Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
July-August 2020: 1–12
© The Author(s) 2020
DOI: 10.1177/1729881420945958
journals.sagepub.com/home/arx
SAGE

Broadcast control of multi-robot systems with norm-limited update vector

Mohamad Hafizulazwan Mohamad Nor[1], Zool Hilmi Ismail[1] and Mohd Ashraf Ahmad[2]

Abstract

This article addresses a problem in standard broadcast control framework which leads to an unstable solution in a certain motion-coordination task. First, the unstable phenomenon in a certain motion-coordination task is illustrated using standard broadcast control framework. This issue calls for modification to the standard broadcast control framework by limiting the norm of the update vector of robots' positions into a constant value. Then, we demonstrate that the modified broadcast controller achieves the convergence with the probability of 1. Finally, we illustrate in numerical simulations that the modified broadcast controller can effectively solve the instability issue and also may improve the convergence time as compared to the standard broadcast controller.

Keywords

Multi-robot systems, broadcast control, instability, simultaneous perturbation stochastic approximation (SPSA), norm-limited update vector

Date received: 14 September 2019; accepted: 6 July 2020

Topic Area: Mobile Robots and Multi-Robot Systems
Topic Editor: Nak-Young Chong
Associate Editor: Changjoo Nam

Introduction

Nowadays, multi-robot systems are of a great interest among researchers, compared to individual systems. This is because multi-robot systems can effectively carry out more tasks. Previously, many researchers were focusing on robot-to-robot communication framework to solve various multi-robot coordination problems. Instead of using robot-to-robot communication framework, some researchers explored a new idea to solve multi-robot coordination problems by considering one-to-all communication framework or better known as broadcast control framework. The example is illustrated in Figure 1. This one-to-all communication framework means that the same signal is sent indiscriminatingly to all robots. The consideration of this broadcast framework has been motivated by its advantages, such as practical motion-coordination task for large-scale multi-robot systems such as swarm robot that does not need robots' identification, and consuming minimal energy during execution task execution. Readers may refer to Ismail and Sariff[1] for more examples of multi-robot applications.

[1] Centre for Artificial Intelligence and Robotics, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
[2] Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Pekan, Pahang, Malaysia

Corresponding author:
Zool Hilmi Ismail, Centre for Artificial Intelligence and Robotics, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia.
Email: zool@utm.my

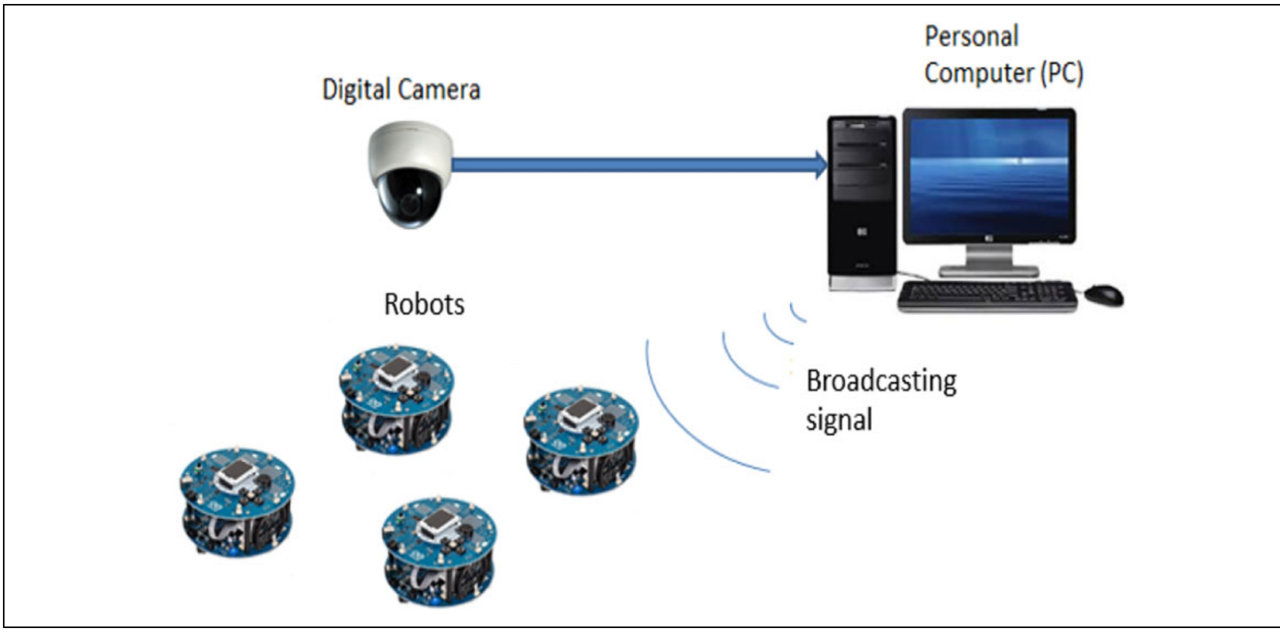

Figure 1. One-to-all robot communication framework for multi-robot systems.

The idea of broadcast control framework was originally inspired by Ueda et al.,[2] where such framework had been used to control a bio-inspired actuator system of many cellular units. A number of results regarding the broadcast control frameworks have been obtained.[3–5] Aside from biological system, the idea of broadcast control was also implemented in controlling a group of multi-agent systems[6–14] and recently, to control connected and automated vehicles at merging highway,[15,16] unmanned aerial vehicles,[17] as well as radar surveillance system.[18] In particular, a standard broadcast control framework was synthesized in Azuma and Sugie[8] to solve certain motion-coordination task for multi-agent systems, such as coverage problem. This work was supported in detailed proof by Azuma et al.[9] The interesting problem encountered in the standard broadcast control framework is in determining methodology to perform any motion-coordination task by sending the same signal to each agent indiscriminatingly. Here, they came out with a novel solution by designing global and local controllers as a feedback system. These global and local controllers' functions were, respectively, to generate commands to the agents by calculating the cost function and move the agents randomly and deterministically. Several solutions to the problem in the standard broadcast control framework were obtained. In their study, Tanaka et al.[12] focused on designing time-varying gain embedded in the local controllers for fast convergence. This work will also enable us to easily tune the gain. As a digital camera is commonly used as global controller's sensor, Tanaka et al.[13] had modified the standard broadcast control framework to cancel out the quantize effect caused by digital camera. Besides that, Tanaka et al.[14] had proposed to combine the standard broadcast framework with one-to-one agent communication known as mix environment. This combination aims to overcome the limited communication range and to reduce convergence time. However, their results were validated for coverage[8,13] and consensus tasks only,[12–14] and it was unsure whether the standard broadcast control framework can work for other motion-coordination tasks. In this study, we have simulated the standard broadcast control framework with some other motion-coordination tasks and identified that it only works for certain multi-robot coordination tasks. This is because in some coordination tasks, the broadcast control framework often encounters instability phenomenon. In this article, the instability phenomenon means that the robots are moving away from their preassigned target positions and never converged to the target positions due to unacceptable broadcasting input signal. For example, a group of robots (each employing the standard broadcast controller) are tasked to go to some target positions which are highly contaminated. Due to the instability phenomenon, these robots will move away from the target positions and probably will go to the place where the condition is unsuitable for the robots and might cause a serious damage to them. From this example, it is obvious that the instability phenomenon is important to be addressed and solved.

