Multi-robot Systems Using Broadcast Control Framework with Visual-Based Tracking

Bhuvendhraa Rudrusamy
School of Engineering & Physical Sciences
Heriot-Watt University Malaysia
Putrajaya, Malaysia
b.rudrusamy@hw.ac.uk

Chun Wei Tan
School of Engineering & Physical Sciences
Heriot-Watt University Malaysia
Putrajaya, Malaysia
ct100@hw.ac.uk

Mun Ping Yang
School of Engineering & Physical Sciences
Heriot-Watt University Malaysia
Putrajaya, Malaysia
my32@hw.ac.uk

Rohindaran Ramunathan
School of Engineering & Physical Sciences
Heriot-Watt University Malaysia
Putrajaya, Malaysia
rqr2000@hw.ac.uk

Yu Jin Ng
School of Engineering & Physical Sciences
Heriot-Watt University Malaysia
Putrajaya, Malaysia
yn13@hw.ac.uk

Florence Chiao Mei Choong
School of Engineering & Physical Sciences
Heriot-Watt University Malaysia
Putrajaya, Malaysia
f.choong@hw.ac.uk

Mohamad Hafizulazwan Mohamad Nor
Technology Disciplines
Petronas Research Sdn Bhd
Bangi, Selangor, Malaysia
mohamadhafizula.moha@petronas.com.my

2022 7th International Conference on Mechanical Engineering and Robotics Research (ICMERR) | 978-1-6654-9051-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMERR56497.2022.10097802

***Abstract*—The multi-robot application is becoming more critical in achieving complex collaborative tasks. This requires an efficient multi-agent robot system that supports distributed tasks, time and energy consumption. One of the developing control methods is the broadcast control framework. We found that variants of broadcast control techniques were validated on simulation only and lacked experimental work. Therefore, this paper focus on the experimental investigation of a variant of the broadcast control framework. To create a similar environment to the application, we developed the system using Raspberry Pi, AlphaBots, a webcam, and Bluetooth as the central controller, robots, visual tracking, and wireless communication. We successfully validated the system in an indoor environment. The experimental finding shows the travelling trajectory of the robots differs from the MATLAB simulation results. Nonetheless, the broadcast control framework system was able to achieve the final target accurately. This paper provides implementation details and challenges for future research and development of a broadcast control framework. We propose to include obstacle avoidance in robots as a new study direction.**

Keywords—broadcast control, multi-agent system, multi-robot system, simulated and application, visual tracking

I. INTRODUCTION

In the era of Industrial Revolution 4.0, interconnectivity and automation [1] have become essential in enabling the ecosystem. Technologies such as advanced robotics with artificial intelligence transform physical and biological information into digital information [2], shifting the operation of the supply chain and production line, including automation of machine-to-machine communication (M2M) via the internet of things(IoT) [3]. Such implementation drastically improves automation, communication, and self-monitoring, minimising human input.

The increase in automation demands more robots with a better communication system. Multi-robot systems are much more effective than individual robotic systems, where a broadcast control framework solves multi-robot coordination problems. For example, the broadcast control framework encourages the merging of vehicles from various lanes on the highway can be connected and controlled [4], reducing traffic accidents. Another broadcast control application supports sustainability in smart cities, such as surveillance radar systems [5] and aerial vehicles [6].

However, to our best knowledge, most of the past work focuses on simulation, and few implemented it as proof-of-concept (POC), validating the practical performance. To prove the theoretical broadcast control framework [7] for a coordination problem, we implemented it on a multi-robot system using the following hardware: (i) Raspberry-PI 4 as master controller, (ii) Arduino robots as multi-agent, (iii) Bluetooth as means of wireless communication, and (iv) a single RGB web camera as visual tracking of robot movement or as feedback of a system. The POC experiment was successfully conducted indoor environment.

II. PREVIOUS WORK

A. Broadcast Control Framework

Broadcast control framework has been a hot topic among researchers as benefits can be achieved using multi-agent robots, such as load-sharing, planetary exploration, inspection in a hazardous area, and swam robot application related to a cooperative task. Inspired by the vast number of sarcomeres as a functional unit in skeletal muscle that works together to carry out human movement, Ueda, et al. [8] first proposed broadcast control for distributed stochastic cellular actuator system that has many units. In that system, the central controller sends a command to individual units. Each individual unit responds with a stochastic decision and returns to the central controller, aggregating the units' error to the reference signal. This can be viewed as a global and local controller problem by converging cost functions deterministically. The simulation results proved the ensembled cellular units were able to respond within the given trajectory. Since then, many researchers have contributed to the development of the broadcast control framework, and the following are recent advancements.

