FUZZY COOPERATIVE CONTROL FOR MULTIPLE MOBILE ROBOTS

Authors: Salman Zaffar[1], Hussain Aftab[1]

[1]DHA Suffa University
Karachi, Pakistan, salmanfar@gmail.com (Salman Zaffar), hussain.aftab@gmail.com (Hussain Aftab),

Received: 22-June-2022 / Revised: 05-July-2022 / Accepted: 05-July-2022
Karachi Institute of Economics and Technology || Technology Forces Journal, Volume 4, Issue 1, 2022

ABSTRACT

This work focuses on the formation control for a team of mobile robots. Path following and cooperation seems an easy task from a human point of view, however for a robot it requires complex control algorithms. Therefore to mimic the human way of thinking, the proposed control algorithm is based on fuzzy logic which closely matches the human thought process. Hierarchal control architecture is used with two levels of controllers which are fuzzy and proportional integral and derivative (PID) based controllers. Fuzzy controller is the high level controller and it perform two tasks, path following and cooperation between robots, whereas the PID controller is responsible for the accurate tracking of speed for the robot wheel motors. Every robot has its own path that is pre-determined by the trajectory planner, as the robot move along their respective path the control algorithm adjust their linear and angular velocities such that they move in formation and reach their destinations at same time regardless of the length and curvature of path. The proposed control scheme is implemented on simulation and the results are obtained.

Keywords: Fuzzy Cooperative Control, Mobile Robots.

1. INTRODUCTION

Formation control of multi mobile robots has been a topic of interest for past several years. A lot of research is being done on how to effectively control multiple mobile robots (MMR) to perform a collective task. The idea of formation control is universal and applies to almost all kinds of robots. For example, a fleet of autonomous air vehicles use the formation control algorithms to test the aerodynamics of such vehicles. Control of Multi mobile robots have a vast range of applications such as unmanned air vehicles [1 –3], autonomous ground vehicles [4-5], mine sweepers [6] etc.

The primary motivation of using multi mobile robots is to improve the overall effectiveness of the system. MMRs can easily perform difficult task which a single robot or a team of independent (Non – Cooperative) robots cannot perform such as to carry a heavy object. Furthermore, MMRs are cost effective in comparison of building one powerful robot several; identical small robots are much more cost effective and efficient.

Cooperative control of mobile robot is a challenging task, in this problem the robots are subjected to both path following and formation maintaining constraints. The idea of real time path following and formation control seems easy for humans, but for a robot it is a difficult task specially to decide between the individual and group goals.

The idea is to make such a model which can perform individual robot path following simultaneously maintaining an inter robot formation. The objective is that the robots should be able to follow their respective paths as well as maintain an inter robot formation. The robots may have different paths with different lengths and curvatures still the robots should track their path, maintain formation and reach the target at the same time regardless of the path length and their initial position.

The goal was to make use of the human thought process for the cooperative behavior for this purpose several things needs to be considered. The robot should be able to track the path and send its location to the controller; therefore the robot must have localized feedback. Several approaches are being used for localization of a robot in its environment [7-9].

To achieve cooperation between robots three different models were studied to test their performance and applications.

- Behavior based model
- Leader follower method
- Virtual structure

In behavior based model the robot consist of several behaviors, each of which is responsible for a specific situation [10-12]. The problem with this type of model is that it does not support decomposition of task or the modeling of sub task. Furthermore it difficult to achieve precise cooperative control as it is mathematical analysis is difficult.

In the leader follower method one of the robots is the leader and the others are the followers [13]. In this scheme the leader is the one which responsible for the formation, all other robots are supposed to follow the leader and maintain a certain distance (relative position) from the leader. The disadvantage of this scheme is that there is no feedback available from the follower robots and hence if anyone of them encounters an obstacle then it is difficult to keep up with the formation.

In virtual structure approach the system consider the team of mobile robots as a rigid structure (virtual structure) [14-15]. This approach is easier to implement as it all the robots are part of virtual structure hence the cooperation is easier, but it can be also a point of failure.

Due to its simplicity and ease of implementation the virtual structure approach was selected. However, to compensate for the weakness of this method a more streamlined control architecture was required.

Background research for the control algorithm reveals that the control algorithm used for cooperative control of mobile robots are based on Nonlinear Control techniques which often require complex mathematical equations to be implemented, which are difficult to be realized in a microcontroller environment, therefore Fuzzy control was selected as it mimics human logic and it is easier to implement and it has fast decision making process which makes it an ideal candidate for multiple mobile robots scenario.

The model used in this thesis is based on the virtual structure with a fuzzy controller. There has been a lot of work being done artificial intelligence for path following of mobile robots. The fore study of path following techniques tells that Neural Network and Fuzzy Logic Controllers are commonly used for path following of mobile robots for both “Known” and “unknown” environments [16-19]. There are researches which exploited both these techniques also known as neurofuzzy controllers [20-21]. However, most of these researches were carried out on single mobile robots but with a little modification these can be implemented in a cooperative control environment.

It was decided to implement a Hierarchal Control structure in which there are two different hierarchies namely High level Fuzzy Controller and Low Level PID controller. The Fuzzy Controller is the main controller which is responsible for the cooperation, path following and localization of robots. While Low level controller is a PID controller which is responsible for the speed tracking for each of the robot's wheel motor.

The organization of this paper is as follows, section 2 presents the modeling of a differential drive robot along with the system constraints. Section 3 presents the control architecture, defines the cooperative control problem and discuss in detail the fuzzy cooperative controller. In section 4 simulation results are presented and section 5 concludes the paper.

2. MODELING OF DIFFERENTIAL DRIVE ROBOT

The kinematic model of a differential drive robot is given in(1). Where $v(t)$ is the linear velocity and $\omega(t)$ is the angular velocity of the robot while $\varphi(t)$ is the orientation of the robot.$v(t)$ and $\omega(t)$ are the control variables.

$$\begin{cases} \dot{x}(t) = v(t)\cos\varphi(t) \\ \dot{y}(t) = v(t)\sin\varphi(t) \\ \dot{\varphi}(t) = \omega(t) \end{cases} \tag{1}$$

Fig. 1 shows a differential drive robot with two wheels mounted on a common axis. If the wheels are rotating considering there is no lateral slip then there exist a point ICC (instantaneous center of curvature) provided that ($v_r \neq v_l$). Where v_r and v_l are the linear velocities of the right and left wheels respectively. By varying v_r and v_l we can change the ICC and hence we can control the movement of the robot. R is the distance from the ICC to the center of the two wheels which is also considered as the center of the robot. L is the distance between the two wheels. And θ is the orientation of the robot from x axis.