To illustrate the cause of the instability issue, consider a standard broadcast control system as shown in Figure 2, which comprises robots, local controllers, and a global controller. In the standard broadcast control framework,[8,9] the position of the robot is maneuvered by the local controller that connotes the random and the deterministic input, alternately. In particular, for iteration $k = 1, 2, ...$, the input is a random vector when k is even. When k is odd, it is a vector comprising prespecified parameters, as shown in Figure 3.

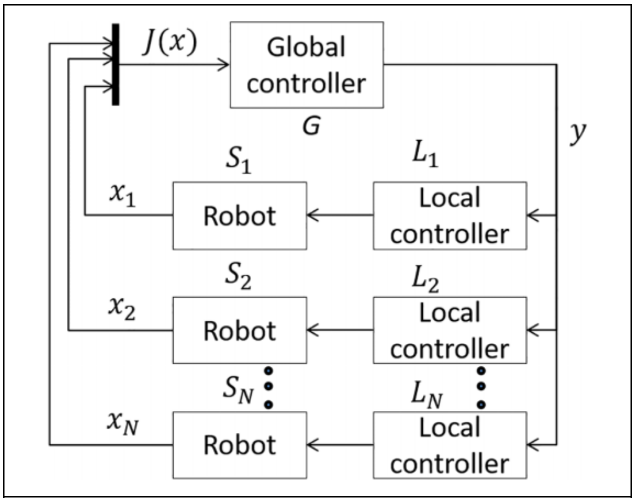

Figure 2. Standard broadcast control framework.

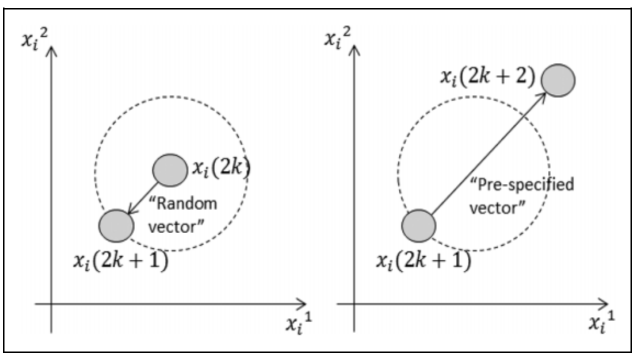

Figure 3. Motion of robots by standard broadcast control framework.

Note that the random vector is generated by the local controller itself, while the prespecified vector is uniquely determined from the output sequence of the global controller, corresponding to the assigned motion-coordination task. This way the robot is expected to achieve the given motion-coordination task as $k \to \infty$. However, in some coordination tasks, that is, role-assignment task (the coordination task is represented as a cost function and we denote it as $J(x)$ in this article), there is a possibility that the deterministic input becomes too large to execute and numerically diverges as $k \to \infty$. As a result, the local controller steers the robot in an unstable manner and cannot achieve the given motion-coordination task. The detailed numerical example of this instability problem will be shown in the second section.

In this article, we solved the instability problem by modifying the standard broadcast control framework which consisted of global and local controllers. In our modified broadcast control framework, a new local controller is derived so that it can limit the amount of the deterministic input in updating the robots' position. Note that the derivation is based on simultaneous perturbation stochastic approximation (SPSA) with norm-limited update vector algorithm adopted from Tanaka et al.[19] The algorithm was adopted because of the practicality in solving various model-free optimization problems.[19–21] From this fact, we believe that the same idea can be applied to solve the instability problem in the standard broadcast control framework. On the other hand, we proved that the modified broadcast controller asymptotically had achieved the task with a probability of 1. Besides that, the numerical simulation also showed that the modified broadcast control can effectively handle the given task. In addition, it is expected that our proposed broadcast control framework may also reduce the convergence time in performing the motion-coordination tasks.

Our contributions are summarized as follows:

(1) Modified an existing broadcast control framework (referred as standard broadcast control framework) based on SPSA with norm-limited update vector algorithm to overcome the instability problem of the existing broadcast control framework in solving some of the motion-coordination tasks (e.g. role-assignment tasks). By solving the instability problem, we believe that a large class of motion-coordination tasks can be catered by the broadcast control framework such as containment,[22] flocking,[23] and formation[24] control.
(2) Proved that the modified broadcast controller can achieve the motion-coordination task with a probability of 1.
(3) Generated simulation results showing our modified broadcast controller also may have faster convergence time as compared to the standard broadcast controller while being competitive with another existing broadcast control framework.

Notation: Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$, and $\mathbb{Z}_{0+}$ be the set of real numbers, set of positive real numbers, set of integers, and set of nonnegative integers, respectively. For the vector x, $||x||$ is defined as the Euclidean norm. When other kind of norm is used, then the norm is denoted as $||x||_p$ for the p-norm. Next, we use x^{-1} to denote a non-zero value or elementwise inverse of the element. For the differentiable function $J : \mathbb{R}^n \to \mathbb{R}$, that is, $\left[\partial J^{(j)}(x)/\partial x_1^{(j)}\right.$ $\left.\partial J^{(j)}(x)/\partial x_2^{(j)} \ldots \partial J^{(j)}(x)/\partial x_N^{(j)}\right] \in \mathbb{R}^n$, it is expressed by $\nabla J^{(j)}(x)$ for $j = 1, 2, ..., N$ and $\dot{z}(k)$ is denoted as the differential equation of $\mathrm{d}z(k)/\mathrm{d}k$. On the other hand, we denote $\gamma : \mathbb{R}^n \to \mathbb{R}^n$ as a saturation function, that is, $\gamma(\theta) := [\mathrm{sign}(\theta_1)\min(|\theta_1|, e)\ \ \mathrm{sign}(\theta_2)\min(|\theta_2|, e) \ldots \mathrm{sign}(\theta_n)\min(|\theta_n|, e)]^{\mathrm{T}}$ where $\mathrm{sign}(\theta_n)$ is the signum function of the element $\theta_i \in \mathbb{R}$. Here, θ_i is the ith element of $\theta \in \mathbb{R}^n$ and $\min(|\theta_n|, e)$ is the minimum value between $|\theta_n|$ and positive integer $e \in \mathbb{R}_+$. The Kronecker product of matrix A and B is denoted as $A \otimes B$. Finally, we denote $\Pr(a)$ and $\mathrm{E}(b)$ as the probability and expectation of event a and random variable b.

Broadcast control framework

This section briefly explains the standard broadcast control framework. Then, an example of unstable solution using the standard broadcast control is illustrated.

Standard broadcast control framework

Based on standard broadcast control system P in Figure 2, the state of the robot (e.g. discrete time model of an omnidirectional mobile robot) $S_i(i = 1, 2, \ldots, N)$ is given by

$$S_i : x_i(k+1) = x_i(k) + u_i(k) \tag{1}$$

where $k \in \mathbb{Z}_{0+}$ is the iteration, $x_i(k) \in \mathbb{R}^n$ is the n-dimensional space position of the robot, and $u_i(k) \in \mathbb{R}^n$ is the input of the robot. The collective position of the robots $x(k) \in \mathbb{R}^{nN}$ and the initial position $x(0) \in \mathbb{R}^{nN}$ are denoted as $x(k) := \left[x_1(k)^{\mathrm{T}} \quad x_2(k)^{\mathrm{T}} \quad \ldots \quad x_N(k)^{\mathrm{T}}\right]^{\mathrm{T}}$ and $x(0) := \left[x_1(0)^{\mathrm{T}} \quad x_2(0)^{\mathrm{T}} \quad \ldots \quad x_N(0)^{\mathrm{T}}\right]^{\mathrm{T}}$, respectively.