Sariff and Ismail [9] proposed event-based triggered broadcast control into a simultaneous perturbation stochastic algorithm to reduce the broadcast control energy considering optimal iteration and trajectory. The findings showed that the proposed method is robust and performs better than traditional sampling methods. In addition, the number of channels has been better preserved compared to its predecessors.

Darmaraju, et al. [10] proposed a multi-step broadcast control system in deploying a group of autonomous mobile agents. This method overcomes the single-step global perturbation resulting in sub-optimal performance, where the weighted average was used to compensate for the immediate steps for faster convergence compared to non-weighted perturbation. At the same time, the final error rate remains the same.

Segall and Bruckstein [11] considered the agents' bearing and distance (velocity) to guide and lead the agents toward the convergence of objective functions for swarm mobile agent applications. The direction of agents was obtained from the compass, where the velocity determines the leading agent arbitrarily.

Many of these proposals are conceptual and were verified on a simulation basis. This paper investigates the principles of the broadcast control framework in a proof-of-concept (POC) fashion and compares it to the simulation outcome. To avoid the convergence issue, we implemented a variation of the broadcast control framework proposed by Mohamad Nor, et al. [7] that solves the instability issue. Hence, it prevents the robots from drifting away from their desired goal and being unable to converge to the objective function.

B. Localization

Localisation is an essential issue for mobile robots to determine their position before carrying out any task or instructions, as well as keeping track of their trajectory. In terms of global localisation, the system must determine the position of a mobile robot within its environment without any prior assumption of its position. Hence, the system works by preparing the environment, dependency on certain features of the robot or environment, high-quality sensors and computing hardware requirements. Robots within the map are usually designed with certain characteristics for a known environment. In such a situation, static cameras mounted within the map simplify the complexity of localisation.

Scales, et al. [12] proposed visual-based localisation using a camera to determine the position of the robot while navigating the robot to the desired position visually. In this application, a ceiling camera was implemented to localise the robot activities, where a neural network was used to correspond between robot activity and environment. However, the appearance-based method for multi-robot localisation fails for large environments due to the camera viewpoint. Therefore, Zhou, et al. [13] proposed a semantic histogram-based graph matching method for both homogeneous and heterogeneous robots. The experimental findings showed that the proposed method was robust to viewpoint variation that can accurately localise multi-robots in large environments.

For competitive environments such as soccer robot competition (RoboCup), Ma, et al. [14] used the omnidirectional camera for localisation. The visual-based feedback was critical for robots to estimate their position, other robot's positions and goal position so the strategy could be decided. The experimental result showed that visual-based localisation was sufficient for RoboCup soccer games.

In improving localisation accuracy, Thivanka Perera, et al. [15] showed that a visual-based sensor (fisheye camera) with laser scanners could be employed for accurate localisation.

For implementing the visual-based localisation, image pre-processing is needed to remove the background noise before determining the position of the mobile robot. Thereafter, the coordinates of the robot can be calculated based on grid points (x,y) to obtain the position of the robot. Depending on the environment, a unique algorithm will be developed to extract feature information to create the robot map [16]. In addition, camera calibration is also required to map the visual for localisation. For example, Se, et al. [16] and Li, et al. [17] used a webcam for indoor localisation. In addition to image pre-processing to remove the background, the image (x,y) grid position was mapped to the robot position (or distance). Also, comparing the mapping principle with actual location via experiment reduces the mapping formulation assumptions, simple and effective method, and reliable approach using relatively low-resolution and low-cost camera. To avoid camera calibration, Yasuda, et al. [18] proposed stereo camera to obtain three dimensional coordinates of the robot. However, the stereo camera solution was costly compared to web camera.

Deshpande R and Nair A [19] proposed colour and shape detection method for differentiating multiple robots within an environment, which was found to be a faster and easier way to detect. However, a limited number of objects can be placed within the environment as there must be a significant difference between the object and the background.

There are more advances in object detection and tracking. Most methods [12, 20, 21] use the artificial intelligence method, which requires dataset and training for accurate detection or tracking and requires high computing power. For this work, we found it was not necessary.

III. METHODOLOGY

A. System Overview

This investigation aims to demonstrate the principle of the broadcast control framework, and an overview of the proposed multi-robot broadcast-controller system is shown in Fig. 1. The system has Raspberry Pi as the main controller and is connected to an LCD screen as a user display interface. Two (2) AlphaBot2 were used as robots, each with Arduino Uno as a local controller. A single central static Logitech C920 Full HD webcam was used as top-down visual feedback for the system. The main controller broadcasts the commands to robots via Bluetooth wireless protocol. The main controller, camera, and screen are wired.

