


RESEARCH ARTICLE

Coverage control of mobile agents using multi-step broadcast control

Shalini Darmaraju¹ , Md Abdus Samad Kamal², Madhavan Shanmugavel³ and Chee Pin Tan^{1,*}

¹School of Engineering, Monash University Malaysia, Selangor, Malaysia, ²Graduate School of Science and Technology, Gunma University, Gunma, Japan, and ³Department of Mechatronics Engineering, SRM Institute of Science and Technology, Tamil Nadu, India

*Corresponding author. E-mail: tan.chee.pin@monash.edu

Received: 24 February 2021; **Revised:** 18 October 2021; **Accepted:** 21 January 2022;

First published online: 28 February 2022

Keywords: broadcast control, multi-agent system, area coverage control, wireless coverage

Abstract

This paper proposes a novel multi-step broadcast control (MBC) scheme to deploy a group of autonomous mobile agents for accomplishing coverage tasks in a bounded region. Traditional broadcast control (BC) schemes use a one-to-all communication framework to transmit a uniform signal to all agents, making it cost-effective compared with any all-to-all communication-based scheme for a multi-agent system. However, as BC schemes are based on a single-step view of the environment for decision-making, the environment's varying distribution density is not known immediately to the agents, resulting in suboptimal performance. To overcome this drawback, this paper proposes an MBC scheme, where agents use a predictive multi-step view and are able to detect the varying densities in the environment ahead of time. The local controller output is estimated using a weighted averaging technique which assigns a higher weight to immediate steps; this feature compensates for any decrease in prediction accuracy as the number of steps increases. We demonstrate the effectiveness of the proposed MBC scheme using a coverage task over a region with uneven population density. Compared to existing BC schemes, the proposed MBC scheme shows superior convergence characteristics in task accomplishment and deployment efficiency.

1. Introduction

Nature shows one of the best displays of group coordination through flocks of birds, schools of fish and herds of animals. The collective motion of a group of animals is not a planned script but instantaneous decisions and responses by individual members [1]. In artificial man-made settings, these members can be represented as intelligent mobile agents. The research interest in mobile agents [2, 3, 4, 5, 6] is growing rapidly, as collective groups of agents are able to perform tasks that individual agents cannot. Recent research in this area focused on the motion and control planning of cooperative mobile ground robots and also aerial systems such as quadrotors [7, 8, 9, 10], which may provide services such as coverage, search and rescue, delivery, tracking, and inspection [11, 12, 13, 14, 15].

Coverage control [16, 17, 18] is a benchmark problem and is the topic of interest in this paper. In general, the coverage control task aims to optimally deploy a group of mobile agents over an area of interest or environment. Control laws distribute agents to different parts of the coverage environment, usually partitioned into Voronoi cells. A well-known method for coverage problems is the gradient descent method [16], which proposes continuous and discrete-time algorithms that converge to the centroids of the Voronoi partition of the search domain. Similar approaches using Voronoi partitions for coverage control have been applied successfully [19, 20, 21]. Other methods

such as feedback control [22], receding horizon control [18] and game theory method [23] have also been used.

In all the above-mentioned methods, the global task is visible and known to all agents, and individual agents are aware of all other agents' locations either partially or fully. However, the drawback of these methods is that collecting and storing all these information requires high communication data volume. Besides, with the global task and progress made known to individual agents, attacks on any one agent can compromise the mission's security. The methods mentioned above modelled the mobile agents' cooperative behaviour using either deterministic control techniques [24, 25] or stochastic techniques [26, 27, 28]. In deterministic techniques, the output is predictable and is fully determined by the system input and initial conditions. However, in stochastic techniques, the outputs vary with inherent randomness in the system. The randomness in stochastic models could stem from its system dynamics, switching in network topology, issues in network communication such as time delays. Stochastic techniques are better representation of the natural world but are considerably more complicated.

The broadcast control (BC) scheme [29] and its variant pseudo-perturbation-based broadcast control (PBC) [30] are recently proposed stochastic optimisation methods that have been applied to multi-agent coordination tasks. The BC scheme aims to solve the problem of multi-agent systems using one-to-all communication protocols that replace standard agent-to-agent-based communication protocols. The goal is to complete the mission by observing group performance and sending the same signal to all agents indiscriminately. The agents do not require specific hardware nor do they need to expend energy for information transmission. Although the BC/PBC system utilises minimal hardware and reduced communication volume, the broadcast scheme is susceptible to broadcast-based challenges. An example is when the broadcasted signal reaches an agent, it can affect other agents and result in undesirable configurations. The PBC converges twice as fast and displays improved control performance compared to the BC scheme. These schemes are able to overcome the drawbacks of the earlier mentioned methods as they do not involve communication among agents. Furthermore, the global mission is unknown to individual agents in the BC schemes.

In this paper, we introduce a novel variant of the BC scheme termed *multi-step broadcast control (MBC)* for coverage control of mobile agents with non-uniform density distribution in the study environment. Unlike the previous BC schemes studied for uniform coverage control, this paper considers non-uniformity in the coverage area and compares the performance of our proposed scheme against the previous BC schemes.

The proposed MBC scheme is inspired from model predictive control (MPC) theory, where the control strategy is based on a prediction horizon in the future. To the best of our knowledge, this is the first work that employs a MBC of multi-agent systems in a detailed non-uniform coverage task study. The proposed MBC scheme takes multiple predictive steps in a horizon to predict future variables several steps ahead in an environment. An immediate step is given a higher weight, and as the number of steps increases, the respective weightage is decreased. More agents are deployed to dense sections (higher density) than sparsely populated sections (lower density) to provide equitable services to an environment with uneven density distribution. Using multi-step forward views, dense sections are made known to MBC agents earlier than existing BC schemes, which are limited to a one-step forward view (PBC uses multiple one-step views from different directions). Thus, the proposed MBC increases the stochastic gradient accuracy much faster and outperforms the existing BC schemes in both coverage task achievement and deployment efficiency.

This paper is organised as follows. In Section 2, the Voronoi-based coverage control problem is described and formulated. We review the existing BC schemes and our proposed MBC scheme in Section 3. Section 4 presents the numerical example to demonstrate the effectiveness of the MBC scheme in coverage tasks with an uneven density distribution. Lastly, we present our conclusions and directions for future research in Section 5.