The local controllers $L_i(1, 2, \ldots, N)$ embedded in each of the following robots are described as

$$L_i : \begin{cases} \xi_i(k+1) = \phi(k, \xi_i(k), y(k)) \\ u_i(k) = \psi(k, \xi_i(k), y(k)) \end{cases} \tag{2}$$

where $\xi_i(i = 1, 2, \ldots, N) \in \mathbb{R}^{2n+1}$ are the temporary memorizing variables, $y(k) \in \mathbb{R}$ and $u_i(k) \in \mathbb{R}^n$ are, respectively, the input and output of the local controller, and $\phi : \mathbb{Z}_{0+} \times \mathbb{R}^{2n+1} \times \mathbb{R} \to \mathbb{R}^{2n+1}$ and $\psi : \mathbb{Z}_{0+} \times \mathbb{R}^{2n+1} \times \mathbb{R} \to \mathbb{R}^n$ are functions given by

$$\phi(k, \xi_i(k), y(k)) := \begin{cases} \begin{bmatrix} \beta_i(k) \\ c(k)\beta_i(k) \\ y(k) \end{bmatrix} & \text{if } k \text{ is even} \\ \xi_i(k) & \text{if } k \text{ is odd} \end{cases} \tag{3}$$

$$\psi(k, \xi_i(k), y(k)) := \begin{cases} c(k)\beta_i(k) & \text{if } k \text{ is even} \\ -\xi_{i2}(k) - a(k)\left(\dfrac{\Delta J}{c(k)}\xi_{i1}^{(-1)}(k)\right) & \text{if } k \text{ is odd} \end{cases} \tag{4}$$

where $\beta_i(k) \in \mathbb{R}^n$ is a Bernoulli distributed random vector, $c(k)$ and $a(k)$ are gains sequence, the components of $\xi_i(k)$ are $\xi_{i1}(k) \in \mathbb{R}^n$, $\xi_{i2}(k) \in \mathbb{R}^n$, $\xi_{i3}(k) \in \mathbb{R}$, that is

$$\xi_i(k) = \begin{bmatrix} \xi_{i1}(k) \\ \xi_{i2}(k) \\ \xi_{i3}(k) \end{bmatrix}$$

and $\Delta J = y(k) - \xi_{i3}(k) = J(x(k) + c(k)\beta_i(k)) - J(x(k))$.

The global controller G is given by

$$G : y(k) = f(J(x(k))) \tag{5}$$

where $J(x(k)) \in \mathbb{R}_{0+}$ and $y(k) \in \mathbb{R}$ are the input and output of the global controller, respectively, and $J : \mathbb{R}^{nN} \to 0 \cup \mathbb{R}_+$ and $f : \mathbb{R}_{0+} \to \mathbb{R}$ are functions. Note that the function f is introduced so that we have sufficient flexibility in designing the broadcast controller. On the other hand, function J is an objective function that corresponds to the given motion-coordination task. Then, we can indicate that this standard broadcast control achieves the given motion-coordination task if

$$\lim_{k\to\infty} J(x(k)) = \min_{x\in\mathbb{R}^{nN}} J(x)$$

for every initial positions of the robots $x(0) \in \mathbb{R}^{nN}$.

Unstable solution using standard broadcast control framework

In demonstrating the instable phenomenon in the standard broadcast control framework, we chose role-assignment task[9,25] given by

$$J(x) = \sum_{i=1}^{N}\prod_{j=1}^{N} \| x_i - p_j \| + \lambda\left(\prod_{i=1}^{N-1}\prod_{j=i+1}^{N} \| x_i - p_j \|\right)^{-1} \tag{6}$$

where $p_j(j = 1, 2, \ldots, N) \in \mathbb{R}^n$ is the target position and $\lambda \in \mathbb{R}$ is an approximate accuracy. Note that the overlapping function at the right-hand side of equation (6), that is, $\lambda\left(\prod_{i=1}^{N-1}\prod_{j=i+1}^{N} \| x_i - p_j \|\right)^{-1}$, is added so that the robots do not converge to the same place. The parameters of the standard broadcast controller (equations (1) to (5)) are tabulated in Table 1. Note that the design of gains $a(k)$ and $c(k)$ in Table 1 follows the similar way as in Azuma et al.[9] and the detailed explanation on how we choose the

Table 1. Parameters of standard broadcast control.

Parameter	Value
n	2
N	6
$x(0)$	$\begin{bmatrix} 90 & 60 & 80 & 80 & 60 & 90 \\ 40 & 90 & 20 & 80 & 10 & 60 \end{bmatrix}^T$
p_j	$\begin{bmatrix} 50 & 80 & 40 & 60 & 60 & 60 \\ 30 & 40 & 50 & 40 & 70 & 40 \end{bmatrix}^T$
$\beta_{ij}(k)(i=1,2,\ldots,N)$ $(j=1,2,\ldots,n)$	$\begin{cases} \Pr(\beta_{ij}(k)=1)=0.5 \\ \Pr(\beta_{ij}(k)=-1)=0.5 \end{cases}$
$a(k)$	$\begin{cases} \dfrac{a_0}{\left(\dfrac{k}{2}+1+a_v\right)^{a_p}} & \text{if } k \text{ is even} \\ a(k-1) & \text{if } k \text{ is odd} \end{cases}$ $a_0 = 2.1,\ a_v = 150,\ a_p = 1.55$
$c(k)$	$\begin{cases} \dfrac{c_0}{\left(\dfrac{k}{2}+1\right)^{c_p}} & \text{if } k \text{ is even} \\ c(k-1) & \text{if } k \text{ is odd} \end{cases}$ $c_0 = 6,\ c_p = 0.86$
λ	1×10^{30}

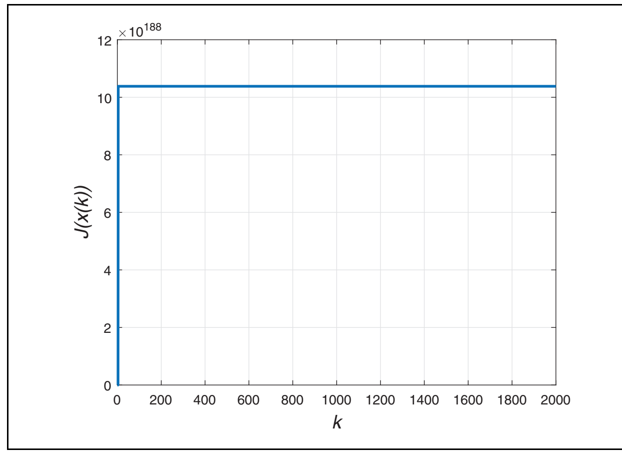

Figure 4. Evolution of the cost function for role-assignment task using standard broadcast control framework.

parameters of these two gains will be described in the fourth section. From Figure 4, it can be seen that the evolution of the role-assignment task diverges from x^* where x^* is the optimal solution. Recall that this divergence or instability phenomenon occurred due to the deterministic input in the second condition of equation (4), where it becomes too large to execute as $k \to \infty$.

This example proved that the standard broadcast control framework does not always resolve the motion-coordination task issue. Hence, it encouraged us to adopt a modified version of broadcast control framework.

