Enhancement of Non-Holonomic Leader-Follower.pdf

by

Enhancement of Non-Holonomic Leader-Follower Formation Using Interval Type-2 Fuzzy Logic Controller

4 authors, including:

Some of the authors of this publication are also working on these related projects:

Faculty Grant 2011 View project

Data Modelling for Public Awareness towards air quality View project

Enhancement of Non-Holonomic Leader-Follower Formation Using Interval Type-2 Fuzzy Logic Controller

Bambang Tutuko, Siti Nurmaini(✉), Saparudin, Gita Fadila Fitriana
Intelligent System Research Group, Universitas Sriwijaya, Indonesia
sitinurmaini@gmail.com

Abstract—Robotics control system with leader-follower approach has a weakness in the case of formation failure if the leader robot fails. To overcome such problem, this paper proposes the formation control using Interval Type-2 Fuzzy Logic controller (IT2FLC). To validate the performance of the controller, simulations were performed with various environmental systems such as open spaces, complexes, circles and ovals with several parameters. The performance of IT2FLC will be compared with Type-1 Fuzzy Logic (T1FL) and Proportional Integral and Derivative (PID) controller. As the results found using IT2FLC has advantages in environmental uncertainty, sensor imprecision and inaccurate actuator. Moreover, IT2FLC produce good performance compared to T1FLC and PID controller in the above environments, in terms of small data generated in the fuzzy process, the rapid response of the leader robot to avoid collisions and stable movements of the follower robot to follow the leader's posture to reach the target without a crash. Especially in some situations when a leader robot crashes or stops due to hardware failure, the follower robot still continue move to the target without a collision.

Keywords—Leader-Follower Robot, Interval type-2 fuzzy logic, Formation control, Swarm robot

1 Introduction

Formation control of multiple autonomous mobile robots a has been studied extensively and it became a challenging topic among multi-robot research issues [1]. This is because there are many potential advantages of such systems over a single robot, including greater flexibility, adaptability to unknown environments and robustness [2], [3]. Formation control is defined as the coordination of the group of robots that maintain a formation within specified geometrical shapes as an obstacle. Contrary to the single robot, the multi-robot maneuvering for keeping in formation is complicated, requiring additional conditions to be considered to make them work cooperatively [1]. The formation control has been expected to be employed in various applications such as manipulation of large objects [4], [5], intelligent highway systems [6], [7], surface vehicle formation [8], flight formation systems [9], [10], formation of multiple spacecraft [11] and surveillance systems [12], [13], machine vision [14].

Multi-robot formation methods can be partitioned into three class approaches such as virtual structure approach, behavioral approach, and leader-follower approach. Each of them has several advantages and weaknesses. The virtual structure approach treats the entire formation as a single virtual rigid structure [5], [11]. The main disadvantage of the current virtual structure implementation is the centralization, which leads a single point of failure for the whole system. By behavior-based approach, several desired behaviors are prescribed for each robot, and the final action of each robot is derived by weighting the relative importance of each behavior [15], [16]. The limitation of such approach, it is difficult to analyze mathematically, therefore it is hard to guarantee a precise formation control. In the leader-follower approach [17], [18], [19], one of the robots is designated as the leader, with the rest being followers. Unfortunately, it centralized control, if the leader fails then it can be influenced by the all system performance [20], [22]. Due to, the limitation has explained above hence, the selected method to enhance the formation control performance, how to improve it by overcoming the drawbacks is desirable. In a certain sense, the formation control problem can be seen as a natural extension of the traditional trajectory-tracking problem.

Several technique have proposed to give a solution with good performance [21], [5], [15], [17], [23], [24], [25], [26], [27], [28], [29]. However, in the robotic application, there are some uncertainties in the system, due to the sensors imprecision, inaccurate actuator and environment change every time [28]. In terms of robot's position and orientation, it produces the accumulation error of robot formation. The fuzzy logic control algorithm can overcome the uncertainty problems [30]. Unfortunately, the type-1 fuzzy logic controller (T1FLC) can't ensure the performance, because the uncertainty is crisp value. To the best our knowledge, only a few of existing results have been presented to solve the problem of leader-follower formation control based on the interval type-2 fuzzy logic controller (IT2FLC) [30], [18], [16]. This paper aims to investigate how to design the leader-follower formation control based on the IT2FLC for achieving robust formation against the leader faults. The rest of paper is organized as follows: Section 2 briefly discusses the leader-follower kinematic model while Section 3 describes our proposed controller and material. To demonstrate the usefulness of the proposed control algorithm, the simulations and the result are presented in Section 4, and finally, the conclusion of the study is given in Section 5.

2 The Non-Holonomic Leader-Follower System

Several research about the multi-robots using a cooperative approach like a leader-follower approach has been proposed [31], [32]. Based on previous results, there are some problems with such approach. These problems appear when the robots want to regulate the position and orientation when they move in the groups to reach the target. Such problems include the accuracy of tracking the robot when the leader is moving, the distance to avoid collisions between the leader and the follower and the capability the leader to avoid the obstacles, due to if the leader crash or fail can influence the

follower. Besides, the response time of the follower when the moment the leader moving.
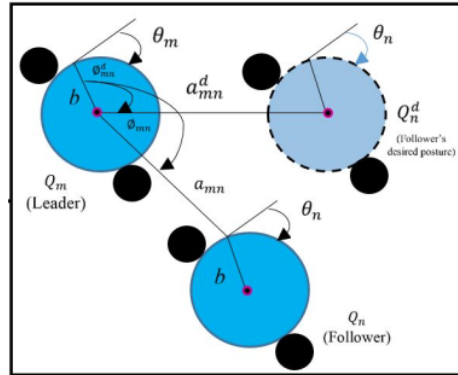

Fig. 1. Leader-follower robot posture

As presented in Figure 1, the leader-follower kinematic system with the general equation of non-holonomic system on a single robot can be expressed in the equation (1) as follows,

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{pmatrix} = T_{NH}(q)\, u(t) \tag{1}$$

Where $q(t)$ is the general variable of an initial position of the robot $q(t) = [x(t), y(t), \theta(t)]^{\mathrm{T}}$, T_{NH} is non-holonomic transformation matrix, and $u(t)$ is forward kinematic matrix which is used to estimate position and speed in the Cartesian space. The non-holonomic transformation of the mobile robot can be seen through the change of the three initial robotic position variables $q(t)$. By solving equation (1) to the change in the speed of the right wheel and the left wheel, the single robot kinematic equation can be transformed into equation (2) as follow,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} cos\theta(t) & 0 \\ sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \tag{2}$$