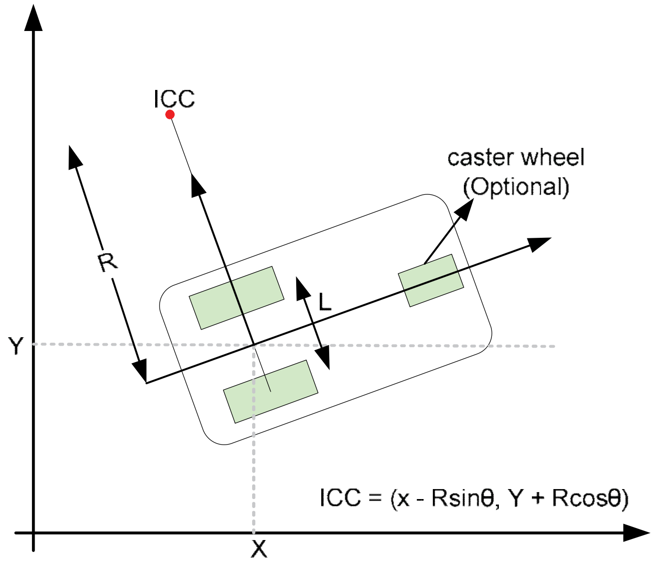

Figure 1: Differential drive mobile robot

We can write the equation for v_r and v_l using the relationship $v = R\omega$.

$$v_r = (R + \frac{L}{2})\omega$$
$$v_l = (R - \frac{L}{2})\omega \tag{2}$$

Solving (2) for R and ω and using the relation $v = R\omega$ we get the following relations.

$$\omega = \frac{(v_r - v_l)}{L}$$
$$v = \frac{(v_r + v_l)}{2} \tag{3}$$

To obtain v_r and v_l in terms of v and ω we can rewrite (3) as follows.

$$v_r = \frac{(2v + \omega L)}{2}$$
$$v_l = \frac{(2v - \omega L)}{2} \tag{4}$$

Thus the mathematical model of the differential drive robot can be implemented by using(1), (3) and(4).

The differential drive robot is subject to nonholonomic constraints such as it cannot move lateral along is axel. This can be mathematically written as.

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{5}$$

Similarly the robot's linear and angular velocities are also bounded by the following constraints.

$$|v(t)| \leq v_{\max} \qquad |\omega(t)| \leq \omega_{\max}$$
$$|\dot{v}(t)| \leq \dot{v}_{\max} \qquad |\dot{\omega}(t)| \leq \dot{\omega}_{\max} \tag{6}$$

3. COOPERATIVE CONTROL STRUCTURE

Fig. 2 shows the control structure. The proposed scheme comprises of two level of control namely.

- High level fuzzy cooperative controller
- Low level PID controller

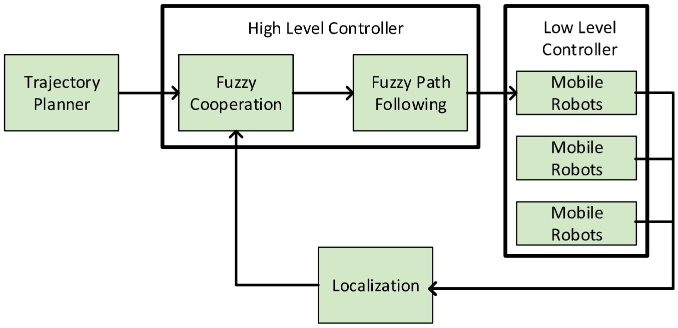

Figure 2: Control structure

The trajectory generator is responsible for generating paths for the robots, which is the reference for fuzzy controller. The output of fuzzy controller is used as a reference for low level PID controller. The low level controller is responsible for accurate tracking of robot's left and right wheel motors whereas fuzzy controller is responsible for path following of individual robot as well as cooperation between robots. Feedback is collected from robot's wheel motors in the form of speed and then a localization algorithm is used to locate each robot's position and orientation which is then sent back to fuzzy controller.

3.1 High Level Fuzzy Control

The High Level Fuzzy Controller is a Takagi sugeno type fuzzy controller which is responsible for path following of individual mobile robots as well as group cooperation between robots. The defuzzification scheme used in the fuzzy controller is weighted average. Fig. 3 shows the Fuzzy Control structure.

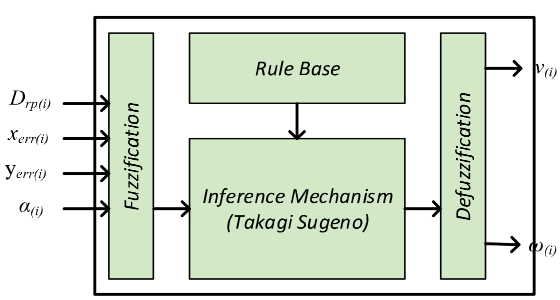

Figure 3: Fuzzy control structure

Refer to Fig. 4: $i = 1, \ldots, k$ is the robot number. The position and orientation of the robot can be described by the vector $P_i = [x_i, y_i, \varphi_i]^T$.The path to be followed by robots is divided into n discrete set of points, where $n = 0, \ldots, f$. Each point on the path can be described by the vector $Q_{di(n)} = [x_{di(n)}, y_{di(n)}, \xi_{di(n)}]^T$. Where $Q_{di(0)}$ being starting point of the path and $Q_{di(f)}$ being the final point. Whereas $Q_{di(n)}$ represent the nth sample point on the path.