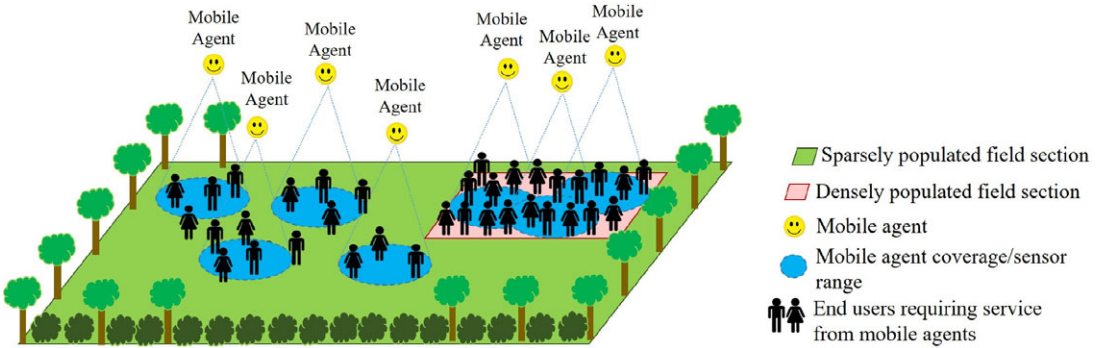


Figure 1. Mobile agents provide coverage service over a bounded area (field). The bounded area consists of densely and sparsely populated sections.

2. Coverage problem formulation

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. 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, that is, not all agents are placed at an equal distance from each other. For example, more agents need to be deployed to dense sections than to sparsely populated sections.

The objective function for the coverage task is [16]

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

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 $x_i = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$. 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:

$$J(x(t)) = \min_{x \in \mathbb{R}^{nN}} J(x) \tag{2}$$

The minimum of J is achieved when N agents are placed optimally in the space. The Voronoi tessellation method [31], which partitions the entire area into sub-areas, is used to compute and visualise the coverage achievement. Created by points (p_1, \dots, p_n) , the optimal partition of Q follows the Voronoi partition $V(P) = \{V_1, \dots, V_n\}$ that is given by:

$$V_i = \{q \in Q \mid \|q - x_i\| \leq \|q - x_j\|, \forall j \neq i\} \tag{3}$$

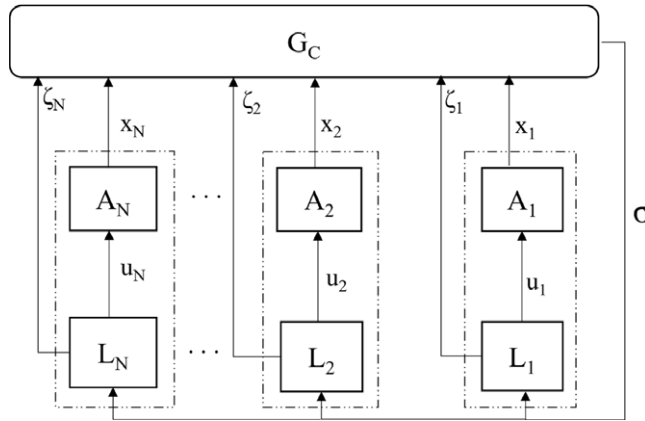


Figure 2. Framework of BC/PBC/MBC of a multi-agent system. The broadcast system consists of N agents A_i , local controllers L_i , and a global controller G_C . In the BC scheme, no local controller L_i has to send its state ζ_i to the global controller, G_C .

3. Proposed scheme : Multi-step Broadcast Control (MBC)

3.1. Review of existing BC schemes

In this section, the BC [29] and PBC [30] schemes are reviewed. The BC multi-agent scheme consists of a global controller, G_C , with N agents, each denoted as $A_i, i = 1, 2, \dots, N$, and the respective local controllers, L_i . The connectivity between the agents and the controllers is shown in Fig. 2.

In this paper, *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(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:

$$A_i : x_i(t + 1) = x_i(t) + u_i(t), \quad i = 1, 2, \dots, N \tag{4}$$

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:

$$L_i : \begin{cases} \zeta_i(t + 1) = \alpha(\zeta_i(t), \sigma_B(t), t) \\ u_i(t) = \beta(\zeta_i(t), \sigma_B(t), t) \end{cases} \tag{5}$$

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

$$\alpha(\zeta_i(t), \sigma_B(t), t) := [\Delta_i(t)^T, \sigma_B(t)]^T \tag{6}$$

$$\beta(\zeta_i(t), \sigma_B(t), t) := \begin{cases} c(t)\Delta_i(t), & \text{if } t \in \{0, 2, 4, \dots\}, \\ -c(t)\zeta_{i1}(t) - a(t) \left(\frac{\sigma_B(t) - \zeta_{i2}(t)}{c(t)} \right) \zeta_{i1}^{[-1]}(t) & \text{if } t \in \{1, 3, 5, \dots\} \end{cases} \tag{7}$$

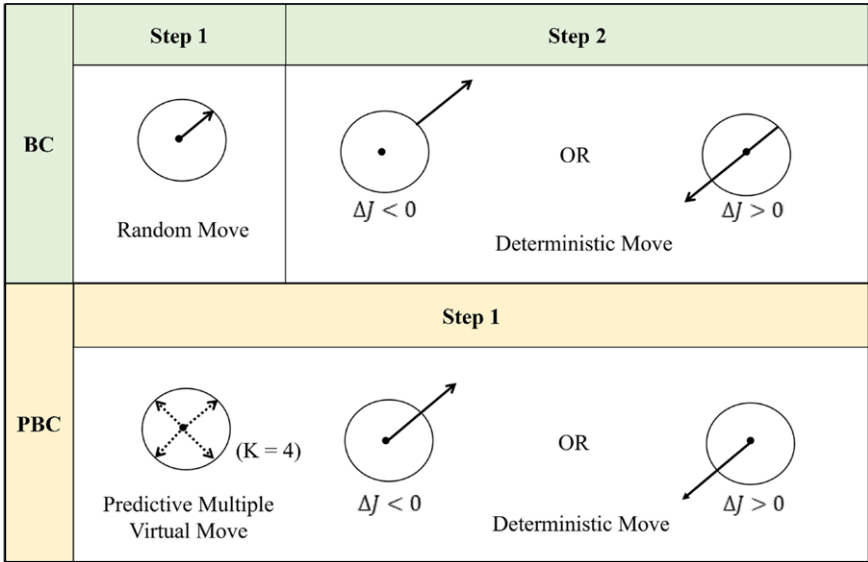


Figure 3. (a) Two-step agent movements in the BC framework. Depending on the feedback on the random move (ΔJ), an agent continues to move in the same or opposite direction in the next step. The process repeats until the convergence. (b) Single-step agent movement in the PBC framework.