Modified broadcast controller

In order to solve the instability phenomenon above, we propose a *modified broadcast controller*. In particular, the function ψ in the local controller is modified as

$$\psi(k,\xi_i(k),y(k)) := \begin{cases} c(k)\beta_i(k) & \text{if } k \text{ is even} \\ -\xi_{i2}(k) - \gamma(d(k,\xi_i(k),y(k))) & \text{if } k \text{ is odd} \end{cases} \tag{7}$$

where $\gamma : \mathbb{R}^n \to \mathbb{R}^n$ is a saturation function and the update vector $d(k,\xi_i(k),y(k))$ is described as

$$d(k,\xi_i(k),y(k)) = a(k)\left(\frac{\Delta J}{c(k)}\xi_{i1}^{-1}(k)\right) \tag{8}$$

Meanwhile, the function and the global controller are set similar to the standard broadcast controller in equations (3) and (5), respectively. Next, the convergence result of our proposed modified broadcast controller in equations (1), (3), (5), and (7) is shown.

Theorem 1. For the modified broadcast control system P, suppose that $(L_1, L_2, \ldots, L_N, G)$ are given by equations (3), (5), and (7). Should the saturation function γ and objective function J are given, then

$$\lim_{k\to\infty} x(k) = x^*$$

where $x^* \in \mathbb{R}^n$ is the solution to $x(k)$ subject to the following conditions:

C1: For almost of $x(k)$ (at each $k \geq K$ for some $K < \infty$) and some ω_0, J is assumed to have twice continuously differentiable, $\nabla J^{(2)}(x)$ bounded by ω_0 for all x.
C2: $\| x(k) \| < \infty$ for all $k \in \mathbb{Z}_{0+}$.
C3: x^* is an asymptotically stable solution of the differential equation $\dot{z}(k) = -\nabla J(z(k))$ where $z(k) \in \mathbb{R}^{nN}$ and the stability is in the Lyapunov sense.
C4: $a(k) = a(k+1) > 0, c(k) = c(k+1) > 0$ for all $k \in \{0,2,4,\ldots\}$, $\lim_{k\to\infty} a(k) = 0$, $\sum_{k=0}^{\infty} a(k) = \infty$, $\lim_{k\to\infty} c(k) = 0$, $\sum_{k=0}^{\infty} c(k) = \infty$, and $\sum_{k=0}^{\infty} (a(k)/c(k))^2 < \infty$.
C5: $\beta_{ij}(k)(i=1,2,\ldots,N)(j=1,2,\ldots,n)$ are i.i.d. random numbers drawn from the Bernoulli

distribution with outcome ± 1 and have equal probabilities given by

$$\begin{cases} \Pr\big(\beta_{ij}(k) = 1\big) = 0.5 \\ \Pr\big(\beta_{ij}(k) = -1\big) = 0.5 \end{cases} \tag{9}$$

where $\beta_{ij}(k)$ is the Jth element of $\beta_{ij}(k)$. Note that the $\beta_{ij}(k)$ is independent from $x(k)(k = 0, 1, 2, \ldots)$ and symmetrically distributed close to zero with $|\beta_{ij}(k)| < \infty$, $|\beta_{ij}^{-1}(k)| < \infty$, and $|\beta_{ij}^{-2}(k)| < \infty$. This condition is identical to the convergence condition for SPSA algorithm in Spall[26] where the design of this broadcast controller is inspired from.

Proof. The following facts prove the theorem.

(i) For all $k \in \mathbb{Z}_{0+}$, the relation from equations (1), (2), (3), (5), (7), and (8) can be written as

$$\begin{aligned} x_i(2k+1) &= x_i(2k) + u_i(2k) \\ &= x_i(2k) + \psi(2k, \xi_i(2k), y(2k)) \\ &= x_i(2k) + c(2k)\beta_i(2k) \end{aligned} \tag{10}$$

$$\begin{aligned} x_i(2k+2) &= x_i(2k+1) + u_i(2k+1) \\ &= x_i(2k+1) + \psi((2k+1), \xi_i(2k+1), y(2k+1)) \\ &= x_i(2k+1) - \xi_{i2}(2k+1) \\ &\quad - \gamma(d((2k+1), \xi_i(2k+1), y(2k_1))) \end{aligned} \tag{11}$$

$$x_i(2k+2) = x_i(2k+1) - \xi_{i2}(2k+1) - \hat{a}\left(\frac{\Delta J}{c(2k+1)}\xi_{i1}^{(-1)}(2k+1)\right) \tag{12}$$

where

$$\hat{a}(2k) = \hat{a}(2k+1) := \begin{cases} e\dfrac{c(2k+1)}{|\Delta J|} & \text{if} |\Delta J| > \dfrac{c(2k+1)e}{a(2k+1)} \\ a(2k+1) & \text{otherwise} \end{cases} \tag{13}$$

If k is even, the position of the robot in equation (1) is updated with random movement $c(2k)\beta_i(2k)$ as shown in equation (10). Then, the resulted position from equation (10) is updated with deterministic movement $-\xi_{i2}(2k+1) - \gamma(\cdot)$ if k is odd as appears in equation (11). From the definition of the saturation function $\gamma(\cdot)$ in equations (7) and (8), we can obtain equation (12) where the design of gain $\hat{a}$ (the same gain is used if k is even and odd) is given in equation (13). The value of gain $\hat{a}$ obtained from equation (13) determines how far the robot will perform the deterministic movement. The detailed derivation of gain $\hat{a}$ is described in the following paragraph.

Note that equation (12) corresponds to a general stochastic approximation algorithm in Appendix 1. From the convergence conditions of the modified broadcast controller algorithm, it is obvious that **C2–C5** imply **D1–D4** in Appendix 1. On the other hand, $\hat{a}(2k+1)$ in equation (13) is a nontrivial fact that must be proven to show that it satisfies the condition in **C4**.

(ii) Conditions from **C1**, **C2**, and **C4** imply **C4′** $\hat{a}(k) = \hat{a}(k+1) > 0$, $c(k) = c(k+1) > 0$ for all $k \in \{0, 2, 4, \ldots\}$, $\lim_{k\to\infty}\hat{a}(k) = 0$, $\lim_{k\to\infty}c(k) = 0$, and $\sum_{k=0}^{\infty}\hat{a}(k)^2/c(k)^2 < \infty$. Moreover, $\sum_{k=0}^{\infty}\hat{a}(k) = \infty$ w.p.1.

(iii) From equations (7) and (8) and γ notation in the first section 1

$$\begin{aligned} &\gamma(d((2k+1), \xi_i(2k+1), y(2k+1))) \\ &\quad = \min(|d((2k+1), \xi_i(2k+1), y(2k+1))|, e) \\ &\qquad \operatorname{sgn}(d((2k+1), \xi_i(2k+1), y(2k+1))) \end{aligned}$$

since $a(2k+1)\left(\frac{\Delta J}{c(2k+1)}\xi_{i1}^{(-1)}(2k+1)\right) > e$ is equivalent to

$$e\operatorname{sgn}\left(\Delta J\xi_{i1}^{(-1)}(2k+1)\right) \tag{14}$$

and since the parameter that we want to control is ΔJ, thus

$$\gamma(d((2k+1), \xi_i(2k+1), y(2k+1))) = \begin{cases} e\operatorname{sgn}\left(\Delta J\xi_{i1}^{(-1)}(2k+1)\right) & \text{if} |\Delta J| > e\dfrac{c(2k+1)}{a(2k+1)} \\ a(2k+1)\left(\dfrac{\Delta J}{c(2k+1)}\xi_{i1}^{(-1)}(2k+1)\right) & \text{otherwise} \end{cases} \tag{15}$$

where $e \in \mathbb{R}_+$ is a predefined positive integer. The next step is to reconstruct equation (14) so that it follows the standard SPSA algorithm as shown in equation (8). From signum properties

$$\begin{aligned} &e\operatorname{sgn}\left(\Delta J\xi_{i1}^{(-1)}(2k+1)\right) \\ &\quad = e\frac{c(2k+1)}{c(2k+1)}\frac{|\Delta J|}{|\Delta J|}\operatorname{sgn}\left(\Delta J\xi_{i1}^{(-1)}(2k+1)\right) \\ &\quad = e\frac{c(2k+1)}{|\Delta J|}\frac{\Delta J}{c(2k+1)}\xi_{i1}^{(-1)}(2k+1) \end{aligned} \tag{16}$$

So, equation (16) shows that it satisfies equations (11) and (12) that we have equation (13).

