Distributed formation control for groups of mobile robots using consensus algorithm

Ryandika Afdila, Fahmi, Arman Sani
Department of Electrical Engineering, Faculty of Engineering, Universitas Sumatera Utara, Medan, Indonesia

Article Info

Article history:

Received Mar 30, 2022
Revised Nov 27, 2022
Accepted Dec 16, 2022

Keywords:

Consensus algorithm
Distributed control
Formation control
Mobile robots
Obstacle avoidance

ABSTRACT

The increasing implementation of autonomous robots in industries and daily life demands the development of a robust control algorithm that enables robots to perform their tasks successfully. Critical tasks such as rescue missions and area exploration require robots to work cooperatively in a formation to accomplish the tasks quickly. Moreover, the existence of obstacles in the environment requires all the robots' to rapidly process environmental changes to maintain the formation pattern. Thus, this paper introduces a distributed robot formation control system using the consensus algorithm that enables a group of robots to establish and maintain formations using only the local information of the robots. Furthermore, an obstacle avoidance algorithm based on the distance and angle between robots and obstacles is introduced to ensure safe navigation for the group of robots. The algorithm's effectiveness is demonstrated by a multi-robot system with randomly generated starting positions and velocities, where it is shown that the robots can agree on the control variables and establish the required formation while also avoiding obstacles in the environment.

Corresponding Author:

Ryandika Afdila
Department of Electrical Engineering, Faculty of Engineering, Universitas Sumatera Utara
Jl. Politeknik, Padang Bulan, Medan Baru, Medan, Indonesia
Email: ryandika@usu.ac.id

1. INTRODUCTION

Formation control systems for groups of robots are generating considerable research attention due to their potential for usage in various disciplines, including land exploration, surveillance, high-precision agriculture, and rescue operations [1]–[3]. The term formation control system refers to the capability of a collection of robots to construct a geometric formation and then travel to a destination while preserving the formation. The purpose of research on the formation control algorithm is to reduce the time and risk for humans in doing time-consuming and hazardous tasks. For example, in area exploration work, a robot's sensors may be focused on a single region while other robots scan other regions by traveling in a formation. As a result, the scan's coverage area is larger and more precise. Additionally, by doing tasks in formation, the robot's time necessary to execute tasks such as search and rescue operations can be reduced [4].

To establish and maintain a specific configuration, each robot in the group must collaborate. A suitable control method for multi-robot formation is indicated to contain the following characteristics [5]: i) high scalability in order to accommodate an increasing number of robots, ii) robust, such that even if one of the robots fails, the formation remains intact, and iii) have a flexible form that allows it to go through passageways and around obstacles in the surroundings. Control strategies for multi-robot formation control can be classified into centralized and distributed control. In a centralized method, a single robot serves as a leader and directs the group's movement. Thus, a centralized computing unit with high computational capabilities is needed to process all the information. On the other hand, the distributed control method only requires the robots' local information, such as velocity, direction, and distance to obstacles, minimizing the resources needed to process the information and negating the need for a central computing station [4], [6].

Several studies have been done on distributed formation control algorithms. A behavior-based formation control system is developed based on the robot's relative location to nearby robots and obstacles [7]. The distance between the robot and the obstacle is used, and the concept of escape angle is introduced to decide when the robot's headings should be adjusted to avoid the obstacle. Another research uses artificial potential field (APF) to control robots' formation [8], [9]. Research by Rezaee and Abdollahi [8] presents a control technique for mobile robots based on the virtual structure where a virtual mobile robot is placed in the middle of the formation and acts as the group leader. Drawing inspiration from electric charges, the virtual mobile robot pulls or pushes other robots toward the desired formation configuration. An obstacle avoidance based on the rotating potential field is also introduced, allowing the mobile robots to steer away from obstacles smoothly without getting stuck in local minima. A control system for vehicle platooning and merging is presented using the APF approach [9]. APF is used to pull or push a vehicle from another vehicle. The automobile uses the control system to follow other cars, shorten the space between them, avoid obstacles, and platoon. The simulation findings indicate a consensus on the control variables is achieved after 18 seconds.

According to Savkin *et al.* [10] implements the concept of consensus variables to update a robot's position, linear velocity, and angular velocity based on the adjacent robots' local information. The consensus algorithm finds similar speeds and heading for the robots to establish a particular formation pattern. The robot's maximum linear and angular speeds are restricted to simulate real-world situations. As a result, the robots could form multiple formation patterns in the environment without obstacles. Another work in [11] considers a system with different types of agents. The consensus algorithm is used with the internal model principle to track reference variables responsible for updating control variables and making the formation scalable. Finally, Hasan and Raafat [12] utilizes the particle swarm optimization (PSO) method to speed up the convergence time of the consensus algorithm for feedback coefficient in multi-agent systems. From the experiments, it is shown that the convergence of the feedback coefficient can be achieved in 115–120 iterations by using PSO compared to 500 iterations without using PSO.