The kinematic constraint of non-holonomic system works only for pure rolling and non-slipping. Such system uses to control mobile robot posture in position and orientation, it can be written,

$$\dot{y}_m \cos\theta_m - \dot{x}_m \sin\theta_m = b\dot{\theta}_m \tag{3}$$

When $\dot{y}_m$ is defined a velocity of leader robot towards y-axis, $\dot{x}_m$ is defined a velocity of leader robot towards x-axis, θ_m which is represented as the angle of robot movement and b represents the actual bearing between the follower and leader robot. In the leader-follower system, two equations of posture robot must solve such as the posture control of leader and the posture control of follower robot. However, the non-holonomic constraint must be calculated in advance. By using Equation (2), the kinematics model of the non-holonomic leader robot can be presented as position and orientation equation. The leader posture in initial position and orientation can be described in Equation (4) as follow,

$$\dot{Q}_m = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \cos \theta_m & -b \sin \theta_m \\ \sin \theta_m & b \cos \theta_m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_m \\ \omega_m \end{bmatrix} \tag{4}$$

Where $\dot{Q}_m$ represents the actual position of leader robot. Meanwhile, the matrix represents a rotation of the leader robot's wheel and $\dot{Q}_m$ generates velocity (v_m) and the angle of the wheel (ω_m). The geometrical relation between the leader and the follower robot is generated from desired follower robots position (Q_n^d). The posture of the leader robot for desired position and orientation is shown in Equation (5),

$$Q_n^d = [x_n^d, y_n^d, \theta_n^d]^T = \begin{bmatrix} x_m - b \cos \theta_m + a_{mn}^d \cos(\phi_{mn}^d + \theta_m) \\ y_m - b \sin \theta_m + a_{mn}^d \sin(\phi_{mn}^d + \theta_m) \\ \theta_m \end{bmatrix} \tag{5}$$

However, the follower robot moves in randomly and continuously follow the leader robot. To force the direction of the follower robot, the Equation (4) is utilized. The value of Q_n^d depends on some values such as desired x axis (x_n^d), desired y axis (y_n^d), and θ_n^d. Meanwhile, to identify actual posture of follower robot can be written as Equation (6).

$$Q_n = [x_n, y_n, \theta_n]^T = \begin{bmatrix} x_m - b \cos \theta_m + a_{mn} \cos(\phi_{mn} + \theta_m) \\ y_m - b \sin \theta_m + a_{mn} \sin(\phi_{mn} + \theta_m) \\ \theta_n \end{bmatrix} \tag{6}$$

The ϕ_{mn} represents actual angle between follower's wheel and leader's wheel. The actual distance between the leader and the follower robot a_{mn} on the x-y Cartesian coordinate can be represented by,

$$a_{mn} = \sqrt{a_{mnx}^2 + a_{mny}^2} \tag{7}$$

where, a_{mnx} the relative distance between the leader and the follower in x coordinate and a_{mny} the relative distance between the leader and the follower in y coordinate. The derivative of Equation (8) and (9) are substituted into Equation (4) become,

$$a_{mnx} = x_m - x_n - b \cos \theta_m = -a_{mn} \cos(\phi_{mn} + \theta_m) \tag{8}$$

$$\dot{a}_{mnx} = \dot{x}_m - \dot{x}_n + b \dot{\theta}_m \sin \theta_m$$

$$= v_m \cos\theta_m - b\omega_m \sin\theta_m - (v_n \cos\theta_n - b\omega_n \sin\theta_n) + b\omega_m \sin\theta_m$$

$$= v_m \cos\theta_m - v_n \cos\theta_n + b\omega_n \sin\theta_n \tag{9}$$

$$a_{mny} = y_m - y_n - b\sin\theta_m = -a_{mn}\sin(\phi_{mn} + \theta_m) \tag{10}$$

$$\dot{a}_{mny} = \dot{y}_m - \dot{y}_n - b\dot{\theta}_m \cos\theta_m$$

$$= v_m \sin\theta_m + b\omega_m \cos\theta_m - (v_n \sin\theta_n + b\omega_n \cos\theta_n) - b\omega_m \cos\theta_m$$

$$= v_m \sin\theta_m - v_n \sin\theta_n - b\omega_n \cos\theta_n \tag{11}$$

Where v_m is a linear velocity and ω_m is an angular velocity of the leader robot R_m and v_n is a linear velocity and ω_n is an angular velocity of the follower robot R_n. By using Equations (7), (8), (9), (10) and (11), the control outputs such as the relative distance of two the robots ($\dot{a}_{mn}$) and the separation bearing angle ($\dot{\phi}_{mn}$) are generated. All parameters of tracking control can be written in Equation (12), (13) and Equation (13), where $\beta_{mn} = \phi_{mn} + \theta_m - \theta_n$

$$\dot{a}_{mn} = -v_m \cos\phi_{mn} + v_n \cos\beta_{mn} + b\omega_n \sin\beta_{mn} \tag{12}$$

$$\dot{\phi}_{mn} = \frac{1}{a_{mn}}(v_m \sin\phi_{mn} - v_n \sin\beta_{mn} + b\omega_n \cos\beta_{mn}) - \omega_m \tag{13}$$

$$\dot{\theta}_n = \omega_n \tag{14}$$

For a single robot, the tracking controller error system presented in Equation (15),

$$\begin{bmatrix} e_x(t) \\ e_y(t) \\ e_\theta(t) \end{bmatrix} = \begin{bmatrix} \cos\theta_c(t) & \sin\theta_c(t) & 0 \\ -\sin\theta_c(t) & \cos\theta_c(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_R(t) & -x_c(t) \\ y_R(t) & -y_c(t) \\ \theta_R(t) & -\theta_c(t) \end{bmatrix} \tag{15}$$