(iv) The detailed proof of fact (ii) is shown here. For the first condition, since $0 < \hat{a}(k) < a(k)$ for all $k \in \mathbb{Z}_{0+}$ holds from equation (13) and **C4**, it follows a straightforward proof from Tanaka et al.[19] which shows that our first condition of **C4′** holds. Now, we prove the second condition of **C4′**, that is,

$\sum_{k=0}^{\infty}\hat{a}(k) = \infty$ w.p.1. For every $k \in \mathbb{Z}_{0+}$, conditions **C1** and **C2** imply that there exist $\delta_0 > 0$ satisfying

$$\| \nabla J^{(1)}(x(2k))\|_{\infty} \leq \delta_0 \tag{17}$$

By applying Taylor's theorem (e.g. Appendix I in Azuma et al.,[27]), **C1**, **C5**, and equation (17)

$$\begin{aligned} |\nabla J| &\leq |c(2k)\nabla J^{(1)}x((2k))\beta(2k) \\ &\quad + \frac{c(2k)^2}{2}\nabla J^{(2)}(x(2k))\beta(2k)\otimes\beta(2k)| \\ &\leq c(2k)n\delta_0 + \frac{c(2k)^2}{2}n^2\omega_0^2 \quad \text{w.p.1} \end{aligned} \tag{18}$$

Substituting equation (18) into equation (13)

$$\begin{aligned} \frac{c(2k)e}{|\Delta J|} &\geq e\left(\frac{c(2k)}{c(2k)n\delta_0 + \frac{c(2k)^2 n^2\omega_0^2}{2}}\right) \\ &\geq e\left(n\delta_0 + \frac{c(2k)n^2\omega_0^2}{2}\right)^{-1} \quad \text{w.p.1} \end{aligned} \tag{19}$$

Finally, from equations (13) and (18), $e > 0$, $n > 0$, $\delta_0 > 0$, and $\omega_0 > 0$

$$\begin{aligned} \sum_{k=0}^{\infty}\hat{a}(2k) &\geq \sum_{k=0}^{\infty}\frac{c(2k)e}{|\Delta J|} \\ &\geq \sum_{k=0}^{\infty} e\left(n\delta_0 + \frac{c(2k)n^2\omega_0^2}{2}\right)^{-1} \\ &= \infty \quad \text{w.p.1} \end{aligned} \tag{20}$$

which verify the proof.

Here we provide comments on conditions **C1–C5**. The twice differentiability, that is, $\nabla J^{(2)}(x)$, in **C1** means that the resulting J is sufficiently smooth and it is required to obtain equation (3A) in Appendix 1 by Taylor's theorem. The easiest way to satisfy this condition is to ensure that the order of J is at least quadratic. In **C2**, the condition is required to obtain Taylor's theorem in equation (18). In **C3**, the condition is required to guarantee that if $x(k)$ does not comply with this condition, the local minimum of $\nabla J(z(k))$ (i.e. $\nabla J(x(k))$) would still exist. That is because the gradient system and broadcast control system are different. So, even if the gradient system is stable, $x(k)$ would not necessarily converge to the solution. A sufficient condition for x^* to be asymptotically stable solution is that the Hessian matrix of $J(x)$ must be nonsingular at each x satisfying $\nabla J(x) = 0$. In summary, this condition is just a condition for gradient system correlated with the broadcast control. Condition **C4** is imposed for users to design the gains a and c. More details are discussed in the fourth section, where by following the typical design would easily satisfy this condition. Finally, condition in **C5** is provided to specify the random variables β_{ij} which can be easily achieved.

Algorithm 1. Modified broadcast controller.

```
Parameters: n, N, x(0), β, a, c, λ, e
Coordination task: J(x), e.g: assignment task (6)
for k = 0:iteration do
    Global controller gets the robots positions, x(k)
    Computes J(x(k))
    Broadcasts y(k) ← J(x(k))
    for i = 1:N do
        if k is even then
            Robot i computes u_i(k) through the first
            condition of (7)
            Update ξ_i1 ← β_i(k), ξ_i2 ← u_i(k),
            ξ_i3 ← y(k)
        else
            Robot i computes u_i(k) through the second
            condition of (7)
        end if
    end for
    Update x(k + 1) through (1)
end for
```

Before explaining the simulation results, we summarize the modified broadcast control in Algorithm 1.

Numerical simulation

Consider the broadcast control system P in Figure 2, for $N := 6$, $n := 2$, $\lambda := 1 \times 10^{30}$, and $e := 0.5$. The cost function considered in this simulation is role-assignment task given in equation (6) where this choice of cost function is assumed to satisfy **C1**. For the broadcast controller $(L_1, L_2, \ldots, L_N, G)$, we employed the modified broadcast controller given by equations (2), (3), (7), and (8) which was proven to approximately achieve the convergence w.p.1 subject to **C1**–**C5**. Note that there is a difference between the local controllers utilized in the standard broadcast controller (i.e. equations (3) and (4)) and the modified broadcast controller (i.e. equations (3), (7), and (8)). This is because the local controller in the standard broadcast controller triggered the instability and resulted in different local controller being utilized in the modified broadcast controller. For this simulation, the gains $a(k)$ and $c(k)$ embedded in the local controllers are set to be the same as in Table 1 where these gains were arbitrarily chosen to satisfy the condition in **C4**. Note that the general guidelines to design the gains $a(k)$ and $c(k)$ can be found in Sections 7.5.1 and 7.5.2 of Spall.[28] For the value of e, we referred to the example given in Tanaka et al.[19] and adjusted this value according to the size of the deterministic movement taken by the robots that the user want, that is, the larger the value of e, the larger the deterministic movement of the robots.

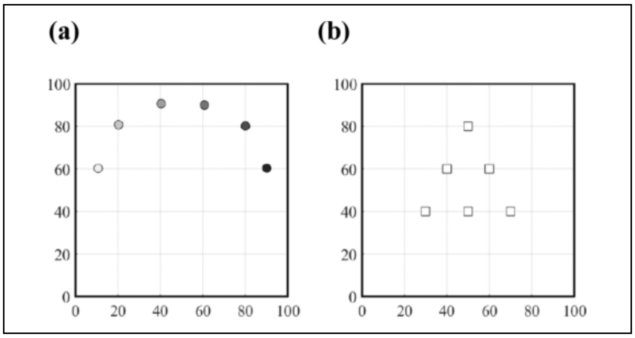

Figure 5. Snapshot of (a) initial position and (b) target position.