where $\zeta_i(t) := [\zeta_{i1}(t)^T, \zeta_{i2}(t)^T]^T$, $\zeta_{i1} \in \{-1, 1\}^n$ with $\zeta_{i2} \in \mathbb{R}$. $\zeta_{i1}^{[-1]}$ denoting the element-wise inverse of ζ_{i1} , namely $\zeta_{i1}^{[-1]} := [1/\zeta_{i1,1}, \dots, 1/\zeta_{i1,n}]^T$. Also, $\zeta_{i1}(t) = \Delta_i(t - 1)$ and $\zeta_{i2}(t) = \sigma_B(t - 1)$. The symbol $\Delta_i(t) = [\Delta_{i,1}(t), \dots, \Delta_{i,n}(t)]^T \in \{-1, 1\}^n$ denotes the vector of random variable where each $\Delta_{i,j}(t)$ ($i \in \{1, \dots, N\}, j \in \{1, \dots, N\}, t \in \{0, 1, \dots\}$) 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:

$$G_C : \sigma_B(t) = J(x(t)) \in \mathbb{R} \tag{8}$$

where

$$x(t) := [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{nN} \tag{9}$$

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:

$$\lim_{t \rightarrow \infty} J(x(t)) = \min_{x \in \mathbb{R}^{nN}} J(x) \tag{10}$$

where $J : \mathbb{R}^{nN} \rightarrow \mathbb{R}$. The optimisation 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 $t \in 0, 2, 4, \dots$, agents take random moves, and at odd time steps $t \in 1, 3, 5, \dots$, they take deterministic moves. This is shown in Fig. 3(a).

In BC, the random movement of agents incurs movement cost and considerable time for convergence. Ito et al. [30] introduced the PBC. In the PBC scheme, each agent takes multiple virtual random moves (Fig. 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.

The state-space equation of agent i in PBC is given as:

$$\hat{x}_i^{(k)}(t + 1) := x_i(t) + \hat{u}_i^{(k)}(t) \text{ for } k = 1, 2, \dots, K \tag{11}$$

$$\hat{u}_i^{(k)}(t) := c(t)\Delta_i^{(k)}(t) \text{ for } k = 1, 2, \dots, K \tag{12}$$

where $\hat{x}_i^{(k)}(t + 1)$ and $\hat{u}_i^{(k)}(t)$ are the virtual input and the virtual predictive state of agents A_i at each step. 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 $\hat{x}^{(k)} := [\hat{x}_1^{(k)T}, \dots, \hat{x}_N^{(k)T}]^T \in \mathbb{R}^{nN}$ and $\Delta^{(k)}(t) := [\Delta_1^{(k)}(t)^T, \dots, \Delta_N^{(k)}(t)^T]^T$.

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

$$\sigma_P(t)F := \begin{bmatrix} J(\hat{x}^{(1)}(t + 1)) - J(x(t)) \\ J(\hat{x}^{(2)}(t + 1)) - J(x(t)) \\ \vdots \\ J(\hat{x}^{(K)}(t + 1)) - J(x(t)) \end{bmatrix} \in \mathbb{R}^K \tag{13}$$

where $\sigma_P^{(k)}(t) := J(\hat{x}^{(k)}(t + 1)) - J(x(t))$ and $\sigma_P(t) = [\sigma_P^{(1)}(t), \dots, \sigma_P^{(K)}(t)]^T$. In PBC, the local controller, L_i determines its state, $\zeta_i(t)$ and the control output, $u_i(t)$ as:

$$\zeta_i(t) := [\Delta_i^{(1)}(t)^T, \dots, \Delta_i^{(K)}(t)^T]^T \in \{-1, 1\}^{nK} \tag{14}$$

$$u_i(t) := -a(t) \frac{1}{K} \sum_{k=1}^K \frac{\sigma_P^{(k)}(t)}{c(t)} \Delta_i^{(k)[-1]}(t) \tag{15}$$

where $\Delta_i^{(k)[-1]}$ denotes the element-wise inverse of vector $\Delta_i^{(k)}$. Compared to (7), 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 real-time coordination of vehicles at a merging road [32].

In the PBC framework, each agent’s virtual random action is limited to a single step ahead of its current location. This does not allow for efficient prediction of unpredictable events that might exist in the task environment beyond a single step ahead, such as varying density distribution. To overcome this limitation, we propose a new scheme in the next section.

3.2. Main result: MBC scheme

This section presents the novel scheme proposed in this paper, the 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 4 depicts the concept of the action taken under the proposed scheme. The inspiration behind this concept comes from the theory of 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 $\Delta_i^{(k)}, k = 0, 1, 2, \dots, K - 1$, and using that the global controller, G_C , calculates the future virtual states $\hat{x}_i^{(k)}$ with $\hat{x}_i^{(0)} = x_i(t)$ as:

$$\hat{x}_i^{(k+1)}(t) := \hat{x}_i^{(k)}(t) + \hat{u}_i^{(k)}(t) \text{ for } k = 0, 1, 2, \dots, K - 1 \tag{16}$$

$$\hat{u}_i^{(k)}(t) := c(t)\Delta_i^{(k)}(t) \text{ for } k = 0, 1, 2, \dots, K - 1 \tag{17}$$

where $\hat{u}_i^{(k)}$ is the corresponding virtual input. We define the collective states of all agents at k as $\hat{x}^{(k)} := [\hat{x}_1^{(k)T}, \dots, \hat{x}_N^{(k)T}]^T \in \mathbb{R}^{nN}$, resulting from corresponding collective random variables