The basic tracking control problem can be extended to a formation control for leader-follower robot with relative distance a_{mn} to achieve the desired angle ϕ_{mn} to the leader robot. By using simple trigonometric identities and further simplification, the error system for follower robot n with respect to the leader robot m can be written as,

$$\begin{bmatrix} e_{nx} \\ e_{ny} \\ e_{n\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta_n & \sin\theta_n & 0 \\ -\sin\theta_n & \cos\theta_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{mn}^d \cos(\phi_{mn}^d + \theta_m) - a_{mn}\cos(\phi_{mn} + \theta_m) \\ a_{mn}^d \sin(\phi_{mn}^d + \theta_m) - a_{mn}\sin(\phi_{mn} + \theta_m) \\ \theta_m - \theta_n \end{bmatrix}$$

$$e_n = \begin{bmatrix} e_{nx} \\ e_{ny} \\ e_{n\theta} \end{bmatrix} = \begin{bmatrix} a_{mn}^d \cos(\phi_{mn}^d + (\theta_m - \theta_n)) - a_{mn}\cos(\phi_{mn} + (\theta_m - \theta_n)) \\ a_{mn}^d \sin(\phi_{mn}^d + (\theta_m - \theta_n)) - a_{mn}\sin(\phi_{mn} + (\theta_m - \theta_n)) \\ \theta_m - \theta_n \end{bmatrix} \tag{16}$$

3   Formation Control Using Interval Type-2 Fuzzy Logic

In this section, the proposed control system for leader-follower formation is discussed. Figure 2 shows the block diagram of IT2FLC which is embedded into the leader-follower robot formation. IT2FLC general processing blocks are similar to T1FLC, the major difference within the blocks is that IT2FLC has a type reducer. Each of these processing blocks will be described in the sequence of this section.
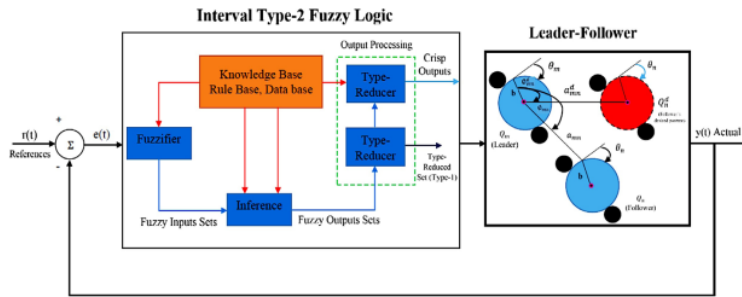

Fig. 2. Block diagram of IT2FL controller for leader-follower formation

The interval type-2 fuzzy sets have the membership functions (MFs) with an area of uncertainty, which bounded between the highest and the lowest values named Footprint of Uncertainties (FOUs). The FOUs are obtained from imprecision of sensor data, due to some noises in the environment cause inaccurate of the robot posture (velocity and steering angle). The highest values of the FOUs are Upper Membership Functions (UMFs), and the lowest values are Lower Membership Functions (LMFs) with the membership degrees is equal to 1 (uniform). In the fuzzification process, the input and output value change from a crisp value become a fuzzy value which is represented as UMFs and LMFs.

The fuzzifier maps a numeric vector $x = (x_1, \dots x_p)^T \in X_1 \times X_2 \times, \dots, \times X_p \equiv$ **X** into an interval type-2 fuzzy sets $\tilde{A}_x$ in **X** [38,39]. Type-2 singleton fuzzifier is utilized, in a singleton fuzzification. In this condition $\mu_{\widetilde{A_x}}(x) = 1/1$ for $x = x'$ and $\mu_{\widetilde{A_x}}(x) = 1/0$ for all other $x \neq x'$. In the fuzzification design, two MFs for input and three MFs for output are utilized by using the symmetric Gaussian function. The Gaussian MFs have the value σ and c will form the set of Gaussian membership functions μ_i, or

$$f(x; \sigma, c) = e^{-\frac{1}{2}\left(\frac{x-c_1}{\sigma}\right)^2} \tag{17}$$

In IT2FLC design, the equation (16) it uses to determine the FOU boundaries (Δ) of the interval type-2 fuzzy sets with upper MFs ($\bar{\mu}$) and lower MFs ($\underline{\mu}_x$) are presented in Equation (18) as follows;

$$\bar{\mu}_x = \begin{cases} \left(e^{-\frac{1}{2}\left(\frac{x+c_1}{\sigma}\right)^2}\right), & x < c_1 \\ 1, & c_1 \leq x \leq c_2 \\ \left(e^{-\frac{1}{2}\left(\frac{x+c_2}{\sigma}\right)^2}\right), & x > c_2 \end{cases} \quad \text{and} \quad \underline{\mu}_x = \begin{cases} \left(e^{-\frac{1}{2}\left(\frac{x+c_2}{\sigma}\right)^2}\right), & x \leq \frac{c_1+c_2}{2} \\ \left(e^{-\frac{1}{2}\left(\frac{x+c_1}{\sigma}\right)^2}\right), & x > \frac{c_1+c_2}{2} \end{cases} \quad (18)$$

The Gaussian function is utilized as input MFs and output MFs, they can be described in Figure 3 and Figure 4 respectively. Figure 3 (a) represents the input MFs of T1FLC and Figure 3 (b) represents the input MFs of IT2FLC. Based on Figure 3 (b), the red-line represents the UMFs value while the blue line represents the LMFs.
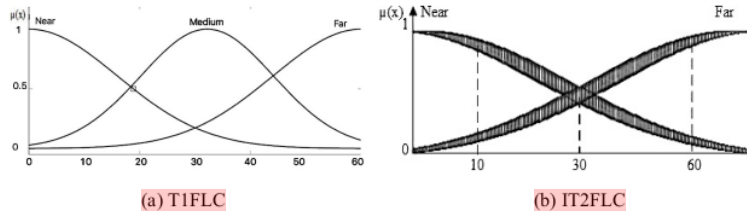

(a) T1FLC   (b) IT2FLC

Fig. 3. Input membership functions

(a) T1FLC velocity   (b) IT2FLC velocity

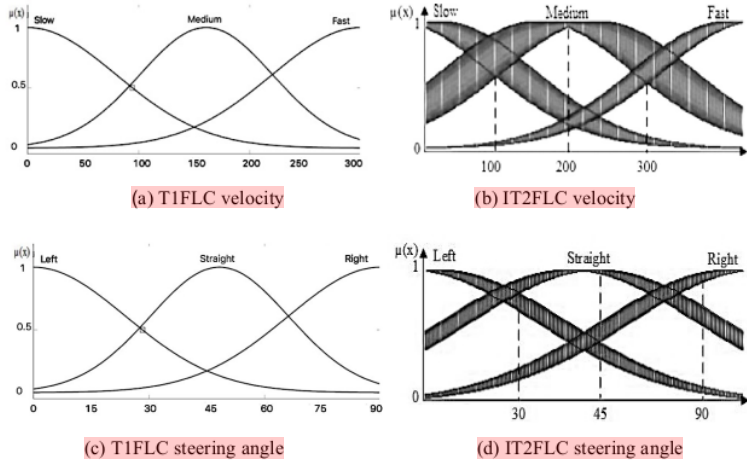(c) T1FLC steering angle   (d) IT2FLC steering angle