While both distributed and centralized control approaches are feasible, distributed control is preferred due to its cheap operational costs, robustness, scalability, and reduced system requirements [4]. As a result, distributed control is utilized in this study to control the robots' formation. Additionally, because several objects in the area might obstruct the robots' movement, the robots require an obstacle avoidance strategy to travel safely to the target in a congested environment. The utilization of obstacle avoidance algorithm in formation control protects the robots from colliding with each other when building the formation and maintain the formation when moving to the target, even if obstacles are present. Sensing modules are used by the robots to locate the presence of the obstacle. The obstacle avoidance algorithm will then advise the robot on the best course of action to avoid the obstacle [13]. This algorithm is divided into reactive control and motion planning. In motion planning, the obstacles and targets positions are known initially; the controller then computes the optimal path towards the target while avoiding an obstacle in their surroundings and orders the robots to move to the target [14]–[16]. On the other hand, using reactive control algorithms, the position of the obstacles is not known in advance; the robot detects them through a sensor as it moves, and the controller promptly directs the robot to avoid them [17]–[19].

This paper applies the algorithm in [10] and adds an obstacle avoidance algorithm based on the distance and angle of the robots toward the obstacles. We assume that the robots have a restricted communication range and that the position and structure of the obstacles are unknown at first and become known when the robot senses them. Moreover, no leader is assigned to the group, and the robots are modeled using the standard mobile robot kinematics equation with certain linear and angular velocities constraints. This work is aimed to develop an algorithm for the formation and navigation of groups of mobile robots used in search and rescue missions.

2. RESEARCH METHOD

2.1. Robot kinematics

Before we can control robots to create the desired pattern, we must first understand their kinematics. In other words, we require knowledge about the robot's movement. A robot may be categorized into two types based on its movement: holonomic and non-holonomic. A holonomic robot can move freely in all directions, such as a robot equipped with a castor wheel. By contrast, a non-holonomic robot, such as mobile robots move using velocity and angular velocity. This implies that the robot will need to speed and change its direction to approach the target. The mobile robot employed in [10] has the motion as (1)-(3):

$$\dot{x}_i(t) = v_i(t)\cos(\theta_i(t)) \tag{1}$$

$$\dot{y}_i(t) = v_i(t)\, sin(\theta_i(t)) \tag{2}$$

$$\dot{\theta}_i(t) = \omega_i(t) \tag{3}$$

$\dot{x}_i(t)$ and $\dot{y}_i(t)$ indicate the robot's coordinate in x-axes and y-axes, respectively, and $\dot{\theta}_i(t)$ represents the robot's orientation, which has a value of [-π, π]. In (1)-(3) demonstrate that the robot's velocity determines its location, but the angular velocity determines how the robot spins. The velocity and angular velocity in these equations are not constrained and may have a significantly large value. However, in real world, mobile robots do not have a high linear velocity, and a high angular velocity would result in a short turning radius, which is unrealistic for an actual mobile robot [6]. Thus, for the robot to accurately portray how a vehicle would operate, some limitations must be included [10]:

$$-\omega^{max} \leq \omega_i(t) \leq \omega^{max}, \ \forall t \geq 0 \tag{4}$$

$$-V^m \leq v_i(t) \leq V^M, \ \forall t \geq 0 \tag{5}$$

Following that, the robots' position and velocities are then converted into two-dimensional vectors, $z_i(t)$ and $V_i(t)$ [10].

$$z_i(t) := \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}, \quad V_i(t) := \begin{pmatrix} v_i(t)cos\big(\theta_i(t)\big) \\ v_i(t)sin\big(\theta_i(t)\big) \end{pmatrix} \tag{6}$$

2.2. Robots communication

The communication channel allows the robot to communicate information about its environment with other robots. However, the range of communication is limited due to energy limitations for transmitting and receiving data. As a result, in multi-robot systems, local information of a robot is passed to its neighbor within the communication range before it can be sent to other robots outside the communication range. The robot interaction within the group can be depicted using graph theory where the robots' locations are represented by the network vertices and the information flow between the robots are indicated by the graph's edges [20]. The graph theory technique has been utilized to preserve the formation in groups of robots exchanging local information and ensure the scalability and stability of the formation [21]–[23].

This work assumes that a wireless communication network with a coverage area of r_c is used by the robots for exchanging local information. The robot can only communicate with other robots inside its coverage area. However, the robot may still get information about robots not covered by the coverage area from another robot within the communication range. The communication issue that arises when various robot systems communicate is illustrated in Figure 1. The undirected graph $\mathcal{G}$(k) can describe communication between robots, with the robots acting as nodes and the communication link between two robots acting as an edge.