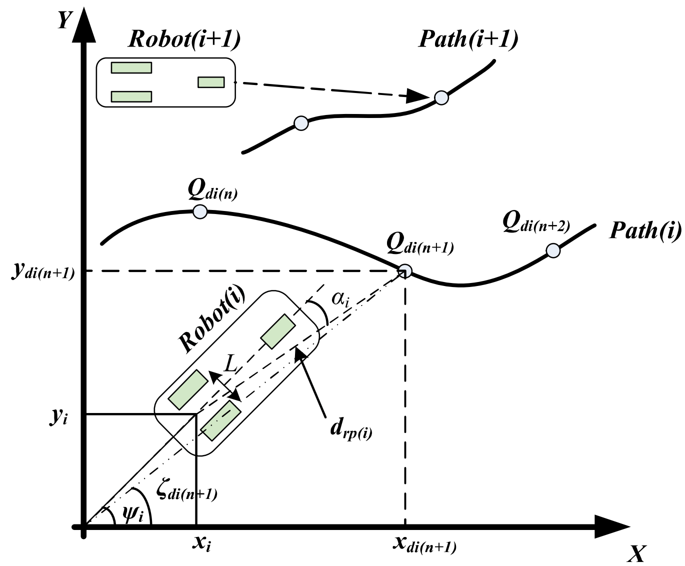

Figure 4: Robot path following parameter

The Inputs to the Fuzzy Controller are $D_{rp(i)}$, $x_{err(i)}$, $y_{err(i)}$ and $\alpha_{(i)}$. Where

- ☐ $D_{rp(i)}$ is the distance of the actual position of the ith robot from its next desired point on the path.
- ☐ $x_{err(i)}$ is the error of the ith robot's position the X direction.
- ☐ $y_{err(i)}$ is the error of the ith robot's position in the Y direction.
- ☐ $\alpha_{(i)}$ is the error in orientation of ith robot from the next desired point.

$D_{rp(i)}$ is calculated using the distance formula as shown in the following equation.

$$D_{rp(i)} = \sqrt{(x_{di} - x_i)^2 + (y_{di} - y_i)^2} \qquad (7)$$

Where x_{di}, y_{di} are the coordinates of the next desired point on the path while x_i, y_i are the coordinates of the actual position of the robot.

$x_{err(i)}, y_{err(i)}$ and $\alpha_{(i)}$ are in the robot's reference frame and are calculated from the following equation.

$$\begin{bmatrix} x_{err(i)} \\ y_{err(i)} \\ \alpha_{(i)} \end{bmatrix} = \begin{bmatrix} \cos\varphi_{(i)} & \sin\varphi_{(i)} & 0 \\ -\sin\varphi_{(i)} & \cos\varphi_{(i)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{di} - x_i \\ y_{di} - y_i \\ \zeta_{di} - \varphi_i \end{bmatrix} \quad (8)$$

Where φ_i is the robots orientation which is adjusted to $\pm\pi$ radians. ζ_{di} is the orientation of the next desired point which can be calculated via using the following equation.

$$\zeta_{di} = tan^{-1}\left(\frac{y_{di}-y_i}{x_{di}-x_i}\right) \quad (9)$$

The Fuzzy controller generates the required linear and angular velocities v_i, ω_i of the robot. The output of the fuzzy controller is used as a set point the low level PID controller.

3.2 Low Level PID Control

The Low level control is a PID controller which is responsible for the accurate tracking of the Left and right wheel motors velocities $\omega_{r(i)}, \omega_{l(i)}$. Fig. 5 shows the structure of the low level PID controller.

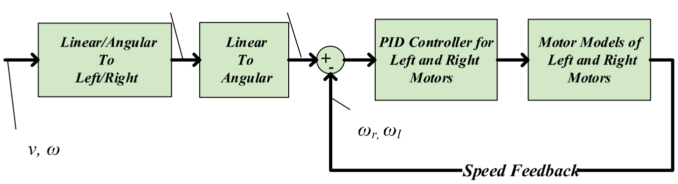

Figure 5: Low level PID

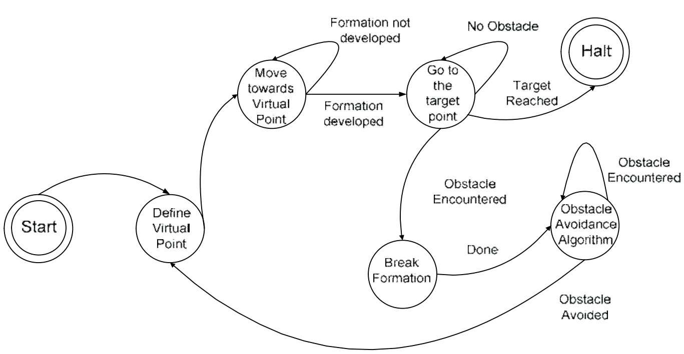

Figure 6: Finite state machine

The Output from the Fuzzy Controller is in the form of Linear and Angular velocities v_i, ω_i of Robot therefore the velocities must first be converted to Left and Right wheel motor velocities $v_{r(i)}, v_{l(i)}$. This conversion is carried out using (4)

The above conversion gives the linear velocities of the left and right wheel but the PID controller needs the angular velocity as reference. Therefore the angular velocities $\omega_{r(i)}, \omega_{l(i)}$ of the right and left motors can be obtained using the relation $v = r\omega$ where r is radius of the wheel.

The velocities $\omega_{r(i)}, \omega_{l(i)}$ are given to the PID controller as reference signal. The PID controller tracks these velocity references with desired accuracy and within certain time limits. The complete functionality of the High Level Fuzzy Cooperative Control is dependent on the assumption that the inner PID controller accurately tracks these velocity references.

3.3 Problem Description

The two main tasks of the robot is to follow a desired path as well as maintain a formation with other robots along the journey. The main idea behind the robot following a continuous path is to analyze the path in a set of discrete points. Each point serves as an intermediate destination to the robot, tracking each point will make the robot appear to be moving smoothly in a continuous path. The paths are being generated by the trajectory generator, the paths are modeled using a fifth order polynomial to generate reasonably difficult curved paths for the robots to follow.

The task of the robot is to move smoothly in a continuous path with best precision possible, passing through every point is not necessary but the robot should pass within the vicinity of a sampling point. The path following is divided into two categories.

1. When the robot is initially placed on the predefined path it then follows the path.
2. When the robot is initially placed away from the predefined path, it then moves forward to reach the path and track it.