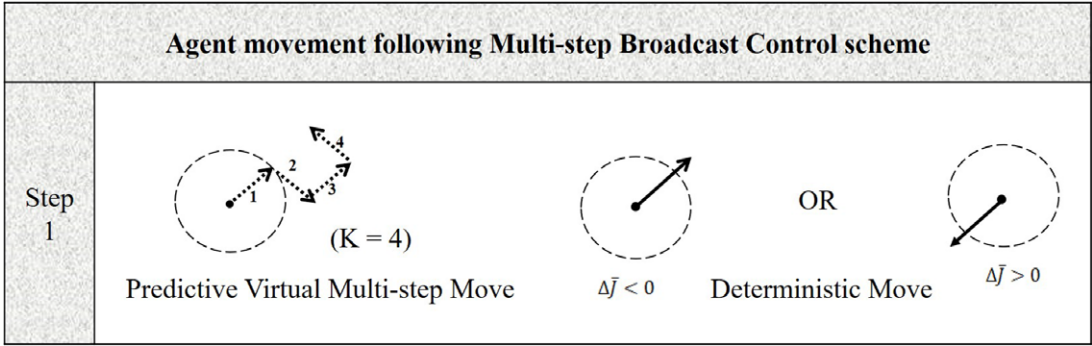


Figure 4. Agent movement taking multi-steps along a horizon in the MBC scheme.

$\Delta^{(k)}(t) := [\Delta_1^{(k)}(t)^T, \dots, \Delta_N^{(k)}(t)^T]^T$. The global controller computes the broadcast signal, σ_M , through a compounding method as follows:

$$\sigma_M(t) := \begin{bmatrix} J(\hat{x}^{(1)}(t)) - J(\hat{x}^{(0)}(t)) \\ J(\hat{x}^{(2)}(t)) - J(\hat{x}^{(1)}(t)) \\ \vdots \\ J(\hat{x}^{(K)}(t)) - J(\hat{x}^{(K-1)}(t)) \end{bmatrix} \in \mathbb{R}^K \tag{18}$$

where $\sigma_M^{(k)}(t) := J(\hat{x}^{(k+1)}(t)) - J(\hat{x}^{(k)}(t))$ and $\sigma_M(t) = [\sigma_M^{(0)}(t), \dots, \sigma_M^{(K-1)}(t)]^T$. The state, $\zeta_i(t)$, and the control output, $u_i(t)$, of the local controller, L_i , are given as:

$$\zeta_i(t) := [\Delta_i^{(0)}(t)^T, \dots, \Delta_i^{(K-1)}(t)^T]^T \in \{-1, 1\}^{nK} \tag{19}$$

$$u_i(t) := -a(t) \frac{1}{\sum_{k=0}^{K-1} \lambda^{(k)}} \sum_{k=0}^{K-1} \frac{\sigma_M^{(k)}(t) \lambda^{(k)}}{c(t)} \Delta_i^{(k)[-1]}(t), \tag{20}$$

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 (18)–(20) 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. [33] that the communication volume of the PBC scheme is about half of that of centralised 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 BC scheme as random physical action is eliminated, and agents’ state converges quicker.

3.3. Convergence analysis of proposed MBC

In this subsection, we present an analysis on the convergence of the proposed MBC scheme. The convergence proof presented herein is similar in nature to that of BC [29] and PBC [30] and is summarised with appropriate modifications.

Theorem 1. (Convergence of the MBC Scheme) Consider the multi-agent system shown in Fig. 2, an objective function $J(x)$, and the MBC scheme in (18), (19) and (20) with $K > 1$. If the following conditions (c1)–(c3) hold, then $x(t)$ in (9) converges to a (possibly sample-path-dependent) solution set to $\partial_x J(x) = 0$ with probability 1.

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

$$J(x(t)) := \rho(\|x\|)J_{obj}(x) + (1 - \rho(\|x\|))x^T x, \tag{21}$$

$$\rho(\|x\|) := \begin{cases} 1(\|x\| \leq l_1) \\ 0(\|x\| \geq l_2) \end{cases} \tag{22}$$

where $J(x)$ and $J_{obj}(x)$ in (1) is non-negative C^2 continuous on \mathbb{R}^{nN} , and there exists a solution to $\partial_x J(x) = 0$. $x^T x$ is a non-decreasing 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 $\partial_z Jz = 0$ (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 [29].

(c3) $\lim_{t \rightarrow \infty} a(t) = 0$, $\sum_{t=0}^{\infty} a(t) = \infty$, $\lim_{t \rightarrow \infty} c(t) = 0$, $\sum_{t=0}^{\infty} (a(t)/c(t))^2 < \infty$ 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 (9) has and reaches a local minimum value for the coverage problem.

Remark 3 Condition (c1) holds as J_{obj} in (1) 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 $\partial_x J(x) = 0$ [34]. Condition (c3) is applied to prevent gains $a(t)$ and $c(t)$ from reaching a very low value; in Section 4 (Numerical Simulations), the gains will be designed as in (33) to satisfy condition (c3).

Proof. The local controller output, $u_i(t)$ of BC as in (7) performs a two-stage state transition as follows:

$$x(t + 2) = x(t) - a(t) \frac{\sigma_B(t + 1) - \sigma_B(t)}{c(t)} \Delta^{[1]}(t) \tag{23}$$

where

$$\sigma_B(t) = J(x(t)) \tag{24}$$

$$\sigma_B(t + 1) = J(x(t) + c(t)\Delta(t)) \tag{25}$$

hold for $t \in \{0, 2, 4, \dots\}$. Then the system (23) becomes

$$x(t + 2) = x(t) - a(t)g(t, \Delta(t)), t \in \{0, 2, 4, \dots\} \tag{26}$$

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

$$g(t, \Delta) := \left(\frac{J(x(t) + c(t)\Delta) - J(x(t))}{c(t)} \right) \Delta^{[1]}(t) \tag{27}$$

If $J(x)$ is a C^2 continuous function with $c(t) \rightarrow 0$ (as implied by conditions (c1) and (c3)), then the expected value of $g(t, \Delta(t))$ (which is E in (28)) reduces to a stochastic approximate gradient of $J(x)$ based on simultaneous perturbation stochastic approximation (SPSA) [35] as follows:

$$E[g(t, \Delta(t))|x(t)] = \partial_x J(x(t)) + O(c(t)) \tag{28}$$

Similarly, following (26), the state transition of multi-steps in the MBC scheme is given by:

$$x(t + 1) = x(t) - a(t) \frac{1}{\sum_{k=0}^{K-1} \lambda^{(k)}} \sum_{k=0}^{K-1} g(t, \Delta^{(k)}(t)) \lambda^{(k)}, \forall t \tag{29}$$

