



Control of mobile robot formations using A-star algorithm and artificial potential fields

Nelson Luis Manuel *, Nihat İnanç, Mustafa Yasin Erten

*Department of Electrical & Electronics Engineering, Kinkkale University
Kinkkale City, 71450, Turkey*

Received 22 November 2021; Accepted 16 December 2021; Published online 31 December 2021

Abstract

Formations or groups of robots become essential in cases where a single robot is insufficient to satisfy a given task. With an increasingly automated world, studies on various topics related to robotics have been carried out in both the industrial and academic arenas. In this paper, the control of the formation of differential mobile robots based on the leader-follower approach is presented. The leader's movement is based on the least cost path obtained by the A-star algorithm, thus ensuring a safe and shortest possible route for the leader. Follower robots track the leader's position in real time. Based on this information and the desired distance and angle values, the leader robot is followed. To ensure that the followers do not collide with each other and with the obstacles in the environment, a controller based on artificial potential fields is designed. Stability analysis using Lyapunov theory is performed on the linearized model of the system. To verify the implemented technique, a simulator was designed using the MATLAB programming language. Seven experiments are conducted under different conditions to show the performance of the approach. The distance and orientation errors are less than 0.1 meters and 0.1 radians, respectively. Overall, mobile robots are able to reach the goal position and maintaining the desired formation in finite time.

©2021 Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences. This is an open access article under the CC BY-NC-SA license (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Keywords: nonholonomic WMR; leader-follower approach; formation control; A-star algorithm; artificial potential fields.

I. Introduction

The formation of robots is an imitation of a group of behaviors observed in various creatures in the biological world that tend to work cooperatively in order to achieve a common goal [1]. While most industrial robots can only perform specific movements in the workspace, mobile robots can move freely within the predefined workspace [2]. This feature (mobility) expands the possibilities for the application of mobile robots in different environments [3][4][5]. There are certain circumstances in which, due to the complexity of the task, a single robot may not be able to satisfy several tasks simultaneously and/or efficiently. To overcome this, the formation of these robots can be considered [2]. Several advantages are pointed out with the implementation of robotic formations, such as better precision, increased efficiency and robustness against external agents [2]. The methods of controlling the formation of mobile robots can be

divided into three classes: behavioral approach, virtual structure approach, and leader-follower approach [6][7]. In the leader-follower approach, one of the robots plays the role of leader and the rest are followers. The idea behind this approach is that the followers adjust their states according to the position of the leader robot.

There are different leader-follower topologies, for example, approaches that consider multi-leaders, forming a chain (vehicle tracks vehicle), among others [2]. It is possible to find different control techniques for leader-follower robotic formations in the literature. Fuzzy logic controller is presented for the formation of wheeled mobile robots (WMRs) [8]. Input-output (I/O) feedback linearization along with potential fields is used to control the formation of robots in the leader-follower approach [9]. Linear model predictive control (MPC) along with feedback linearization is applied to achieve the task of a formation of WMRs [10]. A non-linear MPC based on neurodynamic-optimization is presented to control the formation of WMRs in a leader-follower architecture [11][12]. A control strategy based on Lyapunov theory and Sliding Mode is applied to

* Corresponding Author. Tel: +90-552-5824071
E-mail address: nelsonluismanuel@gmail.com

multiple WMRs in a leader-follower structure [13]. The leader-follower approach has advantages such as simplicity, no need for global knowledge, and consequently, reduced computerization cost [13].

In this paper, a technique for controlling differential mobile robot formations based on the A-star and artificial potential fields algorithms is presented. The approach used is leader-follower, where the leader receives information about the safest and lowest cost route in an environment with the presence of obstacles from the A-star algorithm. The leader's position and orientation are constantly monitored by the followers. Using a controller based on artificial potential fields, the leader's trajectory is followed, maintaining the desired separation and orientation while avoiding collisions with each other and the obstacles present in the environment.

II. Materials and Methods

A. Kinematic model of differential mobile robot

With the exception of a few, most control techniques depend on the mathematical model of the system to be controlled. Therefore, the first step before deducing the control laws is to define the mathematical model of the differential mobile robot. From Figure 1, applying some trigonometry concepts, the system of equations (1) can be determined.

$${}^I \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

Here v is the linear velocity, ω is the angular velocity, \dot{x} is the velocity with respect to the x-axis, \dot{y} is the velocity with respect to the y-axis, and $\dot{\theta}$ is the bearing angle.

A differential drive wheeled mobile robot is usually composed of two independently controlled wheels and an additional wheel (or caster) for balance purposes. The direction of movement, speed, and orientation of the robot is defined by the speed ratios between its wheels [14]. Figure 1 is a schematic of a differential drive wheeled mobile robot in an inertial frame, where X_I and Y_I represent the inertial (or global) frame axes and X_R and Y_R represent the robot frame (or local frame) axes. The robot's linear velocity $v(t)$ is always parallel to the

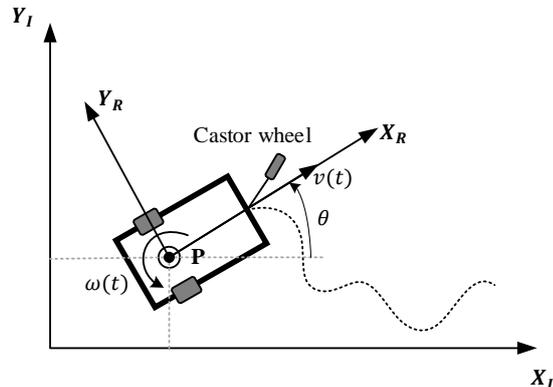


Figure 1. Differential drive mobile robot in an inertial frame

X_R axis of the robot frame due to the nonholonomic constraint. The robot's orientation is measured between the X_I and X_R axes. The change in orientation $\theta(t)$ of the robot represents the angular speed $\omega(t)$. The robot's equivalent position on the inertial frame is shown by the point P. System (1) represents the mathematical model of the differential mobile robot shown in Figure 1.

In general terms, it can be said that to design a controller for this mobile robot system, it is necessary to find the law that provides the relationship between the output vector (the robot position) and the input vector of the system (linear and angular velocities). Such a relationship can be considered as a transformation matrix. The purpose of the controller layout is to find, if it exists, a 2×3 control matrix K , as shown in equation (2) [15].

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}, \quad k_{ij} = k(t, e) \quad (2)$$

Because of the non-linearity of the robot model, the elements (k_{ij}) of the control matrix (K) are time-varying and error dependent, as expressed in equation (2). The matrix K has to be able to generate control signals $v(t)$ and $\omega(t)$ in such a fashion that the error $e(t)$ tends to zero as time t tends to infinity, i.e., equation (3) has to be satisfied.