As mentioned previously P_i being the Robot's position and Q_{di} being the next desired point we have considered v_i and ω_i as robot's linear and angular velocities. The objective of the path following controller is to generate an output such that the robots velocity $u_i = [v_i, \omega_i]^T$ tracks the desired reference velocity $u_{di} = [v_{di}, \omega_{di}]^T$. As the robot's velocity u_i tracks the desired reference velocity u_{di} the error minimizes i.e. $\|u_i - u_{di}\| \to 0$ which will eventually make $\|P_i - Q_{di}\| \to 0$

Now consider a team of mobile robots each of which having its own path following controller like the one described above, hence every individual robot will follow its desired path. For the cooperation problem every robot must follow their respective path in such a way that they maintain an inter robot formation along their journey as well as they must reach their final goal at the same time regardless of their path lengths.

3.4 Fuzzy Cooperative Controller

The Fuzzy Controller of Fig. 3 is responsible for the path following and cooperation of multiple mobile robots have two outputs v_i being the linear velocity and ω_i being the angular velocity for the robot to track. The fuzzy controller is based on Takagi - Sugeno Fuzzy Model. The control law equations are of the form.

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} f_1(D_{rp(i)}, x_{err(i)}, y_{err(i)}, \alpha_{(i)}) \\ f_2(D_{rp(i)}, x_{err(i)}, y_{err(i)}, \alpha_{(i)}) \end{bmatrix} \quad (10)$$

As mentioned previously, the task of the fuzzy controller is to make the robot pass within the vicinity of the desired sampling point if not through it. The controller is designed such that if the sampling points are placed close to each other, then the robot will move at a slower speed but with higher precision. However if the sampling points are placed far from each other, then the robot movement will be less precise but with higher speed.

The membership function of the input $D_{rp(i)}$ is shown in Fig. 7, it is the distance between robot's current position and the next desired point.

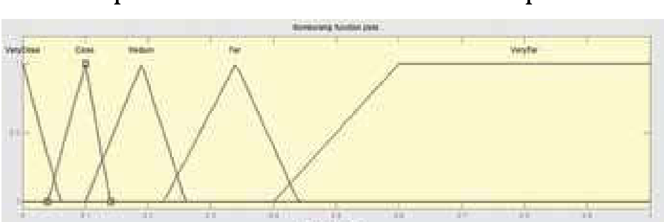

Figure 7: Membership function of $D_{rp(i)}$

The membership function of the input α_i is shown in Fig. 8. it is the error in orientation of robot from the next desired point.

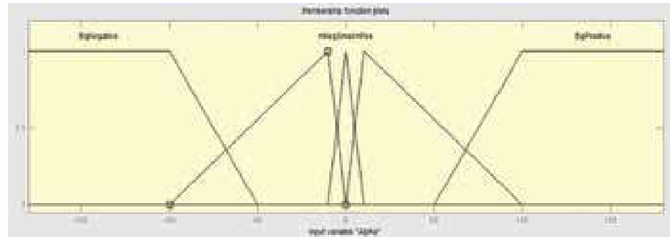

Figure 8: Membership function of α_i

The linear velocity obtained using $D_{rp(i)}$ and α_i is shown in Fig. 9. As can be seen from figure that linear velocity increases as the distance between robot and next sampling point increases and vice versa. However, α_i input has minimal effect on linear velocity.

The angular velocity profile obtained related to the inputs $D_{rp(i)}$ and α_i is shown in Fig. 10, it can be seen that angular velocity is dominated by α_i as the error in orientation increases so does the angular velocity to overcome the error and vice versa.

One of the problems of cooperative path following is that all robots should reach their final goal at the same time. Let us consider a case where the robot is at position $P_i = [x_i, y_i, \varphi_i]^T$has to move from point $Q_{di(n)} = [x_{di(n)},\ y_{di(n)},\ \xi_{di(n)}]^T$ to $Q_{di(n+1)}$ this means that robot's next desired sampling point is $Q_{di(n+1)}$. If robot passes the point $Q_{di(n+1)}$ and $Q_{di(n+2)}$ then when moving to next step $Q_{di(n+1)}$ is left behind from the actual position of robot or mathematically $(x_i > x_{di} \Rightarrow x_{err(i)} < 0)$. This problem is explained in Fig. 4. To avoid the condition of robot moving backwards we use the input $x_{err(i)}$. The fuzzy rule base contains the following rule to avoid the the above mentioned condition.

Rule: If $x_i > x_{di}$ then robot will stop, until the condition $x_i < x_{di}$ is met

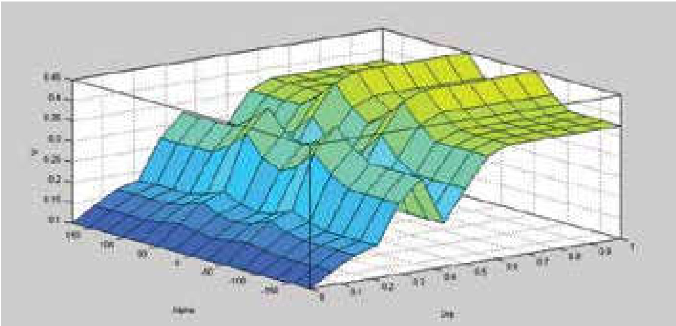

Figure 9: Linear velocity obtained by fuzzy controller

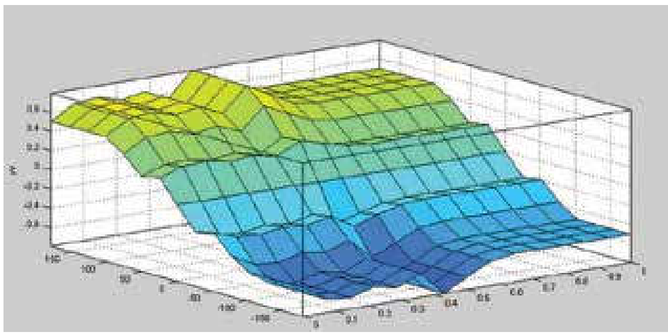

Figure 10: Angular velocity obtained by fuzzy controller

Another problem related to path following that need to be addressed, caused the addition of the input $y_{err(i)}$. If there is an error in the vertical position of robot $y_{err(i)}$ but not in the orientation then robot will travel parallel to the desired path and will never reach it. This can be seen from Fig. 15. The input $D_{rp(i)}$ has minimal effect on the angular velocity therefore it cannot turn robot towards the target point there should be an error in orientation α_i to turn the robot towards the target point.