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

$$e(t) := \frac{1}{\sum_{k=0}^{K-1} \lambda^{(k)}} \sum_{k=0}^{K-1} g(t, \Delta^{(k)}(t)) \lambda^{(k)} - E \left[\frac{1}{\sum_{k=0}^{K-1} \lambda^{(k)}} \sum_{k=0}^{K-1} g(t, \Delta^{(k)}(t)) \lambda^{(k)} \Big|_{x(t)} \right] \tag{30}$$

Substituting (28) and (30) into (29), then as $c(t) \rightarrow 0$, the transition of $x(t)$ under the MBC scheme is given by:

$$x(t + 1) = x(t) - a(t) \{ \partial_x J_x(x(t)) + e(t) + O(c(t)) \} \tag{31}$$

The system (31) is identical to the stochastic approximation algorithm in equation (A.1) (in Appendix A.1) of Azuma et al. [29]. Then, Lemma 2 of Azuma et al. [29] 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 [29] for (30), and thus if (c1) and (c3) hold, then it follows that (31) converges to a compact connected internally chain transitive invariant set $\dot{z}(\tau) = -\partial_z J(z(\tau))$; then, if (c2) holds, (31) converges to $\partial_x J(x) = 0$ and the proof is complete. \square

Remark 4 Convergence to a solution set to $\partial_x J(x) = 0$ shows that the proposed MBC scheme (18)–(20) is our solution to the coverage problem in (10).

Remark 5 MBC is similar to PBC in that it excludes random physical motion. However, the difference lies in that now the virtual random motion is designed to be multi-step. Furthermore, these predictive multi-steps are designed to be along the horizon and compounds from each other. The accuracy of gradient estimation can be increased using the MBC scheme in coverage task with non-uniform density distribution, for example, differing population density. The state transition of PBC is given by:

$$x(t + 1) = x(t) - a(t) \frac{1}{K} \sum_{k=1}^K g(t, \Delta^{(k)}(t)), \forall t \tag{32}$$

Comparing (29) and (32), it is clear that the MBC scheme utilises a weighted average technique, where else the PBC uses the arithmetic mean based on the central tendency.

4. Numerical simulations

The number of agents is set to $N = 25$ in the two-dimensional state space, that is $n = 2$. The size of the coverage workspace is set to 200 by 200 square units. Initially, all agents are uniformly distributed within x -coordinate (80:120) and y -coordinate (0:40). The terminal simulation time is set as 5000 iterations and the number of multi-steps used is $K = 10$. The controller gains a and c are set as following:

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

$$c(t)|_{MBC} = \frac{c_0}{\left(\frac{t}{2} + 1\right)^{c_p}}, \tag{33}$$

for $t \in \{0, 1, 2, \dots\}$ 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 increased number of agents, from 15 to 25 agents.