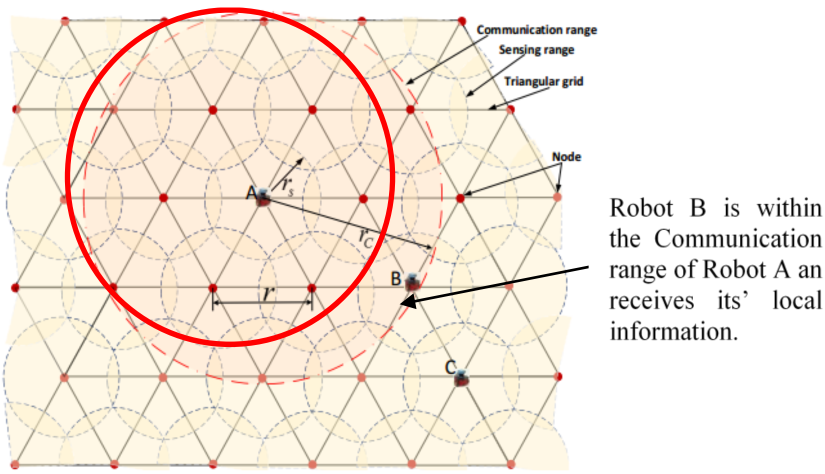


Figure 1. Communication in multi-robot system, adapted from [6]

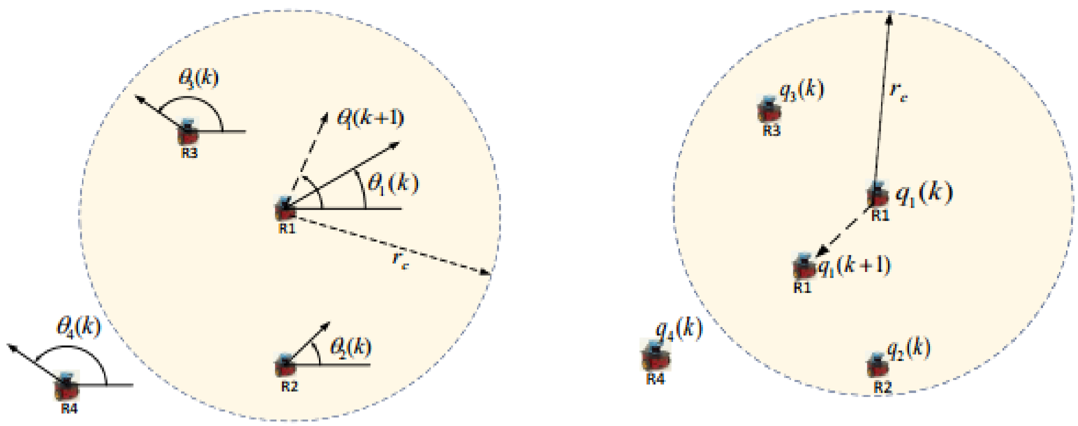
2.3. Formation building algorithm

Formation building for the groups of robots is done using the consensus algorithm which allows robots in multi-robot systems to agree on a specific value of variables required to drive the robots, such as position, velocity, and orientation. The consensus algorithm is utilized in a variety of formation control approaches, from the centralized approach [24], [25] where the leader's control variables become the reference for the follower robots, to the distributed approach [26], [27] where local information of the robots are exchanged to update the groups' control variables. Consensus variables are employed to get the formation's common orientation $\tilde{\theta}_i(k)$, initial coordinates $\tilde{x}_i(k)$ and $\tilde{y}_i(k)$, and linear speeds $\tilde{v}_i(k)$. At time k=0, the robots have varying orientations, velocities, and initial positions. However, as time continues, the consensus variables are updated according to the exchanging information and the robots finally agree on a particular velocity and heading. The following formula is used to update the consensus variables [10]:

$$
\begin{aligned}
\tilde{\theta}_i(k+1) &= \frac{\tilde{\theta}_i(k) + \sum_{j\in\mathcal{N}_i(k)} \tilde{\theta}_j(k)}{1 + |\mathcal{N}_i(k)|} \\
\tilde{x}_i(k+1) &= \frac{x_i(k) + \tilde{x}_i(k) + \sum_{j\in\mathcal{N}_i(k)} (x_j(k) + \tilde{x}_j(k))}{1 + |\mathcal{N}_i(k)|} - x_i(k+1) \\
\tilde{y}_i(k+1) &= \frac{y_i(k) + \tilde{y}_i(k) + \sum_{j\in\mathcal{N}_i(k)} (y_j(k) + \tilde{y}_j(k))}{1 + |\mathcal{N}_i(k)|} - y_i(k+1) \\
\tilde{v}_i(k+1) &= \frac{\tilde{v}_i(k) + \sum_{j\in\mathcal{N}_i(k)} \tilde{v}_j(k)}{1 + |\mathcal{N}_i(k)|}
\end{aligned}
\tag{7}
$$

The equations above are explained in Figures 2(a) and 2(b), where it is assumed that at time k, robot R1 can obtain information from robots R2 and R3 but not from R4. Thus, when robot R1 updates its orientation (θ_1) headings' information of R2 and R3 will be utilized, as shown in Figure 2(a). Hence, the mean of θ_1, θ_2, and θ_3 at time k gives the value of $\theta_1(k+1)$ [6]. Similarly, Figure 2(b) shows the scenario where robot R1 updates its position in accordance with the position of R2 and R3 because R1 was only able to receive information from R2 and R3. Formation pattern can be achieved by introducing a configuration ($\mathcal{C}$) parameter that specifies the coordinates of the desired configuration in the x- and y-axes; $\mathcal{C}$={X_1, X_2, X_3, …, X_n, Y_1, Y_2, Y_3, …, Y_n}. For instance, a rectangular formation with four robots will have the configuration variable of $\mathcal{C}$={0,0,2,2,0,1,0,1}.