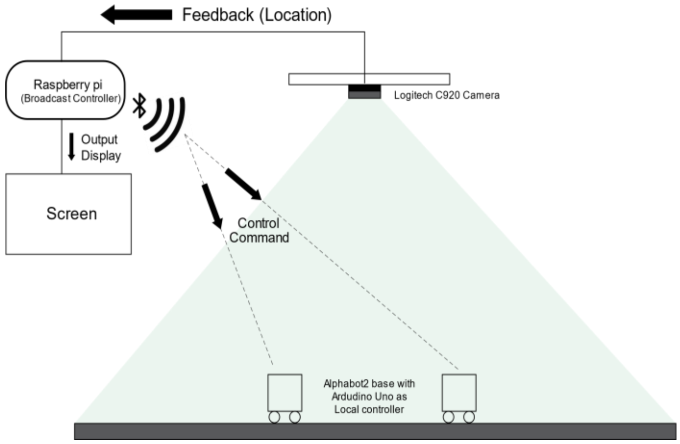


Fig. 1. An overview of a multi-robot broadcast-controlled system.

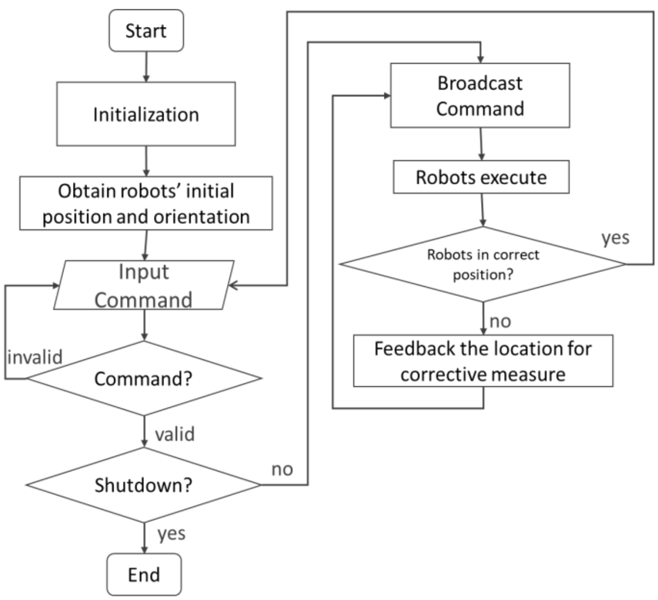


Fig. 2. Flowchart of a multi-robot broadcast-controlled system.

Fig. 2 shows the multi-robot broadcast-controlled system flowchart. First, the system will be initialised to ensure the components are aligned for operation, such as camera connection, Bluetooth connection and robot detection. After that, the system will capture an image to obtain the robot's initial position and orientation and then wait for the user's command. The system will be turned off if it receives a "shutdown" command from the user; otherwise, it shall wait for the broadcast command.

For broadcast command, the user shall provide the robot's desired position as input. The broadcast control framework shall compute the input and cast the command to robots via Bluetooth. Then, each robot shall process the broadcast input and execute the command. At the same time, the central camera monitors the robot's movement. The system acquires images at the appropriate interval to sense the new robot's position. If the robots are not within the acceptable position range, corrective measures shall be taken by the control framework. Otherwise, the system shall return waiting for the following user input.

B. *Broadcast Control*

A broadcast control can be described as guiding a multiple-agent system in a single instruction with zero communication within robots. Hence, a typical broadcast control system has a global controller that broadcasts commands to all the robots. Each robot then processes the instruction, and the local controller reacts accordingly [22].

In this work, we used a modified broadcast control framework with a norm-limited update vector [7], as shown in Fig. 3. This investigation aims to study the modified broadcast control framework behaviour based on simultaneous perturbation stochastic approximation (SPSA) with norm-limited update vector as a practical solution to various types of model-free optimisation issues. From past simulation work [7], the proposed modified broadcast control framework achieved a motion coordination task with a probability of 1, which makes it a perfect candidate for experimental work.