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \lim_{t \rightarrow \infty} e(t) = 0 \quad (3)$$

Finding the matrix K in (2) is not that simple as it depends on the current value of the error and its elements can change over time. To overcome this, the approach used in [15] will be considered.

A schematic of a robot and a desired position (or goal) is illustrated in Figure 2. The frame of the desired position (X_G, Y_G) is set to match the inertial frame (X_I, Y_I). The Euclidian distance between the robot and the desired position is symbolized by ρ ; Δx and Δy are the differences between the x and y coordinates of the robot's position and the desired position, respectively. The bearing angle which orients the robot towards the goal position is

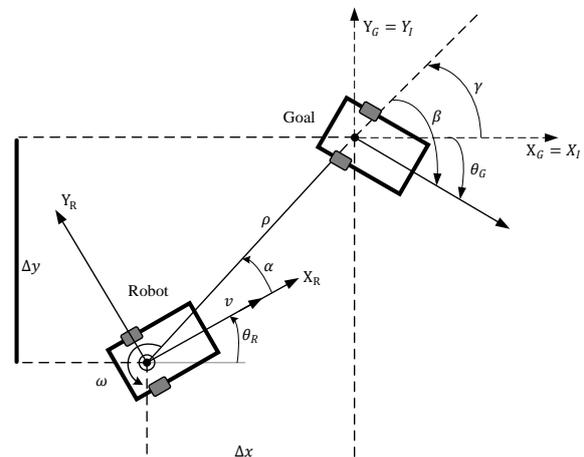


Figure 2. Kinematics of differential mobile robot

indicated by the angle α . The difference between the desired orientation (θ_G) and the robot orientation (θ_R), when $\alpha = 0$ is represented by the angle β . From Figure 2, the following closed-loop parameters can be determined:

$$\begin{aligned} \rho &= \sqrt{\Delta x^2 + \Delta y^2} \\ \alpha &= -\theta_R + \gamma \\ \beta &= \theta_G - \gamma \end{aligned} \quad (4)$$

where $\gamma = \text{atan2}(\Delta y, \Delta x)$, as can be clearly concluded from Figure 2. Unlike the arctangent (atan) function, atan2 function operates in all four quadrants [16]. Since the robot has orientations in all quadrants, the atan2 function is preferred over atan , which only returns results in two quadrants.

To analyze the behavior of the parameters (ρ, α, β) presented in equation (4) as time passes, their time derivatives are computed. Since these parameters are not directly expressed as functions of time, their derivatives are obtained through the chain rule. Therefore, the dynamics of the Euclidean distance (ρ), applying the chain rule, can be expressed as in equation (5).

$$\dot{\rho} = \frac{d\rho}{dt} = \frac{d\rho}{dx_R} \frac{dx_R}{dt} + \frac{d\rho}{dy_R} \frac{dy_R}{dt} + \frac{d\rho}{d\theta_R} \frac{d\theta_R}{dt} \quad (5)$$

The $\dot{\rho}$ on the left side of equation (5) represents the time derivative of the Euclidean distance (ρ), and the addends on the right side are the partial derivatives of ρ with respect to x , y , and θ , respectively. In equations (6), (7), and (8), the derivatives of the addends present on the right side of equation (5) are demonstrated.

$$\frac{d\rho}{dx_R} = -\frac{x_G - x_R}{\sqrt{(x_G - x_R)^2 + (y_G - y_R)^2}} = -\frac{\Delta x}{\rho}; \quad (6)$$

$$\frac{dx_R}{dt} = \dot{x}_R$$

$$\frac{d\rho}{dy_R} = -\frac{y_G - y_R}{\sqrt{(x_G - x_R)^2 + (y_G - y_R)^2}} = -\frac{\Delta y}{\rho}; \quad (7)$$

$$\frac{dy_R}{dt} = \dot{y}_R$$

$$\frac{d\rho}{d\theta_R} = 0; \quad \frac{d\theta_R}{dt} = \dot{\theta}_R \quad (8)$$

where x_R , y_R represent the robot position coordinates on the X_I and Y_I axes and x_G , y_G are the goal position coordinates on the X_I and Y_I axes of the inertial frame. Substituting the results obtained in equations (6), (7), and (8) in equation (5), the equation (9) is derived.

$$\dot{\rho} = -\frac{\Delta x \cdot \dot{x}_R + \Delta y \cdot \dot{y}_R}{\rho} \quad (9)$$

Furthermore, replacing Δx by $\rho \cos(\gamma)$, Δy by $\rho \sin(\gamma)$, x_R by $v \cos(\theta)$, y_R by $v \sin(\theta)$, and then applying the trigonometric identity $\cos(a - b) =$

$\cos(a) \cdot \cos(b) + \sin(a) \cdot \sin(b)$, equation (9) becomes equation (10), which represents the rate of change of the Euclidean distance (ρ) with respect to time.

$$\dot{\rho} = -v \cos(\alpha) \quad (10)$$

Applying the same idea used to obtain the equation (10) to the other two parameters presented in equation (4), the system (11) can be derived.

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos(\alpha) & 0 \\ \frac{\sin(\alpha)}{\rho} & -1 \\ -\frac{\sin(\alpha)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (11)$$

B. Stability analysis using Lyapunov

The feedback system given in (11), presents a discontinuity at $\rho = 0$. To solve this, the inputs (v and ω) are selected as shown in the system (12).

$$\dot{V}(x) = \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos(\alpha) \\ k_\rho \sin(\alpha) - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin(\alpha) \end{bmatrix} \quad (12)$$

where $V(x)$ is the Lyapunov function applied to the differential mobile robot model, k_ρ is the linear velocity adjustment gain, k_α and k_β are the robot orientation adjustment gains.

The system (12) is now continuous at the point $[\rho, \alpha, \beta] = [0, 0, 0]$, which is the point of interest (the goal). However, as this system is not linear, the stability analysis becomes complex. To facilitate this, the analysis is carried out at the equilibrium point of interest. Local stability analysis allows the approximation of the behavior of a non-linear system to an equivalent linear system at the point of interest. Applying Jacobian to the system (12) in the goal position ($[\rho, \alpha, \beta] = [0, 0, 0]$), system (13) is derived. The matrix A of the system (13) is linear and time-invariant, therefore, the stability analysis becomes simplified.

