \zeta_i(t+1) = \alpha(\zeta_i(t), \sigma_B(t), t) \\ \\ u_i(t) = \beta(\zeta_i(t), \sigma_B(t), t) \end{cases} \tag{3.7}$$

where $\zeta_i(t) \in \mathbb{R}^{n_\zeta}$ and $u_i(t) \in \mathbb{R}^n$ are the state and output of the local controller, L_i, respectively. The initial state of L_i is set to $\zeta_i(0) = 0$. $\sigma_B(t) \in \mathbb{R}$ is the broadcast signal provided by G_C.

The controller functions $\alpha : \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_{\sigma_B}} \times \mathbb{N} \to \mathbb{R}^{n_\zeta}$ and $\beta : \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_{\sigma_B}} \times \mathbb{N} \to \mathbb{R}^n$ are defined as follows:

$$\alpha(\zeta_i(t), \sigma_B(t), t) := [\Delta_i(t)^T, \sigma_B(t)]^T \tag{3.8}$$

$$\beta(\zeta_i(t), \sigma_B(t), t) :=$$

$$\begin{cases} c(t)\Delta_i(t), & \text{if } t \in \{0, 2, 4, \ldots\}, \\ -c(t)\zeta_{i1}(t) - a(t)\left(\dfrac{\sigma_B(t) - \zeta_{i2}(t)}{c(t)}\right)\zeta_{i1}^{[-1]}(t) & \text{if } t \in \{1, 3, 5, \ldots\} \end{cases} \tag{3.9}$$

where $\zeta_i(t) := [\zeta_{i1}(t)^T, \zeta_{i2}(t)]^T, \zeta_{i1} \in \{-1, 1\}^n$ with $\zeta_{i2} \in \mathbb{R}$. $\zeta_{i1}^{[-1]}$ denoting the element-wise inverse of ζ_{i1}, namely $\zeta_{i1}^{[-1]} := [1/\zeta_{i1,1}, ..., 1/\zeta_{i1,n}]^T$. Also, $\zeta_{i1}(t) = \Delta_i(t-1)$ and $\zeta_{i2}(t) = \sigma_B(t-1)$.

The random variable is defined as

$$\Delta_i(t) = \begin{bmatrix} \Delta_{i,1}(t) \\ \vdots \\ \Delta_{i,n}(t) \end{bmatrix} \in \{-1, 1\}^n \tag{3.10}$$

where each $\Delta_{i,j}(t)(i \in \{1, ..., n\}, j \in \{1, ..., N\}, t \in \{0, 1, ...\})$ is represented as a Bernoulli distribution with outcome ± 1 with equal probabilities. Additionally, $a(t) \in \mathbb{R}_+$ and $c(t) \in \mathbb{R}_+$ are the time-varying gains of controller L_i.

The broadcast signal from G_C is described as

$$G_C : \sigma_B(t) = J(x(t)) \in \mathbb{R} \tag{3.11}$$

where

$$x(t) := \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{nN} \tag{3.12}$$

is the collective state of all agents in the system. As time t tends to infinity, the

multi-agent system approaches the optimal J as follows:

$$\lim_{t\to\infty} J(x(t)) = \min_{x\in\mathbb{R}^{n\mathcal{N}}} J(x) \tag{3.13}$$

where $J : \mathbb{R}^{nN} \to \mathbb{R}$. The optimization process is based on the gradient of the objective function with respect to each individual action. However, the objective function is not known to individual agents; this is an advantageous feature of the BC scheme where it accomplishes a task without providing the task details or objective to the individual agents. Therefore, in the BC system, agents take random and deterministic control actions alternately. At even time steps $t \in 0, 2, 4, ...,$ agents take random moves, and at odd time steps $t \in 1, 3, 5, ...,$ they take deterministic moves. This is shown in Fig. 3.3(a).

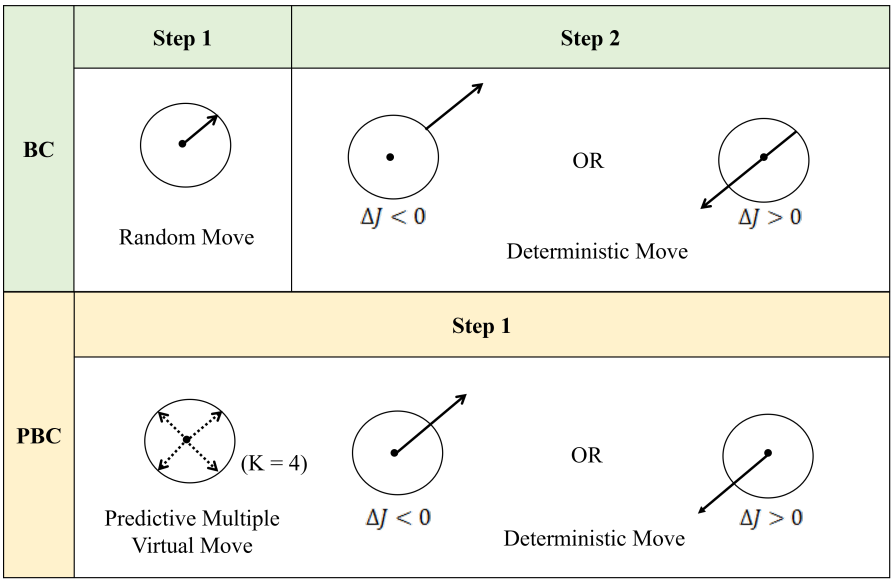


Figure 3.3: (a) Two-step agent movements in the BC framework. Depending on the feedback on the random move (ΔJ), an agent continues to move in the same or opposite direction in the next step. The process repeats until the convergence. (b) Single-step agent movement in the PBC framework.

In BC, the random movement of agents incurs movement cost and considerable time for convergence. Ito et al. [54] introduced the Pseudo-perturbation based

Broadcast Control (PBC). In the PBC scheme, each agent takes multiple virtual random moves (Fig. 3.3(b)) within a single physical step. K is the total number of multiple virtual random moves taken by an agent within a step. Through this, an agent omits the costly single physical random move as taken by the agent in the BC scheme. This causes PBC agents to accomplish a similar outcome in a single step compared to two steps taken by agents using BC.

3.4.2 Pseudo-perturbation based Broadcast Control (PBC)

Next, the PBC [54] is reviewed. The state-space equation of agent i in PBC is given as

$$\hat{x}_i^{(k)}(t+1) := x_i(t) + \hat{u}_i^{(k)}(t) \text{ for } k=1,2,\ldots,K \tag{3.14}$$

$$\hat{u}_i^{(k)}(t) := c(t)\Delta_i^{(k)}(t) \text{ for } k=1,2,\ldots,K \tag{3.15}$$

where $\hat{x}_i^{(k)}(t+1)$, $\hat{u}_i^{(k)}(t)$ and $\Delta_i^{(k)}(t)$ are the virtual input, virtual predictive state and random action of agents A_i at each step. Each component of $\Delta_i^{(k)}(t) \in \{-1,1\}^n$ (for each $i \in \{1,...,N\}$, $k \in \{1,...,K\}$ and $t \in \{0,1,...\}$) independently obeys the Bernoulli distribution with outcome ± 1 with equal probabilities. The global controller, G_C, calculates $\hat{u}_i^{(k)}(t)$ and $\hat{x}_i^{(k)}(t+1)$ using $(x_i(t), \Delta_i^{(k)}(t))$ sent from the agent A_i. Additionally, $\hat{x}^{(k)}$ and $\Delta^{(k)}(t)$ are given as

$$\hat{x}^{(k)} := \begin{bmatrix} \hat{x}_1^{(k)} \\ \vdots \\ \hat{x}_N^{(k)} \end{bmatrix} \in \mathbb{R}^{nN} \tag{3.16}$$

and

$$\Delta^{(k)}(t) := \begin{bmatrix} \Delta_1^{(k)}(t) \\ \vdots \\ \Delta_N^{(k)}(t) \end{bmatrix} \tag{3.17}$$

The G_C in PBC calculates the objective function, J for all virtual steps taken as follows:

$$\sigma_P(t) := \begin{bmatrix} J(\hat{x}^{(1)}(t+1)) - J(x(t)) \\ J(\hat{x}^{(2)}(t+1)) - J(x(t)) \\ \vdots \\ J(\hat{x}^{(K)}(t+1)) - J(x(t)) \end{bmatrix} \in \mathbb{R}^K \tag{3.18}$$

where ${\sigma_P}^{(k)}(t) := J(\hat{x}^{(k)}(t+1)) - J(x(t))$ and $\sigma_P(t)$ is given as

$$\sigma_P(t) = \begin{bmatrix} \sigma_P^{(1)}(t) \\ \vdots \\ \sigma_P^{(K)}(t) \end{bmatrix} \tag{3.19}$$

In PBC, the local controller, L_i determines its state, $\zeta_i(t)$ and the control output, $u_i(t)$ as

$$\zeta_i(t) := [{\Delta_i}^{(1)}(t)^T, \ldots, {\Delta_i}^{(K)}(t)^T]^T \in \{-1, 1\}^{nK} \tag{3.20}$$

$$u_i(t) := -a(t)\frac{1}{K}\sum_{k=1}^{K}\frac{{\sigma_P}^{(k)}(t)}{c(t)}\Delta_i^{(k)[-1]}(t) \tag{3.21}$$

where $\Delta_i^{(k)[-1]}$ denotes the element-wise inverse of vector $\Delta_i^{(k)}$. Compared to (3.9),

we can notice that the term $c(t)\Delta_i(t)$ (which represents the physical random action) is removed in PBC.

In PBC, the global controller, G_C, also receives the predicted random state ζ_i from the local controller, L_i, whereas in BC, L_i does not share its state with G_C. Even with this extra cost of communication, PBC converges to the optimal point twice as fast as BC. To date, the PBC framework has been effectively applied for the real-time coordination of vehicles at a merging road [74].

3.5 Development of Multi-step Broadcast Control (MBC) Scheme

This section presents the novel scheme proposed in this research, the Multi-step Broadcast Control (MBC) scheme, which incorporates a predictive multi-step move in a horizon within a single physical step. The multiple steps taken is useful to predict uncertainties several steps ahead, which could be a dense section to move towards to. Figure 3.4 depicts the concept of the action taken under the proposed scheme. The inspiration behind this concept comes from the theory of model predictive control (MPC). Like MPC, the proposed multi-steps are driven along a horizon to anticipate and consider future events to optimise the current iteration/time.

In the case of MBC, K denotes the maximum number of predictive virtual steps taken along the horizon. At each iteration, an agent provides its current state $x_i(t)$ and its random action, $\Delta_i^{(k)}, k = 0, 1, 2, \ldots, K-1$, and using that the global

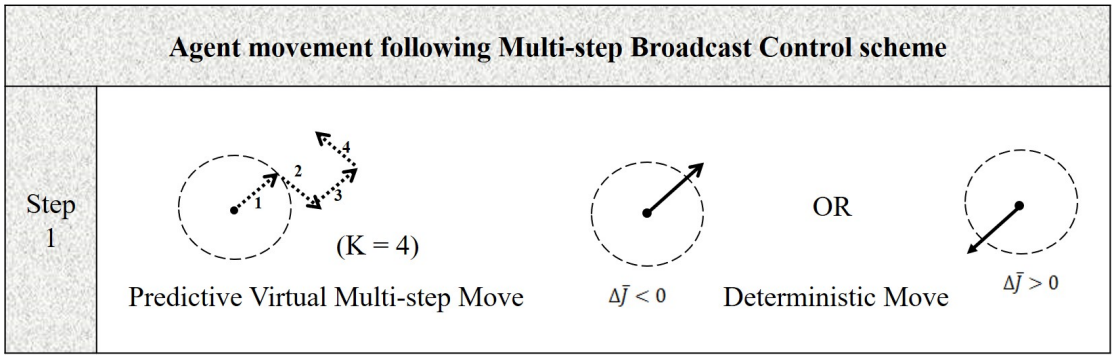


Figure 3.4: Agent movement taking multi-steps along a horizon in the MBC scheme.

controller, G_C, calculates the future virtual states $\hat{x}_i^{(k)}$ with $\hat{x}_i^{(0)} = x_i(t)$ as

$$\hat{x}_i^{(k+1)}(t) := \hat{x}_i^{(k)}(t) + \hat{u}_i^{(k)}(t) \text{ for } k=0,1,2,\ldots,K\text{-}1 \tag{3.22}$$

$$\hat{u}_i^{(k)}(t) := c(t)\Delta_i^{(k)}(t) \text{ for } k=0,1,2,\ldots,K\text{-}1 \tag{3.23}$$

where $\hat{u}_i^{(k)}$ is the corresponding virtual input. The collective states of all agents at k is defined as

$$\hat{x}^{(k)} := \begin{bmatrix} \hat{x}_1^{(k)} \\ \vdots \\ \hat{x}_N^{(k)} \end{bmatrix} \in \mathbb{R}^{nN} \tag{3.24}$$

resulting from corresponding collective random variables defined as

$$\Delta^{(k)}(t) := \begin{bmatrix} \Delta_1^{(k)}(t) \\ \vdots \\ \Delta_N^{(k)}(t) \end{bmatrix} \tag{3.25}$$

The global controller computes the broadcast signal, σ_M, through a compounding method as follows:

$$\sigma_M(t) := \begin{bmatrix} J(\hat{x}^{(1)}(t)) - J(\hat{x}^{(0)}(t)) \\ J(\hat{x}^{(2)}(t)) - J(\hat{x}^{(1)}(t)) \\ \vdots \\ J(\hat{x}^{(K)}(t)) - J(\hat{x}^{(K-1)}(t)) \end{bmatrix} \in \mathbb{R}^K \tag{3.26}$$

where ${\sigma_M}^{(k)}(t) := J(\hat{x}^{(k+1)}(t)) - J(\hat{x}^{(k)}(t))$ and $\sigma_M(t)$ is given as

$$\sigma_M(t) = \begin{bmatrix} \sigma_M^{(0)}(t) \\ \vdots \\ \sigma_M^{(K-1)}(t) \end{bmatrix} \tag{3.27}$$

The state, $\zeta_i(t)$, and the control output, $u_i(t)$, of the local controller, L_i, are given as

$$\zeta_i(t) := [\Delta_i^{(0)}(t)^T, \ldots, \Delta_i^{(K-1)}(t)^T]^T \in \{-1, 1\}^{nK} \tag{3.28}$$

$$u_i(t) := -a(t)\frac{1}{\sum_{k=0}^{K-1} \lambda^{(k)}} \sum_{k=0}^{K-1} \frac{{\sigma_M}^{(k)}(t)\lambda^{(k)}}{c(t)} \Delta_i^{(k)[-1]}(t), \tag{3.29}$$

where $\Delta_i^{(k)[-1]}$ denotes the element-wise inverse of vector $\Delta_i^{(k)}$ and $\lambda \in (0, 1)$ is a discount factor of a weighted average technique. Through the use of λ, immediate action is given a higher weight, and as the number of steps increases, the respective weightage is decreased gradually. The reduction in weights is designed as such because the probability of accurate prediction in an uncertain environment decreases with an increase in steps in forward prediction. While the use of the

arithmetic average technique is a quick general representation of a data set (as used in PBC), the use of the weighted average technique in (3.26)-(3.29) is more descriptive of a specific problem and is capable of yielding better accuracy.

The computing complexity of BC schemes can be expressed in terms of communication volume. The MBC scheme will have similar communication data with that of PBC if the number of steps, K, is the same. It is shown in Ito et al [87] that the communication volume of the PBC scheme is about half of that of centralized unicast protocols. On the other hand, BC will have slightly lower communication volume than both MBC and PBC as it does not transmit random variable (K virtual steps). Nevertheless, both MBC and PBC have better control performance than the BC scheme as random physical action is eliminated, and agents' state converges quicker.

3.5.1 Convergence analysis

Theorem 1 (Convergence of the MBC Scheme)
Consider the multi-agent system shown in Fig. 3.2, an objective function $J(x)$, and the MBC scheme in (3.26), (3.28) and (3.29) with $K > 1$. If the following conditions (c1)-(c3) hold, then $x(t)$ in (3.12) converges to a (possibly sample-path-dependent) solution set to $\partial_x J(x) = 0$ with probability 1.

(c1) The objective function $J(x) : \mathbb{R}^{nN} \to \mathbb{R}$ is defined as

$$J(x(t)) := \rho(\|x\|)J_{obj}(x) + (1 - \rho(\|x\|))x^T x, \tag{3.30}$$

$$\rho(\|x\|) := \begin{cases} 1(\|x\| \leq l_1) \\ 0(\|x\| \geq l_2) \end{cases} \tag{3.31}$$

where $J(x)$ and $J_{obj}(x)$ in (3.3) is nonnegative C^2 continuous on $\mathbb{R}^{nN}$, and there exists a solution to $\partial_x J(x) = 0$. $x^T x$ is a nondecreasing function with respect to the distance from the origin, and ρ is the switching function where l_1 and l_2 specifies the environment.

(c2) The compact connected internally chain transitive invariant sets of a gradient system $\dot{z}(\tau) = -\partial_z J(z(\tau))$ are included in the solution set to the equation $\partial_z J z = 0$ (i.e., to $\partial_x J(x) = 0$), and there exists an asymptotically stable equilibrium for the gradient system, where $z(\tau) \in \mathbb{R}^{nN}$ and the stability are in the Lyapunov sense [25].

(c3) $\lim_{t\to\infty} a(t) = 0$, $\sum_{t=0}^{\infty} a(t) = \infty$, $\lim_{t\to\infty} c(t) = 0$, $\sum_{t=0}^{\infty}(a(t)/c(t))^2 < \infty$ and $\sum_{t=0}^{\infty}(a(t)c(t))^2 < \infty$.

Remark 2

Convergence of the proposed MBC scheme is proven by showing that the transition of the MBC scheme converges to $\partial_x J(x) = 0$, which implies that the system of $x(t)$ in (3.12) has and reaches a local minimum value for the coverage problem.

Remark 3

Condition (c1) holds as J_{obj} in (3.3) is twice differentiable and $x^T x$ is a quadratic

potential function, resulting in $J(x(t))$ which is sufficiently smooth. Condition (c2) holds as the Hessian matrix of $J(x)$ is nonsingular at each x satisfying $\partial_x J(x) = 0$ [88]. Condition (c3) is applied to prevent gains $a(t)$ and $c(t)$ from reaching a very low value; in Section 3.6 (Numerical Simulations and Analysis), the gains will be designed as in (3.41) to satisfy condition (c3).