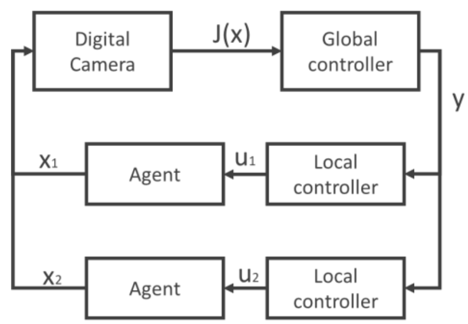
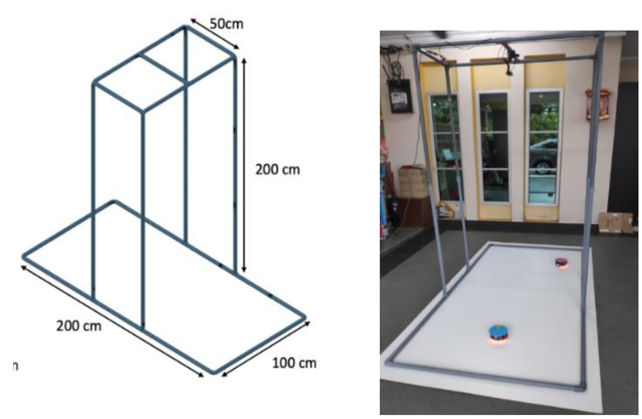


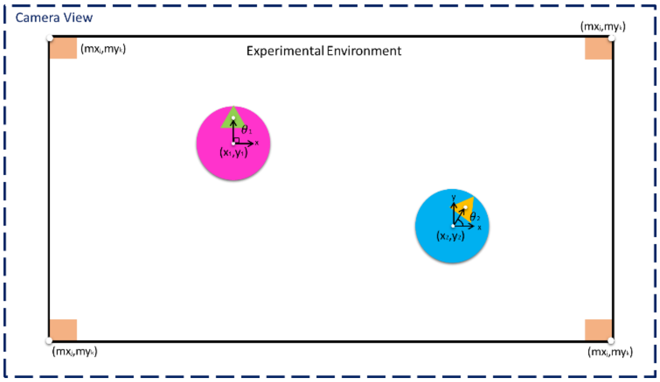
Fig. 3. Flowchart Broadcast Control Framework [7].

C. *Robot Detection & Localization*

A web-based Logitech C920 Full HD camera was used as a visual tracking mechanism for the broadcast control framework. As Bluetooth was used as a wireless communication peripheral between the central controller and the robots, an indoor experiment was conducted with approximately 2m length and 1m width area. Since the webcam has a horizontal field of view of 78°, it must be placed at least 1.23m away from the 2m by 1m work area, as shown in Fig. 4. A simple frame was built to place the camera above the investigation area using PVC pipes to map and coordinate the robots.



(a) (b)



(c)

Fig. 4. Experimental Environment. (a) CAD layout, and (b) actual implementation. (c) Floor layout where the robot is round in shape with colour. In addition, an arrow is placed on the robot for orientation detection.

Since the setup was done indoors, proper lighting condition was required for the investigation environment to reduce shadow or illumination effect. As it is a basic robot, the floor surface must be flat to reduce robot navigation errors. Although the broadcast control framework is done in a closed-loop fashion using visual tracking, such restriction would eliminate the external effect on this investigation, so the performance of the framework can be measured accurately. Referring to Fig. 4 (c), the camera view is placed broader than the experimental environment to reduce the bending distortion due to the camera or lens effect.

The robots were differentiated with different colours to ease navigation using the existing visual camera method. In addition, a simple arrow was marked on the robots to indicate the robot's movement direction. Alternatively, one can consider a hardware upgrade to identify the robot's orientation or localisation.

Referring to Fig. 4 (c), let the angle of an arrow on the robot (or robot orientation) be denoted θ_i where $i = 1, 2$, since we used two robots for this investigation. Then, a simple colour-based detection using the computer vision method was used to obtain the robot position, where the center of the robot is denoted x_i, y_i, where $i = 1, 2$, indicates the position of each robot.

The corners of the floorplan were also detected using computer vision to map the robot position and let it be denoted as mx_j, my_k, where $j, k = 1, 2$ indicates the location of the corners of the investigation area. Hence, the robot position is mapped from the vision (grid or image pixel) to the length (meter) domain.

D. Colour-Based Detection

Computer vision has endured many advancements. In this investigation, we used colour for object detection. Although the steps seem simple, one needs to understand the fundamentals to do the more challenging task using computer vision.

The most common way to represent colour is by using three primary illumination colours that are red, green, and blue (RGB). However, when each RGB component interferes with the other, the chrominance and luminance are mixed, resulting in sensitivity to illumination variation. Another colour space that is least sensitive to illumination is Hue, Saturation and Value (HSV). As the components in HSV colour space are independent, it is more suitable for colour-based detection; and is implemented in this investigation. For efficiency, typically stored as a 24-bits number using 8-bits for each colour component (0 to 255). The colour can be detected accurately by determining a range of HSV values. Therefore, the default RGB image acquired from the webcam is now converted to HSV colour space. In addition, to reduce the image noise, image pre-processing techniques such as filters, morphological technique, and boundary method is used for image enhancement.