Each robot (i) is set to travel toward a fictional target (T_i) rather than the actual target. The fictional target is positioned in front of the robot at a distance specified by $c = \frac{2V^M}{\omega^{max}}$. The robot is set to move toward the fictional target because when the distance between the robot and the actual target decreases, the $\omega_i(t)$ will increase and violate the restriction in (4). Figure 3 illustrates the control algorithm's architecture.



(a) (b)

Figure 2. Consensus variables update strategy (a) headings update and (b) positions update [6]

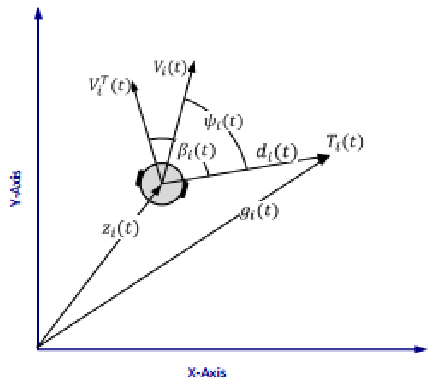


Figure 3. Parameters for the system [6]

The location of the fictional target is denoted by the two-dimensional vector $g_i(t)$ and shown by [10]:

$$g_i^x(t) = \begin{cases} h_i(t) + c & if\ x_i(t) \leq h_i(t) \\ x_i(t) + c & if\ x_i(t) > h_i(t) \end{cases}$$

$$g_i^y(t) = (y_i(t) + \tilde{y}_i(t) + Y_i \tag{8}$$

$$g_i(t) = \begin{pmatrix} g_i^x(t) \\ g_i^y(t) \end{pmatrix}$$

To ensure the mobile robots always navigate toward the target's direction, we created the vector $h_i(t)$ and function f to compare the degree of vectors h_1 and h_2 and decide the robot's moving direction.

$$f(h_1, h_2) = sign\ (a)$$

$$h_i(t) := \left(x_i(t) + \tilde{x}_i(t)\right) + X_i + t\tilde{v}_i(t) \tag{9}$$

Finally, the control algorithm for the system is specified by [10]:

$$v_i(t) = \begin{cases} V^M & if\ x_i(t) \leq h_i(t) \\ V^m & if\ x_i(t) > h_i(t) \end{cases}$$

$$\omega_i(t) = \omega^{max} f\left(V_i(t), d_i(t)\right) \tag{10}$$

Where $f\left(V_i(t), d_i(t)\right)$ is $\psi_i(t)$ which indicates the angle between the robots and targets' headings.

2.3. Obstacle avoidance algorithm

This work uses the technique described in [6], where a range sensor is equipped to a robot to provide the angles and distances between the mobile robots and the obstacles. This information is then utilized to compute the best course of action in avoiding obstacles in the environment. This method considers two scenarios; when the obstacle is perpendicular to the robot and when a curve obstacle is sensed.

Figure 4 illustrates the first scenario. In this scenario, when an obstacle enters the range of the sensor (r_s), the robot quickly adjusts its trajectory to move away from the obstruction, establishing a space between the robot and the obstacle. The robot must maintain this distance to avoid the barrier. The robot's turning radius (R_t) must be evaluated so that the robot may maintain a minimum distance from the barrier. The following equation can be used to compute the robot's maximum turning radius and minimum distance [6]:

$$R_t^{max} = \frac{V^M}{\omega^{min}}$$

$$d_{min} = r_s - R_t^{max} \tag{11}$$

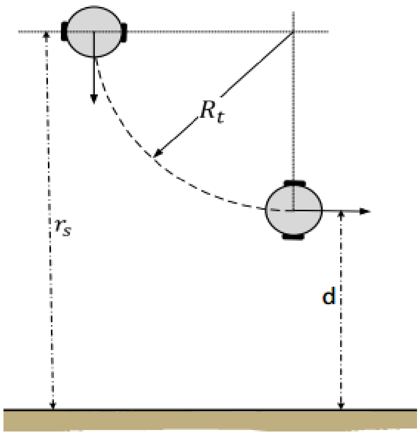


Figure 4. Robot perpendicular to the obstacle [6]

The last case is when the obstacle's curvature is sensed by the robot, as shown in Figure 5. A reference point is the furthest point that the robot can detect. Since the robot is travelling parallel to the obstacle, an angle between the reference point and robot's heading is formed, which is referred to as the avoidance angle (ϕ_0). Because of the obstacles' curvature, the avoiding angle (ϕ) is larger than the avoiding angle when the obstacle is flat (ϕ_0). Thus, to keep the distance, the robot needs to adjust its heading by $\Delta\phi = |\phi - \phi_0|$ towards the obstacle. The distance between the mobile robots and the obstacles is $d_0 = r_s \sin\phi_0$.