In order to address above mentioned problem we introduced a new variable λ_i which is angle measured in degrees. λ_i is added to α_i if there is an error in y coordinate,it will help the robot to turn towards target point even if the $\alpha_i = 0$, it will tell the robot to turn towards the target point and will decrease as $y_{err(i)}$ decreases and will be zero when robot catch the desired path.

$$if\left|y_{err(i)}\right| > 0 \Rightarrow \alpha_i(new) = \lambda_i + \alpha_i \qquad (11)$$

The membership function of $y_{err(i)}$ is shown in Fig. 11. The relationship between $y_{err(i)}$ and λ_i is shown in Fig. 12, it can be seen that λ_i increases as $y_{err(i)}$ increases and vice versa, hence it makes sure that robot turn towards the target point if there exist an error in y-coordinate even when the error in orientation is zero.

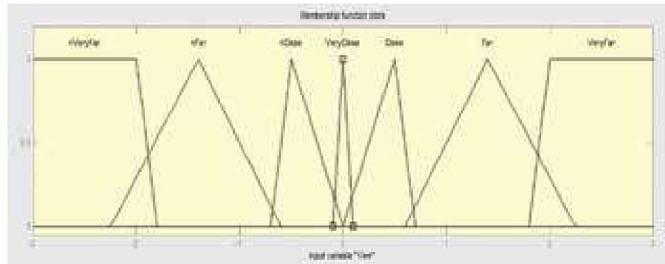

Figure 11: Membership fuction of $y_{err(i)}$

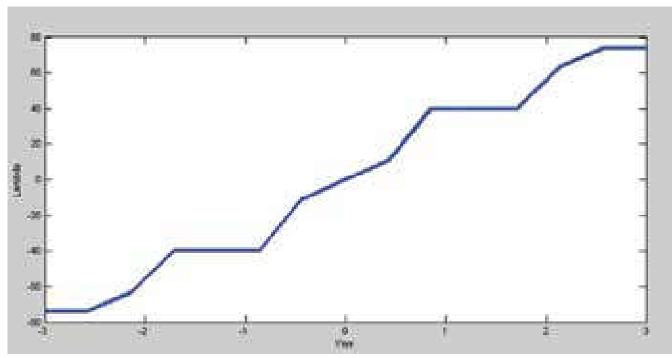

Figure 12: Relationship between $y_{err(i)}$ and λ_i

The path of each robot is divided into equal number of small segment and the parameter ς_i will keep track of the current segment the robot is executing. The robots are required to maintain $\varsigma_i = \varsigma_j$ for all i, j. For cooperation it is necessary that all paths are divided into equal number of segments regardless of their shape and length. If a path is longer than the sampling points will be far from each other or the distance between sampling point will increase and vice versa. Hence the robot will move faster or slower if the sampling points are close or far from each other respectively. Therefore if each robot tracks their respective sampling points then they will be moving in a cooperative behavior and will reach the final point at the same time. If a robot is ahead of its path then it will wait for other robots to catch up and then start following its path. This is achieved by x coordinate error input i.e if $x_{err(i)} < 0$ then robot will stop. The effect of

$x_{err(i)}$ and $D_{rp(i)}$ on linear velocity v_i is shown in Fig. 13 and the obtained angular velocity ω_i from the input $x_{err(i)}$ and α_i is shown in Fig. 14.

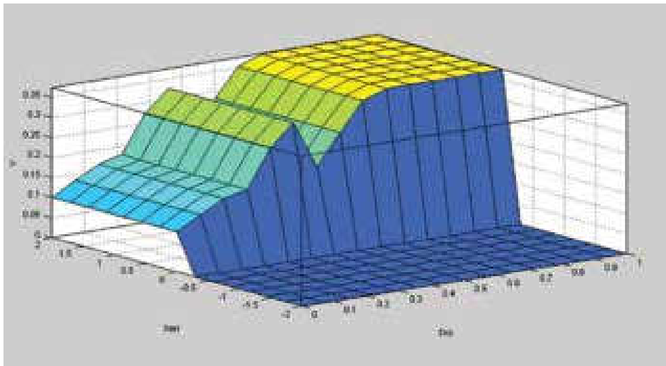

Figure 13: Linear velocity obtained from $x_{err(i)}$ and $D_{rp(i)}$

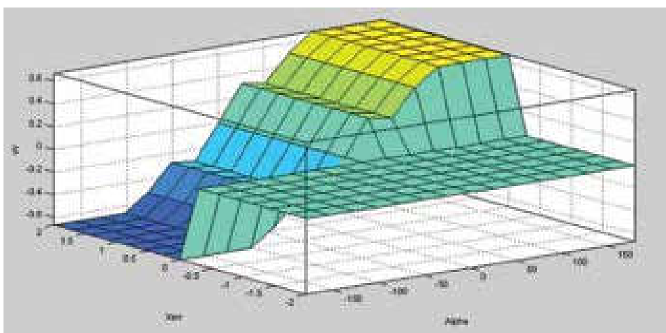

Figure 14: Angular velocity obtained from $x_{err(i)}$ and α_i

The Rule Base of the fuzzy controller is shown in Table 1. On the basis of these rules the fuzzy controller evaluates the inputs and generates the output. All the rules are very much self-explanatory like if the distance between the next sampling point and robot $D_{rp(i)}$ increase the fuzzy controller increases the linear velocity output so that the robot can catch its target. Similarly if the orientation error α_i increases then the fuzzy controller increases the angular velocity output to overcome the orientation error and correct the robot heading direction. Also if the robot is ahead of its path the Fuzzy controller will make both linear and angular velocity outputs zero to stop the robot.