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \underbrace{\begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix}}_A \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} \quad (13)$$

The system is locally exponentially stable, if all eigenvalues of the matrix A presented in (13) have a negative real part. From the matrix A the characteristic polynomial presented in equation (14) is obtained.

$$(\lambda + k_\rho) [\lambda^2 + \lambda(k_\alpha - k_\rho) - k_\rho k_\beta] \quad (14)$$

The λ in equation (14) represents the eigenvalues of matrix A. Therefore, in order for the system to be stable after reaching the destination, the conditions presented in (15) must be met.

$$k_\rho > 0; \quad k_\beta < 0; \quad k_\alpha > k_\rho \quad (15)$$

C. A-star algorithm

A-star, is one of the widely applied computer science algorithms, which takes some inputs (the starting sector and the destination sector), calculates the costs of different possible routes and returns the path with the lowest cost between the starting sector and the destination sector [17]. A-star is based on equation (16):

$$F(x) = g(x) + h(x) \quad (16)$$

where x is a sector on the map, $g(x)$ is the total distance from the starting sector to the current sector x , $h(x)$ is the heuristic function used to calculate the distance from the current position to the desired position, and $F(x)$ is the cost function. Figure 3 is a flowchart that gives a brief explanation of how the applied A-star algorithm works.

D. Artificial potential fields

In artificial potential fields (APF), objects in the workspace are modeled as potential fields that generate forces of attraction and repulsion [18]. Target points are modeled to generate attraction forces, while obstacles are modeled as points that generate repulsion forces to the robot [19]. In this manner, the robot tends to move towards the target while avoiding obstacles present in the environment. The function of the attractive potential field can be

expressed by equation (17).

$$U_{att}(X_R) = \frac{1}{2} K_{att} (X_R - X_D)^2 \quad (17)$$

where $U_{att}(X_R)$ represents the attractive potential field function, K_{att} is a positive constant (defines the influence of the attractive field), X_R is the vector of the robot's position and X_D is the vector of the desired (or goal) position.

The negative gradient of the attractive potential field function results in the corresponding attractive force, as shown in equation (18).

$$\begin{aligned} F_{att} &= -\nabla(U_{att}) = K_{att} (X_D - X_R) \\ &= K_{att} \begin{bmatrix} x_D - x_R \\ y_D - y_R \end{bmatrix} \end{aligned} \quad (18)$$

where x_D and y_D are the desired position (X_D) components and x_R and y_R are the robot position (X_R) components. Similarly, the repulsive potential field function is defined according to equation (19).

$$U_{rep}(X_R) = \begin{cases} \frac{1}{2} K_{rep} \left(\frac{1}{\varepsilon} - \frac{1}{\varepsilon_0} \right)^2, & \varepsilon \leq \varepsilon_0 \\ 0, & \varepsilon > \varepsilon_0 \end{cases} \quad (19)$$

where $U_{rep}(X_R)$ is the repulsive potential field

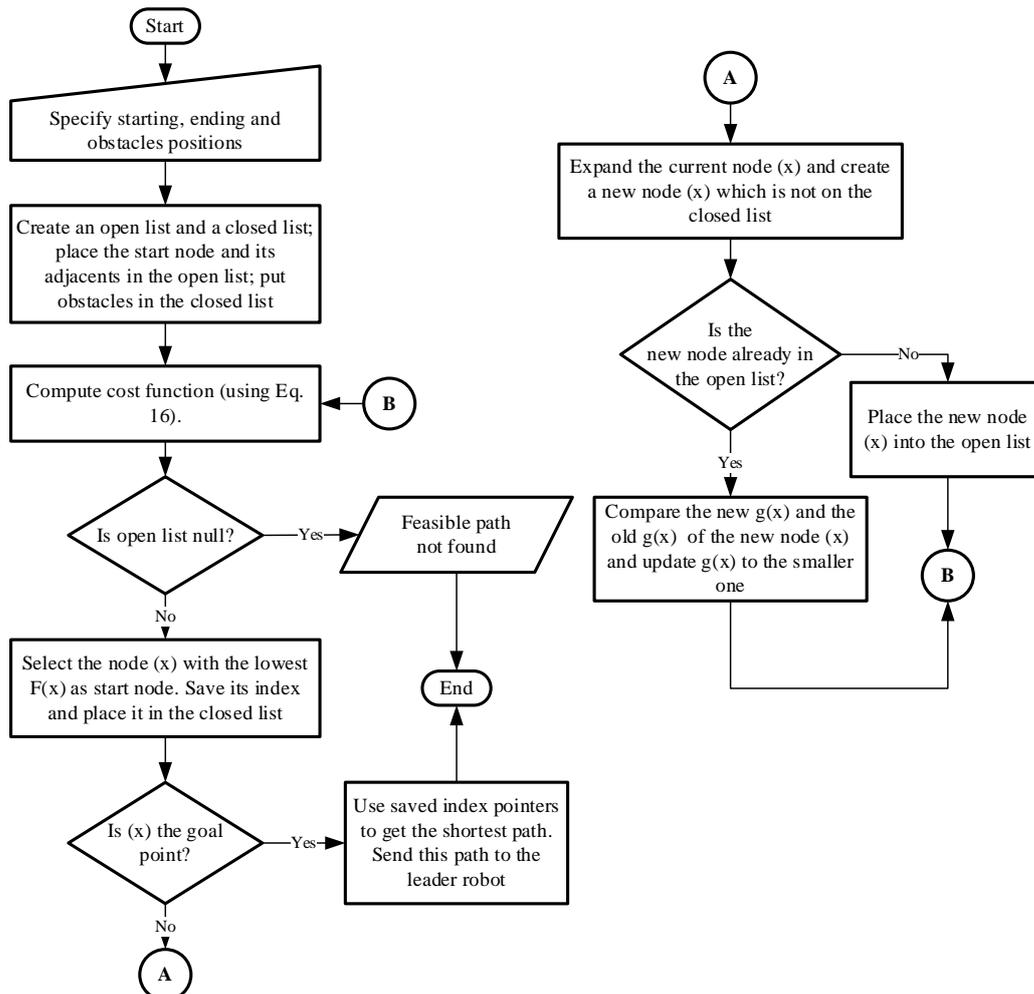


Figure 3. Flowchart of A-star algorithm

function, K_{rep} is a positive parameter which defines the impact of the repulsive potential, ε is the distance between the robot and the obstacle and ε_0 represents the threshold distance of the repulsive potential influence.