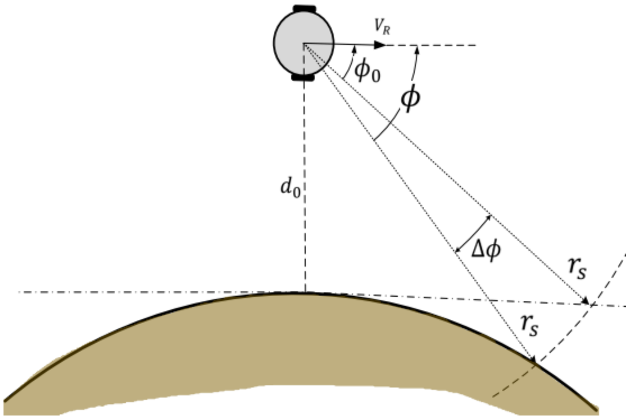


Figure 5. Detecting an arched surface [6]

Thus, the obstacle avoidance method used in this work is given by [6]:

$$\begin{aligned} v_i(t) &= V^M \\ \omega_i(t) &= \omega^{max} sign(\psi_i(t)) \end{aligned} \tag{12}$$

for i=1, 2, 3, …, n, and $\psi_i(t)$ is:

$$\psi_i(t) = \begin{cases} 1 & if\ \phi < \phi_0 \\ 0 & if\ \phi = \phi_0 \\ -1 & if\ \phi > \phi_0 \end{cases} \tag{13}$$

3. RESULTS AND DISCUSSION

In the first experiment, a system where the robots have random initial position and velocities and no obstacle in the environment are considered. The robots have limited range of communication (r_c) and are able to share their local information between each other. Then, the robots must form a formation and navigate towards the target. For this experiment, the desired formations are the square and hexagonal formation. The parameters used in this experiment are shown in Table 1.

Table 1. Control inputs and configuration for the formation

Parameters	Value
Vmax	3 m/s
r_c	3 m
Number of Robots (n)	4 and 6 Robots
Configuration settings	[0.707 0.707; -0.707 0.707; -0.707 -0.707; 0.707 -0.707] (Square formation) $[0\ \ 0;\ 4\ \ 0;\ 6\ \ 2\sqrt{3};\ 4\ \ 4\sqrt{3};\ 0\ \ 4\sqrt{3};\ -2\ \ 2\sqrt{3}]$ (Hexagonal formation)

After the control systems algorithm in (7) is applied, the position and velocities of the robots converge after 120 iterations to the same value as shown in Figure 6 where different velocities of the robots converge to 1.76 m/s. This shows that the consensus algorithm is working. Next, we do the simulation to test the formation-building algorithm based on the parameters from Table 1. We found that 4 robots that initially had random starting positions and speeds could achieve consensus and form a square formation, as shown in Figure 7(a). Furthermore, we also tested the hexagonal formation for 6 robots, as shown in Figure 7(b).

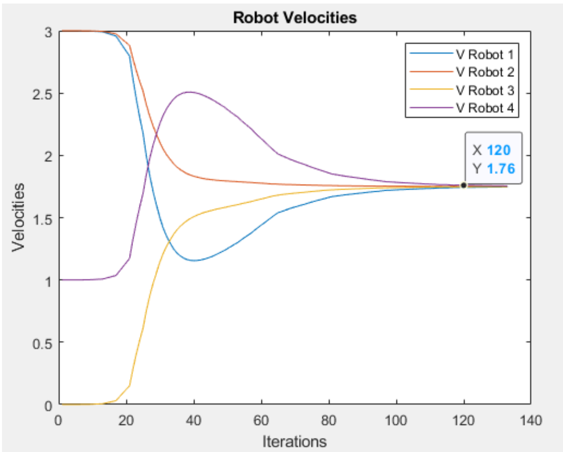
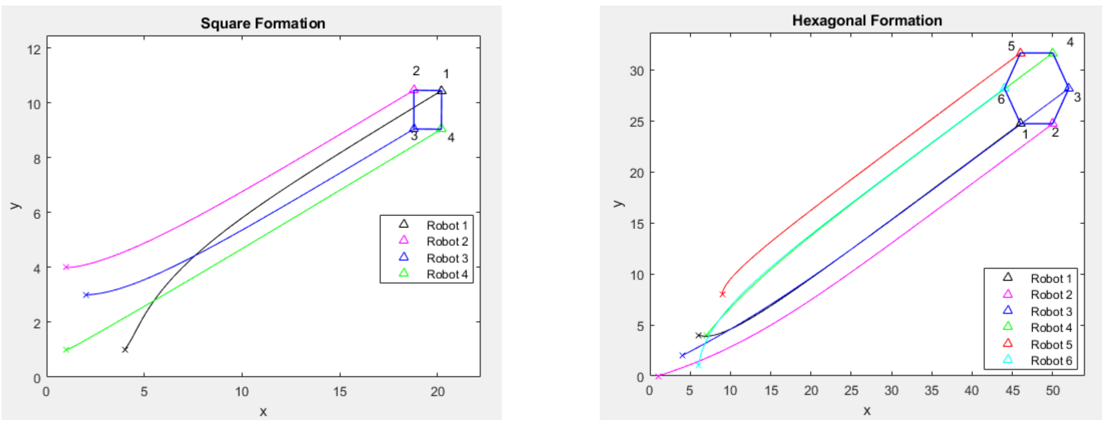


