Broadcast Control of Multi-Agent Systems for Assembling at Unspecified Point with Collision Avoidance

M. H. Mohamad Nor [1], Z. H. Ismail [1], M.A. Ahmad [2]
[1]Centre for Artificial Intelligence and Robotics, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, 51400, Kuala Lumpur, Malaysia.
[2]Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Pekan, 26600, Pahang, Malaysia.
mhafizulazwan@gmail.com

***Abstract*—This paper presents a broadcast control problem for multi-agent systems to perform a motion-coordination i.e. assembling tasks without colliding into each other. In this paper, we use potential energy function for the collision avoidance where the agents will move away from the neighbor agents if they are reaching certain distance. In the simulation, the results show that the agents are successfully achieving the assembling task without colliding into each other.**

***Index Terms*—Broadcast Control; Assembling; Collision Avoidance; Simultaneous Perturbation Stochastic Approximation (SPSA).**

I. INTRODUCTION

The effectiveness of using multi-agent systems in solving various engineering problem has attracted many researchers to explore this field. This is because the assigned task can be executed in a more effective and efficient way if we have many agents. Previously, researchers were focusing on the development of communication between the agents to complete the multi-agent coordination task. Because of several factors that are lacking in the communication between the agents, researchers have started to consider and develop new communication framework, which is based on broadcast strategy, as illustrated in Figure 1. Among the advantages of using this broadcast framework are, the agents only need one same command input, more energy can be save by the agents, and suitable for a large number of agents.

Figure 1: Example of Multi-agent System Based on Broadcast

The broadcast control framework for multi-agent coordination has been introduced by [1] and the detailed proof of this framework is shown in [2]. This new controller can be applied for various multi-agent coordination tasks such as assembly at unspecified location, role assignment [3] and coverage over an environment [4]. Note that the broadcast control algorithm is developed based on Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm which is originally proposed by [5]. Recently, the variations of this controller is focused on quantized problem caused by digital sensor [6], mix-environment case [7] and controller gain selection for fast convergence [8]. Besides that, there are also other recent progresses doing on broadcast control as reported in [7, 9-11].

For the broadcast control framework, the works in [1] claims that their proposed broadcast controller can handle various multi-agent coordination tasks. Thus, the problem to consider in this paper is to assemble the agents at an unspecified point with collision avoidance which have not yet been implemented by other researchers using the same broadcast control framework. The aim of this paper is to present the investigation on the ability of the broadcast controller to handle the motion-coordination task where the desired position of the agents is unknown along with the collision avoidance.

This paper thus validates multi-agent problems based on the assembling and collision avoidance task by using the same broadcast control framework in [1]. Assembling in this problem means to agree over one unknown location for the agents to assemble. On the other hand, collision avoidance means that the agents would not hit into each other. This collision avoidance task has been inspired from region shape control proposed by [12] and has a better performance instead of using overlap function from [1]. This study aims to show that the assembling and collision avoidance task can be implemented using the broadcast control framework. The performance of the broadcast framework is then demonstrated by numerical simulation.

Notation: Let $\mathbf{R}$, $\mathbf{R}_+$, $\mathbf{Z}$, $\mathbf{Z}_{0+}$ 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 define as the Euclidean norm. Besides that, the gradient of the differentiable function $J:\mathbf{R}^n \to \mathbf{R}$ is expressed by $\nabla J(x)$ where $J(x) \in \mathbf{R}_+$. Lastly, we denote $\Pr(a)$ as probability of event *a*.

II. Problem Formulation

A. System Description

Consider the feedback broadcast control system Σ in Figure 2 which consists of a global controller, local controllers, and *N* agents.