Proof

The local controller output, $u_i(t)$ of BC as in (3.9) performs a two-stage state transition as follows:

$$x(t+2) = x(t) - a(t)\frac{\sigma_B(t+1) - \sigma_B(t)}{c(t)}\Delta^{[-1]}(t) \tag{3.32}$$

where

$$\sigma_B(t) = J(x(t)) \tag{3.33}$$

$$\sigma_B(t+1) = J(x(t) + c(t)\Delta(t)) \tag{3.34}$$

hold for $t \in \{0, 2, 4, ...\}$. Then the system (3.32) becomes

$$x(t+2) = x(t) - a(t)g(t, \Delta(t)), t \in \{0, 2, 4, ...\} \tag{3.35}$$

where the stochastic function $g(t, \Delta)$ is defined as

$$g(t, \Delta) := \left(\frac{J(x(t) + c(t)\Delta) - J(x(t))}{c(t)}\right)\Delta^{[-1]}(t) \tag{3.36}$$

If $J(x)$ is a C^2 continuous function with $c(t) \to 0$ (as implied by conditions (c1) and (c3)), then the expected value of $g(t, \Delta(t))$ (which is E in (3.37)) reduces

to a stochastic approximate gradient of $J(x)$ based on simultaneous perturbation stochastic approximation (SPSA) [89] as follows:

$$E[g(t, \Delta(t))|_{x(t)}] = \partial_x J(x(t)) + O(c(t)) \tag{3.37}$$

Similarly, following (3.35), the state transition of multi-steps in the MBC scheme is given by

$$x(t+1) = x(t) - a(t)\frac{1}{\sum_{k=0}^{K-1}\lambda^{(k)}}\sum_{k=0}^{K-1} g(t, \Delta^{(k)}(t))\lambda^{(k)}, \forall t \tag{3.38}$$

Next, we define the following function $e(t)$ for brief notation:

$$e(t) := \frac{1}{\sum_{k=0}^{K-1}\lambda^{(k)}}\sum_{k=0}^{K-1} g(t, \Delta^{(k)}(t))\lambda^{(k)} - E\left[\frac{1}{\sum_{k=0}^{K-1}\lambda^{(k)}}\sum_{k=0}^{K-1} g(t, \Delta^{(k)}(t))\lambda^{(k)}|_{x(t)}\right] \tag{3.39}$$

Substituting (3.37) and (3.39) into (3.38), then as $c(t) \to 0$, the transition of $x(t)$ under the MBC scheme is given by

$$x(t+1) = x(t) - a(t)\{\partial J_x(x(t)) + e(t) + O(c(t))\} \tag{3.40}$$

The system (3.40) is identical to the stochastic approximation algorithm in equation (A.1) (in Appendix A.1) of Azuma et al [25]. Then, Lemma 2 of Azuma et al [25] shows that (A.1) converges to a (possibly sample path dependent) compact connected internally chain transitive invariant set of the gradient system,

$\dot{z}(\tau) = -\partial_z J(z(\tau))$. Conditions (c1) and c(3) imply the conditions for the almost-sure convergence of Lemma 2 [25] for (30), and thus if (c1) and (c3) hold, then it follows that (3.40) converges to a compact connected internally chain transitive invariant set $\dot{z}(\tau) = -\partial_z J(z(\tau))$; then, if (c2) holds, (3.40) converges to $\partial_x J(x) = 0$ and the proof is complete. □

3.6 Numerical Simulations and Analysis

The number of agents in the two-dimensional state space is set to $N = 25$, i.e. $n = 2$. The size of the coverage work area is fixed at 200 x 200 square units. Initially, all agents are evenly distributed within the x-coordinate (80: 120) and y-coordinate (0:40). The terminal simulation time is set to 5,000 iterations, and the number of multi-steps used is $K = 10$. The controller gains a and c are determined as follows:

$$\begin{aligned} a(t)|_{MBC} &= \frac{a_0}{\left(\frac{t}{2} + 1 + a_v\right)^{a_p}}, \\ c(t)|_{MBC} &= \frac{c_0}{\left(\frac{t}{2} + 1\right)^{c_p}}, \end{aligned} \tag{3.41}$$

for t $\in \{0, 1, 2, ...\}$ where $a_0 > 0$ and $c_0 > 0$ hold to satisfy $a(t) > 0$ and $c(t) > 0$.

The performance study to evaluate the effectiveness of MBC against BC and PBC is conducted through three different scenarios. The first scenario is for a single *dense section*, secondly for *two dense sections* and thirdly for an increased number of agents, from 15 to 25 agents.

3.6.1 Coverage with a single dense section

The performance of MBC for a single dense section is first discussed. For this scenario, the value of a_o and c_o are 1.2 and 14.5, respectively. a_v, a_p and c_p are set as 15.5, 0.7 and 0.16 for all scenarios. Figure 3.5 shows the corresponding agent distributions during specific intervals of iteration of 500, 1,500, 2,500 and 5,000 in all three schemes. BC is represented by the left column, followed by PBC in the middle and proposed MBC in the right column. As the number of iterations increase, it can be observed that the agents begin to disperse from their initial positions and distribute themselves equally in the environment. The number of agents in the dense section increases as the iteration progresses. The dense shaded section holds two agents by iteration 5,000 for all the three schemes. However, it is clear that MBC achieves this first by iteration 1,500, followed by PBC by 2,500 and lastly by BC by iteration 5,000. These show that the schemes can achieve the same allocated number of agents in the dense section, making them comparable for convergence study.

Figure 3.6 shows the evolution of the objective function for BC, PBC and MBC. From the chart, it can be observed that both PBC and MBC have similar initial steeper gradient descent compared to BC. However, after the initial drop, MBC shows the quickest convergence to the final value, followed by PBC and finally by BC.

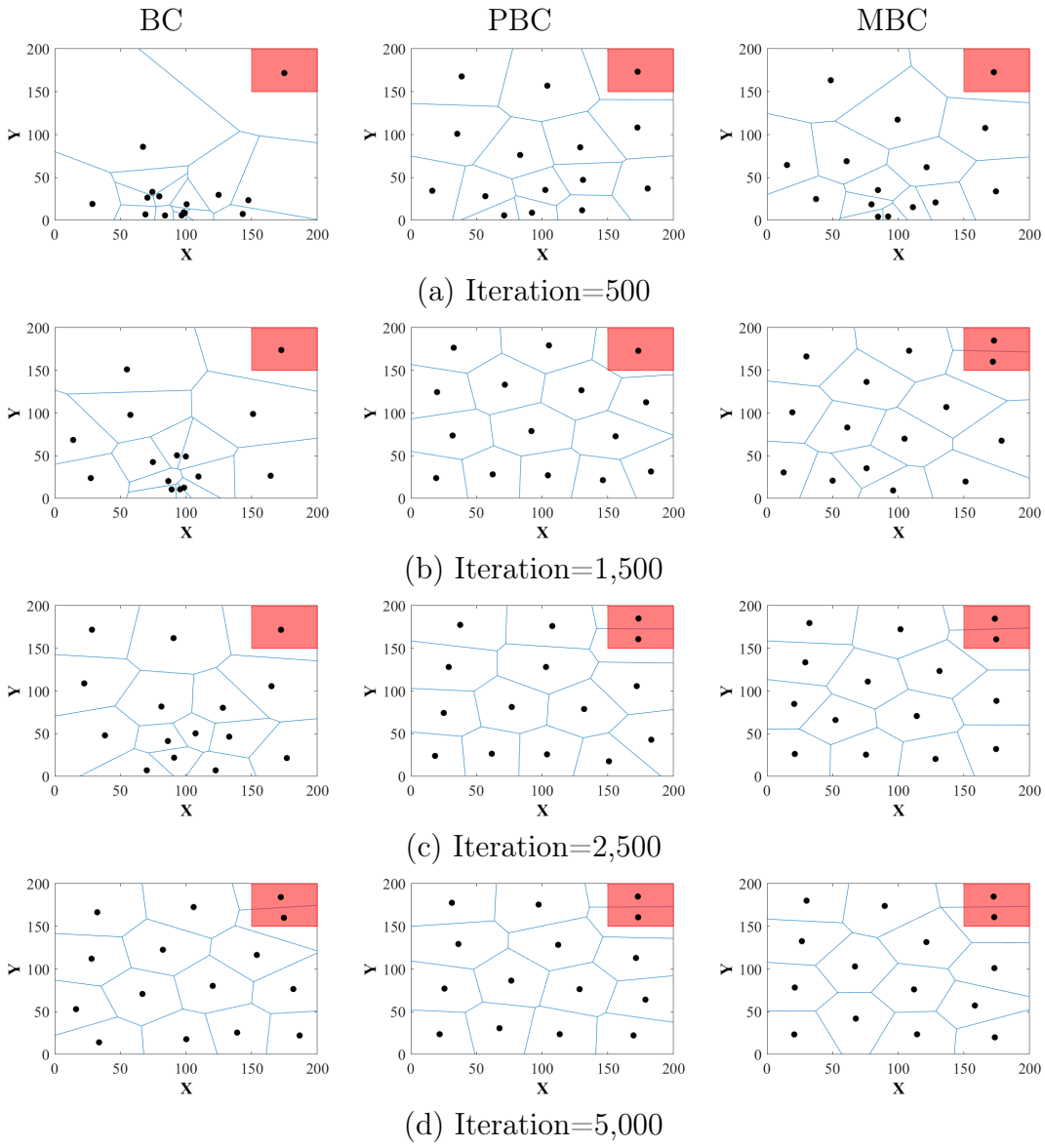


Figure 3.5: The coverage task's achievement with a single dense section is shown by agents' positions in the Voronoi diagram at various iterations. Comparison of three different schemes: BC (left), PBC (middle), and proposed MBC (right).

This confirms that the proposed MBC shows better convergence performance than the PBC scheme based on the execution difference (23) and (24). The performance of BC is very different compared to both PBC and MBC as it takes two physical steps to accomplish what PBC and MBC achieve through a single physical step with multiple virtual steps. Therefore, for the subsequent scenarios,

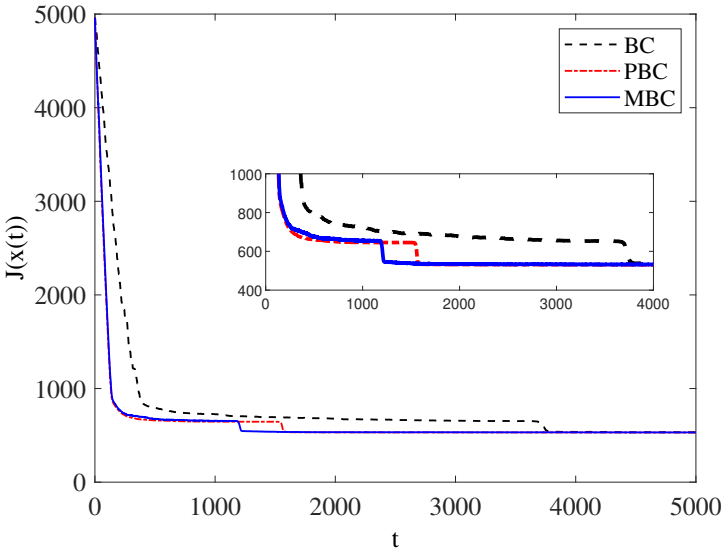


Figure 3.6: Evolution of objective function for BC, PBC and MBC for coverage task with a dense section.

a comparison study will be conducted only for PBC and MBC. (Note: BC is able to reach the same performance albeit much more slowly).

3.6.2 Coverage with two dense sections

The earlier section's coverage study can be expanded to multiple dense sections, and here two dense sections are studied. For this specific study, the value of a_o is 0.55 and c_o is 12.5.

Figure 3.7 shows the comparison between PBC agent distribution on the left column and MBC on the right column for iteration 1,000, 2,000 and 4,000. At iteration 1,000, both PBC and MBC have equally acquired a single agent in the right dense section. By iteration 2,000, MBC has managed to allocate two agents in both the dense sections, while PBC only manages two agents in the left dense section. However, by iteration 4,000, PBC has managed to catch up and converge

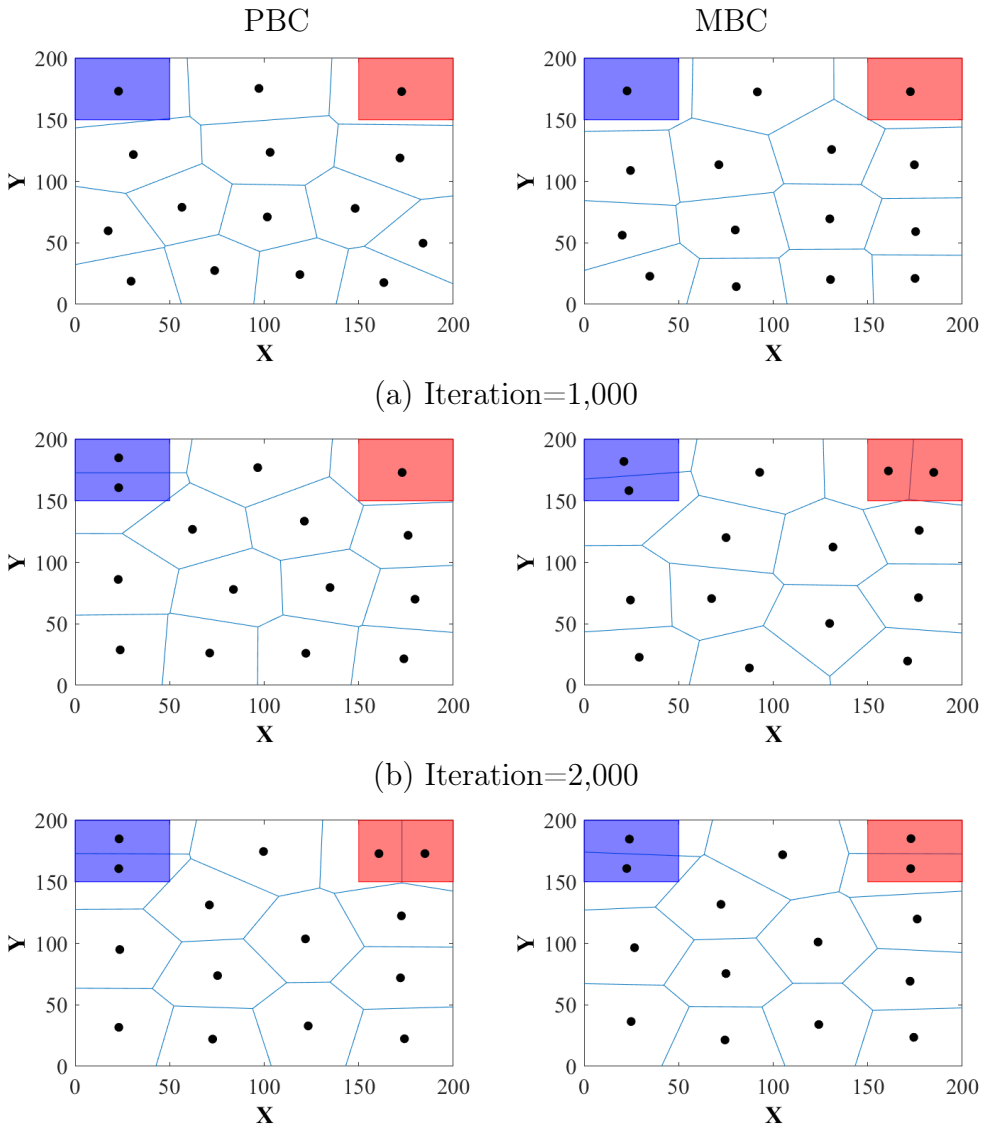
similarly to MBC.

The objective function in Fig. 3.8 shows these results clearly where MBC has converged faster by iteration 1,500 and PBC at about iteration 3,000. This shows the MBC outperforms PBC in a coverage task with double dense sections.

3.6.3 Coverage for an increased number of agents

The two scenarios discussed above for single and double dense sections utilise 15 agents. This section studies the performance of the proposed MBC scheme with an increased number of agents. Here, 25 agents are used, and the initial positions of the agents are now five rows. Two extra rows of agents are included in the existing three rows of agents. The values of initial gain utilised are $a_o = 0.3$ and $c_o = 7.5$.

With an increased number of agents for coverage with a single dense section, the comparison of agent distribution for PBC and MBC is shown in Fig. 3.9. The distribution is captured at iterations 1,500, 2,500 and 4,000. Intuitively, we can expect that by using dense sections of the same size and increasing the number of agents, more agents will be allocated in the sections. Figure 3.9 shows that this is true, as now at convergence, there are three agents in the dense section. Referring to the same figure, by iteration 1,500, MBC has two agents allocated in the dense section compared with PBC, which has one. At iteration 2,500, the number of agents for MBC within the dense section has increased to three, where else for PBC, it has increased to two. Finally, by iteration 4,000, both methods show similar distribution. This is presented in Fig. 3.10 where the MBC scheme



(a) Iteration=1,000

(b) Iteration=2,000

(c) Iteration=4,000

Figure 3.7: The coverage task's achievement with double dense sections is shown by agents' positions in the Voronoi diagram at various iterations. Comparison of two different schemes: PBC (left) and proposed MBC (right).