Fig. 4. Output membership functions

Figure 4 (a), 4(b), 4(c) and 4(d) respectively, describe the output MFs of T1FLC and IT2FLC for velocity and steering angle. The velocity $\dot{a}_{mn}$ of each wheel for the robot to perform rolling motion and to make the robot heading in the desired direction $\dot{\phi}_{mn}$. The value of variable linguistics in fuzzification process is generated by using Equation (17) and Equation (18). The linguistics variable of the robot velocity is represented as Slow (SL), Medium (MD), and Fast (FS), the values are set at the interval from 0 rpm to 400 rpm. The reference velocity of IT2FLC about 9,16 pixels/sec, while reference type of both T1FLC about 8,99 pixels/sec. The variable linguistics of the robot steering angle is represented as Left (LF), Straight (ST) and Right (RG) and the values are set at the interval from -180 ° to 180 °.

Table 1. The fuzzy rule bases of IT2FLC

Sensor 1	Sensor 2	Sensor 3	Velocity	Steering
N	N	N	SL	ST
N	N	F	SL	RG
N	F	N	SL	ST
N	F	F	MD	RG
F	N	N	SL	LF
F	N	F	MD	LF
F	F	N	MD	ST
F	F	F	FS	FS

In this paper, the IT2FLC rule bases only 8 rules are utilized to control the robot formation (See Table 1). The results performance compare to T1FLC with 8 rules and 27 rules. Each row of the rule is used to determine the fuzzification value to be stored during the inference process. The fired set $f_i = [\ \bar{f}_i, \underline{f}_i\ ]$ of the system by the number of rules is using the Extended Sup-Star Composition is defined as Equation (19):

$$\underline{f_i} = \frac{sup}{x \in X} \int_{X_1} \cdots \int_{X_p} \left[\underline{\mu}_{\tilde{X}_1}(x_1) * \underline{\mu}_{\tilde{F}_k^i}(x_1) \right] * \ldots * \left[\underline{\mu}_{\tilde{X}_1}(x_p) * \underline{\mu}_{\tilde{F}_k^i}(x_p) \right] / x \qquad (19a)$$

$$\bar{f_i} = \frac{sup}{x \in X} \int_{X_1} \cdots \int_{X_p} \left[\bar{\mu}_{\tilde{X}_1}(x_1) * \bar{\mu}_{\tilde{F}_k^i}(x_1) \right] * \ldots * \left[\bar{\mu}_{\tilde{X}_1}(x_p) * \bar{\mu}_{\tilde{X}_1}(x_p) \right] / x \qquad (19b)$$

This process is similar to the max-min composition of T1FLC, the difference only in two values of firing set of fuzzy consequent (upper and lower). The last part of the fuzzy controller to generate the result from the controller based on two steps; (1) type reduction, and (2) defuzzification. In the first step, the membership function of fuzzy type-2 is converted to type-1. The Karnik-Mendel algorithm is used as an iterative algorithm to produce y_r and y_l [32]. Two values kept into the memory and ready to be processed as the controller output. Assumed that y_r^i and y_l^i ($i = 1 \ldots$ M) are arranged ascendingly and calculate all rules fire values of f_i using Equation (20),

$$f_i = \frac{1}{2}[\underline{f_i} + \bar{f}_i] \qquad (20)$$

For i = 1...M, where M is the number of rules and calculate y_r' and y_l' by

$$y_r' = \frac{\sum_{i=1}^{M} y_i^r f_i^r}{\sum_{i=1}^{M} f_i^r} \quad and \quad y_l' = \frac{\sum_{i=1}^{M} y_i^l f_i^l}{\sum_{i=1}^{M} f_i^l} \tag{21}$$

To determine y_r' and y_l' by calculating the value of K (K = 1 ... M-1), in which $y_k^r \leq y' \leq y_{k+1}^r$ and $y_k^l \leq y' \leq y_{k+1}^l$ take into account that y already arranged ascendingly such that $y_1^r \leq y_i^r \leq y_m^r$ and $y_1^l \leq y_i^l \leq y_m^l$. To calculate the value of y_r'' and y_l'' by using Equation (22) as follows,

$$y_r'' = \frac{\sum_{i=1}^{k} y_i \underline{f}_i + \sum_{i=k+1}^{M} y_i \overline{f}_i}{\sum_{i=1}^{k} \underline{f}_i + \sum_{i=k+1}^{M} \overline{f}_i} \quad and \quad y_l'' = \frac{\sum_{i=1}^{k} y_i \overline{f}_i + \sum_{i=k+1}^{M} y_i \underline{f}_i}{\sum_{i=1}^{k} \overline{f}_i + \sum_{i=k+1}^{M} \underline{f}_i} \tag{22}$$

Check whether y_r'' equal to y_r' and y_l'' equal to y_l', if it is equal, then $y_r = y_r''$ and $y_l = y_l''$, if it is not, set $y_r = y_r''$ and $y_l = y_l''$ and repeat until y_r and y_l are found. In the second step of defuzzification, the value of y_r and y_l will be average to determine the crisp value by using Equation (23) as follows,

$$y = \frac{y_l + y_r}{2} \tag{23}$$

4   Discussion

In this section, the simulation is done to evaluate and to validate the fuzzy controller implementation in the leader-follower formation control. The simulation is determined based on transformation between cm to a pixel, and the environment for the experiment is a rectangular area bounded by six walls which some position of x,y. The scale of the testing environment is 1.8:1 cm, where 1 cm in the real environment is represented by 1.8 pixels on the simulation. The boundary of six walls created a space of 500 x 450 pixels. The sensors in the simulation are distance sensors, which the longest reading by 300 pixels or 214 cm. If the nearest obstacle is more than 300 pixels, the sensor will return a value of 300. In the simulation, the mobile robot is a circular shape with 35-pixels diameter circle or 25 cm. There are three locations for the ultrasonic sensor at the front, left and right of the mobile robot and each of sensor is separated by 30^0.