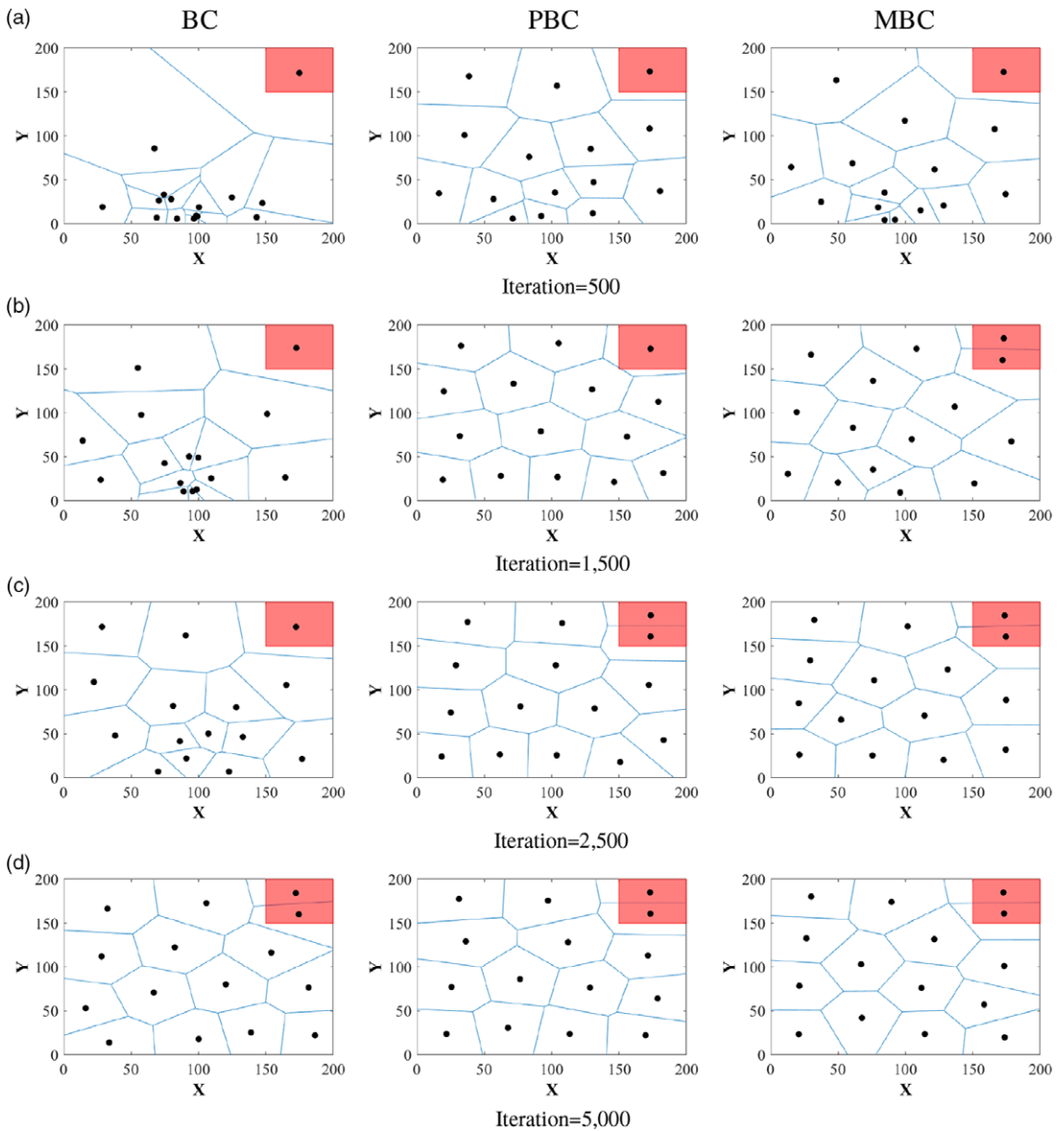


Figure 5. The coverage task’s achievement with a single-dense section is shown by agents’ positions in the Voronoi diagram at various iterations. Comparison of three different schemes: BC (left), PBC (middle), and proposed MBC (right).

4.1. 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, respectively, for all scenarios. Figure 5 shows the corresponding agent distributions during specific intervals of iteration of 500, 1500, 2500 and 5000 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 increases, 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

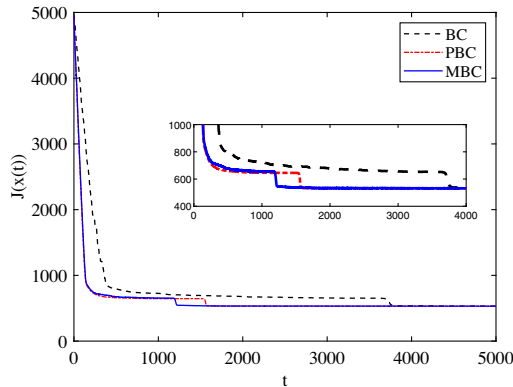


Figure 6. Evolution of objective function for BC, PBC and MBC for coverage task with a dense section.

dense shaded section holds two agents by iteration 5000 for all the three schemes. However, it is clear that MBC achieves this first by iteration 1500, followed by PBC by 2500 and lastly by BC by iteration 5000. These show that the schemes can achieve the same allocated number of agents in the dense section, making them comparable for convergence study.

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

4.2. 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 7 shows the comparison between PBC agent distribution on the left column and MBC on the right column for iteration 1000, 2000 and 4000. At iteration 1000, both PBC and MBC have equally acquired a single agent in the right dense section. By iteration 2000, 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 4000, PBC has managed to catch up and converge similarly to MBC.

The objective function in Fig. 8 shows these results clearly where MBC has converged faster by iteration 1500 and PBC at about iteration 3000. This shows the MBC outperforms PBC in a coverage task with double-dense sections.

4.3. Coverage for an increased number of agents

The two scenarios discussed above for single- and double-dense sections utilises 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 5 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. 9. The distribution is captured at iteration 1500, 2500

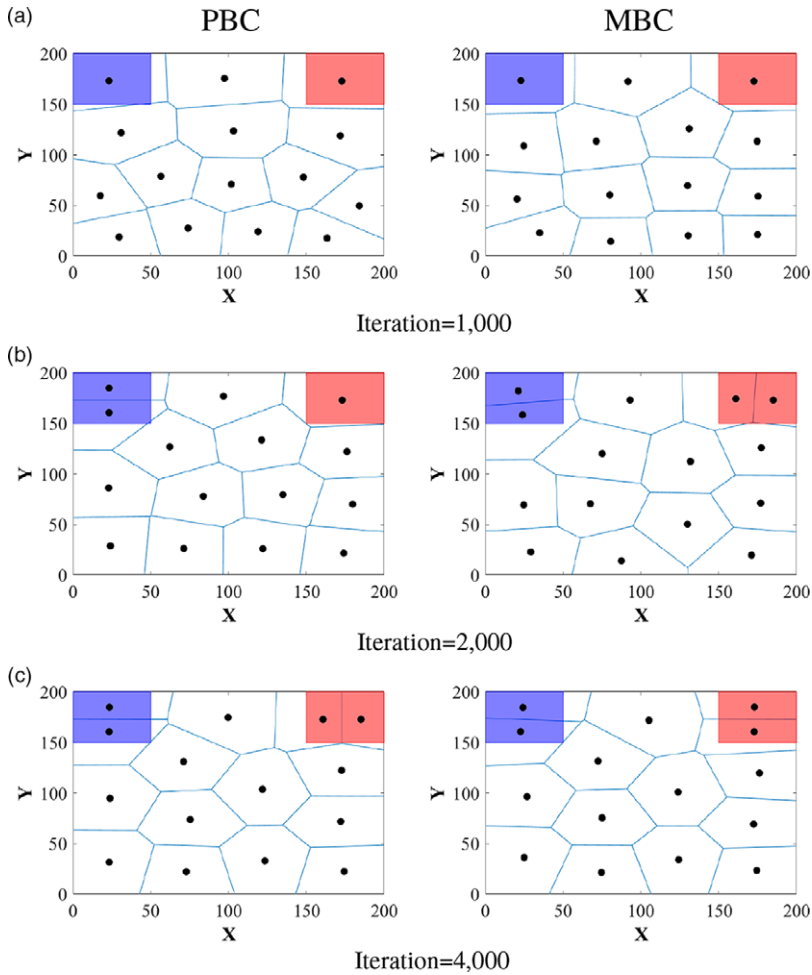


Figure 7. The coverage task's achievement with double-dense sections is shown by agents' positions in the Voronoi diagram at various iterations. Comparison of two different schemes: PBC (left) and proposed MBC (right).

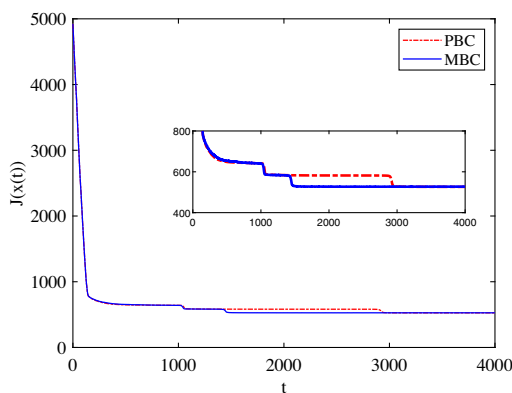


Figure 8. Evolution of objective function for PBC and MBC for coverage task with double-dense sections.