Figure 6. Consensus of the robots' velocities



(a) (b)

Figure 7. Formation building simulation result (a) robots forming square formation and (b) robots forming hexagonal formation

Figures 7(a) and (b) show that the robots can successfully form the desired formation and navigate to the target. From the 4 robots' simulation we found that the initial coordinates of the robots were $\begin{bmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \\ -0.707 & -0.707 \\ 0.707 & -0.707 \end{bmatrix}$ the final coordinates were $\begin{bmatrix} 20.215 & 10.449 \\ 18.792 & 10.462 \\ 18.795 & 9.050 \\ 20.198 & 9.040 \end{bmatrix}$. When the final coordinates of the robots were subtracted, we found that the final coordinates of all the robots in the group are similar $\begin{vmatrix} 19.508 & 9.742 \\ 19.499 & 9.755 \\ 19.502 & 9.757 \\ 19.491 & 9.747 \end{vmatrix}$ which shows that the robot were able to achieve consensus on their position. This is proven by Figure 8 which shows the control input for each robot converge to zero, minimizing the error between the error of the designed control system.

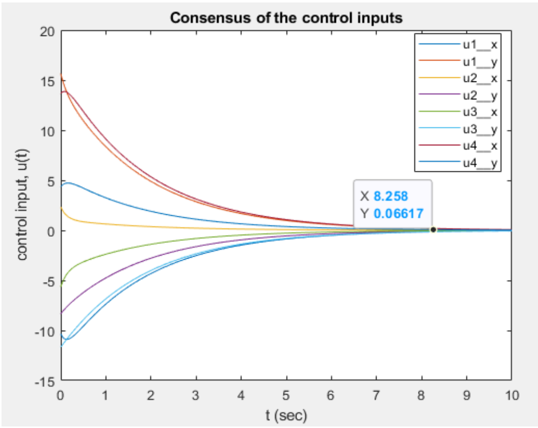
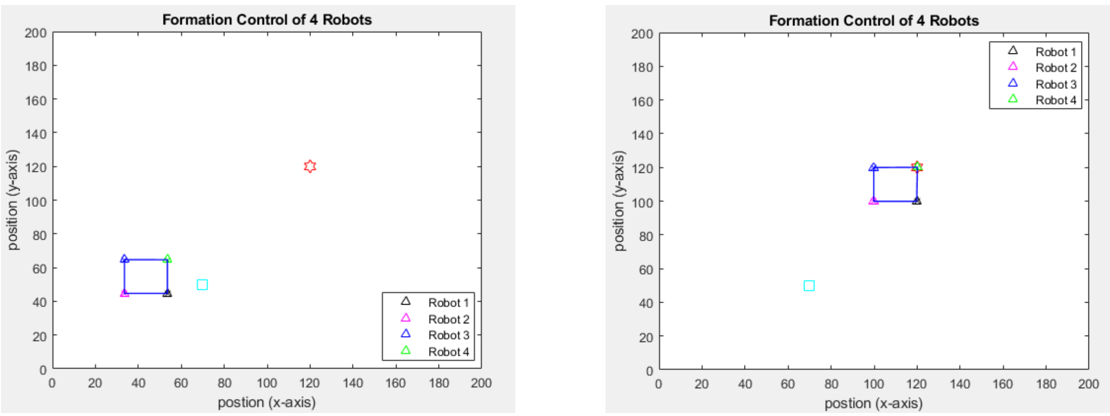


Figure 8. Consensus of the control inputs

Finally, in the second experiment the obstacle avoidance algorithms in (11) and (12) is applied to the control system. We consider a system where robots need to avoid the obstacle while moving through the environment. The robots avoid the obstacle by sensing the distance and the angle between them and the obstacle and then determined whether the sensed distance is within the specified safe distance. The safe distance is used to enable the robots to turn safely if there is an obstacle ahead and it is determined by putting constraint on the robots' turning radius $(\frac{V^M}{\omega_{min}})$. Figure 9(a) shows that the 4 robots forming a square formation can safely avoid the obstacle in the environment while maintaining their formation, and Figure 9(b) shows that the groups of robots arrived at the goal while still in a square formation.