There are two kinds of the fuzzy logic are presented in this paper, such as T1FLC and IT2FLC. The comparison between two controllers is highlighted in some performances in terms of time response and stability movements. However, in a certain sense, the formation control problem can be seen as a natural extension of the traditional trajectory-tracking problem like Proportional, Integral, Derivative (PID) controller. For making a benchmark of the proposed IT2FLC is compared to PID controller in terms of trajectory shape, an output of the controller, and the response for x(t), y(t) and $\theta(t)$.

4.1 Leader-Follower Trajectory

To evaluate the fuzzy controller performance in the leader-follower tracking formation, three environments are utilized such as simple environment, cluttered environment with obstacles in it, and unstructured environment (circle and oval environment). Initial robot position in this experiment is (100,300) with 0^0 as its initial direction. The number of data recorded is 300 data.
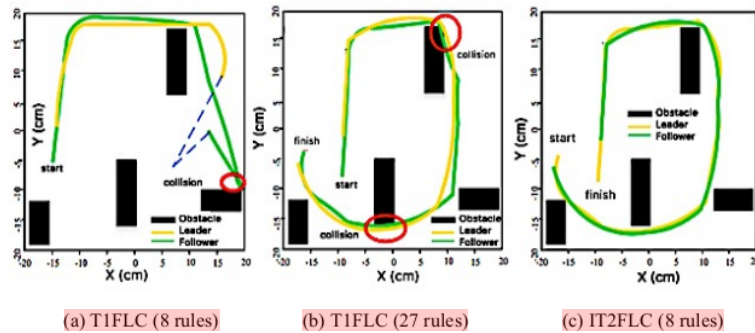

(a) T1FLC (8 rules)   (b) T1FLC (27 rules)   (c) IT2FLC (8 rules)

Fig. 5. Leader-follower tracking formation in environment 1 with 4 obstacles

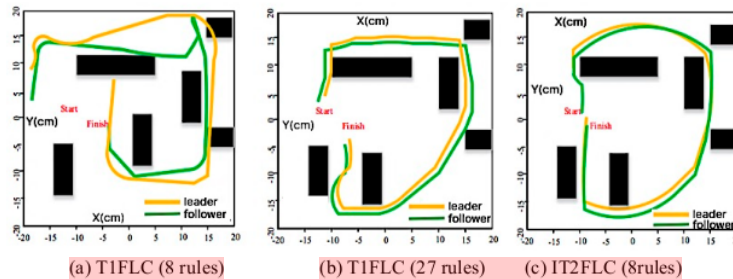(a) T1FLC (8 rules)   (b) T1FLC (27 rules)   (c) IT2FLC (8rules)

Fig. 6. Leader-follower tracking formation in environment 2 with 6 obstacles

Figure 5 and Figure (6) presents, the leader-follower formation control using IT2FLC in several environments with 4 obstacles, and 6 obstacles. The trajectory results from the experiment are compared to T1FLC with a different number of the rule (8 rules and 27 rules) in terms of data generated, time to find the target and the smoothness of the robot motion. By using 8 rules and 27 rules, the T1FLC always close distance to the wall and the follower movement not smooth. The follower keeps the longest distance to avoid the leader and the wall, therefore the trajectory of the follower not the same as the leader. IT2FLC produce a good performance, compare to T1FLC. However, when they move in an environment with 4 obstacles (see Figure (5)), by using T1FLC with 8 rules the leader fails, and the follower can't continue

move to the target. But T1FLC with 27 rules even though the leader fails, the follower still move to the target. It does not happen with IT2FLC with 8 rules, the leader-follower can accomplish the target without collision.
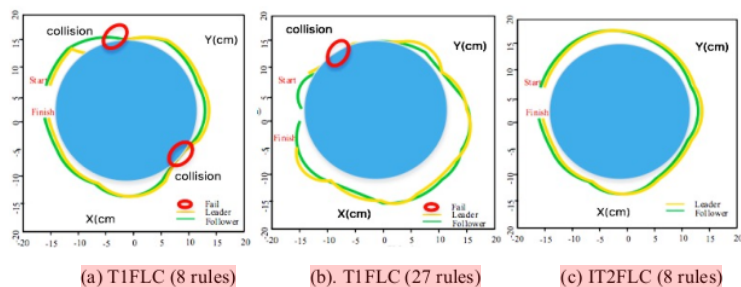

| (a) T1FLC (8 rules) | (b). T1FLC (27 rules) | (c) IT2FLC (8 rules) |

Fig. 7. Leader-follower tracking performance in circle environment

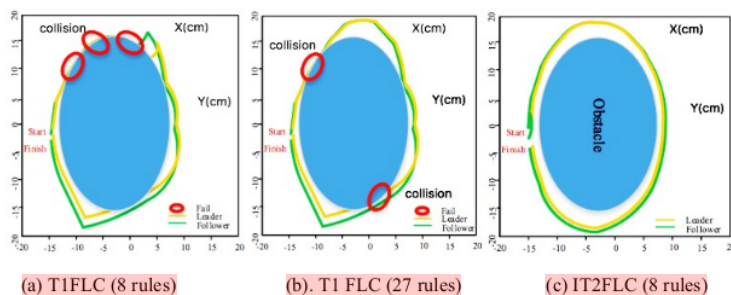| (a) T1FLC (8 rules) | (b). T1 FLC (27 rules) | (c) IT2FLC (8 rules) |

Fig. 8. Leader-follower tracking performance in oval environment

In a complex environment with 6 obstacles, T1FLC with 27 rules and IT2FLC produce good performance (see. Figure 7). However, the leader crash to the obstacle, when they use T1FLC with 8 rules. It shows that by using only 8 rules the leader robot based on IT2FLC has the ability to provide a past response to avoid the obstacles occurrence, while the follower is able to keep maintain the trajectory with the leader and fast response to follow the leader in the smooth trajectory. In this research, proposed IT2FLC is testing in an unstructured environment such as a circle, and oval. As seen in Figure 7 and Figure 8, respectively, only IT2FLC does not hit the wall, while the other controllers crash the wall. However, the number of collisions using T1FLC with 8 rules more than 27 rules. It's happened, due to the IT2FLC has the ability to changes in the position and orientation of the leader robot against the dynamic environment in the interval value. It more robust than T1FLC with one value. In addition, the shape of the resulting trajectory was more smooth.