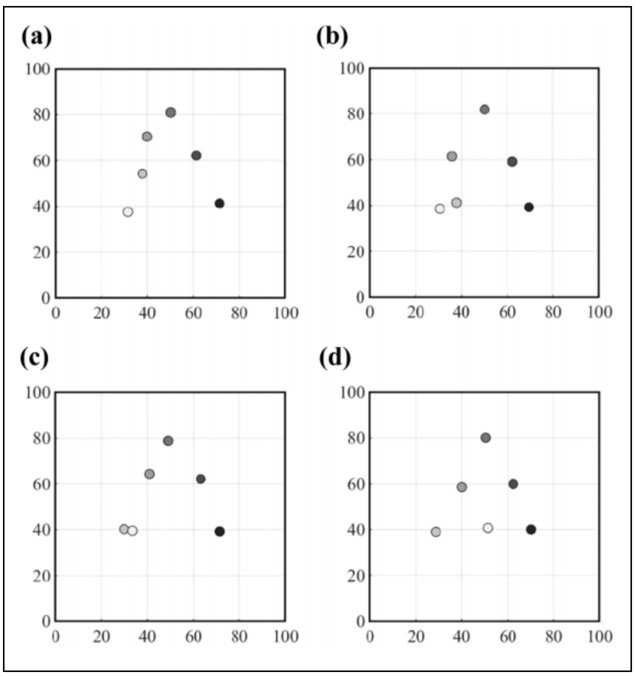

Figure 6. Positions of robots at different iteration k when performing role-assignment task: (a) $k = 500$, (b) $k = 1000$, (c) $k = 1500$, and (d) $k = 2000$.

Figure 5 shows the snapshots of the initial positions of the robots $x_i(0) (i = 1, 2, \ldots, N)$ and the target positions of the robots $p_j (j = 1, 2, \ldots, N)$. The values of these initial and target positions are set to be the same as in Table 1. The circles in gray denoted the robots numbered as $x_1(0)$ until $x_6(0)$ counted from the right-hand side and the small squares are the target positions. On the other hand, Figure 6 shows the positions of the robots $x_i(k) (i = 1, 2, \ldots, N)$ at different iteration $k = (0, 500, \ldots, 2000)$ when performing the role-assignment task using the modified broadcast controller. Note that the desired position for the robots to perform the assignment task is unknown (i.e. it is not necessary for robot 1 to converge to target position 1, for robot 2 to converge to target position 2, etc.) and the robots are required to cooperate with each other to determine their ideal target positions.

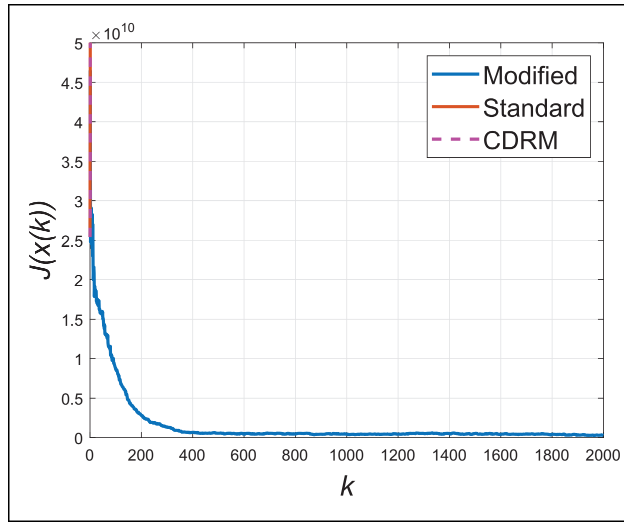

Figure 7. Evolution of the cost function for role-assignment task.

Table 2. Experiment on different gain parameters for role-assignment task.

	Evolution of cost function		
Gain parameters	Modified	Standard	CDRM
$a_0 = 3$, $a_v = 16.5$, $a_p = 0.7$, $c_0 = 2$, $c_p = 0.6$	Converge	Diverge	Diverge
$a_0 = 4.1$, $a_v = 32$, $a_p = 0.95$, $c_0 = 5$, $c_p = 0.36$	Converge	Diverge	Diverge
$a_0 = 3.34$, $a_v = 50$, $a_p = 1.2$, $c_0 = 7$, $c_p = 0.46$	Converge	Diverge	Diverge
$a_0 = 1.1$, $a_v = 12$, $a_p = 0.8$, $c_0 = 1.4$, $c_p = 0.26$	Converge	Diverge	Diverge
$a_0 = 2.3$, $a_v = 120$, $a_p = 1.8$, $c_0 = 1.8$, $c_p = 0.26$	Converge	Diverge	Diverge

CDRM: constant-distance random movement.

Figure 7 provides the evolution of the cost function from Figure 6 to perform the role-assignment task with the modified broadcast controller. The evolution is also compared with the standard and constant-distance random movement[12] (CDRM) broadcast controllers to analyze the performance of our modified broadcast controller in

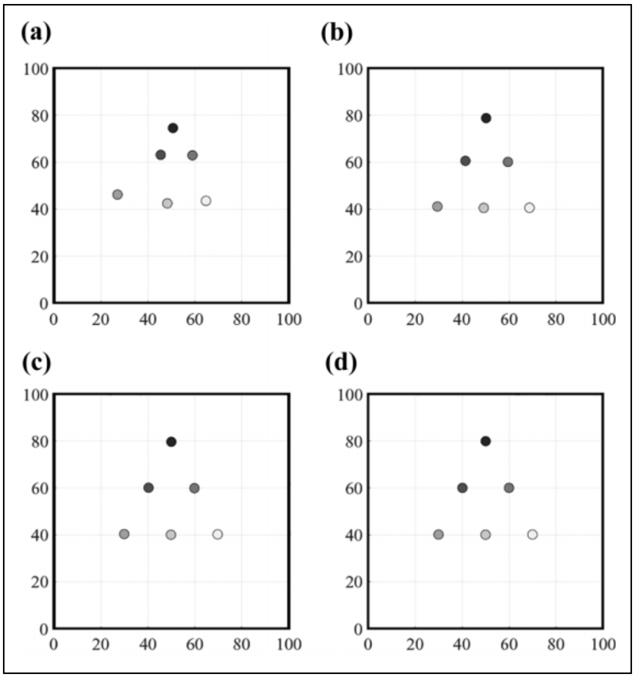

Figure 8. Positions of robots at different iteration k when performing consensus task: (a) $k = 500$, (b) $k = 1000$, (c) $k = 1500$, and (d) $k = 2000$.

performing the role-assignment task. From the figure, the robots with modified broadcast controller begin to converge into the assigned positions at about $k = 400$. Meanwhile, the standard and CDRM broadcast controllers diverge toward the infinite value of $J(x(k))$ even at the beginning of the iteration. We also have tuned the gain parameters of these three broadcast controllers for some values as tabulated in Table 2, but the evolution is still showing the same, that is, the modified broadcast controller is converging while the standard and CDRM broadcast controllers are diverging. Note that the CDRM broadcast controller only limits the random movement of the robots and not the deterministic movement. Because of this fact, the CDRM broadcast controller also diverges toward the infinite value of $J(x(k))$.