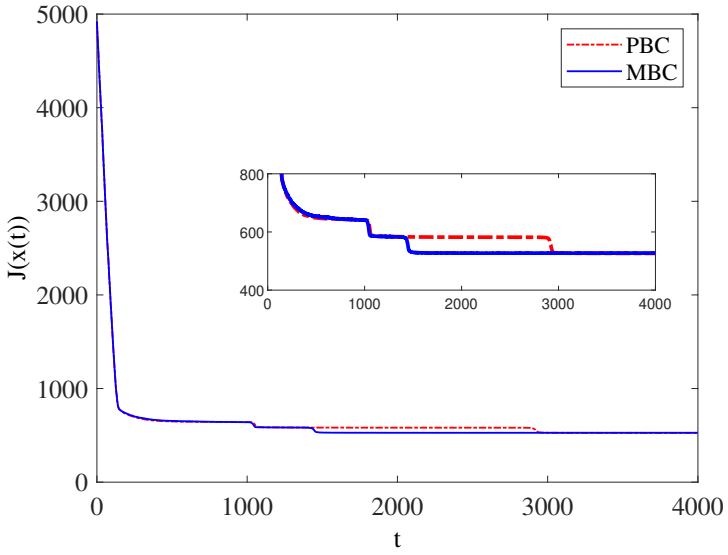
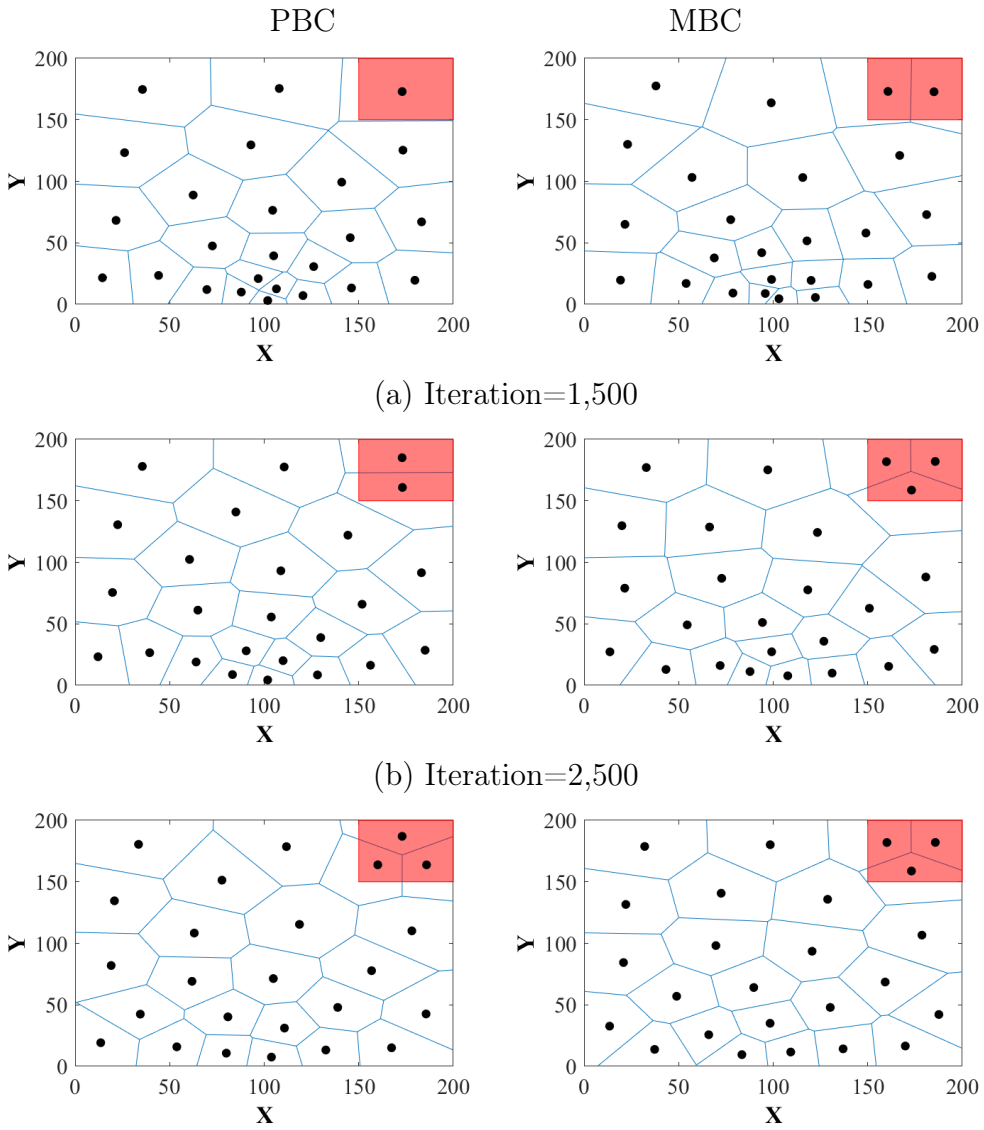


Figure 3.8: Evolution of objective function for PBC and MBC for coverage task with double dense sections.

displays quicker convergence performance than PBC.

In addition, the initial BC scheme proposed by Azuma [25] was tested experimentally for uniform coverage task using mobile ground robots, and the results showed that the numerical scheme is transferable to physical execution. On the same note, the MBC scheme is expected to perform experimentally with similar success as it follows a similar physical execution protocol. The only difference here is the additional virtual computation load and decrease in physical steps taken by each robot. This, alongside the numerical simulations conducted in this paper, validates the proposed scheme for successful implementation in a coverage control task.



(a) Iteration=1,500

(b) Iteration=2,500

(c) Iteration=4,000

Figure 3.9: The achievement of 25 agents in a coverage task with a single dense section is shown by agents' positions in the Voronoi diagram at various iterations. Comparison of two different schemes: PBC (left) and proposed MBC (right).

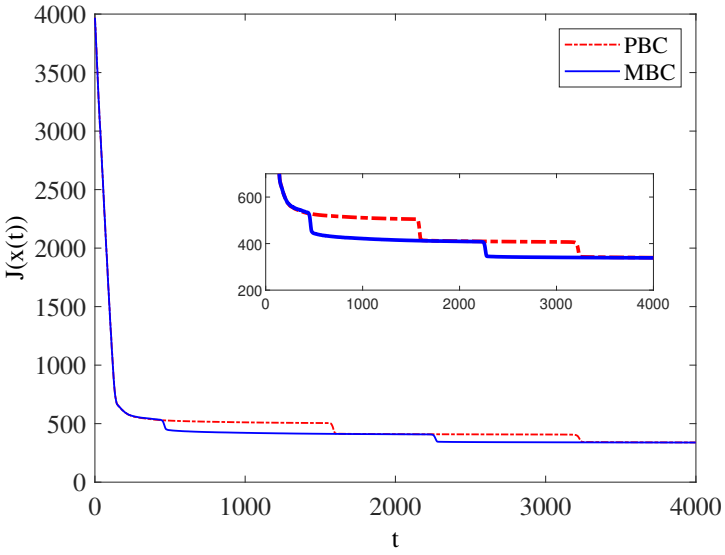


Figure 3.10: Evolution of objective function for PBC and MBC for coverage task with a single dense section using 25 agents.

3.7 Summary

In this chapter, a novel broadcast control-based scheme called Multi-step Broadcast Control (MBC) was developed. This scheme is designed to overcome the limited one-step look-ahead of the existing BC schemes. This was achieved by introducing several virtual random stages along the horizon. By using λ, immediate action is given a higher weight, and the weighting is gradually reduced as the number of steps increases.

Convergence of the proposed MBC scheme is proven by showing that the transition of the MBC scheme converges to $\partial_x J(x) = 0$, which implies that the system (collective agent dynamics) has and reaches a local minimum value for the coverage problem. While using the arithmetic averaging technique is a quick general representation of a data set (as used in PBC), in MBC, the local controller output has been calculated using a weighted averaging technique that better describes a

specific problem and is capable of better accuracy.

The performance of the proposed MBC has been tested in terms of the coverage problem with single and double density sections and for an increasing number of agents. The results of the study suggest that the introduction of multiple virtual steps in a horizon increases task fulfilment and deployment efficiency. It is noteworthy that MBC outperforms the existing BC schemes and converges the fastest as the gradient is calculated from several steps along the horizon where the dense sections appear earlier to the agents compared to BC and PBC. In this chapter, path constraints experienced by physical agents are not modelled into the MBC scheme. Therefore, the next chapter will focus on developing the MBC scheme to accommodate the path constraints of the agents.

4 Development of MBC scheme considering Agent Path Constraints

4.1 Introduction

The previous chapter (Chapter 3) introduced the Multi-step Broadcast Control (MBC) scheme. The MBC scheme is a novel Broadcast Control (BC) scheme employing a weighted averaging technique of multiple virtual steps along a horizon to increase performance parameters in a coverage problem with non-uniformity in the importance of coverage area. However, in the previous chapter, it was assumed that the agents can always reach the predicted position at the future time step from the initial time step. This is not always feasible in a realistic/practical scenario where agents are physical autonomous robots or vehicles.

Any real physical agent is constrained in its motions along the travel path. For example, physical agents are limited in their motion based on its turning limit and heading direction. The original BC scheme does not consider these constraints of the agents. Ideally, control schemes must reflect the actual position that an agent reaches as the starting point for the next iteration.

This chapter develops an integrated framework that outlines how an agent's path constraints can be taken into consideration in the development of the MBC scheme. Here, agents will be assumed as fixed-wing autonomous unmanned aerial vehicles (UAVs). Section 4.2 describes the path constraints for UAVs using subsets Dubin's curve [90]. The following section, Section 4.3, outlines the decision making procedure for trajectory planning. Section 4.4 describes the coverage task suited for UAVs. Nextly, Section 4.5 demonstrates the effectiveness of the developed scheme via numerical simulation of the integrated framework and its corresponding analysis. Finally, Section 4.6 summarises the chapter with its findings and possible improvements for better overall performance.

4.2 Agent Path Constraints

The previous chapter showed that the MBC scheme produces the desired point for each agent at every iteration. Agents were assumed to travel between the points following straight-line paths. Figure 4.1 illustrate these paths.

However, paths of straight-line segments result in discontinuous trajectories, which is not realistic in a physical system. The main assumption presuming agents can always reach the predicted point at each step is unrealistic. This assumption concludes that the agent's heading direction is always in the direction of the predicted point. This requires the physical agents to rotate in the current position to face the new direction before travelling to the desired point following a straight path.

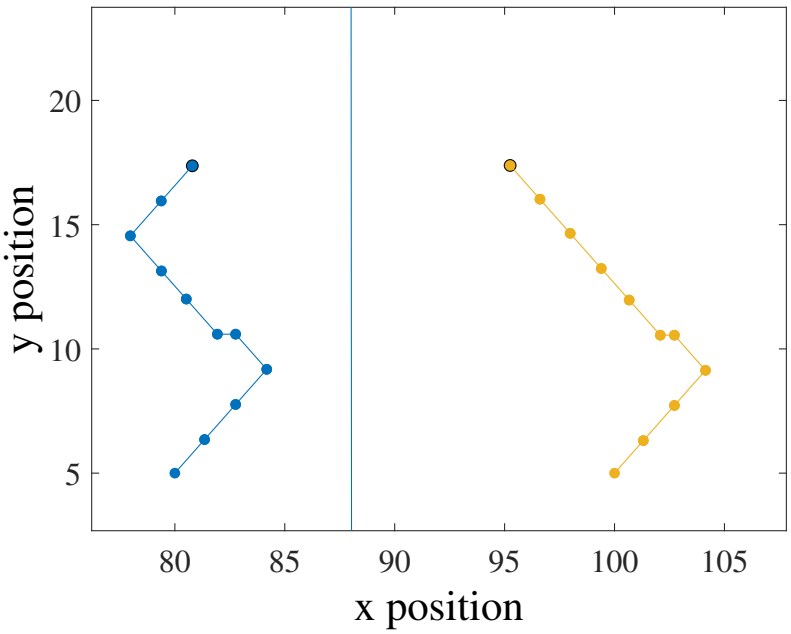


Figure 4.1: Straight-line paths of agents using the BC/PBC/MBC scheme. Paths are shown for 2 agents.

This section describes the path constraints that are introduced for the agents. Two main constraints are introduced; the agent's heading direction and path design. The constraints are detailed as follows.

Heading direction

- The heading direction of each agent is set at the start of the simulation.
- The heading direction is expected to change after each step.
- The heading of an agent is expected to remain the same as the previous step only if the agent travels in straight-ahead motion.
- The heading direction of the agents are logged into the simulation at every iteration continuously.

Path design

- The available paths for the agents are; straight-ahead motion, clockwise (right) turn or an anticlockwise (left) turn.
- The paths are designed as subsets of Dubin's path planning.
- Paths are planned for each iteration.
- The shortest path between the computed and possible points is selected.

Typical path planning procedures in literature are performed differently from the current study. Typically for each agent, the path planning connects a known start position and direction to the desired final position and direction by a path. The goal locations and the initial position of the task is usually determined before the start of the mission. The shortest route between these two points is computed, and the full trajectory of the mission is planned ahead. Examples of these are presented in the works of Shanmugavel et al. [91–93] for multiple UAVs. One of these works is shown in Fig. 4.2. The figure shows that paths designed for the UAVs consist of circular arcs and straight lines connecting the start position(bottom of figure to the goal position(top of figure). Such a path design is known as Dubin's curve [90].

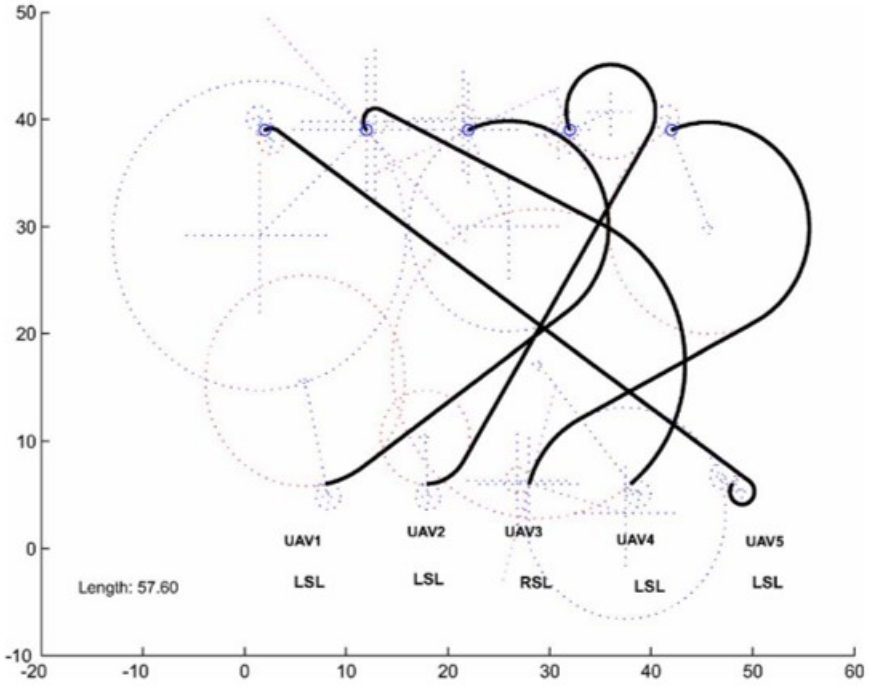


Figure 4.2: Travel paths for multiple UAVs by Shanmugavel et al. [91]

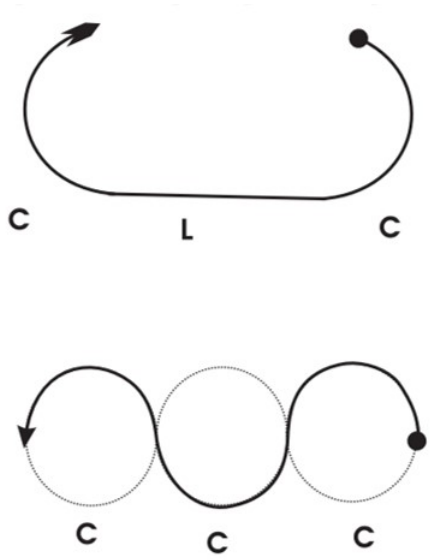


Figure 4.3: Dubin curve's CLC and CCC paths [94]

A Dubins path is the shortest path that connects two configurations in a plane under the restriction of a curvature [90]. In a 2D plane, a line represents the shortest distance between two points, while an arc is the shortest curve of constant curvature—the combination of these two results in the shortest route. Typically the Dubins path is formed by linking two circular arcs with common tangents or three successive tangential circular arcs. The former is a CLC path, and the latter is a CCC path, where C stands for arc and L for the line segment. These paths are depicted in Fig. 4.3. This work focuses on a subset of Dubins's path; either a line or a circular arc. The theory of Dubin state that the path must be a continuously differentiable curve that consists of not more than three pieces, each of which is either a straight line segment or an arc of a circle of radius R. The use of subsets of Dubins curves allows obtaining smooth paths compliant with flight mechanics constraints for the case of UAVs.

Integrating heading direction and path design constraints into to MBC scheme will produce a continuous and feasible path for fixed-wing UAVs. Additionally, in this thesis, the path planning procedure will be conducted for every iteration, as the goal position of the coverage task is not immediately known. For each iteration, the shortest path will also be computed and selected. The method employed to decide on the shortest path will be discussed in the next section.

4.3 Decision Making Procedure for Trajectory Planning

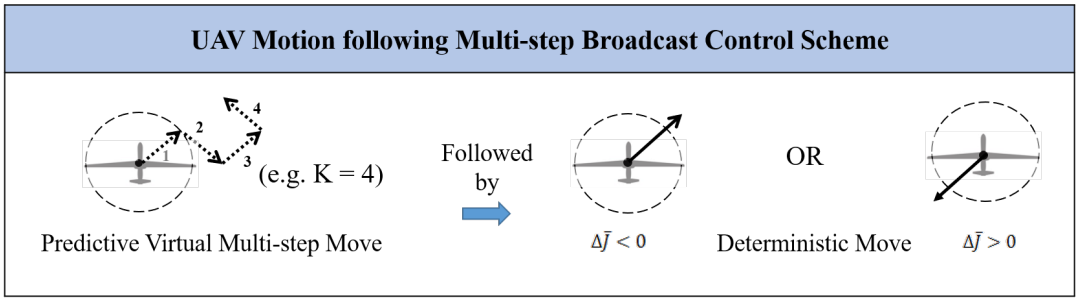


Figure 4.4: UAV taking multi-steps along a horizon in the Multi-step Broadcast Control scheme.

This section describes the process of decision making used to choose between the available paths, which were earlier discussed as path constraints. The alternating virtual and deterministic UAV motion following the MBC scheme is depicted in Fig. 4.4.

Each UAV position in space x, y, z is $\mathbf{x_i}(t)$ and its yaw angle (heading) is $\psi_i(t)$. Throughout the mission, the UAV altitudes are maintained. Therefore, there is no change in z of the UAV pose. The possible paths that the UAV can take are constrained to either a straight-ahead direction, a clockwise (right) or an anti-clockwise (left) turn. The UAV computes the future position of all three path possibilities to make this decision. It calculates the Euclidean distance from these possible points to the point computed by the mission control framework. The path with the shortest difference in distance, dis, is selected. Figure 4.5 illustrates the possible manoeuvres for an UAV.

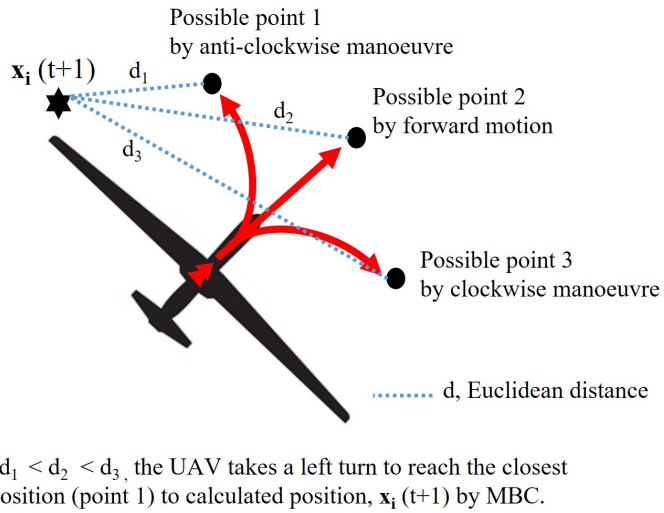


Figure 4.5: Schematic diagram of UAV decision making for 3 possible manoeuvres.