The repulsive force (F_{rep}) is determined by computing the negative gradient of the repulsive field, as shown in equation (20).

$$F_{rep} = -\nabla(U_{rep}) = \begin{cases} \frac{K_{rep}}{\varepsilon^3} \left(\frac{1}{\varepsilon_0} - \frac{1}{\varepsilon} \right) \begin{bmatrix} x_o - x_R \\ y_o - y_R \end{bmatrix}, & \varepsilon \leq \varepsilon_0 \\ 0, & \varepsilon > \varepsilon_0 \end{cases} \quad (20)$$

where x_o and y_o are the components of the obstacle vector. Therefore, the resulting total force (F_{tot}) that moves the robot towards the desired position will have the influences of both attractive and repulsive forces as expressed in equation (21). Figure 4 is a simplified representation of the resulting force on the follower robot in the presence of the attractive field (which pulls the robot towards the leader) and the repulsive field (which arises due to the presence of obstacles).

$$F_{tot}(X_R) = F_{att}(X_R) + F_{rep}(X_R) \quad (21)$$

In Algorithm 1, the pseudocode of the controller applied in this paper based on artificial potential fields is presented.

Algorithm 1: Pseudocode of the APF

Input: leader's position, followers' positions, desired position(s), stopping criterion (δ), max. linear and angular speeds (v_{max} and ω_{max}).

Output: linear and angular speeds (v and ω).

```

1: while ( $\rho > \delta$ ) do
2:    $x_R, y_R, \theta_R \leftarrow \text{Sense the robot position};$ 
3:    $F_{att} = K_{att} \begin{bmatrix} x_D - x_R \\ y_D - y_R \end{bmatrix};$ 
4:    $F_{rep} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$ 
5:    $N_o \leftarrow \text{Sense the obstacle}(s);$ 
6:   if  $N_o > 0$  then
7:      $F_{rep} = K_{rep} \sum_{i=1}^{N_o} \frac{1}{\varepsilon_i^3} \left( \frac{1}{\varepsilon_0} - \frac{1}{\varepsilon_i} \right) \begin{bmatrix} x_o^i - x_R \\ y_o^i - y_R \end{bmatrix};$ 
8:   end
9:    $F_{tot} = F_{att} + F_{rep};$ 
10:   $v \leftarrow \min(\|F_{tot}\|_2, v_{max});$ 
11:   $\omega \leftarrow (\text{atan2}(F_{tot}^{(y)}, F_{tot}^{(x)}) - \theta_R);$ 
12:   $\omega \leftarrow \text{sign}(\omega) \min(|\omega|, \omega_{max});$ 
13:   $\rho \leftarrow \sqrt{(x_D - x_R)^2 + (y_D - y_R)^2};$ 
14:  return  $v$  and  $\omega$ ;
15: end

```

III. Results and Discussions

To demonstrate that the designed controllers work properly, a simulator was designed using the MATLAB programming language. The simulations were carried out using MATLAB R2019a software, installed in a personal computer equipped with an Intel® Core™ i3 2.20 GHz processor and 4.00 GB of RAM.

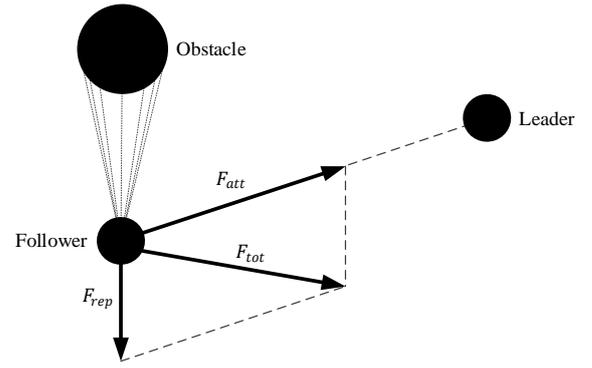


Figure 4. Interaction of forces in the APF

The A-star algorithm is used to obtain the least cost path. This path is sent to the leader. The leader's position is communicated to the followers. Based on this information and according to the values of the desired distances and angles, the followers track the leader robot. Followers are controlled based on the artificial potential fields algorithm, thus avoiding obstacles in real time. As shown in the next subsections, seven experiments were conducted under different conditions. In all experiments, the leader is colored red, and the followers are colored yellow. The robots' positions are represented by a 1×3 vector, that is, $[x(m), y(m), \theta(rad)]$, while the obstacles are represented by a 1×2 vector, i.e., $[x(m), y(m)]$.

A. First experiment

One follower and one leader are used for this experiment. A vertical obstacle is placed in the environment. The leader robot is asked to follow the safe route obtained by A-star. The follower robot is asked to keep an angle of 0 radians and a distance of 1 meter from the leader. The starting positions of leader and follower are $[2.50, 6.50, -0.524]$, and $[2.00, 8.50, 0.175]$, respectively. The goal position is $[10.5, 6.5, 0.873]$, while the positions of the obstacles are $[6.5, 5.5]$, $[6.5, 6.5]$, and $[6.5, 7.5]$. The leader's final position is $[10.44, 6.43, 0.877]$ and the follower's final position is $[9.73, 5.60, 0.858]$. The trajectories of this experiment, which lasted 122.623 seconds to complete, are illustrated in Figure 5.

B. Second experiment

In this experiment, one leader and one follower are taken again. But this time, a horizontal obstacle was placed in the environment. Under these conditions, the leader is asked to reach the target point by following the path obtained by A-star. The follower robot is requested to follow the leader by maintaining a distance of 1.5 meters and an angle of -0.5236 radians relative to the leader. The starting positions of leader and follower are $[1.50, 6.50, -0.524]$, and $[1.00, 8.50, 0.175]$, respectively. The goal position is $[10.5, 6.5, -0.767]$, while the position of the obstacle is $[5.5, 6.5]$. The leader's final position is $[10.43, 6.56, -0.763]$ and the follower's final position is $[8.95, 6.78, -0.753]$. Figure 6 shows this experiment, which lasted 156.099 seconds.