(a) (b)

Figure 9. Obstacle avoidance simulation result (a) robots avoiding obstacle while maintaining formation and (b) robots arrived at goal in formation

At the start of the simulation, robots formed a square formation and navigated towards the goal, indicated by a red-colored star-shaped symbol in Figures 9(a) and 9(b). When an obstacle exists on the robot's path, the robots steer from it while maintaining the formation, as shown in Figure 9(a). Next, the robots continued to move until they reached the goal while still in a square formation, as shown in Figure 9(b). This result proved that the proposed algorithm is suitable for controlling the formation of robots even when obstacles exist in the environment.

4. CONCLUSION

Multiple robot systems have been widely used to complete complex and time-consuming jobs when done by a single robot. To complete the complex work, the robots must be able to coordinate with each other so that they can form a formation that can shorten the time of task execution. The distributed control system is believed to have the best performance to control the behavior of the robot in forming and maintaining the formation due to its robustness, scalability, and flexibility. Therefore, in this study, a formation control system is designed using a consensus algorithm that allows each robot to send their local information and process it without having to have a leader. Based on the simulation results, it is shown that the robots can reach consensus and form the desired formation. In addition, the robots are also able to maintain their formation while avoiding obstacles and navigating to their destination.

ACKNOWLEDGEMENTS

We would like to express our highest gratitude to Universitas Sumatera Utara for supporting this research through TALENTA Research Program under the scheme Penelitian Dasar with contract number 467/UN5.2.3.1/PPM/SPP-TALENTA-USU/2021.

REFERENCES

[1] J. J. Roldán, E. P-Tapia, P. G-Aunon, J. Del Cerro, and A. Barrientos, "Bringing Adaptive and Immersive Interfaces to Real-World Multi-Robot Scenarios: Application to Surveillance and Intervention in Infrastructures," *IEEE Access*, vol. 7, pp. 86319–86335, 2019, doi: 10.1109/ACCESS.2019.2924938.

[2] J. P. Queralta *et al.*, "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020, doi: 10.1109/access.2020.3030190.

[3] Q. Vu, M. Raković, V. Delic, and A. Ronzhin, "Trends in Development of UAV-UGV Cooperation Approaches in Precision Agriculture," in *Lecture Notes in Computer Science*, Springer International Publishing, 2018, pp. 213–221, doi: 10.1007/978-3-319-99582-3_22.

[4] Y. Cao, W. Yu, W. Ren, and G. Chen, "An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, Feb. 2013, doi: 10.1109/tii.2012.2219061.

[5] E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, and M. Dorigo, "Self-organized flocking with a mobile robot swarm: a novel motion control method," *Adaptive Behavior*, vol. 20, no. 6, pp. 460–477, Oct. 2012, doi: 10.1177/1059712312462248.

[6] A. Baranzadeh, "Decentralized Autonomous Navigation Strategies for Multi-Robot Search and Rescue." *arXiv*, May 13, 2016, doi: 10.48550/arXiv.1605.04368.

[7] G. Lee and D. Chwa, "Decentralized behavior-based formation control of multiple robots considering obstacle avoidance," *Intelligent Service Robotics*, vol. 11, no. 1, pp. 127–138, Oct. 2017, doi: 10.1007/s11370-017-0240-y.

[8] H. Rezaee and F. Abdollahi, "A Decentralized Cooperative Control Scheme With Obstacle Avoidance for a Team of Mobile Robots," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 347–354, Jan. 2014, doi: 10.1109/tie.2013.2245612.

[9] E. S-Kazerooni, K. Elferink, J. Ploeg, and H. Nijmeijer, "Multi-objective platoon maneuvering using artificial potential fields," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15006–15011, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.2570.

[10] A. V. Savkin, C. Wang, A. Baranzadeh, Z. Xi, and H. T. Nguyen, "Distributed formation building algorithms for groups of wheeled mobile robots," *Robotics and Autonomous Systems*, vol. 75, pp. 463–474, Jan. 2016, doi: 10.1016/j.robot.2015.08.006.

[11] D. W. Djamari, M. R. Fikri, B. A. Budiman, and F. Triawan, "Formation control of non-identical multi-agent systems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, p. 2721, Jun. 2022, doi: 10.11591/ijece.v12i3.pp2721-2732.

[12] A. M. Hasan and S. M. Raafat, "Optimized formation control of multi-agent system using PSO algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, p. 1591, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1591-1600.

[13] M. A. Kamel and Y. Zhang, "Decentralized leader-follower formation control with obstacle avoidance of multiple unicycle mobile robots," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2015, doi: 10.1109/ccece.2015.7129312.

[14] Y. Zhao, X. Chai, F. Gao, and C. Qi, "Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III," *Robotics and Autonomous Systems*, vol. 103, pp. 199–212, May 2018, doi: 10.1016/j.robot.2018.01.007.