The path for each UAV is calculated following the below steps.

1. Based on the UAV heading, the final pose of the UAV for all three paths of possible manoeuvres is computed.

2. Euclidean distance from all three poses to the predicted point in space computed by the UAV flight controller is determined.

3. The path with the shortest difference in Euclidean distance is selected.

4. UAV performs the selected manoeuvre in Step 3.

5. UAV returns current coordinate to integrated MBC system and the next iteration is repeated.

Following Fig. 4.6, the UAV flies with the selected manoeuvre, and the pose will be either one from the below.

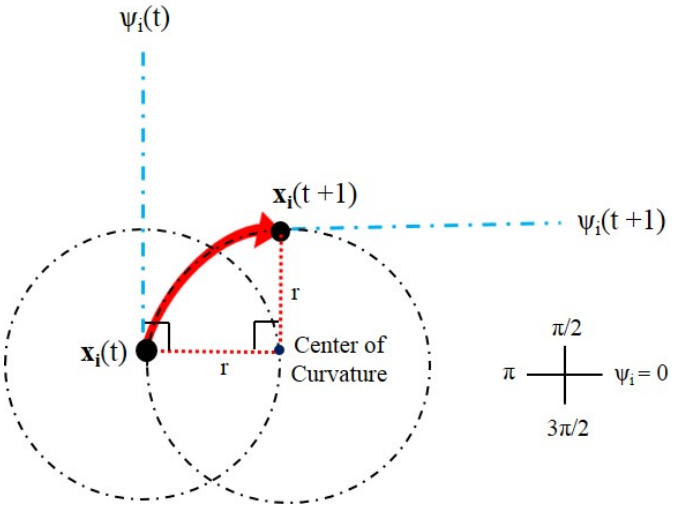


Figure 4.6: Manoeuvre of the UAV : Current position and heading ($\mathbf{x_i}(t)$,$\psi_i(t)$) to next position and heading ($\mathbf{x_i}(t+1)$,$\psi_i(t+1)$).

Pose of UAV at $\mathbf{x_i}(t+1)$ for forward motion:

$$\begin{aligned} A_i : x_i(t+1) &= x_i(t) + r\ \cos(\psi_i(t)), \\ y_i(t+1) &= y_i(t) + r\ \sin(\psi_i(t)). \end{aligned} \tag{4.1}$$

Pose of UAV at $\mathbf{x_i}(t+1)$ for clockwise manoeuvres:

$$\begin{aligned} A_i : x_i(t+1) &= x_i(t) + r\ \cos(\psi_i(t) - (\pi/2)) + r\ \cos(\psi_i(t)), \\ y_i(t+1) &= y_i(t) + r\ \sin(\psi_i(t) - (\pi/2)) + r\ \sin(\psi_i(t)). \end{aligned} \tag{4.2}$$

Pose of UAV at $\mathbf{x_i}(t+1)$ for anti-clockwise manoeuvres:

$$\begin{aligned} A_i : x_i(t+1) &= x_i(t) + r\ \cos(\psi_i(t) + (\pi/2)) + r\ \cos(\psi_i(t)), \\ y_i(t+1) &= y_i(t) + r\ \sin(\psi_i(t) + (\pi/2)) + r\ \sin(\psi_i(t)). \end{aligned} \tag{4.3}$$

4.4 Coverage Problem for a Group of UAVs

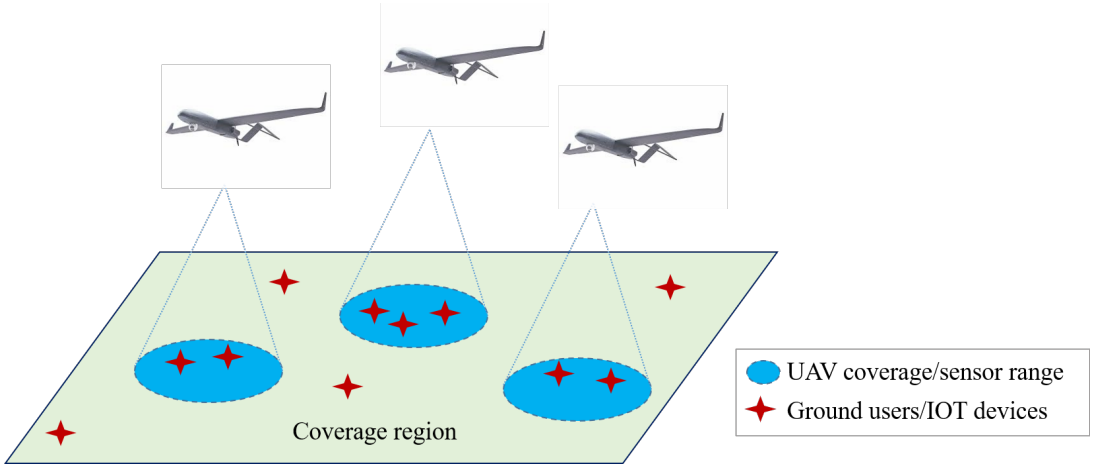


Figure 4.7: Multiple UAVs provide coverage for a region with varying distribution of ground users/IoT devices.

Consider a group of N fixed-wing UAVs ($U_i, i = 1, 2, ..., N$) equipped with wireless ad-hoc capabilities and airborne sensors performing a cooperative area coverage mission in a region, Q such as in Fig. 4.7. The multi-UAVs aim to efficiently provide network coverage for the whole environment considering the ground users (or IoT devices) density distributions. The search environment is taken as a rectangular plane of size l_x x l_y in a two-dimensional Euclidean space.

The following assumptions and requirements about the multi-UAV systems are made for the coverage control problem.

- All the UAVs are homogeneous and fly at the same altitude, H.
- The ground users' position or IoT devices is only known to the cloud/global controller.

- Each UAV calculates its trajectory and flight parameters (heading angles) in real-time.
- To decrease communication load, the UAVs are not connected.
- Mission goal is not made known to individual UAVs to maintain confidentially and security in case of random UAV hijack or loss.

For practical execution of the above assumptions, a cloud system must fully observe the coverage region. Hence, the coverage task can be realistically executed in a smart city environment.

4.4.1 Integration of Loiter Manoeuvre into Coverage Function

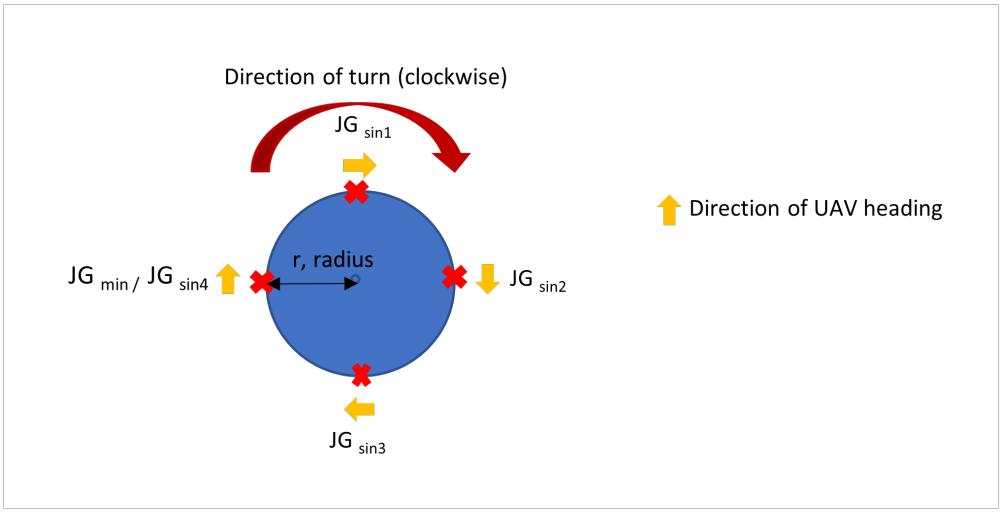


Figure 4.8: Design of loiter manoeuvre.

When the UAVs reach their final positions, they are set to enter a loiter phase. This is done so that the UAVs can fly around their coverage region to provide their services as required to the respective ground users or IoT devices. The UAVs will loiter at the same altitude the loiter was entered into.

Additionally, the loiter phase serves as a transitional phase before the UAVs are assigned to return to the base or to continue towards their next mission. The loiter manoeuvre as represented in Fig. 4.8 can be modelled as a sinusoidal approximation model. This loiter model is included as an additional term in the coverage cost function and is executable at the end of the task. The coverage function inclusive of the loiter model can be represented as:

$$J(x(t)) = J(x(t))_{min} + Asin(t) \tag{4.4}$$

The path taken by a UAV will be a combination of subsets of Dubins' curve and the loiter manoeuvre.

4.5 Results and Discussions

This section presents the results of the integrated MBC scheme with agent constraints. The UAV path trajectories, the evolution of the objective function and the coverage distribution are discussed. The importance of certain sections of the coverage domain is designed using Gaussian density functions. Furthermore, the increase of agents is also modelled to test the adaptability of the scheme.

To begin with, the path taken by a UAV which is a combination of subsets of Dubins' curve and the loiter manoeuvre, is shown in Fig. 4.9. The resulting path travelled by the UAV is continuous and realistic. This is attributed to the fact that the agents (UAVs) are now constrained to move in only two different modes (straight lines or circular arcs) from its heading direction. Therefore, compared to the earlier straight-line paths as in Fig. 4.1 the overall trajectory becomes continuous and less erratic. Figure 4.10 shows the path courses for 4 UAVs.

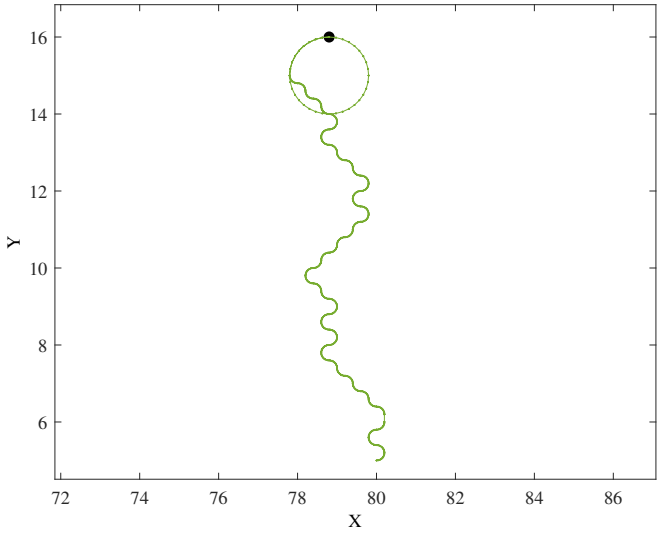


Figure 4.9: Sample path taken by an UAV during the coverage mission. The path's direction is from bottom to top. The circular manoeuvre represents the loiter maneuver of the UAV at the end of the task.

Next, numerical simulations conducted to validate the developed MBC scheme inclusive of path constraints are discussed. Here, four different scenarios are tested. The number of agents in the two-dimensional state space is set to $N = 20$, i.e. $n = 2$. The size of the coverage area is fixed at 100 x 100 square units. The controller gains a and c are determined using the formula below:

$$a(t)|_{MBC} = \frac{a_0}{\left(\frac{t}{2} + 1 + a_v\right)^{a_p}},$$
$$c(t)|_{MBC} = \frac{c_0}{\left(\frac{t}{2} + 1\right)^{c_p}}, \tag{4.5}$$

for t $\in \{0, 1, 2, ...\}$ where $a_0 > 0$ and $c_0 > 0$ hold to satisfy $a(t) > 0$ and $c(t) > 0$. For all scenarios, the value of a_o and c_o are 9.5 and 0.15, respectively. Additionally, the values of a_v, a_p and c_p are set as 15.5, 0.7 and 0.16.

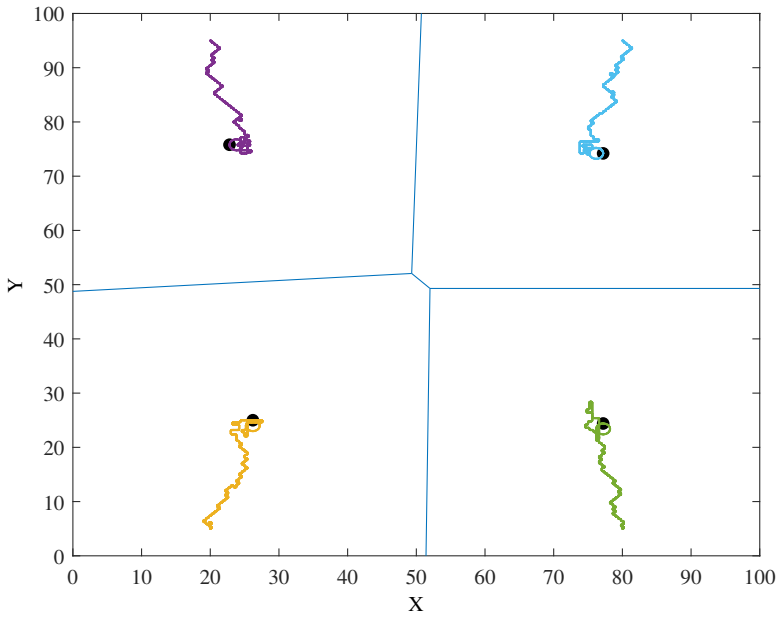
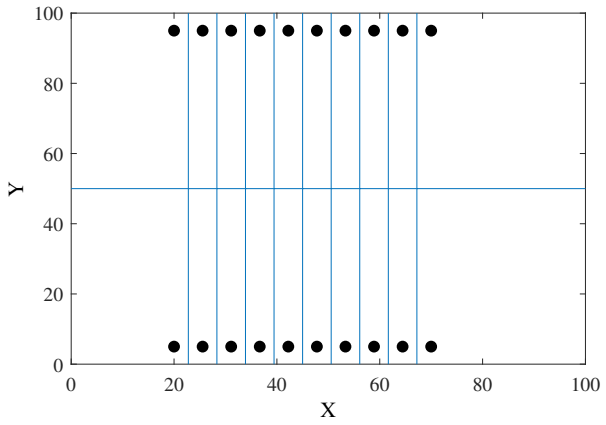


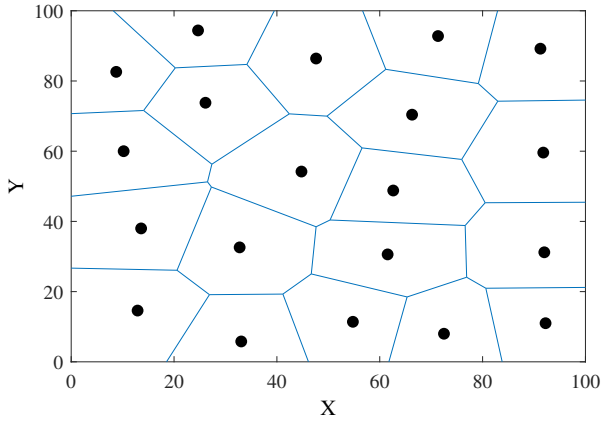
Figure 4.10: Travel path of four UAVs during coverage mission.

1. Uniform density distribution of ground users.

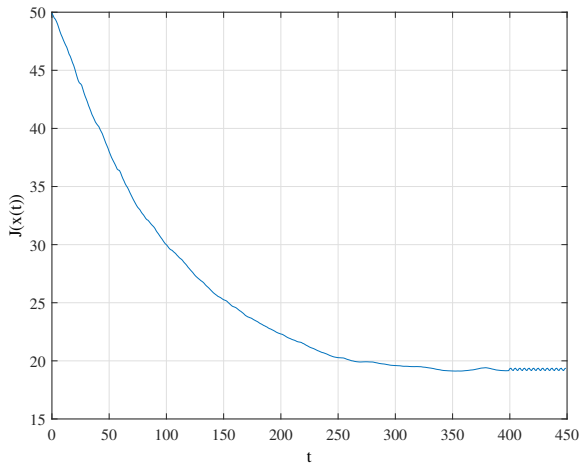
In this task, 20 UAVs are assigned to provide uniform coverage. The UAVs begin from their initial position that is evenly distributed within the x-coordinate (20:70) and y-coordinate (0:100) as shown in Fig. 4.11(a). The UAVs begin to converge to a Voronoi distribution to provide full coverage of the environment from the initial position. The final distribution at convergence is shown in Fig. 4.11(b). Figure 4.11(c) shows that the objective function decreases smoothly. The smooth decrease shows that the MBC scheme can perform seamlessly even when the agents' constraints are applied. From the same figure, minor perturbations of the objective function can be observed from $t = 400$ to $t = 450$, which illustrate loiter manoeuvre being executed.



(a)



(b)



(c)

Figure 4.11: a) Initial Voronoi configuration of 20 UAVs in an environment with a uniform density of ground users/IoT devices. (b) Final centroidal Voronoi configuration of 20 UAVs in an environment with a uniform density of ground users/IoT devices. (c) Mission cost function.

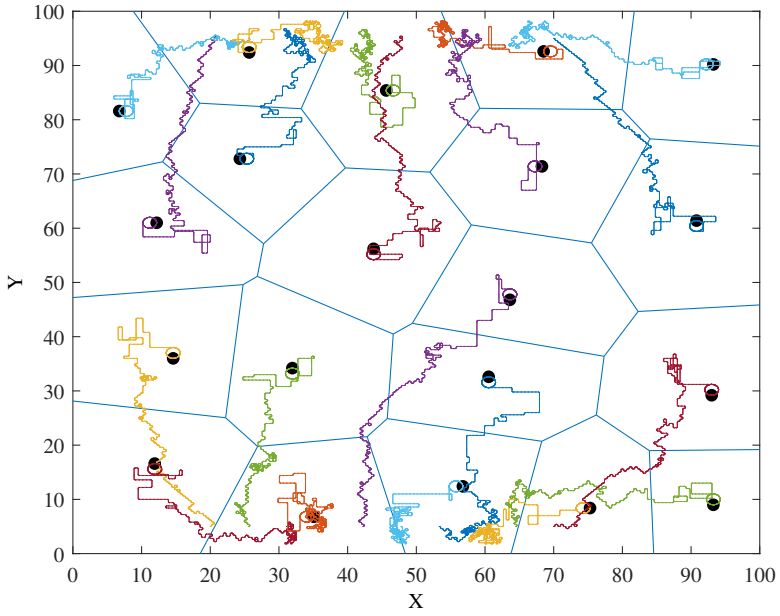


Figure 4.12: Trajectories of UAVs during coverage task with uniform density of ground users/IoT devices.