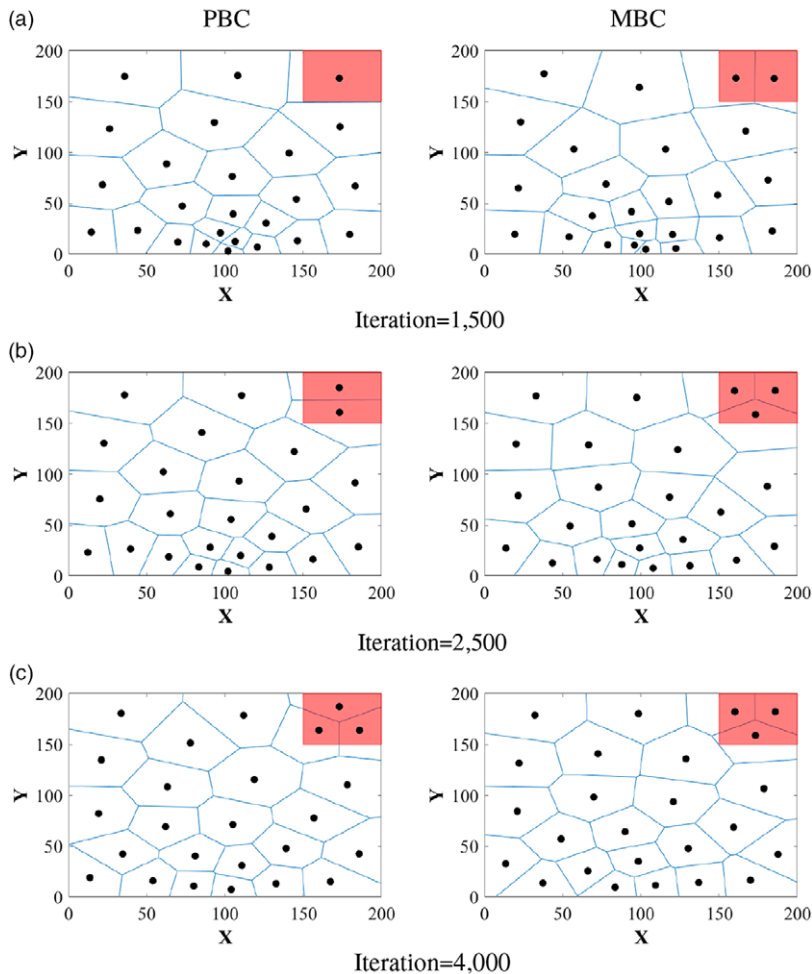


Figure 9. The achievement of 25 agents in a coverage task with a single-dense section is shown by agents' positions in the Voronoi diagram at various iterations. Comparison of two different schemes: PBC (left) and proposed MBC (right).

and 4000. 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 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 1500, MBC has two agents allocated in the dense section compared with PBC, which has 1. At iteration 2500, the number of agents for MBC within the dense section has increased to 3, where else for PBC, it has increased to 2. Finally, by iteration 4000, both methods show similar distribution. This is presented in Fig. 10 where the MBC scheme displays quicker convergence performance than PBC.

In addition, the initial BC scheme proposed by Azuma et al. [29] 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.

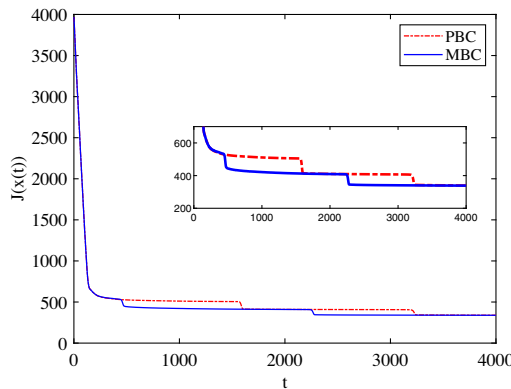


Figure 10. Evolution of objective function for PBC and MBC for coverage task with a single-dense section using 25 agents.

5. Conclusions and future work

In this paper, the MBC scheme has been proposed to overcome the limited one-step forward view of the BC schemes by introducing multiple predictive random steps along the horizon. The proposed scheme has been evaluated against the existing BC schemes in a coverage task over a region with uneven population density with mobile agents. Numerical simulations prove that the MBC scheme outperforms the BC and PBC scheme in task accomplishment and deployment efficiency. MBC converges the quickest as the gradient is computed from multiple steps along the horizon where the dense sections emerge to the agents earlier compared to both BC and PBC, which only uses a one-step forward view. As the MBC scheme utilises broadcast-based communication and takes a different outlook from the recent algorithms utilising agent-to-agent communication protocol [36, 37, 38, 39, 40], the comparison study is deemed appropriate only with the previous BC schemes as proven. In the future, we intend to extend the MBC scheme to some practical applications of multi-agent systems.

Conflicts of interest. None.

Financial support. This research is partly supported by the Fundamental Research Grant of Ministry of Higher Education, Malaysia, FRGS/1/2016/TK04/MUSM/02/1 and the Japan Society for the Promotion of Science (JSPS) Grant-in-Aids for Scientific Research (C) 20K04531.

Ethical considerations. None.

Authors' contributions. S. D., M. K. and C. T. designed the study. S. D. performed the numerical simulations, data analysis and wrote the manuscript with input and feedback from C. T., M. K. and M. S. All authors reviewed the results and approved the final version of the manuscript.

References

- [1] F. L. Lewis, H. Zhang, K. Hengster-Movric and A. Das, *Cooperative Control of Multi-Agent Systems, Communications and Control Engineering* (Springer, London, 2014).
- [2] Y. Cao, W. Yu, W. Ren and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Inf.* **9**(1), 427–438 (2013).
- [3] R. Haghighi and C. Cheah, "Multi-group coordination control for robot swarms," *Automatica* **48**(10), 2526–2534, (2012).
- [4] M. Rubenstein, A. Cornejo and R. Nagpal, "Robotics. Programmable self-assembly in a thousand-robot swarm," *Science (New York, NY)* **345**(6198), 795–799 (2014).