4.2 Fuzzy Logic Controller and PID controller

To find the trajectory tracking error that will be the input on the PID is to subtract the reference position equation with the actual position equation. After that the equation of position error and orientation obtained as follows,

$$x(t) = G_P . e_{nx}(t) + G_I \int_0^t e_{nx}(t) + G_D \frac{de_{nx}(t)}{dt} \tag{24}$$

$$y(t) = G_P . e_{ny}(t) + G_I \int_0^t e_{ny}(t) + G_D \frac{de_{ny}(t)}{dt} \tag{25}$$

$$\theta(t) = G_P . e_{n\theta}(t) + G_I \int_0^t e_{n\theta}(t) + G_D \frac{de_{n\theta}(t)}{dt} \tag{26}$$

PID controller calculates an error value as the difference between a measured variable and desired set-point. The controller attempts to minimize the error by adjusting the process control inputs. The PID controller calculation involves three separate constant parameters G_P is proportional gain, G_I integral gain, and G_D derivative gain. To use this method, each gain should be obtained by several tuning methods for improving the performance. However, PID controllers tuning is difficult, it is a difference with the fuzzy system, by using human cognitive and decision-making process, the developing and tuning of the FIS is more intuitive than the PID controller. In this paper, Ziegler Nichols is utilized to determine the gain of PID.

The performance of the proposed IT2FLC compare to PID controller is depicted in Figure 9. The both of controller can move to the target, but by using PID the trajectory of the leader-follower robot is not smooth because, it controls the angle of the robot bearing with a coarse value resembling the on-off value, while IT2FLC changes the bearing angle with smooth value and it produce stable velocity. Therefore, the trajectory becomes smoother. It can be seen from the simulation in Figure 10 to Figure 12, the leader-follower robot based-on IT2FLC with less rule can provide better control compare to PID controller. It doesn't require a plant model and has better disturbance rejection properties. Fuzzy logic is well suited to processes with poor known dynamics and uncertainties and it compensates by adding the knowledge and experience from personnel related to the process using fuzzy rules. PID controller does not vary with the different stages, namely behaves the same magnitude over the entire operating range, but the two systems are effective against disturbances in dynamic environment.
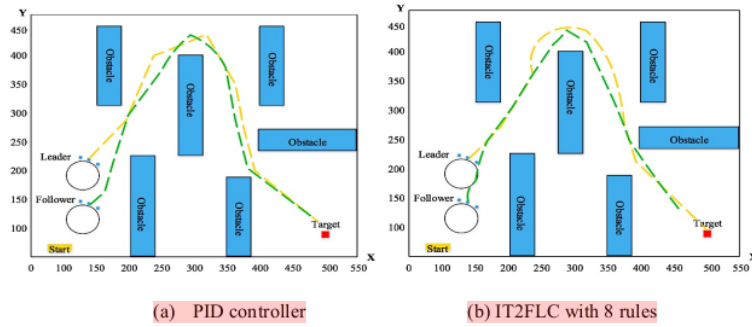
(a)   PID controller   (b) IT2FLC with 8 rules

Fig. 9. PID controller and IT2FLC performance with 6 obstacles

(a)   Linear velocity   (b) Angular velocity

Fig. 10. PID controller
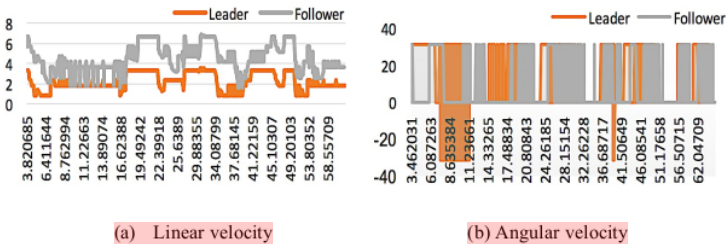

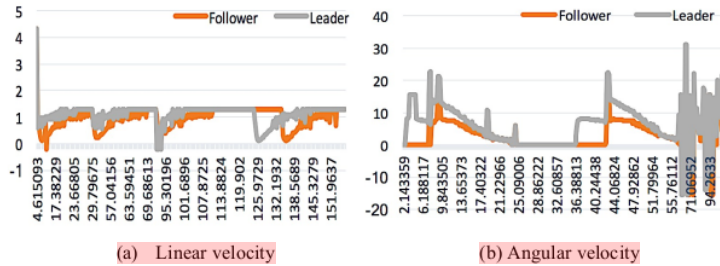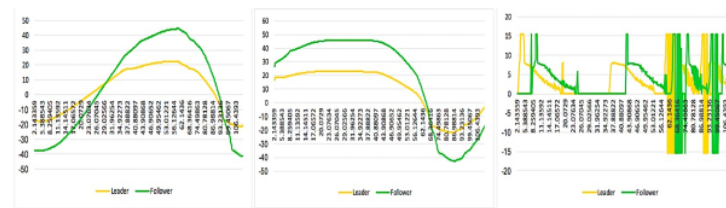(a)   Linear velocity   (b) Angular velocity

Fig. 11. IT2FL controller

(a)   The sample of PID response for x(t), y(t) and $\theta(t)$

(b)   The sample of IT2FLC response for x(t), y(t) and $\theta(t)$

Fig. 12. PID Controller and IT2FLC performance in complex environment

From Figure 10 and Figure 11, the output of both controller in terms of angular velocity and linear velocity have different performance. Based on the IT2FLC produces a velocity with gradually changes in a smoother value, with a max amplitude of about 23 rpm for angular velocity. In contrast to PID controller with gain settings that have been done manually, it changes in incremental velocity with more rough value decrease with max higher amplitude around 30 rpm. The changes value of angular velocity and linear velocity will greatly affect position and orientation of the leader-follower robots (see Figure 12). The interval value of the IT2FLC membership function has a significant effect on the formation of the leader-follower robot, it generates the position and orientation in average values, in contrast to the PID controller which has only one output value as the control parameter of the robot movement. By using IT2FLC the changes in robot positions are smoother with more data generation about 106 for every cycle changes in x (t), y (t) and $\theta(t)$. In contrast to the PID controller data generated only 61 data for a single cycle of movement and changes in the values of x (t), y (t) and $\theta(t)$ which are not as good as the Fuzzy controller. However, both controllers are able to deal with complex environmental conditions and effective against disturbances in the complex environment. From Figure 13 presents the IT2FLC in a square environment has the ability to maintain the distance between an obstacle and move to the target with smooth movement, but PID controller produces the rough movement due to, the velocity of the robot change not gradually but drastically with a large interval.
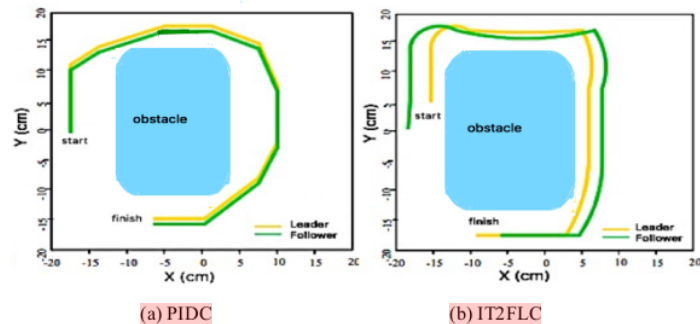
(a) PIDC   (b) IT2FLC