In order to demonstrate the usefulness of our modified broadcast controller in handling various classes of motion-coordination tasks, we also employed it to a motion-coordination task that has been successfully solved using the standard and CDRM broadcast controllers. Consider a consensus task[13] given by

$$J(x) = (x_i - p_i)^{\mathrm{T}}(x_i - p_i) \tag{21}$$

This cost function is different from the role-assignment task where the robots are assigned with their corresponding target positions. For example, robot x_1 is required to go to target position p_1, robot x_2 is required to go to target position p_2, and so on. Readers may refer to Cao et al.,[29] Wasiela et al.,[30] and Huang et al.,[31] for other examples of

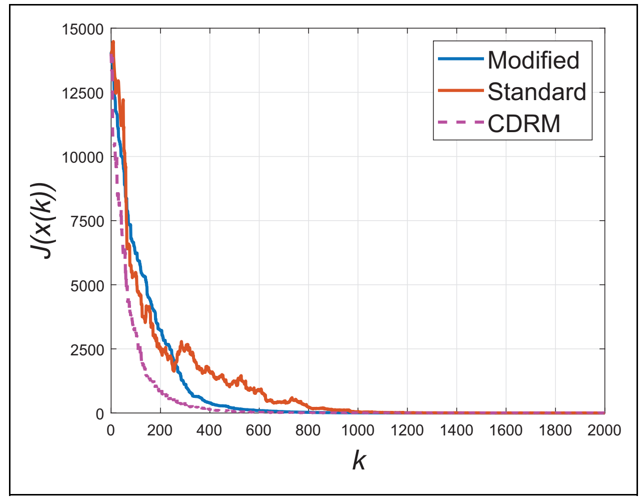

Figure 9. Evolution of the cost function for consensus task.

Table 3. Convergence time for modified, standard, and CDRM broadcast controllers.

	Standard	Modified	CDRM
Minimum	800	886	790
Maximum	1902	1058	1110
Average	1390.7	945.7	934.6
Standard deviation	357.5	63.3	98.0

CDRM: constant-distance random movement.

consensus task. Note that we used the same parameter values as in Table 1 for this task except $a(k)$, $c(k)$, and e. For the gains $a(k)$ and $c(k)$, we set the parameters $a_0 = 2.2$, $a_v = 450$, $a_p = 0.95$, $c_0 = 0.26$, and $c_p = 0.16$ while the value of e is set to be $e := 1$. Figure 8 shows the position of the robots $x_i(k)(i = 1, 2, \ldots, N)$ at iteration $k = (0, 500, \ldots, 2000)$.

Figure 9 shows the performance index of the modified broadcast controller in performing the consensus task. Since the standard and CDRM broadcast controllers can also perform this consensus task, we also include the evolution of the cost function of these two broadcast controllers so that we can compare the convergence time between all broadcast controllers. In this regard, we define the convergence time as the required number of iterations for the objective function $J(x(k))$ to achieve its minimum value and we choose the minimum value to be $J(x(k)) \leq 5$. From Figure 9, we observe that the modified broadcast controller converges faster (converges at around $k = 1093$) as compared to the standard broadcast controller (converges at around $k = 1465$) but it shows a slower convergence compared to the CDRM broadcast controller (which converges at around $k = 858$).

In order to fairly compare the convergence time in performing the consensus task, we run the simulation for 10 times for each broadcast controller and tabulated the results in Table 3. From the table, in particular the average values, it is obvious that the modified broadcast controller significantly improves the convergence time approximately 32% compared to the standard broadcast controller while being competitive when compared to the CDRM broadcast controller. On the other hand, the standard deviation shows that the convergence time for the modified broadcast controller does not varies much from the average in contrast with standard and CDRM broadcast controllers. This convergence time improvement is due to the limit we set for the norm of the update vector of robots' positions (in this simulation, the parameter is e). Consecutively, we can confirm the effectiveness of our modified broadcast controller in handling the motion-coordination task in terms of the instability and also has the possibility to reduce convergence time.

Conclusion

This article has addressed the issue of unstable solution in the standard broadcast control framework to solve certain motion-coordination tasks. Briefly, the standard broadcast control framework was modified, that is, limiting the norm of the update vector of robots' position. As a theoretical result, the proposed modified broadcast controller was proven to approximately achieve the convergence w.p.1. Finally, simulation results were provided to illustrate the effectiveness of our modified broadcast controller in solving the instability issue and the possibility to reduce convergence time. As for future work, collision avoidance should be included into the broadcast controller so that we can prevent damage to the robots. In addition, experiments based on real robotic systems should also be considered to validate the performance of the modified broadcast controller.

Acknowledgement

The authors would like to thank Universiti Teknologi Malaysia for funding this research project.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Universiti Teknologi Malaysia through a research grant (Q.K130000.2543.13H80) titled "Machine Vision Algorithm for Remotely Operated Vehicle in the Inspection of Industrial Reservoirs."

ORCID iD

Zool Hilmi Ismail https://orcid.org/0000-0002-5918-636X

References

1. Ismail ZH and Sariff N. A survey and analysis of cooperative multi-agent robot systems: challenges and directions. In: *Hurtado EG (ed) Applications of mobile robots*. London, UK: IntechOpen, 2018, pp. 8–14.
2. Ueda J, Odner L, and Asada HH. Broadcast feedback of stochastic cellular actuators inspired by biological muscle control. *Int J Robot Res* 2007; 26: 1251–1265.
3. Julius AA, Halasz A, Sakar MS, et al. Stochastic modeling and control of biological systems: the lactose regulation system of *Escherichia coli*. *IEEE Trans Autom Control* 2008; 53: 51–65.
4. Wood LB, Das A, and Asada HH. Broadcast feedback control of cell populations using stochastic Lyapunov functions with application to angiogenesis regulation. In: *Proceedings of American control conference (ACC)*, Seattle, WA, USA, 11–13 June 2008, pp. 6227–6232. New Jersey, US: IEEE.
5. Mesquita AR, Hespanha JP, and Astrom K. Optimotaxis: a stochastic multi-agent optimization procedure with point measurements. In: *International workshop on hybrid systems: computation and control*, St. Louis, MO, USA, 22–24 April 2008, pp. 358–371. Berlin: Springer.
6. Berman S, Halasz A, Hsieh MA, et al. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Trans Robot* 2009; 25(4): 927–937.
7. Das K and Ghose D. Positional consensus in multi-agent systems using a broadcast control mechanism. In: *Proceedings of American control conference (ACC)*, St. Louis, MO, USA, 10–12 June 2009, pp. 5731–5736. IEEE.
8. Azuma SI and Sugie T. Broadcast control of multi-agent systems. In: *Proceedings of decision and control and European control conference (CDC-ECC)*, Orlando, FL, USA, 12–15 December 2011, pp. 3590–3595. IEEE.
9. Azuma SI, Yoshimura R, and Sugie T. Broadcast control of multi-agent systems. *Automatica* 2013; 49(8): 2307–23161.
10. Azuma SI, Sakar MS, and Pappas GJ. Stochastic source seeking by mobile robots. *IEEE Trans Autom Control* 2012; 57(9): 2308–2321.
11. Azuma SI, Baba I, and Sugie T. Broadcast control of Markovian multi-agent systems. *SICE J Control Measure Syst Integrat* 2016; 9(2): 103–112.
12. Tanaka Y, Azuma SI, and Sugie T. Broadcast control with constant-distance random movement. *Trans Soc Instrum Control Eng* 2014; 50: 139–146.
13. Tanaka Y, Azuma SI, and Sugie T. Broadcast control of multi-agent systems with quantized measurements. *IEICE Trans Fundam Electron Commun Comput Sci* 2014; 97(3): 830–839.
14. Tanaka Y, Azuma SI, and Sugie T. Multi-agent consensus under a communication-broadcast mixed environment. *Int J Control* 2014; 87(6): 1103–1116.
15. Ito Y, Kamal MAS, Yoshimura T, et al. Multi-vehicle coordination on merging roads based on pseudo-perturbation-based broadcast control. In: *Proceedings of annual American control conference (ACC)*, Milwaukee, WI, USA, 27–29 June 2018, pp. 4008–4013. IEEE.