- [5] J. Ota, "Multi-agent robot systems as distributed autonomous systems," *Adv. Eng. Inf.* **20**(1), 59–70 (2006).
- [6] Y. Tan and Z.-y. Zheng, "Research advance in swarm robotics," *Defence Technol.* **9**(1), 18–39 (2013).
- [7] J. Raj, K. Raghunwaiya, B. Sharma and J. Vanualailai, "Motion control of a flock of 1-trailer robots with swarm avoidance," *Robotica* **39**(11), 1926–1951 (2021).
- [8] J. Raj, K. S. Raghunwaiya and J. Vanualailai, "Novel Iyapunov-based autonomous controllers for quadrotors," *IEEE Access* **8**, 47393–47406 (2020).
- [9] B. Pradhan, A. Nandi, N. B. Hui, D. S. Roy and J. J. Rodrigues, "A novel hybrid neural network-based multirobot path planning with motion coordination," *IEEE Trans. Veh. Technol.* **69**(2), 1319–1327 (2019).
- [10] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Rob.* **34**(4), 837–855 (2018).
- [11] J. Xie and C.-C. Liu, "Multi-agent systems and their applications," *J. Int. Council Electr. Eng.* **7**(1), 188–197 (2017).
- [12] A. Trotta, M. D. Felice, F. Montori, K. R. Chowdhury and L. Bononi, "Joint coverage, connectivity, and charging strategies for distributed UAV networks," *IEEE Trans. Rob.* **34**(4), 883–900 (2018).
- [13] Y. Rizk, M. Awad and E. W. Tunstel, "Decision making in multi agent systems: A survey," *IEEE Trans. Cognit. Dev. Syst.* **10**(3), 514–529 (2018).
- [14] M. A. Ali and M. Mailah, "Path planning and control of mobile robot in road environments using sensor fusion and active force control," *IEEE Trans. Veh. Technol.* **68**(3), 2176–2195 (2019).
- [15] J. P. Queraltá, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access* **8**, 191617–191643 (2020).
- [16] J. Cortes, S. Martinez, T. Karatas and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Rob. Autom.* **20**(2), 243–255 (2004).
- [17] S. Huang, R. S. H. Teo and W. L. Leong, "Review of Coverage Control of Multi Unmanned Aerial Vehicles," *2017 11th Asian Control Conference (ASCC)* (2017) pp. 228–232.
- [18] F. Mohseni, A. Doustmohammadi and M. B. Menhaj, "Distributed receding horizon coverage control for multiple mobile robots," *IEEE Syst. J.* **10**(1), 198–207 (2016).
- [19] M. Schwager, D. Rus and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Rob. Res.* **28**(3), 357–375 (2009).
- [20] A. Kwok and S. Martinez, "Unicycle coverage control via hybrid modeling," *IEEE Trans. Autom. Control* **55**(2), 528–532 (2010).
- [21] M. Schwager, D. Rus and J.-J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *Int. J. Rob. Res.* **30**(3), 371–383 (2011).
- [22] I. I. Hussein and D. M. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Control Syst. Technol.* **15**(4), 642–657 (2007).
- [23] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang and Y. Xu, "Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework," *China Commun.* **15**(10), 194–209 (2018).
- [24] O. Arslan, D. P. Guralnik and D. E. Koditschek, "Coordinated robot navigation by hierarchical clustering," *IEEE Trans. Rob.* **32**(2), 352–371 (2016).
- [25] D. V. Dimarogonas, E. Frazzoli and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control* **57**(5), 1291–1297 (2011).
- [26] S. Bandyopadhyay, S.-J. Chung and F. Y. Hadaegh, "Probabilistic and distributed control of a large-scale swarm of autonomous agents," *IEEE Trans. Rob.* **33**(5), 1103–1123 (2017).
- [27] H. Li, C. Feng, H. Ehrhard, Y. Shen, B. Cobos, F. Zhang, K. Elamvazhuthi, S. Berman, M. Haber-land and A. L. Bertozzi, "Decentralized Stochastic Control of Robotic Swarm Density: Theory, Simulation, and Experiment," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 4341–4347.
- [28] K. Elamvazhuthi and S. Berman, "Optimal Control of Stochastic Coverage Strategies for Robotic Swarms," *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015) pp. 1822–1829.
- [29] S.-i. Azuma, R. Yoshimura and T. Sugie, "Broadcast control of multi-agent systems," *Automatica* **49**(8), 2307–2316 (2013).
- [30] Y. Ito, M. A. S. Kamal, T. Yoshimura and S.-i. Azuma, "Pseudo-perturbation-based broadcast control of multi-agent systems," *Automatica* **113**, 108769 (2020).
- [31] S. Fortune, "Voronoi Diagrams and Delaunay Triangulations," *In: Computing in Euclidean geometry* (World Scientific, 1992) pp. 193–233.
- [32] 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 Trans. Intell. Transp. Syst.* **20**(9), 3496–3512 (2018).
- [33] 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," *Proceedings of 2018 American Control Conference* (2018).
- [34] C. Robinson, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos* (CRC Press, Boca Raton, FL, 1998).
- [35] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control* **37**(3), 332–341 (1992).
- [36] K. Li, C. Yuan, J. Wang and X. Dong, "Four-direction search scheme of path planning for mobile agents," *Robotica* **38**(3), 531–540 (2020).
- [37] L. Chen, H. G. de Marina and M. Cao, "Maneuvering formations of mobile agents using designed mismatched angles," *IEEE Trans. Autom. Control* (2021) Early access, 17 March 2021. doi: [10.1109/TAC.2021.3066388](https://doi.org/10.1109/TAC.2021.3066388).

- [38] J. Flores-Resendiz and E. Aranda-Bricaire, “A general solution to the formation control problem without collisions for first-order multi-agent systems,” *Robotica* **38**(6), 1123–1137 (2020).
- [39] B. M. Yousuf, A. S. Khan and A. Noor, “Multi-agent tracking of non-holonomic mobile robots via non-singular terminal sliding mode control,” *Robotica* **38**(11), 1984–2000 (2020).
- [40] M. S. Güzel, V. B. Ajabshir, P. Nattharith, E. C. Gezer and S. Can, “A novel framework for multi-agent systems using a decentralized strategy,” *Robotica* **37**(4), 691–707 (2019).