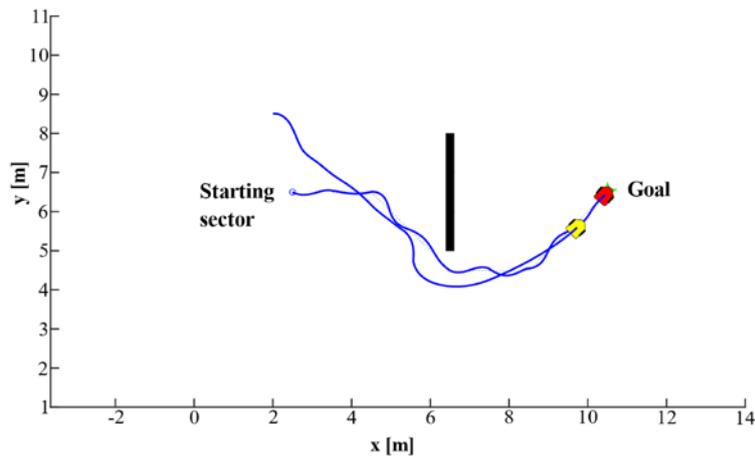


Figure 5. Result of the first experiment obtained using the simulator made in MATLAB

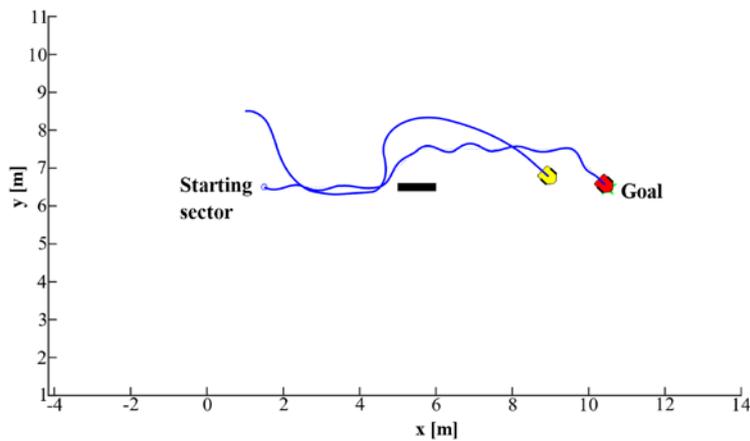


Figure 6. Result of the second experiment obtained using the simulator made in MATLAB

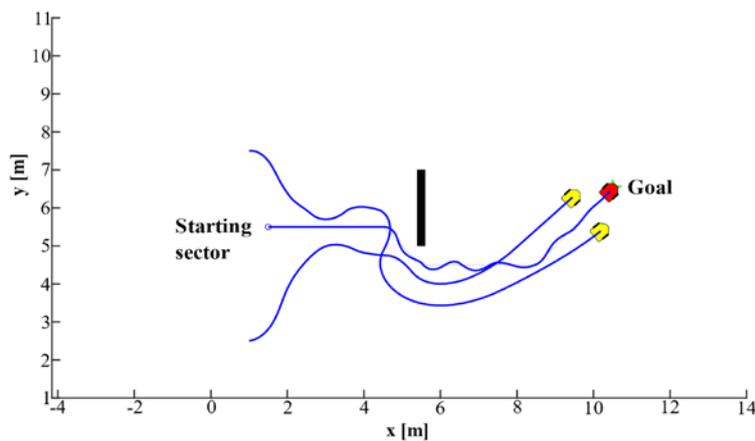


Figure 7. Result of the third experiment obtained using the simulator made in MATLAB

C. Third experiment

Two followers were used in this experiment. A vertical obstacle is included in the environment. Both follower robots are required to be 1 meter away from the leader. At the same time, one of the follower robots is asked to keep -0.5236 radians away from the leader. The other one is asked to keep it at 0.5236 radians. The leader's starting position is $[1.50, 5.50, 0.000]$ and the followers' starting positions are $[1.00, 7.50, 0.087]$, and $[1.00, 2.50, 0.175]$. The goal position is $[10.5, 6.5, 0.775]$, while the positions of the obstacles are $[5.5, 5.5]$, and

$[5.5, 6.5]$. The leader's final position is $[10.43, 6.44, 0.778]$ and the followers' final positions are $[10.19, 5.39, 0.782]$, and $[9.45, 6.28, 0.759]$. The trajectories of this experiment, which lasted 124.426 seconds to complete, are illustrated in [Figure 7](#).

D. Fourth experiment

Two followers and a horizontal obstacle were used in this experiment. The two followers must maintain 0 radian in relation to the leader. To prevent followers from colliding with each other, followers are commanded to have different separation distances from the leader. Thus, the first

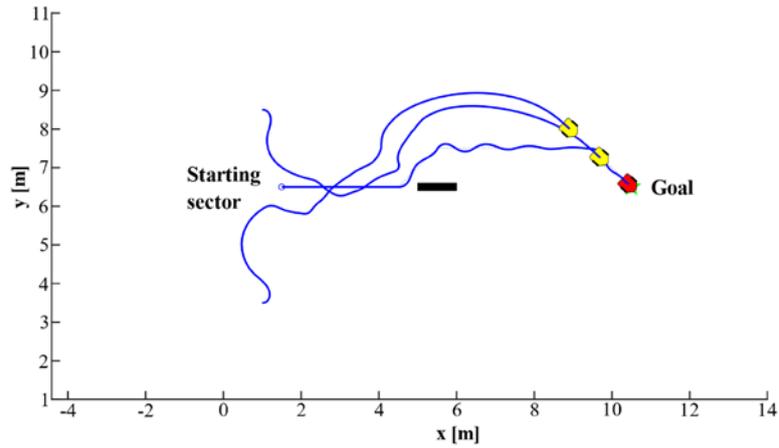


Figure 8. Result of the fourth experiment obtained using the simulator made in MATLAB

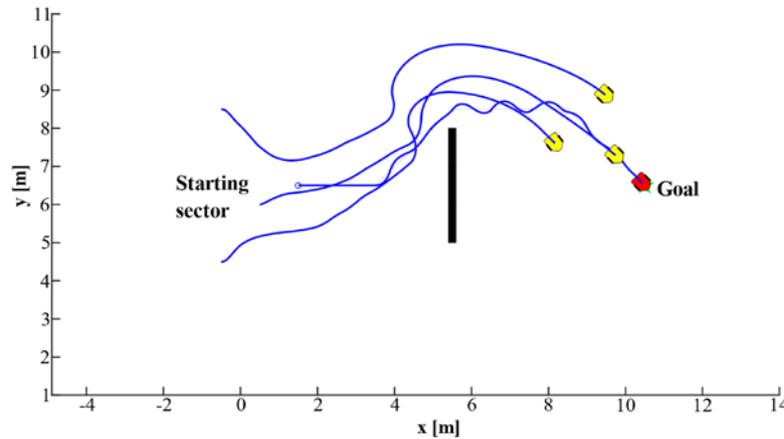


Figure 9. Result of the fifth experiment obtained using the simulator made in MATLAB

follower is commanded to keep 1 meter away from the leader and the other one is asked to keep 2 meters away from the leader. The leader's starting position is [1.50, 6.50, 0.000] and the followers' starting positions are [1.00, 8.50, 0.000] and [1.00, 3.50, 0.000]. The goal position is [10.5, 6.5, -0.768], while the position of the obstacle is [5.5, 6.5]. The leader's final position is [10.43, 6.56, -0.765] and the followers' final positions are [9.71, 7.26, -0.771] and [8.99, 7.95, -0.755]. The trajectories of this experiment, which lasted 245.220 seconds to complete, are illustrated in [Figure 8](#).