The complete path trajectory taken by the UAVs is presented in Fig. 4.12. Even with additional decision-making components integrated into the MBC scheme, the UAVs can follow a continuous travel path to reach their final goal.

2. Gaussian density distribution of ground users.

In this task, 20 UAVs are assigned to provide coverage for an area following a Gaussian density distribution function. The distribution used is described as

$$f(x) = 50\exp(-0.5(((X-100)/40)^2) - 0.5(((Y-100)/40)^2)) \qquad (4.6)$$

This distribution is illustrated in Fig. 4.13.

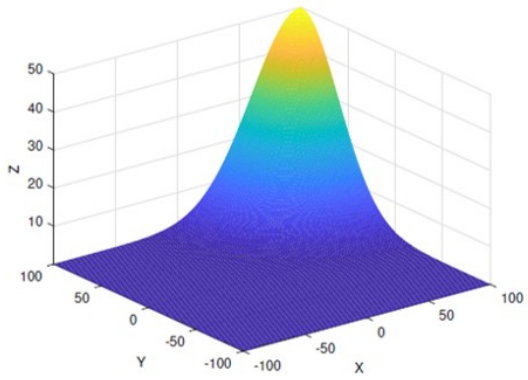
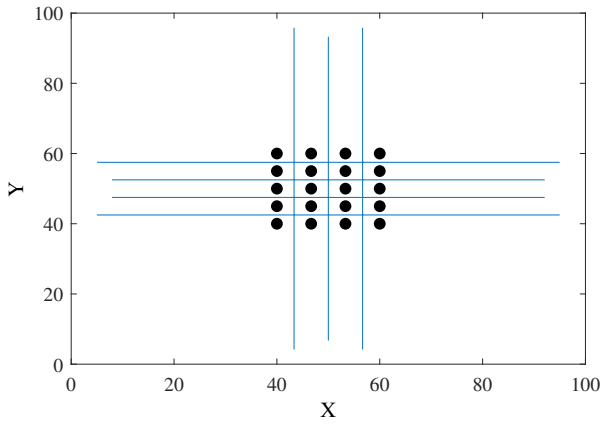


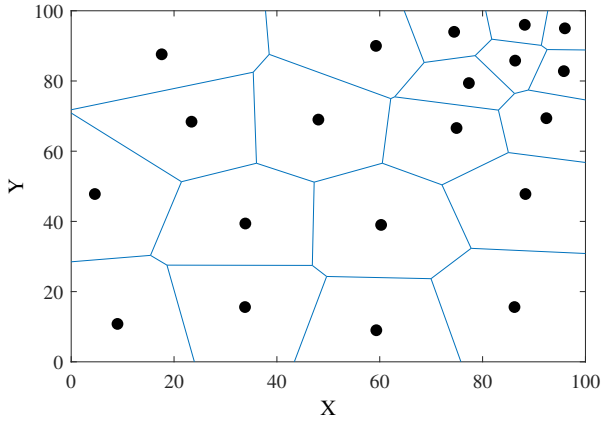
Figure 4.13: Gaussian density distribution.

The UAVs begin from their initial position that is evenly distributed within the x-coordinate (40:60) and y-coordinate (40:60) as shown in Fig. 4.14(a). The UAVs begin to converge to a Voronoi distribution to provide full coverage of the environment from the initial position. The final distribution at convergence is shown in Fig. 4.14(b). Figure 4.14(c) shows that the objective function decreases smoothly. The smooth decrease shows that the MBC scheme can perform seamlessly even when the agent constraints are applied for the case of non-uniform coverage of the domain of interest. Similar in nature to Fig. 4.11(c), minor perturbations of the objective function can be observed from $t = 800$ to $t = 850$ as it illustrates the loiter manoeuvre being performed.

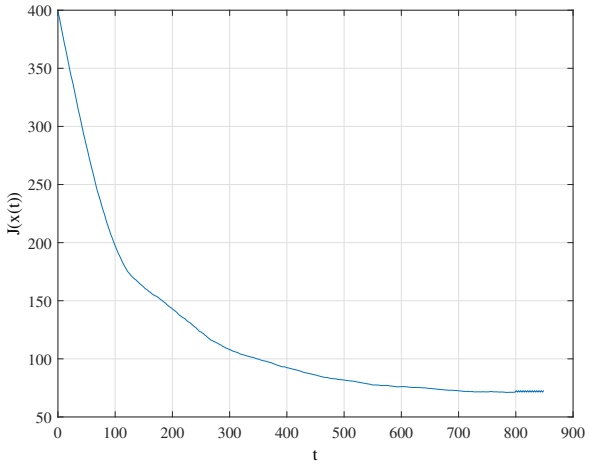
The entire path trajectory taken by the UAVs is presented in Fig. 4.15. The paths show that the UAVs have successfully orientated themselves through their heading direction to move towards the denser part of the coverage environment, located at the top right of the domain, to achieve the coverage mission successfully.



(a)



(b)



(c)

Figure 4.14: (a) Initial Voronoi configuration of 20 UAVs in a non-uniform density of ground users/IoT devices; (b) Final centroidal Voronoi configuration of 20 UAVs in an environment with Gaussian density distribution of ground users/IoT devices; (c) Mission cost function.

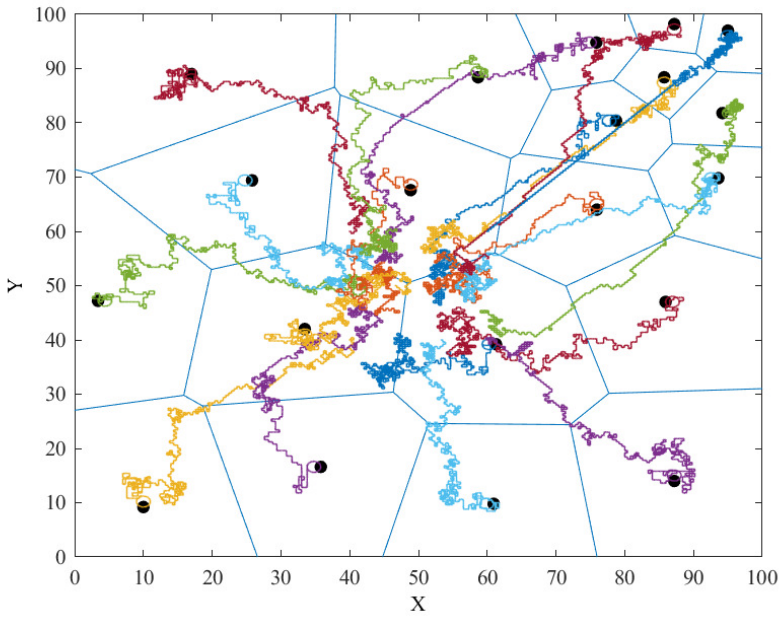
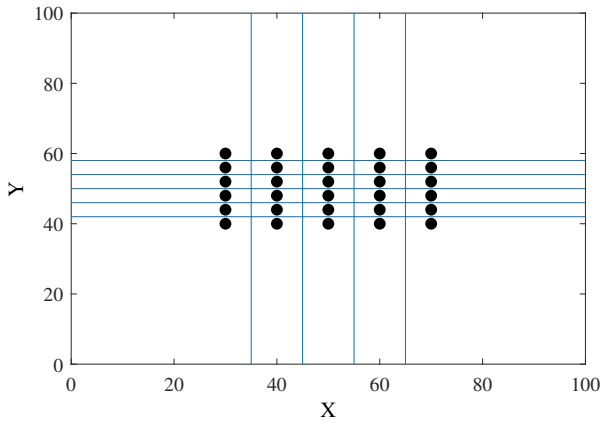


Figure 4.15: Trajectories of UAVs during coverage task with a non-uniform density of ground users/IoT devices expressed using Gaussian function.

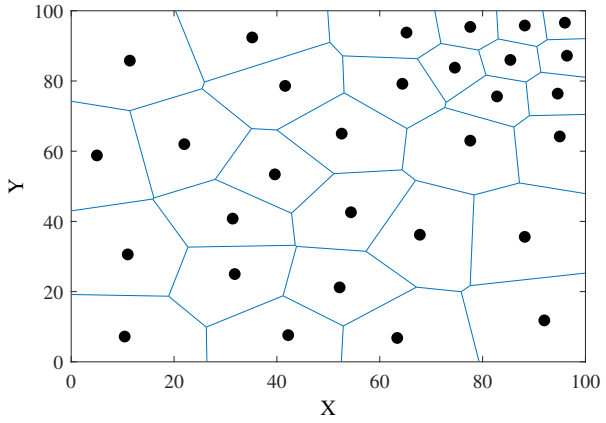
3. Increased number of UAVs (scalability test).

The scalability of the developed scheme in this chapter is validated in this scenario. Here, the number of UAVs being tested is increased to 30. The UAVs begin from their initial position that is evenly distributed within the x-coordinate (30:70) and y-coordinate (40:60) as shown in Fig. 4.16(a). A similar Gaussian density distribution as in Eq. 4.6 is applied to the coverage domain.

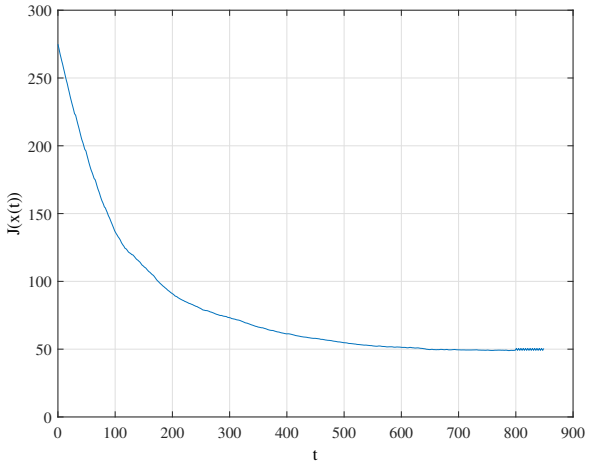
The UAVs begin to converge to a Voronoi distribution to provide full coverage of the environment from the initial position. The final distribution at convergence is shown in Fig. 4.16(b). As expected, the number of agents distributed at the denser section at the top right of the domain has increased compared to the earlier scenario consisting of 20 UAVs.



(a)



(b)



(c)

Figure 4.16: (a) Initial Voronoi configuration of 30 UAVs in an environment with a non-uniform density of ground users/IoT devices; (b) Final centroidal Voronoi configuration of 30 UAVs in an environment with Gaussian density distribution of ground users/IoT devices; (c) Mission cost function.

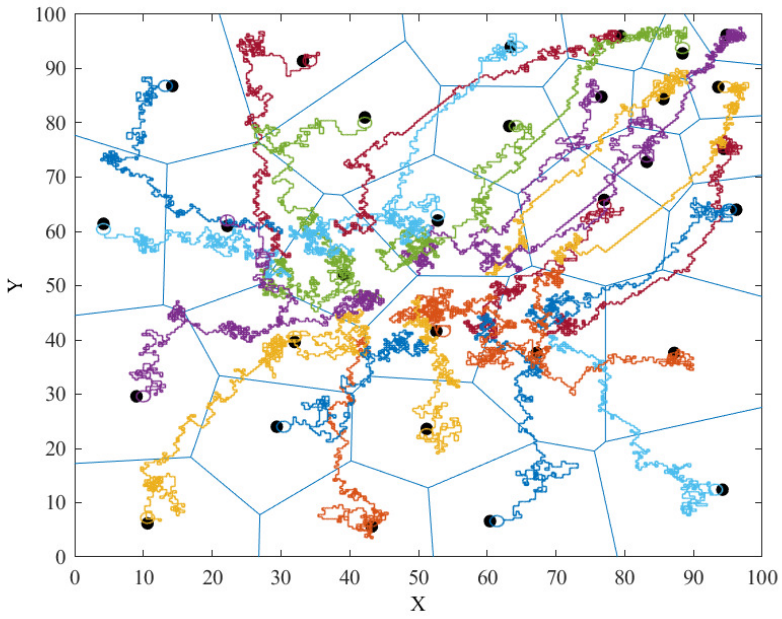


Figure 4.17: Trajectories of UAVs during coverage task with a non-uniform density of ground users/IoT devices expressed using Gaussian function with increased number of agents.

Figure 4.16(c) shows that the objective function decreased smoothly. The smooth decrease shows that the MBC scheme can perform well despite the increase in number of agents in the task. The minor perturbations caused by the loiter manoeuvre can be observed from $t = 800$ to $t = 850$ in Fig. 4.16(c).

The complete path trajectory taken by the UAVs is presented in Fig. 4.17. The paths show that even with the increase in the number of UAVs, the developed MBC scheme can successfully orientate the UAVs to move toward the denser part of the coverage environment while maintaining good coverage throughout the whole domain. However, Fig. 4.15 with 20 UAVs shows visually smoother paths compared to Fig, 4.17 with 30 UAVs.

4. Multiple density distribution of ground users.

The fourth scenario involves varying the density distribution to include multiple centres of densities. Specifically, 20 UAVs are assigned for a mission to provide coverage for an area following a Gaussian density function described as

$$\begin{aligned} f(x) = 50\exp(-0.5 * (((X-60)/40)^2) - 0.5 * (((Y-60)/40)^2)) \\ + 50\exp(-0.5 * (((X+60)/40)^2) - 0.5 * (((Y+60)/40)^2)) \end{aligned} \tag{4.7}$$

This density function is depicted in Fig. 4.18. The corresponding final coverage distribution of the UAVs following the Gaussian function is as shown in Fig. 4.19(a). The coverage distribution shows a good match to the density function used. This shows that the MBC scheme is adaptable to varying the Gaussian functions and is able to provide optimal coverage for the multiple areas in the domain of interest.

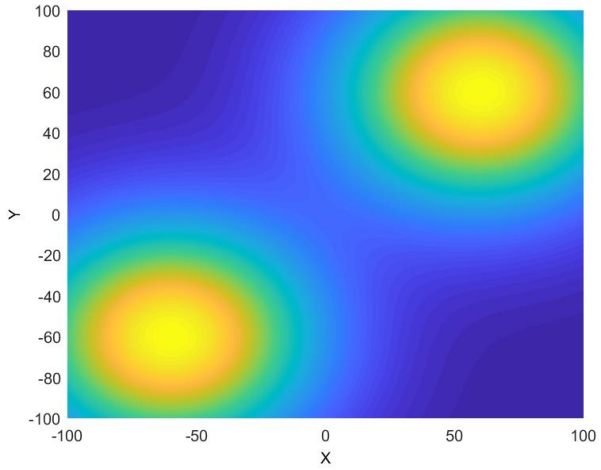
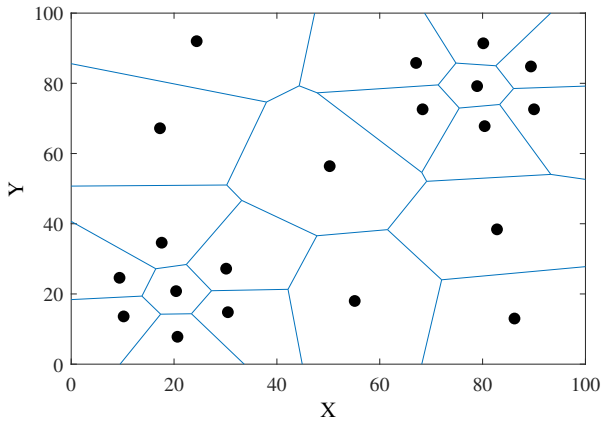
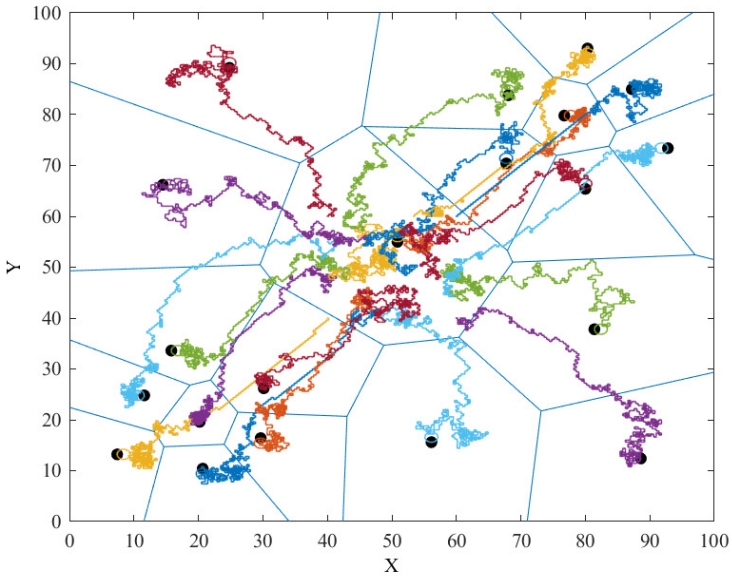


Figure 4.18: Double Gaussian density distribution.



(a)



(b)

Figure 4.19: (a) Final centroidal Voronoi configuration of 20 UAVs in an environment with double Gaussian density of ground users/IOT devices; (b) UAVs' trajectories.

The complete travel path to reach the final distribution is reflected in Fig. 4.19(b). The paths show that the scheme can execute coverage tasks successfully with multiple density centres. This result can be translated to mean that the MBC scheme can effectively allocate the UAVs to the various density distribution of ground users and targets during a practical application.

4.6 Summary

Existing BC schemes have been applied for agents who are flexible in their motions. However, many scenarios require a different outlook if BC schemes were to be used for physical agents. Practical agents are limited in their path constraints. As previous BC schemes did not consider the effect of path constraints on its agents, for the first time in the class of BC algorithms, agent path constraints were introduced in this chapter. Agent path constraints were integrated into the MBC scheme assuming agents as fixed-wing UAVs.

The findings from this chapter show that the integrated MBC scheme is adaptable to changes in path constraints experienced by the agents. For example, constraints such as the heading orientation of agents and the path it takes to reach the desired point have been successfully adapted into the system. The path constraints are able to create a feasible and continuous path for the agents. Contrary to step changes in non-uniformity of the coverage area, as in the previous chapter, continuous distribution functions represent gradual changes in the non-uniformity in the coverage area. The MBC system was shown to be adaptable to such non-uniformity using Gaussian functions and even with the increase in the number of Gaussian centres in an environment.

The numerical simulations also show that the MBC scheme can effectively deploy multiple UAVs, provide coverage uniformly over the mission environment, and adapt to an increased number of UAVs. The next chapter will study the methods to decrease agents' travel distance and aim to reduce execution time.