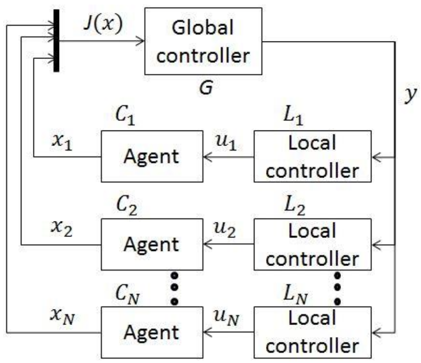


Figure 2: Broadcast Control System Σ

The coordination equation C_i of agents *i*=1,2,…,*N* is described by:

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

where $k \in \mathbf{Z}_{0+}$ is iteration, $x_i(k) \in \mathbf{R}^n$ is the *n*-dimensional space position, and $u_i(k) \in \mathbf{R}^n$ is the input. In the real system, the position of the agents can be tracked by using a sensor for example digital camera.

The coordination (1) can be updated with the following local controller (2) where it is embedded in each agent *N*.

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

where $\xi_i (i = 1,2, \dots, N) \in \mathbf{R}^v$ is the state to memorize the previous value, $y(k) \in \mathbf{R}$ is the input, $u_i(k)(i = 1,2,\dots, N)$ is the output, and $a : \mathbf{Z}_{0+} \times \mathbf{R}^v \times \mathbf{R} \to \mathbf{R}^v$ and $b : \mathbf{Z}_{0+} \times \mathbf{R}^v \times \mathbf{R} \to \mathbf{R}^n$ are functions. The functions *a* and *b* and the initial states $\xi_i(0)$ are assumed to be the *same* for all the local controllers $L_i (i = 1,2, \dots, N)$ and this means that agents are handled with random data. In order to make the notation simpler, the collective position of the agents *x*(*k*) is denoted as follows

$$x(k) := [x_1^{\mathrm{T}} \quad x_2^{\mathrm{T}} \quad \dots \quad x_N^{\mathrm{T}}]^{\mathrm{T}},$$

The function $J : \mathbf{R}^{nN} \to \mathbf{R}_{0+}$ is to express the performance of the group as to whether the agents have achieved their assigning task or not. The agents are said to converge if $\nabla J(x) = 0$.

Lastly, the global controller *G* is given as follows:

$$G : y(k) = f(J(x(k))) \tag{3}$$

where the *broadcast signal* is denoted by $y(k)$, $f : \mathbf{R}_{0+} \to \mathbf{R}$ is a function (see [2] for the reason of introducing *f*), and $J(x(k)) \in \mathbf{R}_{0+}$ is the input.

B. Broadcast Control with Assembling and Collision Avoidance Task

For the potential energy function given by:

$$q(\Delta x_{ij}) = \sum_{j \in \mathrm{N_i}} \frac{m}{2} [\max(0, d(\Delta x_{ij}))]^2, \tag{4}$$

the objective function for assembling with collision avoidance task is represented as follow:

$$J(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} \|x_i - x_j\| - \lambda \frac{\partial q(\Delta x_{ij})}{\partial(\Delta x_{ij})}, \tag{5}$$

where $\lambda \in \mathbf{R}_+$ is a positive constants, $q : \mathbf{R}^{nN} \to \{0\} \cup \mathbf{R}_+$ denote the potential energy function, N_i is a set of neighbors around agents *i* and *m* is a positive constant. Note that, the term $\sum_{i=1}^{N} \sum_{j=1}^{N} \|x_i - x_j\|$ represents the assembling task, while the term $\lambda \partial q(\Delta x_{ij}) / \partial(\Delta x_{ij})$ denotes the collision avoidance. Here, $\partial q(\Delta x_{ij}) / \partial(\Delta x_{ij})$ is the partial derivative of the potential energy function in (4) given by:

$$\frac{\partial q(\Delta x_{ij})}{\partial(\Delta x_{ij})} = \sum_{j \in N_i} m[\max(0, d(\Delta x_{ij})) \left(\frac{\partial q(\Delta x_{ij})}{\partial(\Delta x_{ij})} \right)^{\mathrm{T}}]. \tag{6}$$