Then, the image is binarised (black or white), where black and white are referred to as the background and object of interest, respectively. To better group the white and black pixels, OpenCV prebuilds algorithms such as 'RETR_TREE' and 'CHAIN_APPROX_NONE' to smoothen the black and white contour.

To find the centroid, area, radius, or such property of the object in the image, prebuild algorithm in OpenCV such as 'Moments' can be used to intensify the weighted mean of pixels to identify such personalities. In this investigation, the centroid of the robots and the arrows are determined using Equation (1), which are in round and triangle shapes, respectively. On the other hand, the corners of the floorplan in rectangle shapes are identified using the 'bounding rectangle', where the exact location is determined using its dimension information.

$$C_x, C_y = \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \tag{1}$$

where C, and M indicates centroid in x- and y-coordinate and zero or first order moment, respectively.

Also, OpenCV 'atan2' prebuild algorithm is used to estimate the angle of the arrow (robot orientation) with respect to the centroid of the object (robot).

E. Robot Movement

A simple algorithm is implemented for robot movements. The algorithm will compute the two actions on the robot, which are rotation and forward movement. The algorithm's output will be the wheel's spinning duration, either rotation or forward motion, to reach their respective location.

For rotation movement, the algorithm calculates the robot rotation time required for the robot to rotate towards the destination, and the angles were obtained earlier from computer vision, as shown in Fig. 5(a). Referring to Fig. 5(b), let the robot's (1) desired position, (2) current position, and (3) angle between the current and desired position be denoted as θ, θ_1, and θ_2, respectively. The angles will be calculated once the user enters the broadcast command in the system using Equations (2) and (3). Note that the sign of the angle shall determine clock- or anticlockwise circular movement.

$$Rotational\ speed = \frac{tangential\ speed}{\pi * axle\ length} \tag{2}$$

$$t_r = \frac{abs(\theta)}{Rotational\ Speed * 360} \tag{3}$$

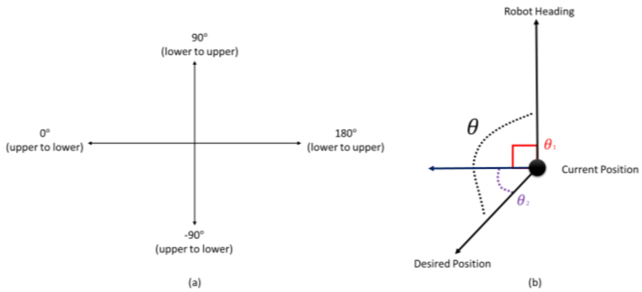


Fig. 5. Rotation movement, (a) angle map (atan2); (b) robot angle orientation.

For forward movement, the algorithm calculates the forward movement time t_f required to reach the destination, which the distance d was obtained earlier from the computer vision. Euclidean distance d is calculated using Equation (4), resulting in the shortest distance, where, x, y and subscript $1, 2$ is in cartesian coordinate, denoting current and destination position.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4}$$

$$t_f = \frac{d}{tangential\ speed} \tag{5}$$

For quantifying the robot's movement, computer vision was used to estimate the distance in meters. The corners were extracted from the image pixel (see Fig. 4 (c)) and mapped to the actual length of the floor. Let the size (length) of the upper corners of the floorplan in the image (pixel) and the actual length of the floor be denoted as a and b, respectively. Obtained the length to pixel (m/px) domain as:

$$ratio = \frac{b}{a} \tag{6}$$

Noted that this is an estimation only, and for accurate measurement, a global position system (GPS) device can be added to the robot at a cost.

IV. Results and Discussion

The broadcast control framework was developed based on the methodology mentioned in this paper. However, before the complete investigation, a few experiments were conducted to set up the environment correctly.

The computer vision method was used for robot detection. Since the colour-based technique was implemented, we applied blue and pink colours with an orange and green arrow on each robot and peach colour for floorplan corners. The experimental finding shows the system can recognise the floorplan, and each robot's position and direction, as shown in Fig. 6. We obtained the HSV threshold for floorplan corners, the blue robot, and the pink robot from (1, 50, 35) to (14, 255, 255), from (131, 70, 70) to (179, 255, 255), and from (90, 100, 100) to (130, 255, 255), respectively, and the experimental outcome is shown in Figure 6(b), (c), and (d). For the objection detection, robot location and direction were obtained using the "Moments" function in the OpenCV library, as shown Fig. 6 (e) and (f), respectively.