5 Development of Receding Horizon based Broadcast Control (RHBC) Scheme

5.1 Introduction

In the previous chapter (Chapter 4), an MBC framework that considers the path constraints of physical agents was established using the Dubins curve and heading control. The results demonstrated that the MBC control scheme is adaptable to changes in path constraints experienced by physical vehicle-like agents. However, the agent's travel distance and execution time can be further optimised. Therefore, this chapter aims to achieve both these goals by integrating the receding horizon control technique and momentum-based learning into the existing scheme by proposing an advanced predictive scheme named Receding Horizon based Broadcast Control (RHBC).

In Chapter 3, the Multi-step Broadcast Control (MBC) scheme was proposed. The MBC scheme gives agents views of multiple steps ahead along a predictive virtual horizon. To extend the multi-step concept in a renewed way, a Receding Horizon based Broadcast Control (RHBC) scheme is introduced. The proposed

method is another form of multi-step control. The scheme will be repeated multiple times into the future (prediction horizon) virtually before an optimised direction is obtained from the final prediction horizon. This will significantly reduce the computation time of the BC schemes. The momentum method [95] is a technique for accelerating gradient descent that accumulates a velocity vector in directions of persistent reduction in the objective across iterations. Hence, momentum-based learning will be incorporated into the developed RHBC scheme to decrease the total travel distance of the agents and improve the overall performance of the scheme. The RHBC scheme can be considered a Cyber-physical System (CPS) where the global coordinator and a set of virtual agents perform the computation in the cyber layer. The positional data will be transmitted wirelessly to the physical agents located in the physical layer.

Section 5.2 will introduce the background theories relevant to the proposed scheme. The development of the RHBC scheme will be detailed in Section 5.3. The numerical simulations and analysis will be presented in Section 5.4. Finally, Section 5.5 will summarise the outcome of the RHBC scheme and its contributions.

5.2 Background Theories

5.2.1 Receding Horizon Control (RHC)

Receding horizon control (RHC) is a feedback control technique that became well known in the 1980s. Receding horizon control is also referred to as model predictive control (MPC) in the chemical process control community. The term “model

predictive" refers to the use of the model to predict system behaviour over the planning horizon at each update. Specifically, RHC is a general-purpose control scheme that repeatedly solves a constrained optimisation problem and selects the control action based on predictions of future costs, disturbances, and constraints over a moving time horizon [96].

In RHC, an optimisation problem is solved at each time step in order to establish an action plan over a fixed time horizon. The current control action in receding horizon control is determined by solving a finite horizon optimal control problem online at each sampling instant. Each optimisation produces an optimal control trajectory, also known as an optimal control plan, and the first sample of the plan is used to control the system until the next sampling instant. The control policy includes feedback as real-time measurements are used to determine the control input. The resample and replan action provide feedback to minimise the effects of system uncertainty. A common source of uncertainty is the mismatch between the system model used for planning and the actual system dynamics. The term “receding horizon" refers to the fact that the planning horizon, which is typically fixed, recedes with each update.

RHC has the ability to include generic models, linear and nonlinear, and constraints in the optimal control problem. Therefore, RHC is a good control candidate, suitable for multi-agent systems as it can handle generic state and control constraints.

5.2.2 Momentum-based learning

There is extensive literature on modifications to the stochastic gradient descent algorithms to improve its theoretical and empirical performance using momentum-based learning. The idea of momentum is derived from physical mechanics, which models object inertia. The goal behind using momentum in stochastic gradient descent is to keep the influence of the previous update direction on the following iteration to some extent. The momentum approach can accelerate convergence when dealing with high curvature, small but constant gradients, or noisy gradients [97]. The most notable momentum-based learning methods are the Heavy Ball Method (HB) by Polyak [95] and Nesterov's Accelerated Gradient Method (NAG) [98]. The application of HB to training neural networks can be found in Rumelhart et al. [99] and recently Sutskever et al. [100], have highlighted the importance of such momentum-based methods in the field of deep learning. Such a modification involves the addition of a momentum factor which defines the current search direction g_k as a combination of the current stochastic gradient d_k and previous search directions, g_{k-1}. The stochastic gradient method is represented in the form

$$\theta^{k+1} = \theta^k - a_k g^k \tag{5.1}$$

where $\theta \in R^n$ represents the parameters of a learning model and $a_k > 0$ is the step size or learning rate. The stochastic variant of Polyak's Heavy Ball method [26] can be represented as

$$g^k = d^k + m_f g^{k-1} \tag{5.2}$$

where m_f is the momentum factor ranging between $[0, 1]$, whereas, Gupal and Bazhenov had presented a normalised version of (5.1) and (5.2) as

$$g^k = (1 - m_f)d^k + m_f g^{k-1} \tag{5.3}$$

Directions with a low curvature have a slower local change in their rate of reduction. Therefore, they tend to persist over iterations and are amplified by the momentum factor. Second-order methods, like first-order methods, amplify steps in low curvature directions. Still, instead of accumulating changes, they reweight the update along each eigen direction of the curvature matrix by reversing the associated curvature. Similarly, it has been shown that in HB, the momentum factor can significantly accelerate convergence to a local minimum, requiring $\sqrt{R}$ times fewer iterations than the steepest descent to achieve the same degree of accuracy. The term R is the condition number of the curvature at the minimum and m_f is set to $(\sqrt{R-1})/(\sqrt{R+1})$.

The general influence of the momentum factor on the convergence properties is as follows :

- If the current gradient is parallel to the previous search direction, then the previous search direction can help to accelerate this search.
- If the current gradient is the inverse of the previous search direction, then the previous search direction delays this search.

- When the learning rate is small, the right momentum can help to accelerate convergence.
- When the learning rate is large, the momentum factor helps to reduce the convergence oscillation.
- When the current stochastic gradient derivative reaches zero, it continues to update the current search direction to achieve equilibrium and is dampened by friction factor, $m_f g^{k-1}$.

It is advantageous to depart from the local minimum during the training process for the search process to converge faster [95, 100]. Choosing the appropriate size of the momentum factor is also a challenge. When the momentum factor is small, it is difficult to achieve the effect of faster convergence. The current point can deviate from the optimal value point when the momentum factor is large. Several experiments have empirically confirmed that 0.9 [101] is the best setting for the momentum factor.

5.2.3 Cyber-Physical System (CPS)

Systems that bridge the cyber world of computers and communication with the physical world are called cyber-physical systems (CPS) [102]. Operations of CPS are coordinated and controlled by a computing centre through networks. The embedded computer and network networks monitor and control the physical processes, usually with feedback loops, where physical processes affect the computation and vice versa. The CPS concept is envisioned as a new paradigm for future control systems. It has attracted research attention from both the aca-

demic and industry. Examples of applications of CPS include intelligent vehicles, aerospace systems, robotic systems, precision agriculture, medical devices and systems, smart spaces, defence systems, process control, building and environmental control, and factory automation.

Multi-agent systems and CPS have a lot in common. In the case of broadcast control applicable to multi-agent systems, the design is comparable to a smart factory for industry 4.0. Industrial processes must achieve high flexibility and efficiency in addition to low energy consumption and costs. The multi-agents system is the most representative of the advanced manufacturing schemes proposed to overcome the drawback of current production lines. Here, the manufacturing resources are described as intelligent agents that negotiate with each other to adopt dynamic reconfiguration to attain flexibility.

Nevertheless, due to the lack of some global coordination, the planned multi-agent schemes are not able to efficiently handle complex manufacturing systems. To overcome this, CPS offers a tremendous opportunity to apply smart manufacturing. The CPS can support the multi-agent systems with emerging technologies such as wireless sensor networks, big data, cloud computing, embedded systems, mobile internet and the Internet of Things (IoT).

5.3 Development of Receding Horizon based Broadcast Control (RHBC) Scheme

This section presents the novel scheme proposed in this research, the Receding Horizon based Broadcast Control (RHBC) scheme. The RHBC is inspired by the technique of receding horizon control and incorporates multiple predictive horizons within an iteration. The multiple predictive horizons can reduce system uncertainty and decrease convergence time, t, significantly.

In RHBC, H denotes the number of predictive horizons. At each iteration, the scheme is repeated multiple times, i.e. for a certain number of horizons, H. The state equation of an agent in the physical layer is represented as follows:

$$A_i: \ x_i(t+1) = x_i(t) + \overline{u}_i(t), \ i = 1, 2, \ldots, N \tag{5.4}$$

where $x_i(t) \in \mathbb{R}^n$ is the position in the n-dimensional space and $\overline{u}_i(t) \in \mathbb{R}^n$ is the control input generated in the cyber layer. The control input, $\overline{u}_i(t)$ is represented as

$$\overline{u}_i(t) = \gamma[x_i^H(t) - x_i(t)] \tag{5.5}$$

where γ represents the "trustworthiness factor". Only a certain percentage of the direction of gradient from the predicted horizon is "trusted" and taken into account for the computation for the physical step. This is because of the limited horizon, there are uncertainties in the system. Only if the predicted horizon is infinite,

100% of its value can be fully trustworthy. The notation $x_i^H(t)$ represents the final state of the agents at horizon H.

The state-space equation of agent i within the cyber layer is given as

$$\hat{x}_i^{(h+1)}(t) := \hat{x}_i^{(h)}(t) + \hat{u}_i^{(h)}(t) \text{ for } h=0,1,2,\ldots,H\text{-}1 \tag{5.6}$$

$$\hat{u}_i^{(h)}(t) := c(t)\Delta_i^{(h)}(t) \text{ for } h=0,1,2,\ldots,H\text{-}1 \tag{5.7}$$

where $\hat{x}_i^{(h+1)}(t)$ and $\hat{u}_i^{(h)}(t)$ are the virtual predictive state and virtual input of agents A_i at each horizon.

The Global Controller, GC, in RHBC calculates the objective function, J, for the virtual step as follows:

$$\sigma_R{}^h(t) := J(\hat{x}^{(h+1)}(t)) - J(\hat{x}^{(h)}(t)) \tag{5.8}$$

where $J(\hat{x}^{(h+1)}(t)) = J(\hat{x}^h(t) + c(t)\Delta_i^{(h)}(t))$.

While, the control output, $u_i(t)$, of the local controller, L_i, at each horizon which is also predictive is given as

$$\hat{u_i}{}^h(t) := -a(t)\frac{\sigma_R{}^{(h)}(t)}{c(t)}\Delta_i^{(h)[-1]}(t), \tag{5.9}$$

Next, applying the momentum factor m_f, $u_i(t)$ is transformed into

$$\hat{u}_i^h(t) = (1 - m_f)\hat{u_i}{}^{(h)}(t) + m_f\hat{u_i}{}^{(h-1)}(t) \tag{5.10}$$

Eq. 5.6 to Eq. 5.10 is repeated until $h = H$. The final agent state ${x_i}^H(t)$ will be used in Eq. 5.5 before the control input $\overline{u}_i(t)$ is transmitted to agents in physical layer as in Eq. 5.4.

The RHBC scheme can be summarised as follows.

1. At time t and the current state x_i, the BC scheme is repeated virtually a number of times, i.e., for the number of horizons, $h = 1, ..H$.
2. The final agent state, ${x_i}^H$ at H is computed.
3. The final agent state is used to compute the local controller output, $\overline{u}_i(t)$.
4. The local controller output, $\overline{u}_i(t)$ is transmitted to physical agents for execution.
5. Repeat the fixed horizon optimisation at time $t+1$ for future virtual horizons, H, starting from the current state $x_i(t + 1)$.

The RHBC is proposed as a CPS model with the control scheme situated in the cyber layer and the physical agents in the physical layer. The position data is broadcasted to the physical agents after multiple iterations in the cyber layer to be executed in the physical layer by the agents. The cyber layer is assumed to be able to observe the physical layer with appropriate sensors and collect positional data from each agent individually. This automation hierarchy is presented in Fig. 5.1 and the detailed CPS model for RHBC is given in Fig. 5.2.

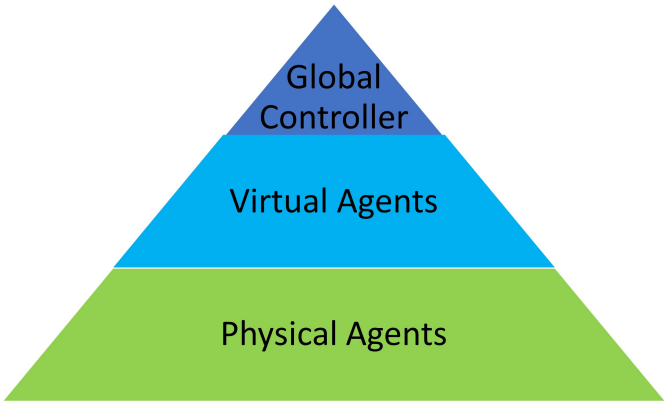


Figure 5.1: The automation hierarchy for the proposed RHBC scheme in a CPS setting.

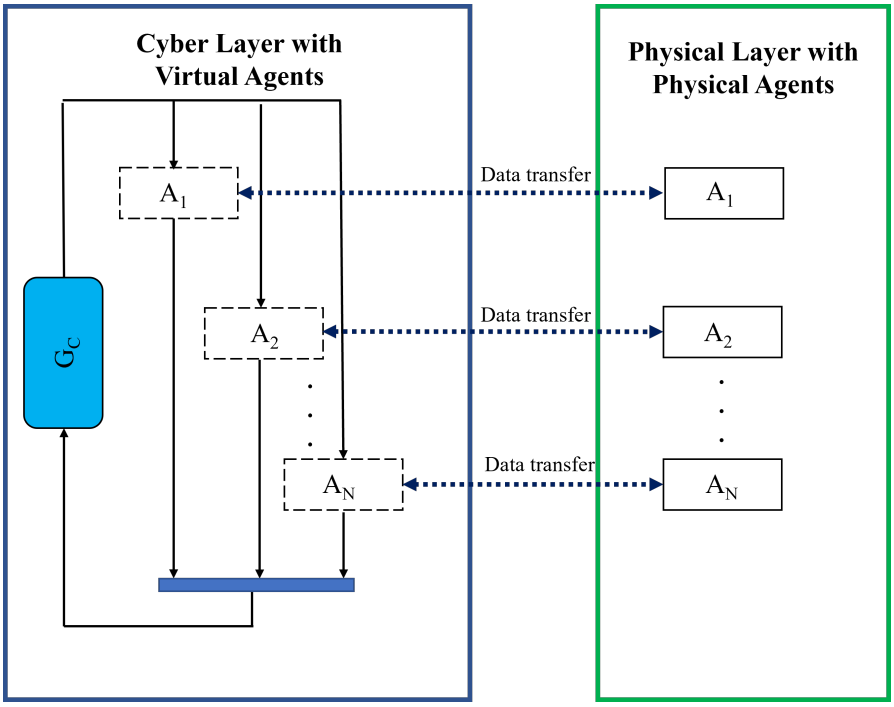


Figure 5.2: The CPS model of RHBC with numerical computation by the global controller and virtual agents in the cyber layer communicating positional data to physical agents in the physical layer.

5.4 Numerical Simulations and Analysis

This section evaluates the effectiveness of the proposed RHBC scheme. The RHBC scheme was developed as an extension to the multi-step BC scheme proposed in Chapter 3. It can be generally considered a multi-step design because, in RHBC, the multi-step is running the whole scheme virtually for a certain number of iterations into the future, i.e. for a certain number of *horizons, H*. Besides the virtual horizons, a *momentum factor*, m_f has been introduced and varied. The effect of these two parameters on the computation time and the travel distance of the agents are investigated.

The number of agents in the two-dimensional state space is set to $N = 25$, i.e. $n = 2$. The size of the coverage work area is fixed at 200 x 200 square units. Initially, all agents are evenly distributed within the x-coordinate (75:125) and y-coordinate (0:25). The values of horizon used are $H = [4, 8, 12, 16, 20]$ and the values of momentum factor used are $m_f = [0.0, 0.3, 0.5, 0.7, 0.9]$. The controller gains a and c are determined using the formula as below:

$$\begin{aligned} a(t)|_{MBC} &= \frac{a_0}{\left(\frac{t}{2} + 1 + a_v\right)^{a_p}}, \\ c(t)|_{MBC} &= \frac{c_0}{\left(\frac{t}{2} + 1\right)^{c_p}}, \end{aligned} \tag{5.11}$$

for t $\in \{0, 1, 2, ...\}$ where $a_0 > 0$ and $c_0 > 0$ hold to satisfy $a(t) > 0$ and $c(t) > 0$. For all scenarios, the value of a_o and c_o are 8.5 and 0.1, respectively. Additionally, the values of a_v, a_p and c_p are set as 15.5, 0.45 and 0.1.

Table 5.1 presents the performance of the proposed RHBC scheme against the existing BC schemes, namely the BC and PBC schemes. The total travel distance, time of convergence (t), computation time per step and total computation time are tabulated. The table shows that the RHBC has the lowest travel distance compared to PBC and BC. It is expected for BC to have more than twice the distance travelled as all the steps taken are taken physically, including the random steps; this is unlike PBC and RHBC, where all random steps are computed virtually, and hence virtual distance is not calculated.

RHBC also shows significant improvement in terms of the time of convergence, where the scheme converges the quickest at $t = 130$, while PBC converges at $t = 450$ and BC at $t = 860$. The BC scheme is the fastest in terms of computation time per step, followed by PBC and RHBC. However, all three schemes still perform in the range of *milliseconds* and the differences can be considered negligible.

Table 5.1: Comparison between BC schemes

Scheme	Total Travel Distance (unit distance)	Convergence at Iteration	Computation Time per Step (ms)
BC	5.26×10^4	860	4
PBC (K=10)	1.73×10^4	450	23
RHBC (H=10)	1.32×10^4	130	36

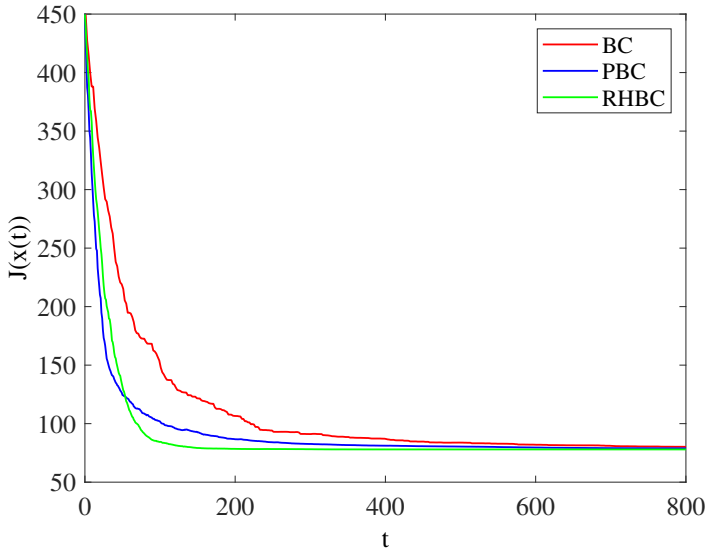


Figure 5.3: Comparison between BC schemes. The proposed RHBC scheme convergences to the solution at the most optimal rate, which is by $t = 130$, followed by PBC and BC respectively at $t = 450$ and $t = 860$.