The potential energy function above actually aims to minimize the distance between the agents define as:

$$d(\Delta x_{ij}) = r^2 - \|\Delta x_{ij}\|^2 \leq 0, \tag{7}$$

where $d : \mathbf{R}^{nN} \to \mathbf{R}$ is the minimum distance function, *r* is the radius defined between two agents and $\Delta x_{ij} = x_i - x_j (i = 1,2,\dots, N-1)(j = i+1,\dots, N)$ is the distance between robot *i* and robot *j*. If the distance between these agents is not satisfying (7) i.e. the value of $d(\Delta x_{ij}) > 0$, then the potential energy function in (6) will be activated to push the agents outside the region r. This means that the agents will not collides into each other.

Problem 1: The objective function or in other word performance index of assembling task with the collision avoidance $J:\mathbf{R}^{nN}\to\{0\}\cup\mathbf{R}_+$ for the system is given. Let x^* be the solution to $\min_{x\in\mathbf{R}^{nN}} J(x^*)$, which is the optimal desired position that the agents need to go. Then, find local controller $L_i\,(i=1,2,\ldots,N)$ and global controller G (i.e., find function *a*, *b*, and *f*) such that:

$$\lim_{k\to\infty} J(x(k))=\min_{x\in\mathbf{R}^{nN}} J(x^*), \tag{8}$$

for every initial state $x(0)\in\mathbf{R}^{nN}$ and $(\xi_1(0),\xi_2(0),\ldots,\xi_N(0))$ satisfying $(\xi_1(0)=\xi_2(0),=\ldots,=\xi_N(0))$.

III. Broadcast Control

As mentioned earlier, the broadcast controller $(L_1,L_2,\ldots,L_N,G)$ used in this paper is the same as the one proposed by [1] given by (2) and (3), where:

$$a(k,\xi_i(k),y(k)):=\begin{cases}\begin{bmatrix}\beta_i(k)\\ y(k)\end{bmatrix} & \text{if } k \text{ is even},\\ \xi_i(k) & \text{if } k \text{ is odd},\end{cases} \tag{9}$$

$$b(k,\xi_i(k),y(k)) :=\begin{cases} c\beta_i(k) & \text{if } k \text{ is even},\\ -c\xi_{i1}(k)-g(k)(\dfrac{y(k)-\xi_{i2}(k)}{c}\,\xi_{i1}{}^{(-1)}(k)) & \text{if } k \text{ is odd},\end{cases} \tag{10}$$

where the components of $\xi_i(k)$ are $\xi_{i1}(k)\in\mathbf{R}^n$ and $\xi_{i2}(k)\in\mathbf{R}$.

$$\xi_i(k)=\begin{bmatrix}\xi_{i1}(k)\\ \xi_{i2}(k)\end{bmatrix}. \tag{11}$$

To be clear $\xi_{i1}(k)=\beta_i(k)$ and $\xi_{i2}(k)=y(k)$. $\xi_{i1}{}^{(-1)}$ is the elementwise inverse of ξ_{i1}, $\beta_i(k)\in(\mathbf{R}\setminus\{0\})^n$, is random vector, $g(k)\in\mathbf{R}_+$ is gain and $c\in\mathbf{R}_+$ is a constant gain.

Here, we provide the steps on how this broadcast control can be implemented:

(i) First, the initial positions of the agents must be predefined.

(ii) Then the sensor will encode the positions of all the agents.

(iii) This information will be sent to the global controller so that it can process the information according to the given objective function in (5). In this paper, a new objective function has been addressed for the assembly task where a collision avoidance has been introduced into the objective function.

(iv) Next, the processed information (scalar value) computed from the global controller will be sent to the local controllers embedded in each agent and the local controllers will store the value at this stage.

(v) After that, the agents will perform random movement and steps i-iv are repeated so that at the second stage the agents can perform deterministic movement.