Fig. 13. Leader-follower tracking performance in square environment

4.3 Robust Formation

To achieve the robustness against the fault of the leader, the proposed fuzzy controller is tested in some situation is shown in Figure 14. As can be seen in Figure 14 (a) and (c) the leader crash with the obstacle and stop moving. The follower still moving, but it becomes unstable because the follower doesn't know the global information about the environment. The follower move based on position and orientation of the leader. When the leader failed, the follower move in new direction. However, in Figure 14 (b), the different situation occurs. The leader and the follower are crashed. They can't continue to finish the work to reach the finish point. All results based on the T1FLC and the performance not good. Due to the T1FLC can't overcome the dynamic environment, the controller only one MFs value, when it works out of the value the performance of the controller is decreased. However, the IT2FLC has the FOU, it can be overcome the uncertain situation better than the T1FLC. The upper MFs and the Lower MFs have some values to produce the interval of environmental uncertainty. It can be seen in Figure 14 (d), when the leader crashes the obstacle, the follower can overcome the situation and move to the target without collision.

As the same results are found in the complex environment. By using the T1FLC with 8 rules and 27 rules is shown in Figure 15 (a) and (b), the follower robot can't finish the task move to the target, due to the leader fails and disappear. In the one value of MFs, the leader hasn't a decision, because it crashes with the obstacle. The information about the position and orientation become zero. Therefore, the follower not stable or it moves a new direction with its own decision. However, if the IT2FLC is used (Figure 15 (c)), the follower has some the MFs value as uncertainty to overcome the zero information from the leader. It can be created if environmental is changed over time by using FOU. The follower can make the own decision by using the information that has from the fuzzy controller, and it reconstructs the position and orientation to the target with respect to the replaced leader and it only utilize 8 rules.
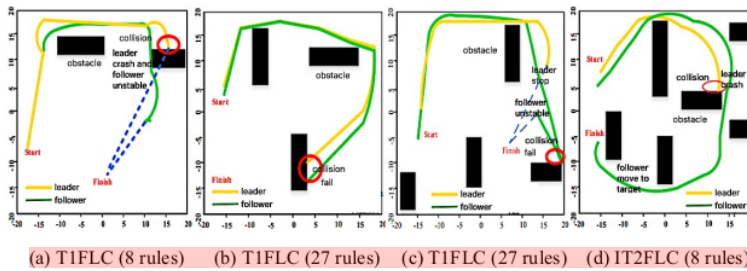
(a) T1FLC (8 rules)   (b) T1FLC (27 rules)   (c) T1FLC (27 rules)   (d) IT2FLC (8 rules)

Fig. 14. Formation tracking control in several environments

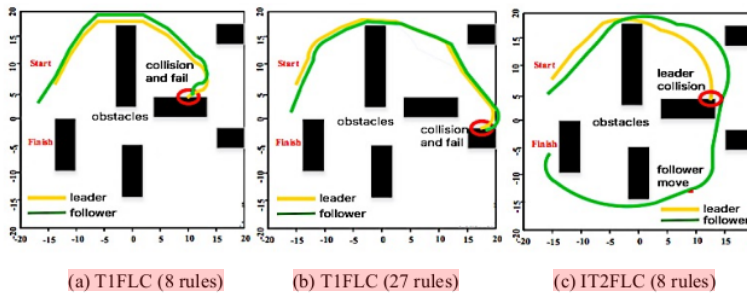(a) T1FLC (8 rules)   (b) T1FLC (27 rules)   (c) IT2FLC (8 rules)

Fig. 15. Tracking formation controller performance in complex environment

The summary of research results of tracking formation in the leader-follower implementation can be seen in Table 2.

Table 2. Fuzzy controller performance in the leader-follower application

Environment	Data (total)		Time (sec)	
	T1FLC	*IT2FLC*	*T1FLC*	*IT2FLC*
Circle condition	1169	486	Fail in 63.9	111.7
Oval condition	360	305	Fail in 20.7	93.5
Six obstacles	1086	415	62.7	156.8
Open-space	1074	836	86.3	107.8

5   Conclusion

By using a proposed IT2FL algorithm, the follower robot can replace the lost (or broken) leader using only 8 rules and it allocates the follower robot to the formation using local information. The multi-robot system employing the proposed algorithm achieves robustness against the member robot faults, decentralized position allocation without high-cost optimization, fast response, small resources and reduction of the moving cost. In this work, we have considered only the kinematics of the differential-drive mobile robot, therefore in future, we will include the dynamics as it is well known that due to a non-holonomic constraint of the differential drive mobile robot, the perfect velocity tracking will not hold, we will have to consider the torque as well. Moreover, we have provided not only simulation results but also the implementation in a real robotic system, to ensure the feasibility and performance of the proposed algorithms in the real multi-robot systems.

6   Acknowledgment

7   References

[1] S. Nurmaini and B. Tutuko, "Intelligent Robotics Navigation System: Problems, Methods, and Algorithm," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3711–3726, 2017.

[2] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *International workshop on swarm robotics*, 2004, pp. 10–20.

[3] J. Pugh and A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, 2007, pp. 332–339.