Figure 5.3 shows the evolution of the objective function, which illustrates the objective function against the time taken for convergence for the BC schemes being compared. It is evident from the figure that the developed RHBC scheme shows the quickest convergence, hence achieving the best performance.

Table 5.2 compares the communication volume between BC, MBC and RHBC. For N number of agents, BC send $2N + 1$ data. Typically both-way communication volume of BC is almost half of that of centralised systems employing unicast protocols. For either PBC or MBC, the data amount is multiplied by the amount of virtual predictive steps sent to the global controller, and therefore the communication volume will be $2NK + 1$. Since the RHBC scheme operates within a cyber-physical layer, the both-way communication volume is greatly reduced compared to PBC and MBC and stands at $4N$.

Table 5.2: Comparison between BC schemes for Communication Volume

Scheme	Communication Volume per Step (N agents)
BC	$2N+1$
PBC/MBC (K virtual steps)	$2NK+1$
RHBC	$4N$

Table 5.3 shows the relation between the change in the number of horizons and the momentum factor. The trend in results illustrates that the travel distance decreases with the increase in momentum factor used. This is in line with the expected performance as the introduction of momentum factor to the gradient computation has increased the scheme's accuracy and hence lowered the total distance travelled by the agents. With the increase of horizon number, the computation time per step for the agents will increase. Nevertheless, these values ranging from $14ms$ for $H = 4$ to $65ms$ for $H = 20$ can be considered a minimal increase.

Table 5.3: Performance based on change of number of horizons, H and momentum factor, m_f

Horizon Number : 4		
Momentum Factor (m_f)	Total Travel Distance (unit length)	Computation Time per Step (ms)
0.0	1.60×10^4	14
0.1	1.51×10^4 (-6%)	
0.3	1.15×10^4 (-27%)	
0.5	8.09×10^3 (-30%)	
0.7	6.19×10^3 (-21%)	
0.9	5.37×10^3 (-13%)	

Horizon Number : 8		
Momentum Factor (m_f)	Total Travel Distance (unit length)	Computation Time per Step (ms)
0.0	1.41×10^4	27
0.1	1.26×10^4 (-11%)	
0.3	9.09×10^3 (-28%)	
0.5	7.07×10^3 (-22%)	
0.7	5.41×10^3 (-24%)	
0.9	4.15×10^3 (-23%)	

Horizon Number : 12		
Momentum Factor (m_f)	Total Travel Distance (unit length)	Computation Time per Step (ms)
0.0	1.22×10^4	41
0.1	1.12×10^4 (-8%)	
0.3	8.46×10^3 (-26%)	
0.5	6.74×10^3 (-20%)	
0.7	4.85×10^3 (-28%)	
0.9	3.93×10^3 (-19%)	

Horizon Number : 16		
Momentum Factor (m_f)	Total Travel Distance (unit length)	Computation Time per Step (ms)
0.0	1.21×10^4	55
0.1	1.09×10^4 (-10%)	
0.3	8.72×10^3 (-20%)	
0.5	6.75×10^3 (-23%)	
0.7	5.00×10^3 (-24%)	
0.9	3.86×10^3 (-23%)	

Horizon Number : 20		
Momentum Factor (m_f)	Total Travel Distance (unit length)	Computation Time per Step (ms)
0.1	1.19×10^4	65
0.1	1.09×10^4 (-8%)	
0.3	8.41×10^3 (-23%)	
0.5	6.34×10^3 (-30%)	
0.7	4.90×10^3 (-23%)	
0.9	3.64×10^3 (-26%)	

Table 5.3 also shows that the performance varies greatly between horizon 4 and 12. Performances of horizons 12, 16 and 20 can be considered similar. This implies that the increase in horizon does not need to be infinite, and after a given horizon number, the performance will tend to plateau. The horizon number where the system's performance will plateau can be identified and selected. Here, the optimal value for the horizon will be $H = 12$. Figure 5.4 presents this result clearly, where we can see that performance of $H = 12$, $H = 16$ and $H = 20$ are similar. A noticeable difference between $H = 4$, $H = 8$ and $H = 12$ can be seen in the figure.

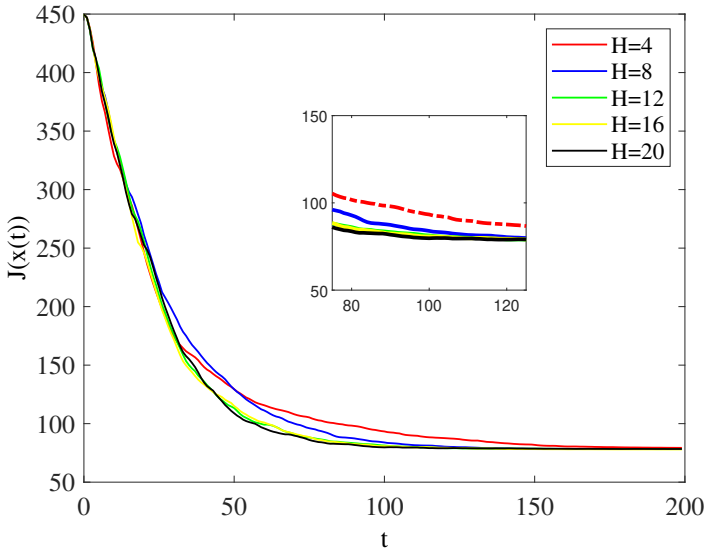
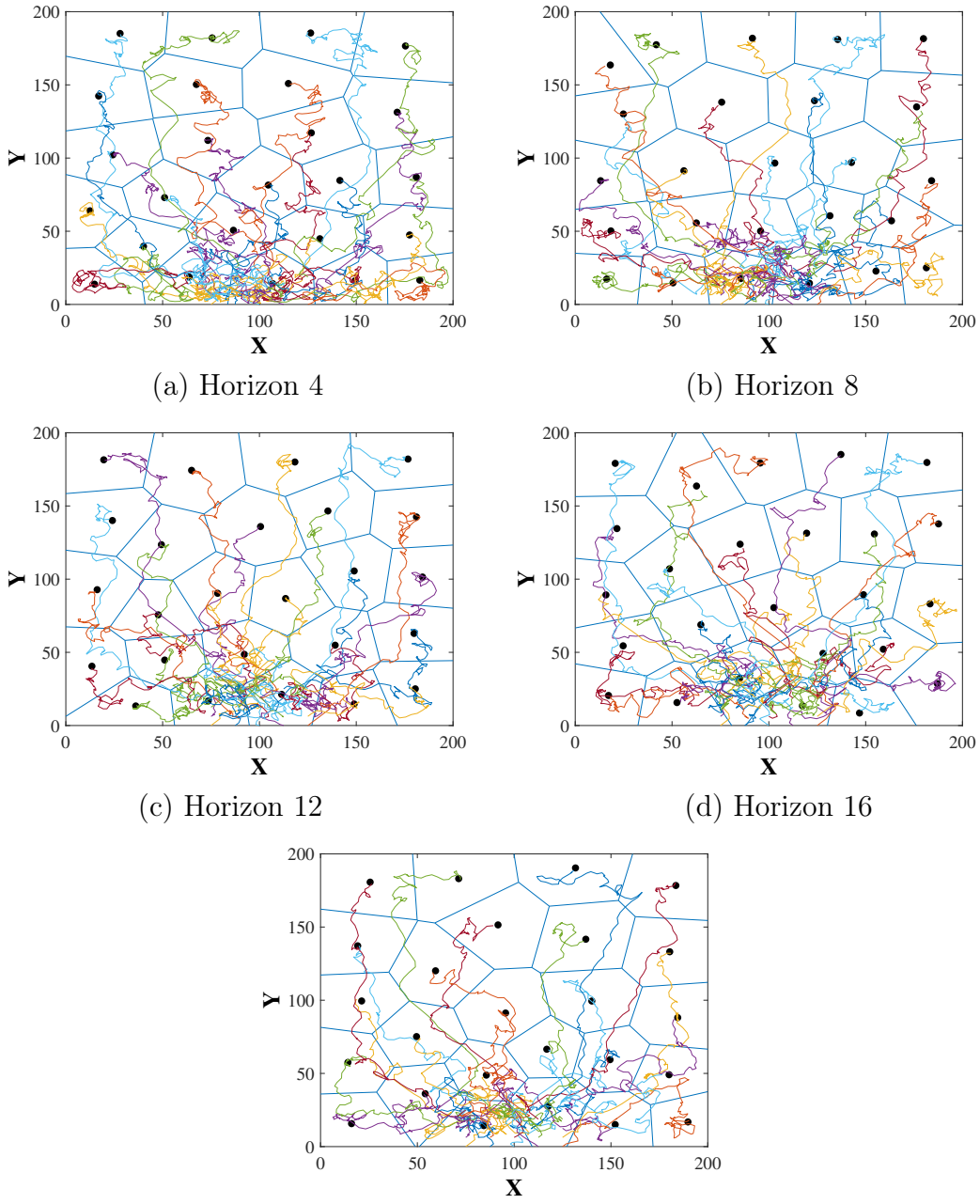


Figure 5.4: Comparison of the evolution of coverage objective function for different horizon numbers, H.

Figure 5.5 shows the travel path agents take to reach their final position for varying horizon numbers from 4 to 20 and a fixed momentum factor of 0.5. Here, through some careful observance, the path travelled by the agents is seen to be becoming smoother (lower in distance value) with horizon number as denoted in Table 5.3. A significant decrease in distance travelled by the agents can be visualised by maintaining a horizon number and varying the momentum factor. This is presented in Fig. 5.6.



(a) Horizon 4

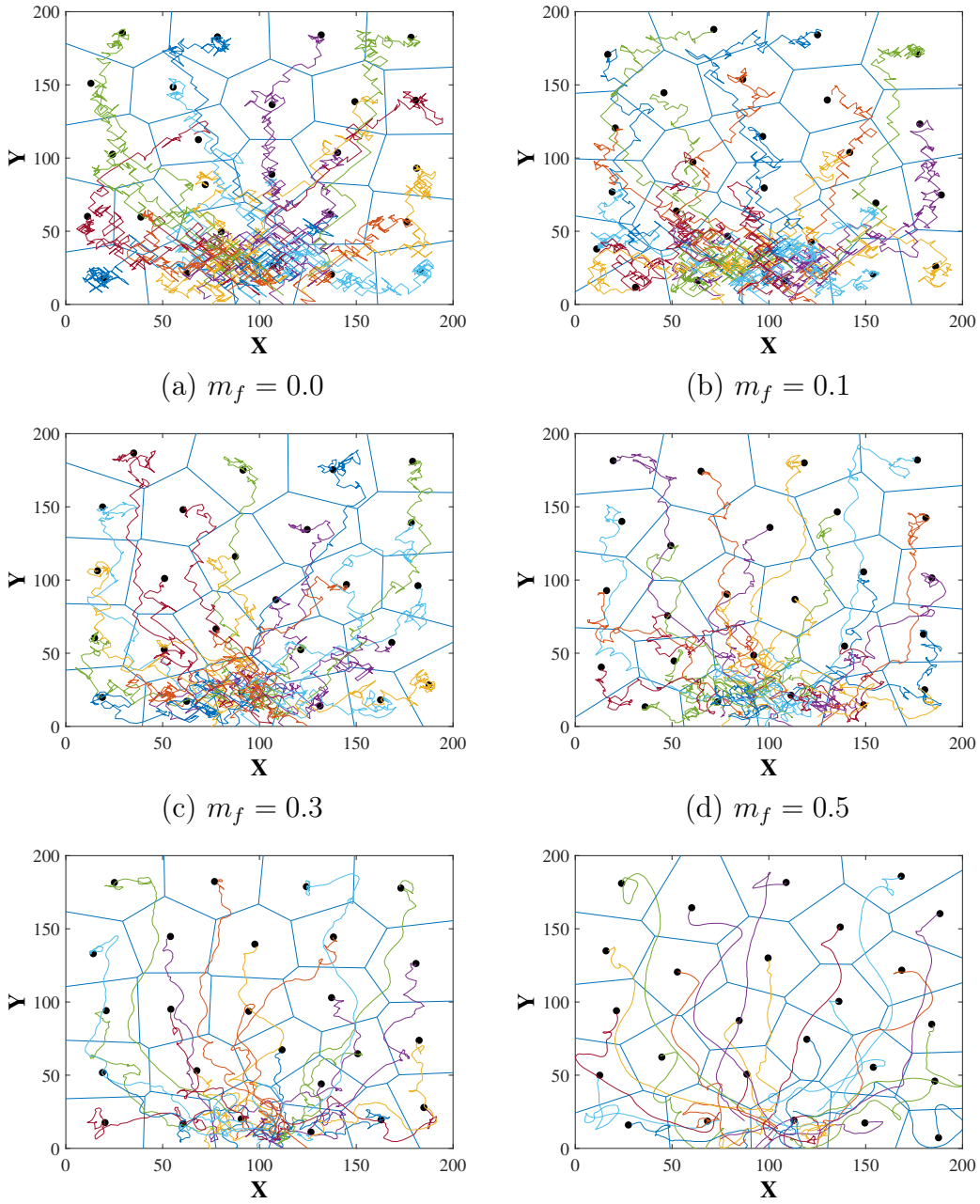
(b) Horizon 8

(c) Horizon 12

(d) Horizon 16

(e) Horizon 20

Figure 5.5: Effect of a varying number of horizons, H on agent travel path. The final agent distribution and travel path at convergence are shown for $m_f = 0.5$



(a) $m_f = 0.0$ (b) $m_f = 0.1$

(c) $m_f = 0.3$ (d) $m_f = 0.5$

(e) $m_f = 0.7$ (f) $m_f = 0.9$

Figure 5.6: Effect of varying momentum factor, m_f on agent travel path. The final agent distribution and travel path at convergence is shown for $H = 12$

Figure 5.6 incrementally illustrates that the agent travel path becomes less erratic as m_f increases from 0 to 0.9, which infers the agents travel a shorter distance. In this study, the horizon number is set to be 12, as chosen by the convergence performance in Fig. 5.4. Momentum factor, m_f value of 0.9 gives the smoothest path and hence the lowest value of total travel distance of 3.93×10^3 in unit length, as shown in Table 5.3 for $H = 12$. Table 5.3 also presents the percentage decrease in the amount of distance travelled for each horizon number as the value of the momentum factor is increased. A substantial reduction in unit distance travelled can be noted from the tabulated data. This indicates that the inclusion of the momentum factor in the proposed RHBC scheme has contributed significantly to the optimal performance of the scheme. The evolution of the coverage objective function is shown in Fig. 5.7.

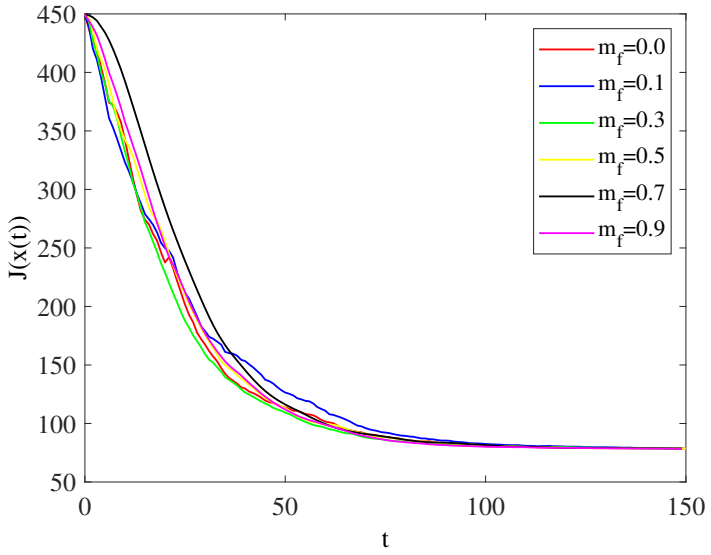


Figure 5.7: Comparison of the evolution of coverage objective function for different momentum factors, m_f.

5.5 Summary

In this chapter, a novel and optimal RHBC scheme was developed. The RHBC supersedes the performance of existing BC schemes in the literature, mainly BC and PBC schemes. The main component that contributed to the success of the RHBC scheme is the inclusion of multiple forward views of the system, inspired by the receding horizon control (RHC) technique, a well-established feedback control technique. Furthermore, the inclusion of the momentum-based learning technique in the proposed RHBC scheme demonstrated superior capability in reducing the distance travelled by the agents.

The RHC technique incorporated into the RHBC scheme includes sampling multiple horizons (multiple repeats of the BC scheme) in the virtual space in a cyber-physical model. The increase of horizon reduces system uncertainty, thus increasing system accuracy. Since the RHC is suitable for generic systems, it is ideal for BC's multi-agent system scheme. The results demonstrated that increasing the horizon increases optimisation accuracy. Horizon number 12 offers the best performance for the scheme as the performance reaches a plateau after a significant increase in horizon number. This shows that the horizon number does not need to be increased infinitely. As expected, the computation time increases with the rise in horizon number. However, the increase from $14ms$ to $65ms$ is acceptable and negligible.

Momentum based learning incorporated in the RHBC system has accelerated the direction towards the final gradient direction and reduced the distance travelled. An increase in the momentum factor significantly reduces the total distance

travelled by the agents. Key findings indicate a decrease of at least 20% of unit distance for a rise in 20% of the value of the momentum factor. This shows that the inclusion of momentum-based learning effectively improves the performance of existing BC systems. The proposed RHBC scheme stands as the most optimised BC scheme to date.

6 | Conclusions and Future Work

6.1 Summary of Main Results

In Chapter 2, multi-agent systems and particularly their application in coverage coordination task was introduced. Then, a review of a Broadcast Control (BC) scheme that admits a general multi-agent state-space model was given. Next, a review of predominant examples of successful implementation of broadcast control combined with the receding horizon technique was explored. Finally, applications of BC-based Cyber-Physical System (CPS) infrastructure were explored.

The inspiration for this thesis is the application of the predictive nature of the receding horizon technique. The generality of the receding horizon control technique made it easy to be modified and adaptable to the broadcast control scheme. As BC schemes themselves operate as a low communication volume problem, the inclusion of the receding horizon technique into the BC scheme improved the performance without a significant increase in computation time. This could assist in adding many agents into the system, and the system will configure even better.