(vi) Lastly, steps i-v are repeated until the system satisfies (8).

For convergence of the agents, below are some conditions that the broadcast controller $(L_1,L_2,\ldots,L_N,G)$ needs to satisfy:

(C1) $g(k)>0$, $c(k)>0$ for all $k\in\mathbf{Z}_{0+}$, $\lim_{k\to\infty} g(k)=0$, $\sum_{k=0}^{\infty} g(k)=\infty$.

(C2) $\beta_i(k)$ are random numbers drawn from the Bernoulli distribution given by (12).

$$\begin{cases}\Pr(\beta_{ij}(k)=1)=0.5,\\ \Pr(\beta_{ij}(k)=-1)=0.5.\end{cases} \tag{12}$$

IV. Numerical Simulation

Consider the broadcast control system Σ in Figure 2 for *N*:=10 and *n*:=2. In this simulation, we will show the difference between convergence of the agents for the assembly task with and without the collision avoidance. For the assembly task only, the objective function considered is $J(x)=\sum_{i=1}^{N}\sum_{j=1}^{N}\|x_i-x_j\|$ while for the assembly task with the collision avoidance, the objective function is given in (5). In this simulation, we used a broadcast controller $(L_1,L_2,\ldots,L_N,G)$ given by (2), (3), (9), and (10). But the gains for the L_i are adopted from [8] which have been proven to achieve fast convergence. The gains for assembly task alone is set as (13).

$$g(k):=\frac{2.2}{\left(\dfrac{k-1}{2}+450\right)^{0.95}},\, c:=1, \tag{13}$$

while the gains and parameters for assembly task with the collision avoidance are set as (14).

$$g(k):=\frac{3.2}{\left(\dfrac{k-1}{2}+860\right)^{0.96}},\, c:=1, \tag{14}$$
$$\lambda:=30,\ m:=1,\ r:=3.$$

Figures 3 and 4 show the snapshots of the initial positions of the agents $x_i(0)$ and the positions $x_i(k)(i=1,2,\ldots,N)$ at $k=(0,500,\ldots,2000)$, respectively, for assembly task. Note that, the circles in grey represent the agents $x_i(k)$. Figure 5 shows the performance index, which indicates the performance degree of the achieving assembly task. In the figure, it is shown that the broadcast controller successfully minimizes the assembly task and it is shown that the agents start to assemble during iteration $k = 350$.

Figure 6 shows the position of the agents $x_i(k)(i = 1,2,..., N)$ at $k = (0,500,...,2000)$ for the assembly task with collision avoidance. In order to prove that agents are not collide when performing the assembling task, the analysis of mean distances between agent i and neighbor agent j are tabulated in Table 1. It shows that all the mean values are satisfying (7) at every iteration k=500, k=1000, k=1500, and k=2000. Meanwhile, The performance index of the assembly task with collision avoidance is shown in Figure 7. As can be seen in the figure, the values of the objective function $J(x(k))$ become large at certain iteration k . This is because, if the agents are reaching certain distance together, the system will increase the value of the objective function and this will make the agents to move away from each other.

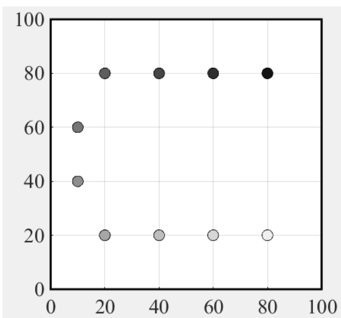
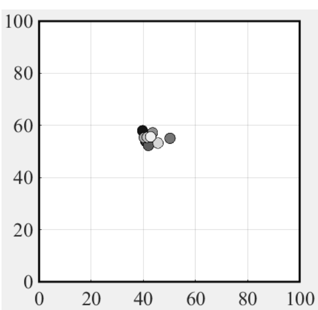
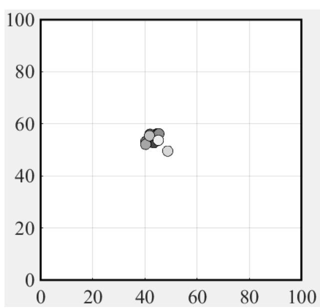