[15] A. Nasrinahar and J. H. Chuah, "Intelligent motion planning of a mobile robot with dynamic obstacle avoidance," *Journal on Vehicle Routing Algorithms*, vol. 1, no. 2–4, pp. 89–104, Jan. 2018, doi: 10.1007/s41604-018-0007-4.

[16] O. Andersson, O. Ljungqvist, M. Tiger, D. Axehill, and F. Heintz, "Receding-Horizon Lattice-Based Motion Planning with Dynamic Obstacle Avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec. 2018, doi: 10.1109/cdc.2018.8618964.

[17] P. Tang, R. Zhang, D. Liu, L. Huang, G. Liu, and T. Deng, "Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle," *Ocean Engineering*, vol. 106, pp. 128–140, Sep. 2015, doi: 10.1016/j.oceaneng.2015.06.055.

[18] C. Berger, P. Rudol, M. Wzorek, and A. Kleiner, “Evaluation of reactive obstacle avoidance algorithms for a quadcopter,” in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov. 2016, doi: 10.1109/icarcv.2016.7838803.

[19] M. Mujahed, D. Fischer, and B. Mertsching, “Tangential Gap Flow (TGF) navigation: A new reactive obstacle avoidance approach for highly cluttered environments,” *Robotics and Autonomous Systems*, vol. 84, pp. 15–30, Oct. 2016, doi: 10.1016/j.robot.2016.07.001.

[20] E. A. Macdonald, “Multi-robot assignment and formation control,” Thesis, Georgia Institute of Technology, 2011. [Online]. Available: https://smartech.gatech.edu/handle/1853/41200. (accessed: Jan. 19, 2023).

[21] G. Wen, H.-T. Zhang, W. Yu, Z. Zuo, and Y. Zhao, “Coordination tracking of multi-agent dynamical systems with general linear node dynamics,” *International Journal of Robust and Nonlinear Control*, 2017, doi: 10.1002/rnc.3753.

[22] S. Shoja, M. Baradarannia, F. Hashemzadeh, M. Badamchizadeh, and P. Bagheri, “Surrounding control of nonlinear multi-agent systems with non-identical agents,” *ISA Transactions*, vol. 70, pp. 219–227, Sep. 2017, doi: 10.1016/j.isatra.2017.06.011.

[23] Y. Ding, X. Wang, Y. Cong, and H. Li, “Scalability Analysis of Algebraic Graph-Based Multi-UAVs Formation Control,” *IEEE Access*, vol. 7, pp. 129719–129733, 2019, doi: 10.1109/access.2019.2938991.

[24] Y. Kuriki and T. Namerikawa, “Consensus-based cooperative formation control with collision avoidance for a multi-UAV system,” in *2014 American Control Conference*, Jun. 2014, doi: 10.1109/acc.2014.6858777.

[25] H. Xiao and C. L. P. Chen, “Leader-Follower Consensus Multi-Robot Formation Control Using Neurodynamic-Optimization-Based Nonlinear Model Predictive Control,” *IEEE Access*, vol. 7, pp. 43581–43590, 2019, doi: 10.1109/access.2019.2907960.

[26] J. A-Mora, E. Montijano, M. Schwager, and D. Rus, “Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, doi: 10.1109/icra.2016.7487747.

[27] L. G-Bonilla, A. Prorok, and V. Kumar, “Formations for Resilient Robot Teams,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 841–848, Apr. 2017, doi: 10.1109/lra.2017.2654550.

BIOGRAPHIES OF AUTHORS

Ryandika Afdila received his bachelor’s degree in electrical engineering from Universitas Sumatera Utara in 2015, and master’s degree in Telecommunication Engineering from University of New South Wales in 2019. Currently he is a lecturer in Department of Electrical Engineering in Universitas Sumatera Utara. His research focus is on wireless communication, IoT, and multi robot systems. He can be contacted at email: ryandika@usu.ac.id.

Fahmi received his B.Sc in Electrical Engineering from ITB Bandung in 2002, and M.Sc in Sensor System Technology FH Karlsruhe in 2005. He gained his Ph.D in Biomedical Engineering at Amsterdam Medical Centre Universiteit Amsterdam in 2015. He worked in several companies in healthcare industry and then joined Universitas Sumatera Utara in 2008 as lecturer. He is currently the Dean of Faculty of Engineering at the same university. His research focus is on sensor system especially in healthcare technology and medical image analysis. He can be contacted at email: fahmimn@usu.ac.id.

Arman Sani received his Bachelor's degree in Electrical Engineering in 1988 from Universitas Sumatera Utara, Medan, Indonesia. He received his Masters's degree from ITB Bandung in 1997 in Telecommunication Engineering. Since 1991, he has worked as a lecturer in the Department of Electrical Engineering Universitas Sumatera Utara. His research interests include telecommunications engineering, power generation, and renewable energy. He can be contacted at email: arman3@usu.ac.id.