E. Fifth experiment

In this experiment, 3 followers and a vertical obstacle were considered. The 3 followers are requested to hold 0.3491, 0, and -0.3491 radians from the leader robot, respectively. Follower 1 should be 2.5 meters away from the leader. Follower 2 should be 1 meter away from the leader. Follower 3 should be 2.5 meters away from the leader. The leader's starting position is [1.50, 6.50, 0.000] and the followers' starting positions are [-0.50, 8.50, 0.00], [0.50, 6.00, 0.000], and [-0.50, 4.50, 0.000]. The goal position is [10.5, 6.5, -0.785] and the obstacles' positions are [5.5, 5.5], [5.5, 6.5], and [5.5, 7.5]. The leader's final position is [10.44, 6.57, -0.803] and the followers' final positions are [9.47, 8.88, -0.894], [9.74, 7.29, -0.802], and [8.16, 7.61, -0.804]. [Figure 9](#)

shows this experiment, which lasted 262.800 seconds.

F. Sixth experiment

In this experiment, 4 followers and a horizontal obstacle were selected. The vector [0.6981, 0, 0, -0.6981] was used to define the desired angles of the four followers. The vector [2.5, 1, 2, 2.5] was used to define the desired distances of the four followers relative to the leader. The leader's starting position is [1.50, 6.50, 0.000] and the followers' starting positions are [-3.50, 8.50, 0.000], [-1.50, 6.00, 0.000], [0.50, 3.50, 0.000], and [-2.50, 4.50, 0.000]. The goal position is [10.5, 6.5, -0.766] and the obstacle's position is [5.5, 6.5]. The leader's final position is [10.43, 6.56, -0.765] and the followers' final positions are [10.91, 9.02, -0.981], [9.71, 7.26, -0.859], [8.99, 7.95, -0.773], and [7.998, 5.99, -0.601]. The trajectories of this experiment, which lasted 277.587 seconds to complete, are illustrated in [Figure 10](#).

G. Seventh experiment

Four follower robots and vertical obstacles were used to simulate this experiment. The vector [0.5236, 0, 0, -0.5236] is used to define the desired angles of the follower robots and the vector [2.5, 1, 2, 2.5] is used to define the distance of the follower robots relative to the leader robot. The leader's starting position is [1.50, 5.50, 0.000] and the followers'

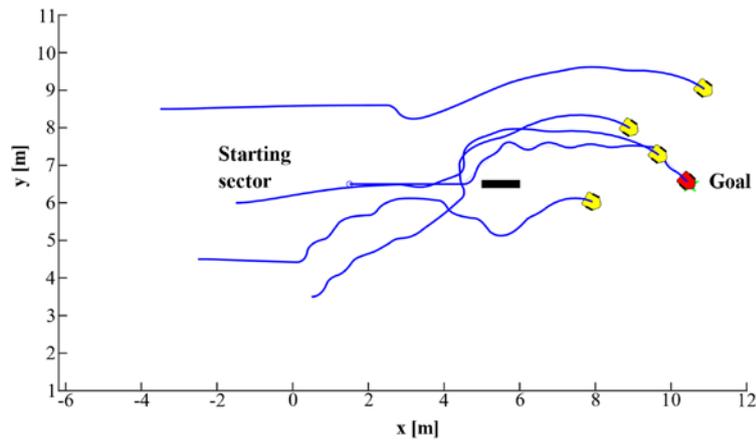


Figure 10. Result of the sixth experiment obtained using the simulator made in MATLAB

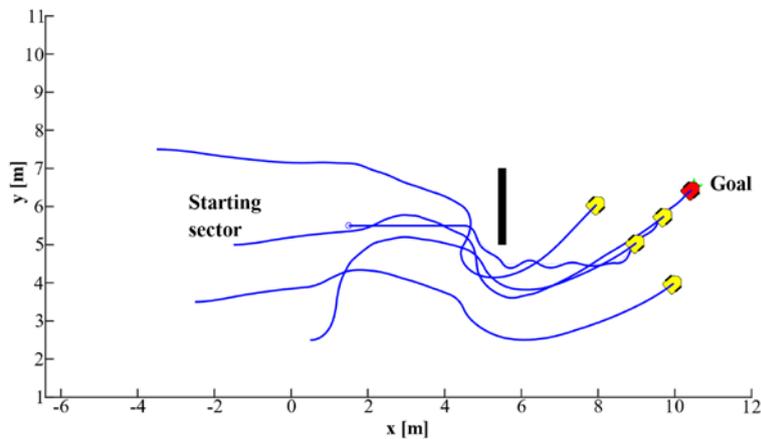


Figure 11. Result of the seventh experiment obtained using the simulator made in MATLAB

starting positions are $[-3.50, 7.50, 0.000]$, $[-1.50, 5.00, 0.000]$, $[0.50, 2.50, 0.000]$, and $[-2.50, 3.50, 0.000]$. The goal position is $[10.5, 6.5, 0.767]$ and the obstacles' positions are $[5.5, 5.5]$, and $[5.5, 6.5]$. The leader's final position is $[10.43, 6.44, 0.768]$ and the followers' final positions are $[7.96, 6.05, 0.778]$, $[9.71, 5.74, 0.743]$, $[8.99, 5.05, 0.735]$, and $[9.96, 3.98, 0.761]$. Figure 11 shows this experiment, which lasted 185.760 seconds.

H. Comparison of distance and orientation errors

In this subsection, the distance and orientation errors measured in each experiment are presented. The errors of the leaders were measured in relation

to the goal position, while the errors of the followers were measured using as reference the desired values of angle and distance in relation to the leaders. Figure 12 to Figure 18 depict the distance and orientation errors for each experiment. The leaders' distance errors remained practically constant in all experiments. The leader's orientation errors are smaller than those of followers in all experiments, except in experiments four and five. With the exception of experiment 1, the followers' distance errors are relatively smaller than the leaders' ones in all experiments. Overall, it can be noticed that the values of both distance and orientation errors do not exceed 0.1 m and 0.1 rad, respectively.