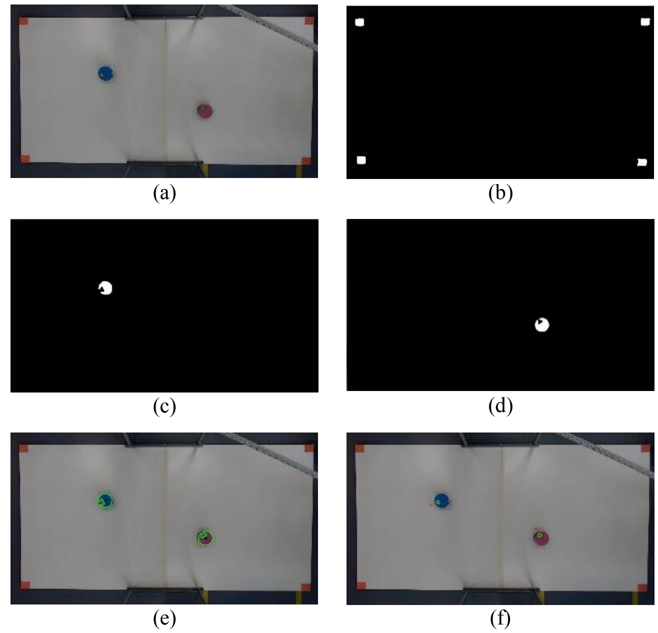


Fig. 6. Object detection using the colour-based method. (a) Image Captured by the system; (b) floorplan corners detection; (c) blue robot detection; (d) pink robot detection; (e) robot localization; and (f) robot direction detection.

Computer vision also was used to obtain the distance of robot movement. However, the experimental finding shows that the corner detection was inaccurate because the fisheyes effect caused the webcam's lens. Camera calibration algorithms and perspective cropping was implemented to minimize the effect. Hence, we obtained the length to pixel domain, as shown in Table I. Few measurements were conducted to verify the accuracy of domain mapping, as shown in Table II. The error measurements ranged from 0.39 cm to 3.43 cm, with an average error of 1.51 cm. The error largely occurs at the corner of the image and reduces towards the centre of the image. Such phenomena were expected due to the webcam lens. Such limitations can be addressed by using high-performance lenses and avoiding using the entire image as the workspace. The inclusion of GPS hardware can improve the deviation. Nonetheless, comparing the robot size with the measurement error, the variation was acceptable as the aims of this investigation were not affected.

The broadcast control framework investigation was conducted upon completing the environment setup. Once the user enters the new robot destination, the system repetitively sends a broadcast command, and it checks the error until it converges to the target, as shown in Fig. 7(a) and (b). To better analyse the behaviour of the broadcast control framework, we sampled the executions by measuring the robot movement. Table III shows the measurement errors of the robots' movement by comparing the actual with expected distance. The experimental finding shows that the error ranged from 0.03 to 6.65 cm, with an average error of 1.93 cm on the x-axis, while 0.11 to 6.45 cm with an average error of 2.02 cm on the y-axis. Furthermore, the error accounted for both forward movement and rotation action, ranging from 2.71 to 24.64 degrees, with an average error of 10.72 degrees.

TABLE I. IMAGE TO LENGTH DOMAIN

Floorplan	Length		Ratio (m/px)
	Pixel (px)	Meter (m)	
Length (x-axis)	1224	2.442	0.001995
Width (y-axis)	601	1.222	0.002023

TABLE II. MAPPING ERROR

Test #	Robot #	Axis	Detected Position (m)	Measured Position (m)	Error (cm)
1	Robot 1	x	1.9536	1.9697	1.61
	(pink)	y	0.9776	1.0012	2.36
2	Robot 2	x	0.2442	0.2099	-3.43
	(blue)	y	0.7332	0.7487	1.55
3	Robot 1	x	1.4797	1.494	1.43
	(pink)	y	0.8381	0.823	-1.51
4	Robot 2	x	0.7327	0.755	2.23
	(blue)	y	0.5102	0.517	0.68
5	Robot 1	x	1.6098	1.635	2.52
	(pink)	y	0.6546	0.645	-0.96
6	Robot 2	x	0.7422	0.759	1.68
	(blue)	y	0.5078	0.512	0.42
7	Robot 1	x	1.4323	1.442	0.97
	(pink)	y	0.7286	0.717	-1.16
8	Robot 2	x	0.7319	0.766	3.41
	(blue)	y	0.4933	0.502	0.87
9	Robot 1	x	1.5706	1.58	0.94
	(pink)	y	0.8239	0.82	-0.39
10	Robot 2	x	0.7147	0.71	-0.47
	(blue)	y	0.4742	0.49	1.58