In Chapter 3, the development of the MBC scheme was introduced and analysed. The MBC scheme is generated to allow individual agents to take multiple

predictive virtual steps along a horizon before a physical step is executed. Through this, the agents were capable of predicting a foresighted target of the environment. This scheme is designed to overcome the limited one-step look-ahead of the existing BC schemes. Using λ, the immediate predictive step is given a higher weight, and the weighting is gradually reduced as the number of steps increases. Convergence of the proposed MBC scheme is proven by showing that the transition of the MBC scheme converges to $\partial J(x) = 0$, which implies that the system (collective agent dynamics) has and reaches a local minimum value for the coverage problem.

In MBC, the local controller output has been calculated using a weighted averaging technique that better describes a specific problem and has better accuracy than PBC, which uses the arithmetic averaging technique as a quick general was tested in terms of the coverage problem with single and double varying density sections and for an increasing number of agents. The study results suggest that the introduction of multiple virtual steps in a horizon increases task fulfilment and deployment efficiency. It is noteworthy that MBC outperforms the existing BC schemes and converges the fastest as the gradient is calculated from several steps along the horizon where the dense sections appear earlier to the agents compared to BC and PBC.

As previous BC schemes did not consider the effect of applying path constraints to their agents, in Chapter 4, this thesis introduced agent path constraints for the first time in the class of BC algorithms. Using fixed-wing unmanned aerial vehicles (UAVs) as agents, path constraints were designed and integrated as subsets of the Dubins model into the MBC scheme. Agent heading orientation was also included as a new feature in the class of BC algorithms. The findings from this chapter show that the MBC scheme is adaptable to changes in path constraints experienced by

the agents.

The path constraints created a feasible and continuous path for the UAVs. Chapter 4 also showed that the MBC scheme is adaptable for varying density distribution using continuous distribution functions such as Gaussian density distribution. Contrary to step changes in non-continuous density distribution as studied in Chapter 3, continuous distribution functions show a constant and smooth decrease in the coverage loss function. The numerical simulations show that the proposed framework can effectively deploy multiple UAVs and provide coverage uniformly over the mission environment and is adaptable to an increased number of UAVs and an increase in the number of Gaussian centres in an environment.

In Chapter 5, a novel and optimal BC scheme, the Receding Horizon based Broadcast Control (RHBC), was proposed and developed. The scheme was executed by sampling multiple horizons (multiple repeats of the BC scheme) in a cyber-physical system framework. Moreover, the momentum-based learning technique integrated into the developed RHBC scheme demonstrated superior capability in reducing the distance travelled by the agents. The proposed RHBC supersedes the performance of existing BC schemes in the literature, mainly BC and PBC schemes. The results show that the increase of horizon reduces system uncertainty, thus increasing optimisation accuracy. The results showed that the horizon number does not need to be increased infinitely for the scheme as the performance reaches a plateau after a significant increase in horizon number. As expected, the computation time increases with the rise in horizon number.

However, the increase of computation time in milliseconds can be considered negligible with the upcoming 5G and 6G technology which offers better bandwidth

and lower latency. Momentum based learning incorporated in the RHBC scheme has accelerated the direction towards the final gradient direction and reduced the distance travelled. An increase in the momentum factor significantly reduces the total distance travelled by the agents. Key findings indicate a decrease of at least 20% of unit distance for a rise of 20% of the value of the momentum factor. The proposed RHBC scheme stands as the most optimised BC scheme to date.

6.2 Future Work

The development in this thesis leads to the advancement of Broadcast Control (BC) schemes and its application in multi-agent systems. However, there are several exciting extensions of the work presented in this thesis – and within the context of multi-agent/robot/vehicle in general.

Regarding the use of coverage control for analysis of the scheme, the coverage density function was assumed to be stationary. Future work can extend the analysis to dynamic density function or moving targets in the environment. The same framework can include the presence of moving obstacles in the coverage environment. Furthermore, homogeneous agents in this thesis could be extended to heterogeneous agents with varying sensing capabilities. Additionally, packet losses in the data transfer process and the other interesting outlooks such as security of the shared information to the agents can be explored.

As there is an increasing number of engineering challenges that can employ receding horizon control due to the utility of the predictive mechanism and the flexibility to include generic constraints and performance objectives, the contribu-

tions of the thesis through the proposed MBC and RHBC schemes bear relevance for multi-agent systems. Other multi-agent algorithms are yet to be evaluated with the same outlook. Nevertheless, the BC schemes for multi-agent systems and its applications remain primarily unexplored, with endless possibilities to be developed in this domain.

7 | References

[1] A. M. Uhrmacher and D. Weyns, *Multi-Agent systems: Simulation and applications*. CRC Press, 2009.

[2] M. Wooldridge, "Agent-based software engineering," *IEE Proceedings-Software Engineering*, vol. 144, no. 1, pp. 26–37, 1997.

[3] J. Shamma, *Cooperative control of distributed multi-agent systems*. John Wiley & Sons, 2008.

[4] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*, pp. 193–200, IEEE, 1997.

[5] Z. Tang and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, 2005.

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

[7] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

[8] P. Sujit and R. Beard, "Distributed sequential auctions for multiple uav task allocation," in *2007 American Control Conference*, pp. 3955–3960, IEEE, 2007.

[9] Y. Wang, I. I. Hussein, and R. S. Erwin, "Awareness-based decision making for search and tracking," in *2008 American Control Conference*, pp. 3169–3175, IEEE, 2008.

[10] Y. Wang and I. I. Hussein, "Bayesian-based decision-making for object search and classification," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1639–1647, 2010.

[11] T. Logenthiran, "Multi-agent system for control and management of distributed power systems," 2012.

[12] T. Arai, E. Pagello, L. E. Parker, *et al.*, "Advances in multi-robot systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655–661, 2002.

[13] J. S. Sandhu, A. M. Agogino, A. K. Agogino, *et al.*, "Wireless sensor networks for commercial lighting control: Decision making with multi-agent systems," in *AAAI Workshop on Sensor Networks*, vol. 10, pp. 131–140, Citeseer, 2004.

[14] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*, vol. 1. Addison-Wesley Reading, 1999.

[15] J. V. Rauff, "Multi-agent systems: An introduction to distributed artificial intelligence," *Mathematics and Computer Education*, vol. 39, no. 1, p. 81, 2005.

[16] M. De Weerdt, Y. Zhang, and T. Klos, "Distributed task allocation in social networks," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1–8, 2007.

[17] L.-H. Ren, Y.-S. Ding, Y.-Z. Shen, and X.-F. Zhang, "Multi-agent-based bio-network for systems biology: protein–protein interaction network as an example," *Amino Acids*, vol. 35, no. 3, pp. 565–572, 2008.

[18] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–8, IEEE, 2009.

[19] J. L. Adler and V. J. Blue, "A cooperative multi-agent transportation management and route guidance system," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5-6, pp. 433–454, 2002.

[20] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.

[21] C. Yadati, C. Witteveen, and Y. Zhang, "Coordinating agents-an analysis of coordination in supply-chain management tasks," in *International Conference on Agents and Artificial Intelligence*, vol. 2, pp. 218–223, SCITEPRESS, 2010.

[22] P. Baran, "On distributed communications networks," *IEEE Transactions on Communications Systems*, vol. 12, no. 1, pp. 1–9, 1964.

[23] D. V. Dimarogonas and K. H. Johansson, "Event-triggered control for multi-agent systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 7131–7136, IEEE, 2009.

[24] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3628–3633, IEEE, 2006.

[25] S.-i. Azuma, R. Yoshimura, and T. Sugie, "Broadcast control of multi-agent systems," *Automatica*, vol. 49, no. 8, pp. 2307–2316, 2013.

[26] D. W. Gage, "Command control for many-robot systems," tech. rep., Naval Command Control and Ocean Surveillance Center Rdt And E Div San Diego CA, 1992.

[27] M. Schwager, D. Rus, and J.-J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 371–383, 2011.

[28] J. O'rourke, *Art gallery theorems and algorithms*, vol. 57. Oxford New York, NY, USA, 1987.

[29] A. Mesbahi, F. Abbasi, and J. M. Velni, "A team-based deployment approach for heterogeneous mobile sensor networks," *Automatica*, vol. 106, pp. 327–338, 2019.

[30] Y. Stergiopoulos and A. Tzes, "Spatially distributed area coverage optimisation in mobile robotic networks with arbitrary convex anisotropic patterns," *Automatica*, vol. 49, no. 1, pp. 232–237, 2013.

[31] C. Zhai and Y. Hong, "Decentralized sweep coverage algorithm for multi-agent systems with workload uncertainties," *Automatica*, vol. 49, no. 7, pp. 2154–2159, 2013.

[32] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[33] S. Miah, B. Nguyen, A. Bourque, and D. Spinello, "Nonuniform coverage control with stochastic intermittent communication," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1981–1986, 2014.

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

[35] M. Todescato, A. Carron, R. Carli, G. Pillonetto, and L. Schenato, "Multi-robots gaussian estimation and coverage control: From client–server to peer-to-peer architectures," *Automatica*, vol. 80, pp. 284–294, 2017.

[36] L. Zuo, Y. Shi, and W. Yan, "Dynamic coverage control in a time-varying environment using bayesian prediction," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 354–362, 2017.

[37] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2542–2547, IEEE, 2005.

[38] X. Sun, C. G. Cassandras, and X. Meng, "Exploiting submodularity to quantify near-optimality in multi-agent coverage problems," *Automatica*, vol. 100, pp. 349–359, 2019.

[39] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[40] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.

[41] J. Hu and Z. Xu, "Brief paper-distributed cooperative control for deployment and task allocation of unmanned aerial vehicle networks," *IET Control Theory & Applications*, vol. 7, no. 11, pp. 1574–1582, 2013.

[42] F. Lekien and N. E. Leonard, "Nonuniform coverage and cartograms," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 351–372, 2009.

[43] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *2008 47th IEEE Conference on Decision and Control*, pp. 3947–3952, IEEE, 2008.

[44] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked

robots," in *2010 IEEE International Conference on Robotics and Automation*, pp. 4982–4989, IEEE, 2010.

[45] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks*. Princeton University Press, 2009.

[46] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[47] M. Schwager, B. J. Julian, and D. Rus, "Optimal coverage for multiple hovering robots with downward facing cameras," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3515–3522, IEEE, 2009.

[48] M. Schwager, J. McLurkin, J.-J. E. Slotine, and D. Rus, "From theory to practice: Distributed coverage control experiments with groups of robots," in *Experimental Robotics*, pp. 127–136, Springer, 2009.

[49] D. Inoue, Y. Ito, and H. Yoshida, "Optimal transport-based coverage control for swarm robot systems: Generalization of the voronoi tessellation-based method," in *2021 American Control Conference (ACC)*, pp. 3032–3037, IEEE, 2021.

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

[51] S. Izumi, S.-i. Azuma, and T. Sugie, "Coverage control inspired by bacterial chemotaxis," in *2014 IEEE 33rd International Symposium on Reliable Distributed Systems Workshops*, pp. 34–39, IEEE, 2014.

[52] D. Inoue, D. Murai, and H. Yoshida, "Stochastic self-organizing control for swarm robot systems," in *International Conference on Swarm Intelligence*, pp. 405–416, Springer, 2019.

[53] S. Huang, R. S. H. Teo, and W. L. Leong, "Review of coverage control of multi unmanned aerial vehicles," in *2017 11th Asian Control Conference (ASCC)*, pp. 228–232, IEEE, 2017.

[54] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S.-i. Azuma, "Pseudo-perturbation-based broadcast control of multi-agent systems," *Automatica*, vol. 113, p. 108769, 2020.

[55] B. Zhu, L. Xie, D. Han, X. Meng, and R. Teo, "A survey on recent progress in control of swarm systems," *Science China Information Sciences*, vol. 60, no. 7, pp. 1–24, 2017.

[56] Z. H. Ismail, N. Sariff, and E. Hurtado, "A survey and analysis of cooperative multi-agent robot systems: challenges and directions," in *Applications of Mobile Robots*, pp. 8–14, IntechOpen, 2018.

[57] I. Segall and A. Bruckstein, "On stochastic broadcast control of swarms," in *International Conference on Swarm Intelligence*, pp. 257–264, Springer, 2016.

[58] J. Ueda, L. Odhner, and H. H. Asada, "Broadcast feedback of stochastic cellular actuators inspired by biological muscle control," *The International Journal of Robotics Research*, vol. 26, no. 11-12, pp. 1251–1265, 2007.

[59] A. A. Julius, Á. Halász, M. S. Sakar, H. Rubin, V. Kumar, and G. J. Pappas, "Stochastic modeling and control of biological systems: the lactose regulation system of escherichia coli," *IEEE Transactions on Automatic Control*, vol. 53, no. Special Issue, pp. 51–65, 2008.

[60] K. Das and D. Ghose, "Positional consensus in multi-agent systems using a broadcast control mechanism," in *2009 American Control Conference*, pp. 5731–5736, IEEE, 2009.

[61] K. Das and D. Ghose, "Broadcast control mechanism for positional consensus in multiagent systems," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1807–1826, 2015.

[62] T. Bretl, "Control of many agents using few instructions.," in *Robotics: Science and Systems*, Citeseer, 2007.

[63] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[64] D. Bertsimas and J. Tsitsiklis, "Introduction to linear optimization, belmont, ma: Athena scientific, 1997," *Dr. Yeung was a member of the Board of Governors of the IEEE Information Theory Society from*, pp. 2011–12, 1999.

[65] D. G. Luenberger, *Introduction to linear and nonlinear programming*, vol. 28. Addison-wesley Reading, MA, 1973.

[66] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, p. 1226, 1995.

[67] S. Berman, A. Halász, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[68] A. R. Mesquita, J. P. Hespanha, and K. Åström, "Optimotaxis: A stochastic multi-agent optimization procedure with point measurements," in *International workshop on hybrid systems: Computation and control*, pp. 358–371, Springer, 2008.

[69] Y. Tanaka, S.-i. Azuma, and T. Sugie, "Broadcast control of multi-agent systems with quantized measurements," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 97, no. 3, pp. 830–839, 2014.

[70] M. M. Nor, Z. Ismail, and M. Ahmad, "Broadcast control of multi-agent systems with instability phenomenon," in *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, pp. 7–12, IEEE, 2016.

[71] S.-i. Azuma, I. Baba, and T. Sugie, "Broadcast control of markovian multi-agent systems," *SICE Journal of Control, Measurement, and System Integration*, vol. 9, no. 2, pp. 103–112, 2016.

[72] S.-i. Azuma, Y. Tanaka, and T. Sugie, "Multi-agent consensus under a communication–broadcast mixed environment," *International Journal of Control*, vol. 87, no. 6, pp. 1103–1116, 2014.

[73] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S.-I. Azuma, "Multi-vehicle coordination on merging roads based on pseudo-perturbation-based broadcast control," in *2018 Annual American Control Conference (ACC)*, pp. 4008–4013, IEEE, 2018.

[74] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S.-i. Azuma, "Coordination of connected vehicles on merging roads using pseudo-perturbation-based broadcast control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3496–3512, 2018.

[75] G. Kumar and M. V. Kothare, "Broadcast stochastic receding horizon control of multi-agent systems," *Automatica*, vol. 49, no. 12, pp. 3600–3606, 2013.

[76] G. Kumar and M. V. Kothare, "Trapping brownian ensemble optimally using broadcast stochastic receding horizon control," *Automatica*, vol. 50, no. 2, pp. 389–398, 2014.

[77] R. Kianfar, P. Falcone, and J. Fredriksson, "A distributed model predictive control approach to active steering control of string stable cooperative vehicle platoon," *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 750–755, 2013.

[78] F. Zhang, G. Tan, C. Yu, N. Ding, and C. Song, "Dynamic feedback power control for cooperative vehicle safety systems," *Wireless Personal Communications*, vol. 90, no. 1, pp. 51–74, 2016.

[79] M. A. S. Kamal, T. Hayakawa, and J.-i. Imura, "Development and evaluation of an adaptive traffic signal control scheme under a mixed-automated traffic scenario," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 590–602, 2019.

[80] M. A. S. Kamal, C. P. Tan, T. Hayakawa, S.-I. Azuma, and J.-I. Imura, "Control of vehicular traffic at an intersection using a cyber-physical multiagent framework," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6230–6240, 2021.

[81] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber–physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016.

[82] D. A. der Technikwissenschaften, *Cyber-Physical Systems: Driving force for innovation in mobility, health, energy and production*. Springer, 2011.

[83] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Integration of agent-based and cloud computing for the smart objects-oriented iot," in *Proceedings of the 2014 IEEE 18th International Conference on Computer Dupported Cooperative Work in Design (CSCWD)*, pp. 493–498, IEEE, 2014.

[84] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From iot system design to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 11, pp. 1939–1956, 2017.

[85] J. M. Bradley and E. M. Atkins, "Optimization and control of cyber-physical vehicle systems," *Sensors*, vol. 15, no. 9, pp. 23020–23049, 2015.

[86] S. Fortune, "Voronoi diagrams and delaunay triangulations," in *Computing in Euclidean Geometry*, pp. 193–233, World Scientific, 1992.

[87] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S.-i. Azuma, "Multi-vehicle coordination on merging roads based on pseudo-perturbation-based broadcast control," in *Proc. of 2018 American Control Conference*, 2018.

[88] C. Robinson, *Dynamical systems: stability, symbolic dynamics, and chaos*. CRC Press, 1998.

[89] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.

[90] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[91] M. Shanmugavel, A. Tsourdos, B. A. White, and R. Żbikowski, "Differential geometric path planning of multiple uavs," 2007.

[92] M. Shanmugavel, A. Tsourdos, B. White, and R. Zbikowski, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice*, 2010.

[93] M. Shanmugavel, A. Tsourdos, and B. A. White, "Collision avoidance and path planning of multiple UAVs using flyable paths in 3D," in *2010 15th International Conference on Methods and Models in Automation and Robotics*, pp. 218–222, IEEE, Aug 2010.

[94] M. Shanmugavel, A. Tsourdos, B. White, and R. Zbikowski, "3d dubins sets based coordinated path planning for swarm of uavs," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 6211, 2006.

[95] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[96] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 52–65, 2011.

[97] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[98] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate o(1/kˆ 2)," in *Dokl. akad. nauk Sssr*, vol. 269, pp. 543–547, 1983.

[99] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[100] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning*, pp. 1139–1147, PMLR, 2013.

[101] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[102] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Design Automation Conference*, pp. 731–736, IEEE, 2010.