Table 1: Fuzzy rule base

Rule 1	*If* $D_{rp(i)}$ *is* ***Very Close*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Very Very Slow***, ω_i *is* ***Zero***
Rule 2	*If* $D_{rp(i)}$ *is* ***Close*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Very Slow***, ω_i *is* ***Zero***
Rule 3	*If* $D_{rp(i)}$ *is* ***Medium*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Slow***, ω_i *is* ***Zero***
Rule 4	*If* $D_{rp(i)}$ *is* ***Far*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Fast***, ω_i *is* ***Zero***
Rule 5	*if* $D_{rp(i)}$ *is* ***Very Far*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Very Fast***, ω_i *is* ***Zero***
Rule 6	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***Big Negative*** *then* v_i *is* ***Very Very Slow***, ω_i *is* ***Big Negative***
Rule 7	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***Medium Negative*** *then* v_i *is* ***Very Very Slow***, ω_i *is* ***Negative***
Rule 8	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***Small*** *then* v_i *is* ***Very Very Slow***, ω_i *is* ***Zero***
Rule 9	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***Medium Positive*** *then* v_i *is* ***Very Very Slow***, ω_i *is* ***Positive***
Rule 10	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***Big Positive*** *then* v_i *is* ***Very Very Slow***, ω_i *is* ***Big Positive***
Rule 11	*if* $X_{err(i)}$ *is* ***Negative*** *then* v_i *is* ***Zero***, ω_i *is* ***Zero***

4. SIMULATION RESULTS

Table 2 shows the parameters of the robot and DC motor model.

The simulation is carried in Simulink with a fixed step size of 1e^{-2}. To test the cooperation problem we use a team of three identical mobile robot models, two different experiments were designed to test the cooperative algorithm. First the robots are tested with paths of different lengths but placed on the path initially, then the robots are given same path but are placed away from the paths, result are collected and discussed below.

Table 2: Parameters of the robot and DC motor model

Parameter	Value	Unit
Robot Model Parameters		
L_i	0.2	m
r	0.045	m
DC Motor Model Parameters		
R	3.07	*ohms*

L	0.04	mh
J	0.0294	$Kg.m^2$
B	0.0141	$N.m.s/rad$
K_e	0.65	$N.m/Amp$
K_t	0.65	$V.s/rad$

4.1 Experiment No. 1

In this experiment, three mobile robots are used each with a different path (different lengths). Robot 1 has the smallest path and Robot 3 has the longest path. The robots are initially placed on the path the initial positions of the robots and path lengths are given below.

$$P_1 = [x_1 \quad y_1 \quad \varphi_1]^T = \left[0.01 \quad 11.4303 \quad -\frac{\pi}{4}\right]^T$$
$$P_2 = [x_2 \quad y_2 \quad \varphi_2]^T = \left[0.01 \quad 6.7207 \quad -\frac{\pi}{4}\right]^T \quad (12)$$
$$P_3 = [x_3 \quad y_3 \quad \varphi_3]^T = \left[0.01 \quad 2.0489 \quad -\frac{\pi}{4}\right]^T$$

The robots are required to follow their respective paths as well as maintain a formation along the journey also they must reach their final destination at the same time regardless of the path lengths.

Fig. 15 shows the cooperation of the mobile robots, as can be seen that robot 1 has the shortest path whereas Robot 3 has the longest path. The round markers are used to highlight the position of each robot at a given time. The first markers on each path will indicate the initial position of the robots at t_0, the second marker will indicate the position of each robot at t_1 and so on. These markers are distributed equally with respect to time. As can be seen that the robot are placed vertically above each other initially, the successive markers shows that the robots are moving along their path but are also maintaining an inter robot formation throughout the journey, and reach their final destination at the same time.

Fig. 16 shows y_{err} or the tracking error, it can be seen that the robots quickly covers the initial tracking error due to the difference in initial orientation and then follows the path with minimal error.

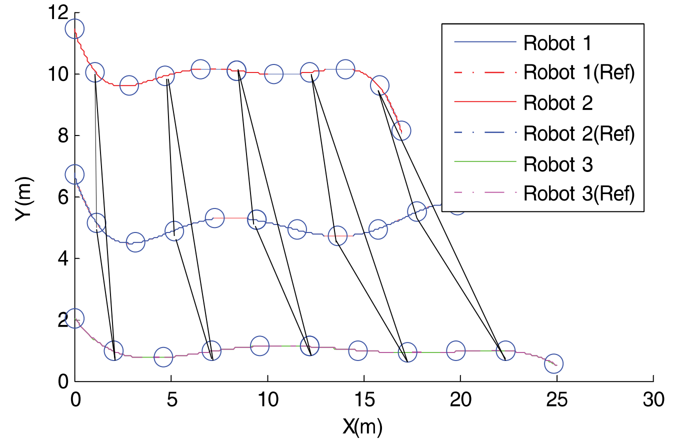

Figure 15: Cooperative path following (Experiment 1)

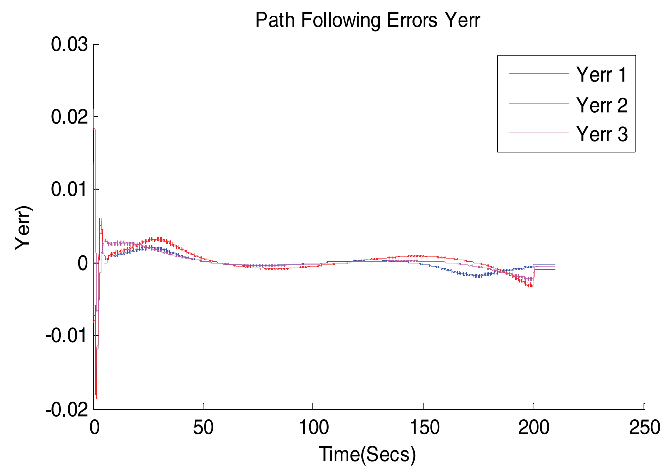

Figure 16: Tracking error

Fig. 17 and 18 shows the linear and angular velocities respectively. It can be seen that robot 1 has the lowest while robot 3 has the highest linear velocity. This is due to the fact that robot 1 has the smallest while robot 3 has the longest path. The angular velocity profiles shows the robots turning with respect to their path curves.