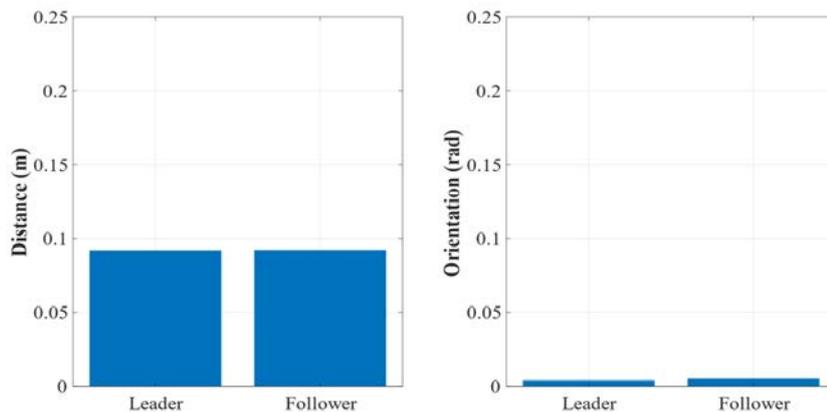


Figure 12. Distance and orientation errors corresponding to the first experiment

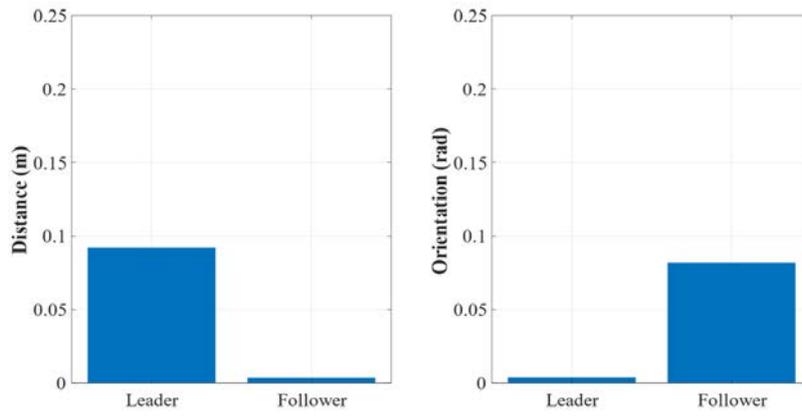


Figure 13. Distance and orientation errors corresponding to the second experiment

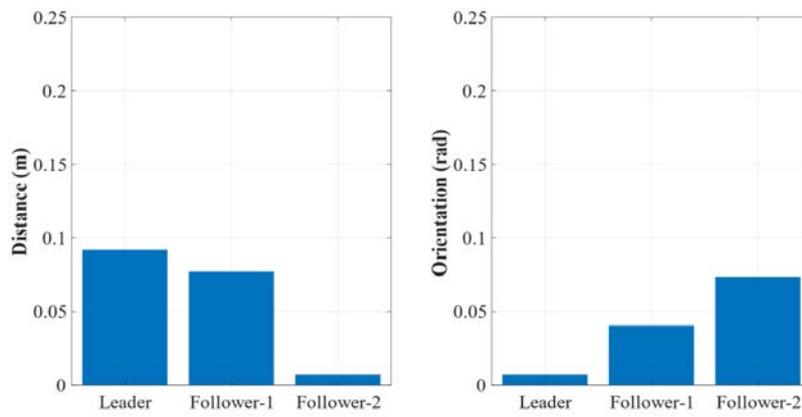


Figure 14. Distance and orientation errors corresponding to the third experiment

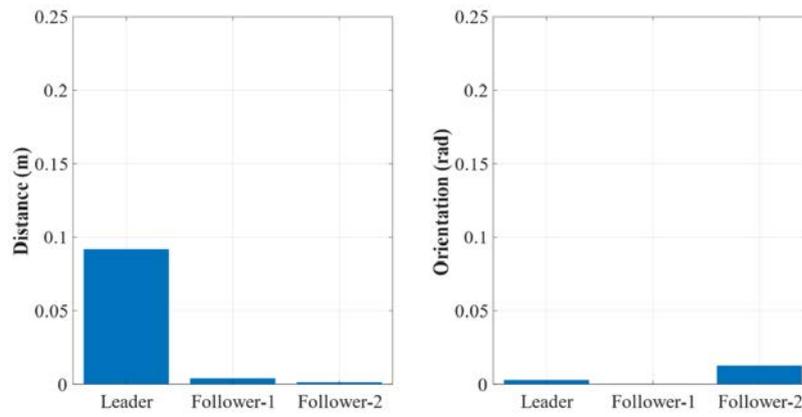


Figure 15. Distance and orientation errors corresponding to the fourth experiment