TABLE III. MEASUREMENT ERRORS OF THE ROBOTS' MOVEMENT

sample #	Robot #	Axis	Desired Distance (m)	Actual Distance (m)	Error (cm)	Rotation Error (deg)	Movement Error (cm)
1	Robot 1	x	-0.4805	-0.4900	0.95	5.26	0.71
	(pink)	y	-0.1687	-0.1631	-0.56		
2	Robot 2	x	0.5049	0.5228	-1.79	-2.71	0.63
	(blue)	y	-0.2609	-0.2385	-2.25		
3	Robot 1	x	0.1916	0.1251	6.65	-6.83	-2.70
	(pink)	y	-0.1564	-0.1814	2.51		
4	Robot 2	x	-0.0050	0.0055	-1.05	8.98	-0.65
	(blue)	y	-0.0112	-0.0018	-0.94		
5	Robot 1	x	-0.1691	-0.1804	1.13	23.13	-1.95
	(pink)	y	0.1314	0.0730	5.84		
6	Robot 2	x	-0.0217	-0.0136	-0.81	12.80	-0.54
	(blue)	y	-0.0147	-0.0158	0.11		
7	Robot 1	x	0.1662	0.1334	3.28	24.64	-0.43
	(pink)	y	0.0328	0.0973	-6.45		
8	Robot 2	x	-0.0263	-0.0200	-0.63	6.12	-0.74
	(blue)	y	-0.0217	-0.0176	-0.41		
9	Robot 1	x	-0.0513	-0.0214	-2.99	7.13	-2.27
	(pink)	y	-0.0373	-0.0347	-0.26		
10	Robot 2	x	-0.0005	-0.0002	-0.03	9.54	0.80
	(blue)	y	0.0003	-0.0086	0.90		

To summarise, the absolute robot movement error ranged from 0.43 to 2.69 cm, with an average error of 1.14cm, considering both forward movement and rotation error. The convergence plot in Fig. 7(c) confirmed that the experimental findings and MATLAB simulation results (travelling trajectory) were different. Although the error occurs due to the combination of computer vision and hardware (robot controller) approximation in the system, the nature of broadcast control convergence behaviour, the closed-loop system is able to achieve the destination successfully, as shown in Fig. 7 (c). This verifies the advantages of using a broadcast control framework for a multi-robot system.

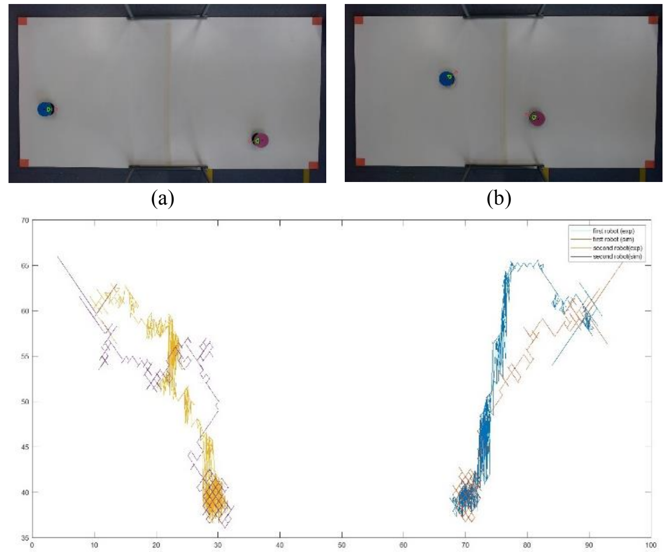


Fig. 7. Results of broadcast control framework. (a) Initial position, (b) final position, and (c) Robot's Heading after execution; (d) Robots' Position after Execution

V. CONCLUSION

This paper presents a laboratory investigation of the behaviour of the broadcast control framework [7]. We used Raspberry Pi, Alphabot2, webcam, and Bluetooth as central controller, robots, visual feedback, and wireless communication for the system, respectively. We implemented this framework in an indoor environment to avoid the external influence of non-constant parameters such as rough terrain for a multi-robot system, inconsistent illumination for visual tracking, etc. We noted several environmental setup challenges that caused differences in travelling trajectory compared to MATLAB simulation. However, the experimental finding shows that the broadcast control framework is able to converge to the target destination successfully. This verifies the advantages of using a broadcast control framework for a multi-robot system. Nonetheless, for an accurate travelling trajectory application, GPS hardware, encoder for robot motors, high-performance lenses, etc., can reduce the differences. For future work, we plan to implement obstacle avoidance in a robot system and consider General Packet Radio Service for wider terrain exploration.