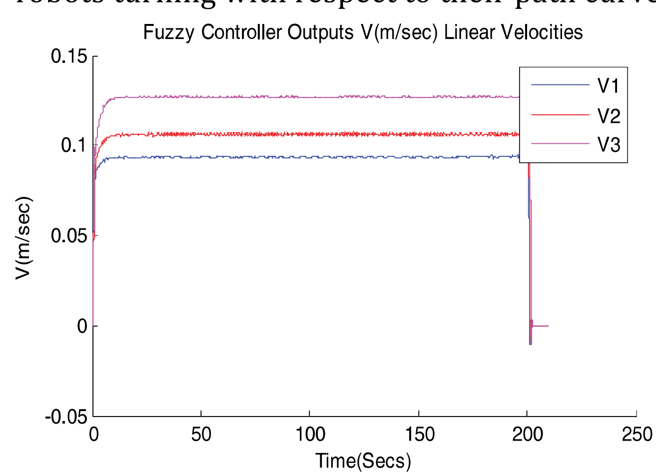

Figure 17: Linear velocity

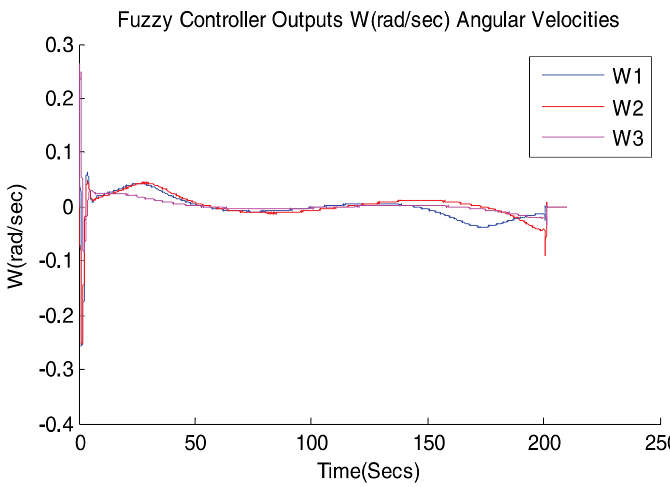

Figure 18: Angular velocity

4.2 Experiment No.2

In this experiment three robots are used, each of which is provided with the same path, but they are placed away from their respective path initially. Robot 1 and robot 3 are placed ahead of their trajectory while Robot 1 is placed behind its trajectory. The robots initial positions are given below.

$$
\begin{aligned}
P_1 &= [x_1 \quad y_1 \quad \varphi_1]^T = \left[1.0 \quad 11.0 \quad -\tfrac{\pi}{2}\right]^T \\
P_2 &= [x_2 \quad y_2 \quad \varphi_2]^T = [-0.5 \quad 8.0 \quad 0]^T \qquad (13) \\
P_3 &= [x_3 \quad y_3 \quad \varphi_3]^T = [1.5 \quad -1.0 \quad 0]^T
\end{aligned}
$$

The robots are required to follow their respective as well as maintain an inter robot formation. Since the robots are not placed on their respective paths, therefore first they need to catch the trajectory, and they must reach their respective end points at the same time. Fig. 19 shows the cooperation of robots in this experiment, it can be seen that Robot 1 and Robot 3 are placed ahead of their respective paths, while Robot 2 is placed behind the trajectory. Note that Robot 2 starts immediately as soon as the simulation starts while Robot 1 and 3 remains idle waiting for the trajectory to move forward. This is because of the assumption that if a robot is ahead of its path then it will wait for other robots to catch up.

The markers on the paths shows that around 3 meter mark, the robots achieve synchronization (formation), then maintains the formation as they move forward along their respective paths and reach their final destination at the same time.

Fig. 20 shows the tracking error y_{err} of the robots. Initially tracking error of Robot 1 and 3 is a function of the trajectory as the robots are standing still. However Robot 2 quickly overcomes its error. It can be seen that as soon as robots starts moving they overcomes y_{err} very quickly. Once the robots achieve formation they follow their respective paths with minimal tracking error.

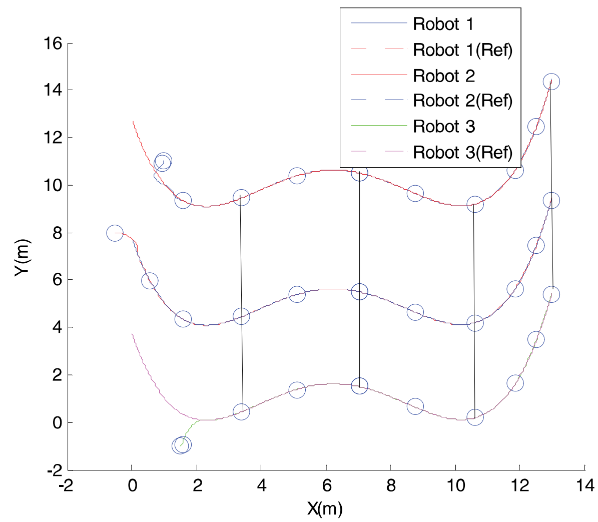

Figure 19: Cooperative path following (Experiment 2)

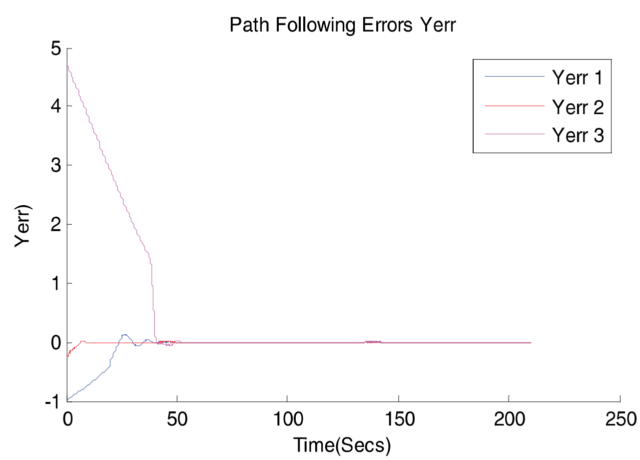

Figure 20: Tracking error

Fig. 21 shows the linear velocities of robots. From figure it can be seen that initially only Robot 2 (RED) is moving (has non zero linear velocity). This is because, Robot 2 is placed behind its trajectory so it has to move fast to catch the trajectory. After about 60 secs Robot 2 starts moving as its trajectory passes its initial position, similarly Robot 3 starts moving at about 90 secs. All three robots have to cover some distance as soon as they start moving to catch the trajectory therefore an initial peak can be seen for all three linear velocities. After catching the trajectory all the robots linear velocities becomes same, this is due to the fact that all the robots have same path similarly all the robots stop (Linear velocity zero) at the same time.

Fig. 22 shows the angular velocities of the robots. As discussed above initially only Robot 2 is moving (non zero angular velocity). The initial peaks in angular velocities represent the controller effort to correct the heading of the robot with respect to the trajectory. Once the robots achieve formation their angular velocities become similar because of same paths.