[4] L. Chaimowicz, V. Kumar, and M. F. M. Campos, "A paradigm for dynamic coordination of multiple robots," *Auton. Robots*, vol. 17, no. 1, pp. 7–21, 2004. https://doi.org/10.1023/B:AURO.0000032935.30271.a5

[5] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Auton. Robots*, vol. 4, no. 4, pp. 387–403, 1997. https://doi.org/10.1023/A:1008814708459

[6] P. Seiler, A. Pant, and K. Hedrick, "Disturbance propagation in vehicle strings," *IEEE Trans. Automat. Contr.*, vol. 49, no. 10, pp. 1835–1842, 2004. https://doi.org/10.1109/TAC.2004.835586

[7] A. Renfrew, A. Gonzalez-Villasenor, and P. Brunn, "Contact bridging dampers for back-up on automated highways," *IET Intell. Transp. Syst.*, vol. 1, no. 2, pp. 167–175, 2007. https://doi.org/10.1049/iet-its:20060054

[8] F. Fahimi, "Sliding-mode formation control for underactuated surface vessels," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 617–622, 2007. https://doi.org/10.1109/TRO.2007.898961

[9] Y. Gu *et al.*, "Design and flight testing evaluation of formation control laws," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 6, pp. 1105–1112, 2006. https://doi.org/10.1109/TCST. 2006.880203

[10] C.-K. Ryoo, Y.-H. Kim, and M.-J. Tahk, "Optimal UAV formation guidance laws with timing constraint," *Int. J. Syst. Sci.*, vol. 37, no. 6, pp. 415–427, 2006. https://doi.org/10.1080/00207720500438340

[11] W. Ren and R. W. Beard, "Formation feedback control for multiple spacecraft via virtual structures," *IEE Proceedings-Control Theory Appl.*, vol. 151, no. 3, pp. 357–368, 2004.

[12] M. Juliá, A. Gil, and O. Reinoso, "Searching dynamic agents with a team of mobile robots," *Sensors*, vol. 12, no. 7, pp. 8815–8831, 2012. https://doi.org/10.3390/s120708815

[13] Q.-H. Meng, W.-X. Yang, Y. Wang, and M. Zeng, "Collective odor source estimation and search in time-variant airflow environments using mobile robots," *Sensors*, vol. 11, no. 11, pp. 10415–10443, 2011. https://doi.org/10.3390/s111110415

[14] R. J. Moreno, O. Aviles, and R. D. H. Beleño, "Humanoid Robot Cooperative System by Machine Vision," *Int. J. Online Eng.*, vol. 13, no. 12, pp. 162–173, 2017. https://doi.org/10.3991/ijoe.v13i12.7594

[15] Y. Yu, X. Chen, Z. Lu, F. Li, and B. Zhang, "Obstacle avoidance behavior of swarm robots based on aggregation and disaggregation method," *Simulation*, vol. 93, no. 11, pp. 885–898, 2017.

[16] S. Nurmaini and B. Tutuko, "Motion coordination for swarm robots," in *ICT For Smart Society (ICISS), 2014 International Conference on*, 2014, pp. 312–315.

[17] W. Zheng and Y. Jia, "Leader-follower Formation Control of Mobile Robots with Sliding Mode," *J. Robot. Netw. Artif. LIFE*, vol. 4, no. 1, pp. 10–13, 2017. https://doi.org/10.2991/jrnal.2017.4.1.3

[18] Y. Huang, Y. Zhang, Q. Zhao, and J. Cheng, "Research on Multi-robot Cooperation Based on Leader-follower Method," 2015.

[19] A. Loria, J. Dasdemir, and N. A. Jarquin, "Leader--follower formation and tracking control of mobile robots along straight paths," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 727–732, 2016. https://doi.org/10.1109/TCST.2015.2437328

[20] X. Jin, "Fault tolerant finite-time leader--follower formation control for autonomous surface vessels with LOS range and angle constraints," *Automatica*, vol. 68, pp. 228–236, 2016. https://doi.org/10.1016/j.automatica.2016.01.064

[21] X. Li, J. Xiao, and Z. Cai, "Backstepping based multiple mobile robots formation control," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 887–892.

[22] L. C. A. Pimenta *et al.*, "Swarm coordination based on smoothed particle hydrodynamics technique," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 383–399, 2013. https://doi.org/10.1109/TRO.2012.2234294

[23] L. Barnes, M. Fields, and K. Valavanis, "Unmanned ground vehicle swarm formation control using potential fields," in *Control & Automation, 2007. MED'07. Mediterranean Conference on*, 2007, pp. 1–8.

[24] D. V Dimarogonas, P. Tsiotras, and K. J. Kyriakopoulos, "Leader--follower cooperative attitude control of multiple rigid bodies," *Syst. Control Lett.*, vol. 58, no. 6, pp. 429–435, 2009. https://doi.org/10.1016/j.sysconle.2009.02.002

[25] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009. https://doi.org/10.1016/j.automatica.2009.06.026

[26] B. Calli, W. Caarls, P. Jonker, and M. Wisse, "Comparison of extremum seeking control algorithms for robotic applications," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3195–3202.

[27] R. Palm and A. Bouguerra, "Particle swarm optimization of potential fields for obstacle avoidance," in *Recent advances in robotics and mechatronics*, 2013, pp. 117–123.

[28] S. Nurmaini, "Modeling of mobile robot system with control strategy based on type-2 fuzzy logic," *Int. J. Inf. Commun. Technol. Res.*, vol. 2, no. 3, pp. 235–242, 2012.

[29] Z. Peng, G. Wen, and A. Rahmani, "Leader-follower formation control of multiple nonholonomic robots based on backstepping," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 211–216.

[30] S. Nurmaini, S. Zaiton, and D. Norhayati, "An embedded interval type-2 neuro-fuzzy controller for mobile robot navigation," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 4315–4321.

[31] C. Wagner and H. Hagras, "Evolving type-2 fuzzy logic controllers for autonomous mobile robots," in *Analysis and design of intelligent systems using soft computing techniques*, Springer, 2007, pp. 16–25. https://doi.org/10.1007/978-3-540-72432-2_3

[32] J. Shao, G. Xie, J. Yu, and L. Wang, "Leader-following formation control of multiple mobile robots," in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, 2005, pp. 808–813.

8   Authors

Bambang Tutuko is a lecture at Computer Engineering Department, Faculty of Computer Science, Sriwijaya University.

Siti Nurmaini is a lecture at Computer Engineering Department, Faculty of Computer Science, Sriwijaya University, Indonesia. She often works as General Chair for international conferences (ICECOS) and Editor in Chief of International Journal Computer Engineering and Applications.

Saparudin is a lecture at Informatics Engineering Department, Faculty of Computer Science, Sriwijaya University.

Gita Fadila Fitriana is a magister student at Informatics Engineering, Faculty of Computer Science, Sriwijaya University, Indonesia.

Enhancement of Non-Holonomic Leader-Follower.pdf