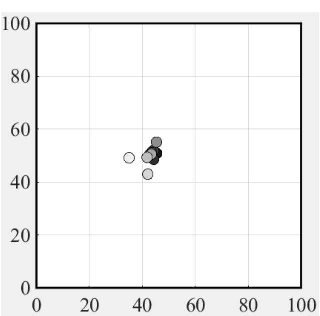
Figure 3: Initial formation $x_i(0)$



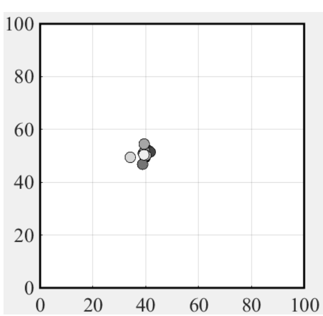
(a) k=500



(b) k=1000



(c) k=1500



(d) k=2000

Figure 4: Positions of agents (Assembly task)

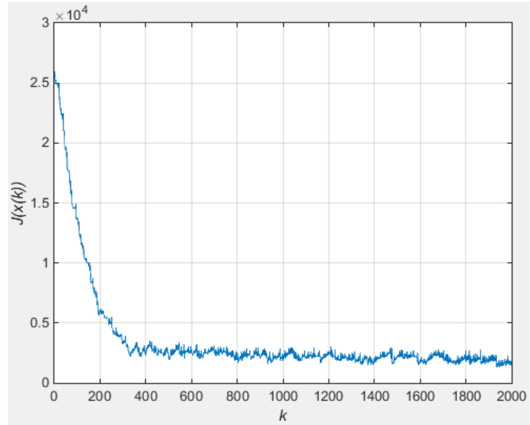
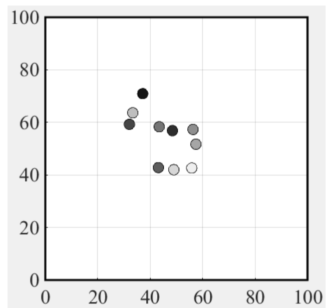
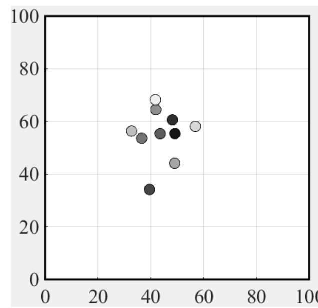


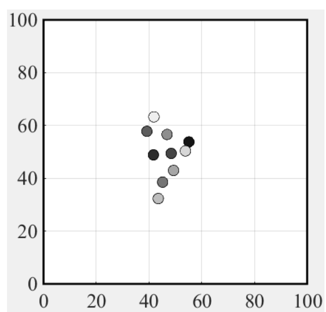
Figure 5: Performance index (Assembly task)



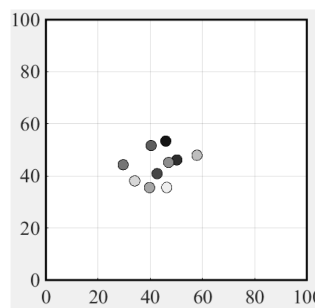
(a) k=500



(b) k=1000



(c) k=1500



(d) k=2000

Figure 6: Positions of agents (Assembly with collision avoidance tasks)