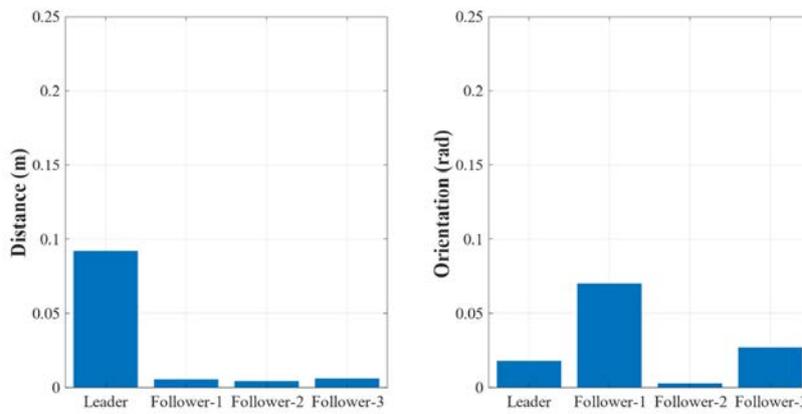


Figure 16. Distance and orientation errors corresponding to the fifth experiment

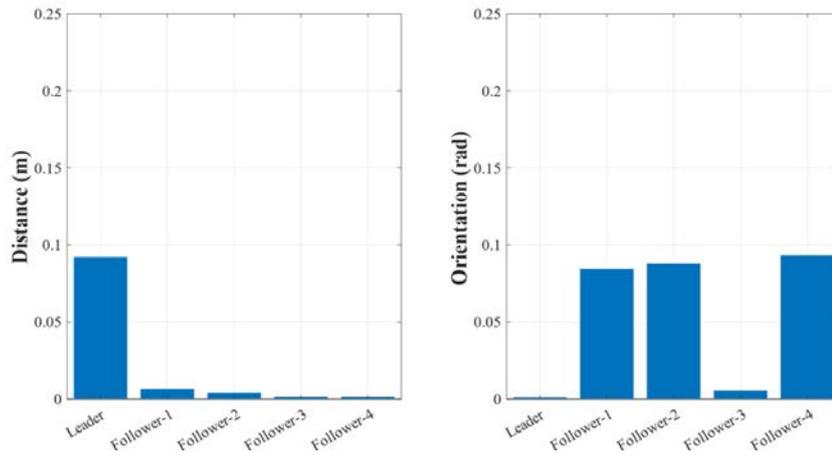


Figure 17. Distance and orientation errors corresponding to the sixth experiment

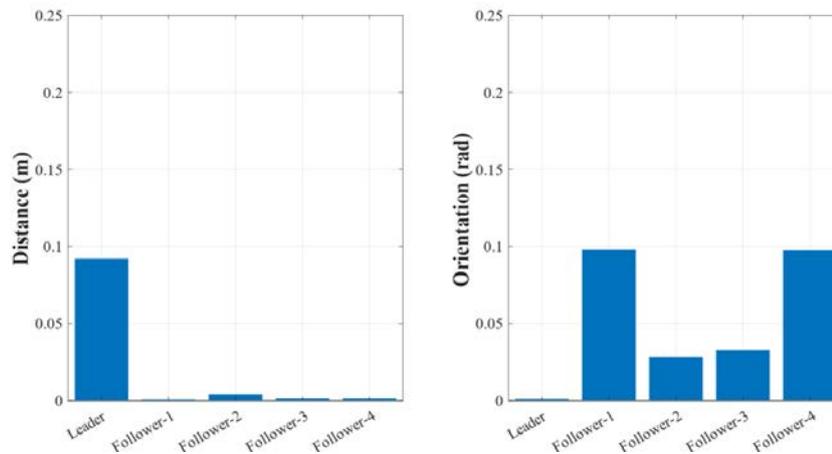


Figure 18. Distance and orientation errors corresponding to the seventh experiment

IV. Conclusion

In this paper, the formation control of differential mobile robots using the leader-follower approach was presented. As seen in the simulations, the A-star algorithm was used to obtain the safe path, then this path was sent to the leader. Follower robots receive information about the leader's position and maintain the desired distance and angles in relation to the leader robot. A controller that allows real-time obstacle avoidance (artificial potential fields) is used to allow followers to travel on safe paths. It is also important to emphasize that the conclusions made here are based on simulations and theoretical knowledge. Although the results obtained are successful in the virtual environment, it should be noted that, in the real world, some parameters of the applied controllers may need to be modified to achieve the same performance as in the virtual environment. This is because in the real world, there are many uncertainties that may not have been considered during the simulations.

Declarations

Author contribution

All authors contributed equally as the main contributor to this study.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information is available at <https://mev.lipi.go.id/>.

Publisher's Note: Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences remains neutral with regard to jurisdictional claims and institutional affiliations.

References

- [1] K. H. Kowdiki, R. K. Barai, and S. Bhattacharya, "Leader-follower formation control using artificial potential functions: A kinematic approach," *IEEE-International Conf. Adv. Eng. Sci. Manag. ICAESM-2012*, pp. 500-505, 2012.
- [2] A. Alouache and Q. Wu, "Performance comparison of consensus protocol and l-phi approach for formation control of multiple nonholonomic wheeled mobile robots," *J. Mechatronics, Electr. Power, Veh. Technol.*, vol. 8, no. 1, pp. 22-32, Jul. 2017.

- [3] L. Pacheco and N. Luo, "Testing PID and MPC performance for mobile robot local path-following," *Int. J. Adv. Robot. Syst.*, vol. 12, no. 11, p. 155, Nov. 2015.
- [4] S. Hiroi and M. Niitsuma, "Building a map including moving objects for mobile robot navigation in living environments," in *2012 Ninth International Conference on Networked Sensing (INSS)*, Jun. 2012, pp. 1–2.
- [5] W.-Y. Lee, K. Hur, K. Hwang, D.-S. Eom, and J.-O. Kim, "Mobile robot navigation using wireless sensor networks without localization procedure," *Wirel. Pers. Commun.*, vol. 62, no. 2, pp. 257–275, Jan. 2012.
- [6] S.-L. Dai, S. He, X. Chen, and X. Jin, "Adaptive leader-follower formation control of nonholonomic mobile robots with prescribed transient and steady-state performance," *IEEE Trans. Ind. Informatics*, vol. 16, no. 6, pp. 3662–3671, Jun. 2020.
- [7] J. Hirata-Acosta, J. Pliego-Jiménez, C. Cruz-Hernández, and R. Martínez-Clark, "Leader-follower formation control of wheeled mobile robots without attitude measurements," *Appl. Sci.*, vol. 11, no. 12, p. 5639, Jun. 2021.
- [8] T. Dewi, C. Sitompul, P. Risma, Y. Oktarina, R. Jelista, and M. Mulyati, "Simulation analysis of formation control design of leader-follower robot using fuzzy logic controller," in *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Oct. 2019, pp. 68–73.
- [9] J. Dong, S. Liu, and K. Peng, "Formation control of multirobot based on I/O feedback linearization and potential function," *Math. Probl. Eng.*, vol. 2014, pp. 1–6, 2014.
- [10] 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, pp. 406–411.
- [11] 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, no. 9, pp. 43581–43590, 2019.
- [12] H. Xiao, Z. Li, and C. L. Philip Chen, "Formation control of leader-follower mobile robots' systems using model predictive control based on neural-dynamic optimization," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5752–5762, Sep. 2016.
- [13] Y. Zhao, Y. Zhang, and J. Lee, "Lyapunov and sliding mode based leader-follower formation control for multiple mobile robots with an augmented distance-angle strategy," *Int. J. Control. Autom. Syst.*, vol. 17, pp. 1314–1321, 2019.
- [14] J. Heikkinen, T. Minav, and A. D. Stotckaia, "Self-tuning parameter fuzzy PID controller for autonomous differential drive mobile robot," *Proc. 2017 20th IEEE Int. Conf. Soft Comput. Meas. SCM 2017*, pp. 382–385, 2017.
- [15] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*, 2nd ed. Cambridge, MA: MIT press, 2011.
- [16] T. Zhang, P. W. Stackhouse, B. Macpherson, and J. C. Mikovitz, "A solar azimuth formula that renders circumstantial treatment unnecessary without compromising mathematical rigor: Mathematical setup, application and extension of a formula based on the subsolar point and atan2 function," *Renew. Energy*, vol. 172, pp. 1333–1340, 2021.
- [17] I. M. Zidane and K. Ibrahim, "Wavefront and a-star algorithms for mobile robot path planning," *Adv. Intell. Syst. Comput.*, vol. 639, pp. 69–80, 2018.
- [18] R. L. Galvez et al., "Obstacle avoidance algorithm for swarm of quadrotor unmanned aerial vehicle using artificial potential fields," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Nov. 2017, pp. 2307–2312.
- [19] A. Azzabi and K. Nouri, "Path planning for autonomous mobile robot using the potential field method," in *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, Jan. 2017, pp. 389–394.