REFERENCES

[1] M. S. R. A. Hamid, N. R. Masrom, and N. A. B. Mazlan, "The key factors of the industrial revolution 4.0 in the Malaysian smart manufacturing context," *International Journal of Asian Business and Information Management,* vol. 13, 2022.

[2] E. O. Pyzer-Knapp, J. W. Pitera, P. W. J. Staar, S. Takeda, T. Laino, D. P. Sanders*, et al.*, "Accelerating materials discovery using artificial intelligence, high performance computing and robotics," *npj Computational Materials,* vol. 8, p. 84, 2022/04/26 2022.

[3] S. Munirathinam, "Chapter Six - Industry 4.0: Industrial Internet of Things (IIOT)," in *Advances in Computers*. vol. 117, P. Raj and P. Evangeline, Eds., ed: Elsevier, 2020, pp. 129-164.

[4] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S. Azuma, "Coordination of Connected Vehicles on Merging Roads Using Pseudo-Perturbation-Based

Broadcast Control," *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, pp. 3496-3512, 2019.

[5] Y. Fujishima and S. Azuma, "Broadcast Control of Radar Surveillance System," in *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2018, pp. 83-88.

[6] S. Darmaraju, M. Shanmugavel, M. A. Samad Kamal, and C. P. Tan, "Area Coverage by a Group of UAVs using the Broadcast Control Framework," *IFAC-PapersOnLine,* vol. 52, pp. 370-375, 2019/01/01/ 2019.

[7] M. H. Mohamad Nor, Z. H. Ismail, and M. A. Ahmad, "Broadcast control of multi-robot systems with norm-limited update vector," *International Journal of Advanced Robotic Systems,* vol. 17, p. 1729881420945958, 2020.

[8] J. Ueda, L. Odhner, and H. H. Asada, "Broadcast Feedback of Stochastic Cellular Actuators Inspired by Biological Muscle Control," *The International Journal of Robotics Research,* vol. 26, pp. 1251-1265, 2007.

[9] N. Sariff and Z. H. Ismail, "Broadcast Event-Triggered Control Scheme for Multi-Agent Rendezvous Problem in a Mixed Communication Environment," *Applied Sciences,* vol. 11, p. 3785, 2021.

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

[11] I. Segall and A. M. Bruckstein, "Broadcast Guidance of Multi-Agent Systems," *International Journal Of Computers Communications & Control,* vol. 17, 2022-01-14 2022.

[12] P. Scales, M. Rimel, and O. Aycard, "Visual-based Global Localization from Ceiling Images using Convolutional Neural Network," presented at the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: VISAPP, 2021.

[13] Z. Zhou, W. Tang, Z. Wang, L. Wang, and R. Zhang, "Multi-robot Real-time Cooperative Localization Based on High-speed Feature Detection and Two-stage Filtering," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2021, pp. 690-696.

[14] Y. Ma, K. Watanabe, H. Kono, and H. Suzuki, "Verification of Robustness against Noise and Moving Speed in Self-localization Method for Soccer Robot," *Journal of Robotics, Networking and Artificial Life,* vol. 8, p. 5, 2021.

[15] R. A. Thivanka Perera, C. Yuan, and P. Stegagno, "A PHD Filter Based Localization System for Robotic Swarms," in *Springer Proceedings in Advanced Robotics*, Cham, 2022, pp. 176-189.

[16] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001, pp. 2051-2058 vol.2.

[17] I.-H. Li, M.-C. Chen, W.-Y. Wang, S.-F. Su, and T.-W. Lai, "Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments," *Sensors,* vol. 14, pp. 2089-2109, 2014.

[18] S. Yasuda, T. Kumagai, and H. Yoshida, "Calibration-free Localization for Mobile Robots Using an External Stereo Camera," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1-6.

[19] Deshpande R and R. A. Nair A, "Color based Object Tracking Robot," *Robot Autom Engineering Journal,* vol. 2, p. 5, 2018.

[20] Ching-Long Shih and Y.-T. Ku, "Image-Based Mobile Robot Guidance System by Using Artificial Ceiling Landmarks," *Journal of Computer and Communications,* vol. 4, p. 14, 2016.

[21] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor Localization of Mobile Robots Through QR Code Detection and Dead Reckoning Data Fusion," *IEEE/ASME Transactions on Mechatronics,* vol. 22, pp. 2588-2599, 2017.

[22] S.-i. Azuma, R. Yoshimura, and T. Sugie, "Broadcast control of multi-agent systems," *Automatica,* vol. 49, pp. 2307-2316, 2013/08/01/ 2013.