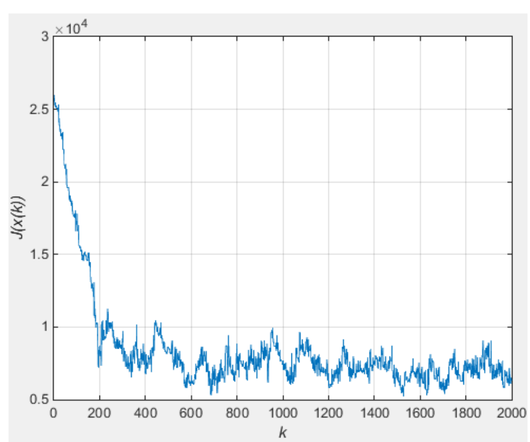


Figure 7: Performance index (Assembly with collision avoidance tasks)

Table 1
Mean of Distance between Agent *i* and Neighbor Agent *j*

Agent *i* and, *j*=*i*+1: *N*	mean($d(\Delta x_{ij})$) at *k*=500	mean($d(\Delta x_{ij})$) at *k*=1000	mean($d(\Delta x_{ij})$) at *k*=1500	mean($d(\Delta x_{ij})$) at *k*=2000
1	-496.67	-150.40	-189.38	-183.33
2	-137.94	-170.19	-99.20	-131.88
3	-320.29	-448.65	-89.77	-72.04
4	-158.42	-71.51	-170.59	-130.17
5	-119.01	-104.61	-119.84	-166.45
6	-106.99	-86.03	-91.27	-54.80
7	-94.74	-128.23	-66.27	-53.98
8	-164.31	-79.46	-137.37	-93.47
9	-3.82	-32.23	-30.02	-14.76

V. CONCLUSION

A broadcast control framework for assembly and collision avoidance task has been addressed. It is shown and validated that the broadcast controller is capable to control the agents in achieving the assembly task. Although the broadcast control framework is shown to successfully handling the assembly task, it is also needs to be included with the collision avoidance for practical use. In this paper, the development of the controller for the assembly task with collision avoidance is still in the early stage and some improvements need to be done for example smooth value of the objective function. This is to avoid the system from increasing the value of the objective function when the agents are already in their optimal positions. For this issue, one may explore this problem for example introducing gain in the global controllers as the tuning gain in the local controller is still not giving a better result.

ACKNOWLEDGMENT

The authors would like to thank Centre for Artificial Intelligence and Robotics (CAIRO), Universiti Teknologi Malaysia for funding this research project.

REFERENCES

[1] S. Azuma, R. Yoshimura, and T. Sugie, "Broadcast control of multi-agent systems," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 3590-3595.

[2] S.-i. Azuma, R. Yoshimura, and T. Sugie, "Broadcast control of multi-agent systems," *Automatica*, vol. 49, pp. 2307-2316, 2013.

[3] M. Ji, S.-i. Azuma, and M. B. Egerstedt, "Role-assignment in multi-agent coordination," 2006.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2002, pp. 1327-1332.

[5] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *Automatic Control, IEEE Transactions on*, vol. 37, pp. 332-341, 1992.

[6] Y. Tanaka, S.-i. Azuma, and T. Sugie, "Broadcast Control of Multi-Agent Systems with Quantized Measurements," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* vol. 97, pp. 830-839, 2014.

[7] S.-i. Azuma, Y. Tanaka, and T. Sugie, "Multi-agent consensus under a communication–broadcast mixed environment," *International Journal of Control,* vol. 87, pp. 1103-1116, 2014.

[8] Y. Tanaka, S.-I. Azuma, and T. Sugie, "Broadcast Control with Constant-distance Random Movement," *Transactions of the Society of Instrument and Control Engineers,* vol. 50, pp. 139-146, 2014.

[9] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, pp. 245-252, 2013.

[10] M.-G. Yoon, J.-H. Moon, and T. K. Ha, "Broadcasting Stabilization for Dynamical Multi-Agent Systems," *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering,* vol. 8, pp. 879-882, 2014.

[11] B. Zhu, L. Xie, and D. Han, "Recent Developments in Control and Optimization of Swarm Systems: A Brief Survey."

[12] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, pp. 2406-2411, 2009.