16. Ito Y, Kamal MAS, Yoshimura T, and Azuma SI. Coordination of connected vehicles on merging roads using pseudo-perturbation-based broadcast control. *IEEE Trans Intelligent Transp Syst* 2018; 20(9): 3496–3512.
17. Darmaraju S, Shanmugavel M, Kamal MAS, et al. Area coverage by a group of UAVs using the broadcast control framework. *IFAC-PapersOnLine* 2019; 52(12): 370–375.
18. Fujishima Y and Azuma SI. Broadcast control of radar surveillance system. In: *Annual conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Nara, Japan, 11–14 September 2018, pp. 83–88. IEEE.
19. Tanaka Y, Azuma SI, and Sugie T. Simultaneous perturbation stochastic approximation with norm-limited update vector. *Asian J Control* 2015; 17: 2083–2090.
20. Ahmad MA, Azuma SI, and Sugie T. Performance analysis of model-free PID tuning of MIMO systems based on simultaneous perturbation stochastic approximation. *Exp Syst Appl* 2014; 41(14): 6361–6370.
21. Baba I, Azuma SI, and Sugie T. Controller design for optimizing fuel consumption of hybrid electric vehicles: a model-free approach based on simultaneous perturbation stochastic approximation. *Trans Soc Instrum Control Eng* 2013; 49(9): 887–894.
22. Xiong Q, Lin P, Chen Z, et al. Distributed containment control for first-order and second-order multi-agent systems with arbitrarily bounded delays. *Int J Robust Nonlinear Control* 2019; 29(4): 1122–1131.
23. Liu W and Gao Z. A distributed flocking control strategy for UAV groups. *Comput Commun* 2020; 153: 95–101.
24. Liu A, Zhang WA, Yu L, et al. Formation control of multiple mobile robots incorporating an extended state observer and distributed model predictive approach. *IEEE Trans Syst Man Cybern Syst* 2018; 1–11.
25. Ji M, Azuma SI, and Egerstedt MB. Role-assignment in multi-agent coordination. *Int J Assist Robot Mechatronics* 2006; 7(1): 32–40.
26. Spall JC. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans Autom Control* 1992; 37(3): 332–341.
27. Azuma SI, Baba I, and Sugie T. Broadcast control of group of Markov chains. In: *Annual conference on decision and control (CDC)*, Maui, HI, USA, 10–13 December 2012, pp. 2059–2064. IEEE.
28. Spall JC. *Introduction to stochastic search and optimization: estimation, simulation, and control* (Vol. 65). Hoboken: John Wiley and Sons, 2005.
29. Cao Y, Zhang L, Li C, et al. Observer-based consensus tracking of nonlinear agents in hybrid varying directed topology. *IEEE Trans Cybern* 2017; 47(8): 2212–2222.
30. Wasiela S, Kasshyap N, Pan YJ, et al. Realization of consensus with collision and obstacle avoidance in an unknown environment for multiple robots. In: *International conference on robotics and biomimetics (ROBIO)*, Dali, China, 6–8 December 2019, pp. 741–746. IEEE.
31. Huang Z, Pan YJ, and Bauer R. Leader–follower consensus control of multiple quadcopters under communication delays. *J Dyn Syst Meas Control* 2019; 141(10): 1–10.
32. Kushner HJ and Clark DS. *Stochastic approximation methods for constrained and unconstrained systems* (Vol. 26). Berlin: Springer Science and Business Media, 2012.

Appendix 1

A general stochastic approximation algorithm, that is, Robbins–Monro algorithm, is given by

$$\zeta(t+1) = \zeta(t) - a(t)(F(\zeta(t)) + \sigma(t) + e(t)) \tag{1A}$$

where $t \in \mathbb{Z}_{0+}$ is discrete time, $\zeta(t) \in \mathbb{R}^d$ is the state, $a(t) \in \mathbb{R}_+$ is the time-varying gain, $F : \mathbb{R}^d \to \mathbb{R}^d$ is a function that needs to be minimized, and $\sigma(t)$ as well as $e(t) \in \mathbb{R}^d$ are random variables. For detailed convergence analysis of this algorithm, refer to Kushner and Clark.[32]

Considering the algorithm in equation (1A), the solution for $\zeta^* \in \mathbb{R}^d$ and $\min F(\zeta(t))$ can be obtained if the following conditions are satisfied.

D1: There exist a compact set $S \subseteq \mathbb{R}^d$ where this set is the stability region for $\dot{\zeta}(t) = -F(\zeta(t))$. The $\dot{\zeta}(t) = -F(\zeta(t))$ with $\zeta(0)$ results in $\zeta(\infty) = \zeta^*$ such that $\zeta(t) \in \mathbf{S}$ occurs infinitely often for every $\zeta(0) \in \mathbf{S}$ and almost all sample points.

D2: $\lim_{t\to\infty} a(t) = 0$ and $\sum_{t=0}^{\infty} a(t) = \infty$.

D3: $\sup_{t\in\mathbf{Z}_{0+}} \| \zeta(t) \| < \infty$ w.p.1 for every $\zeta(0) \in \mathbf{S}$.

D4: For every $\zeta(0) \in \mathbf{S}$ and $t \in \mathbb{R}_+$

$$\lim_{t\to\infty} \Pr\left(\sup_{l\geq i} \| \sum_{t=i}^{l} a(i)e(i) \| \geq t \right) = 0$$

From equation (12) and **C4′**

$$\begin{aligned} x_i(2k+2) &= x_i(2k+1) - \xi_{i2}(2k+1) \\ &\quad -\hat{a}(2k+1)\left(\frac{\Delta J}{c(2k+1)}\xi_{i1}^{(-1)}(2k+1)\right) \\ &= x_i(2k) - \hat{a}(2k) \\ &\quad \left(\frac{J(x(2k+1)) + c(2k)\beta(2k) - J(x(2k))}{c(2k)}\xi_{i1}^{(-1)}(2k)\right) \\ &= x_i(2k) - \hat{a}(2k)d(2k, \xi_i(2k), y(2k)) \end{aligned} \tag{2A}$$

and by applying Taylor's theorem to ΔJ (note the difference between ΔJ and ∇J) and taking similar way as in Ji et al.[25] and Spall[26] resulted in

$$\begin{aligned} &\mathrm{E}(d(2k,\xi_i(2k),y(2k)|x_i(2k))) \\ &\quad = \nabla J(x(2k)) + O(c(2k)) \quad (c(2k)\to 0) \end{aligned} \tag{3A}$$

This means

$$E(x_i(2k+2)|x_i(2k)) \approx x_i(2k) - \hat{a}(2k)\nabla J(x(2k)) \tag{4A}$$

The form in equation (4A) can be transformed into

$$\begin{aligned} x_i(2k+2) &= x_i(2k) - \hat{a}(2k)[\nabla J(x(2k))] \\ &\quad - \mathrm{E}(d(2k,\xi_i(2k)),y(2k)|x(2k)) + O(c(2k)) \end{aligned} \tag{5A}$$

where $x_i(2k+2)$, $x_i(2k)$, $\hat{a}(2k)$, $\nabla J(x_i(2k))$, $\nabla J(x_i(2k)) - \mathrm{E}(d(2k,\xi_i(2k)),y(2k)|x(2k))$, and $O(c(2k))$ correspond to $\zeta(t+1)$, $\zeta(t)$, $a(t)$, $F(\zeta(t))$, $\sigma(t)$, and $e(t)$, respectively, in equation (1A).