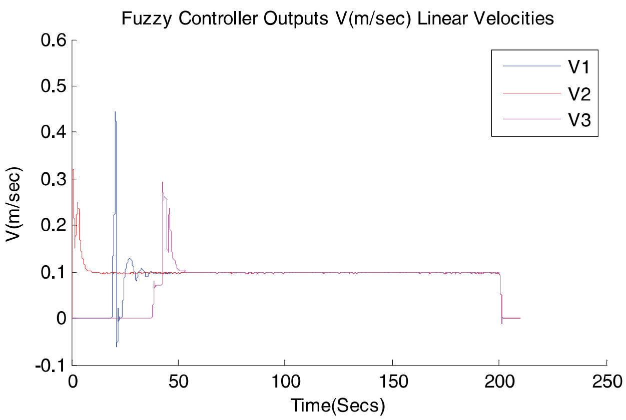

Figure 21: Linear velocities

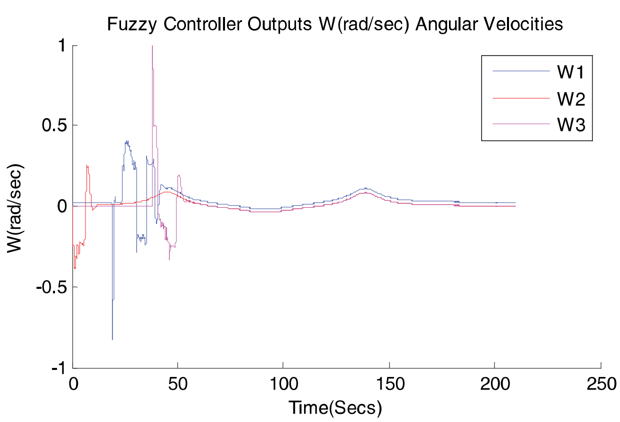

Figure 22: Angular velocities

5. CONCLUSION

This work covers the formation control of mobile robots. The proposed design consists of a hierarchal control architecture a low level PID controller and high level fuzzy cooperative controller. The PID controller tracks the reference speed of the robot's wheel motors; it was tuned for the motor parameters to effectively track the reference speed with minimum tracking time and steady state error.

The fuzzy controller is where the path following and cooperation is done. The cooperative behavior was testedusing a team of three identical robots were used. Different simulation experiments were conducted in which the robots were subjected to different initial conditions and they wer supposed to follow their respective paths and maintain a formation amongst them. The results shows that the Fuzzy Cooperative Controller effectively achieves its goals of individual path following and group cooperation, it has been shown that the robots tracks their respective paths with minimal error and maintain the formation while travelling.

This work is a small effort towards the Formation control of mobile robots. Recommended future work will be to improve the fuzzy controller by incorporating obstacle avoidance and the ability to move in known as well as unknown environments.

REFERENCES

[1] Beard, Randal W., Jonathan Lawton, and Fred Y. Hadaegh. "A coordination architecture for spacecraft formation control." IEEE Transactions on control systems technology 9, no. 6 (2001): 777-790.

[2] Seanor, Brad, Yu Gu, Marcello R. Napolitano, Giampiero Campa, Srikanth Gururajan, and Larry Rowe. "3-aircraft formation flight experiment." In Control and Automation, 2006. MED'06. 14th Mediterranean Conference on, pp. 1-6. IEEE, 2006.

[3] Beard, By Randal W., Timothy W. McLain, Derek B. Nelson, Derek Kingston, and David Johanson. "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs." Proceedings of the IEEE 94, no. 7 (2006): 1306-1324.

[4] Wu, Naiqi, and MengChu Zhou. "Modeling and deadlock control of automated guided vehicle systems." Mechatronics, IEEE/ASME Transactions on 9, no. 1 (2004): 50-57.

[5] Howard, A., L. Parker, and G. Sukatme. "Experiments with a large heterogeneous mobile robot team." Int. J. Robotics Research 25, no. 5-6 (2005): 431-447.

[6] Healey, Anthony J. "Application of formation control for multi-vehicle robotic minesweeping." In Decision and Control, 2001. Proceedings of the 40th IEEE Conference on, vol. 2, pp. 1497-1502. IEEE, 2001..

[7] Bevly, David M., and Bradford Parkinson. "Cascaded Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles." Control Systems Technology, IEEE Transactions on 15, no. 2 (2007): 199-208.

[8] Han, Soonshin, HyungSoo Lim, and JangMyung Lee. "An efficient localization scheme for a differential-driving mobile robot based on RFID system." Industrial Electronics, IEEE Transactions on 54, no. 6 (2007): 3362-3369.

[9] Purvis, Keith B., Karl J. Åström, and Mustafa Khammash. "Estimation and optimal configurations for localization using cooperative uavs." Control Systems Technology, IEEE Transactions on 16, no. 5 (2008): 947-958.

[10] Balch, Tucker, and Ronald C. Arkin. "Behavior-based formation control for multirobot teams." Robotics and Automation, IEEE Transactions on 14, no. 6 (1998): 926-939.

[11] Li, Hao, and Simon X. Yang. "A behavior-based mobile robot with a visual landmark-recognition system." Mechatronics, IEEE/ASME Transactions on8, no. 3 (2003): 390-400.

[12] Gu, Dongbing, and Huosheng Hu. "Integration of coordination architecture and behavior fuzzy learning in quadruped walking robots." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on37, no. 4 (2007): 670-681.

[13] Huang, Jiangyang, Shane M. Farritor, Ala′ Qadi, and Steve Goddard. "Localization and follow-the-leader control of a heterogeneous group of mobile robots." Mechatronics, IEEE/ASME Transactions on 11, no. 2 (2006): 205-215.

[14] Lewis, M. Anthony, and Kar-Han Tan. "High precision formation control of mobile robots using virtual structures." Autonomous Robots 4, no. 4 (1997): 387-403.

[15] Ghommam, Jawhar, Hasan Mehrjerdi, Maarouf Saad, and FaiçalMnif. "Formation path following control of unicycle-type mobile robots." Robotics and Autonomous Systems 58, no. 5 (2010): 727-736.

[16] Chang, Yeong-Chan, and Bor-Sen Chen. "Robust tracking designs for both holonomic and nonholonomic constrained mechanical systems: adaptive fuzzy approach." Fuzzy Systems, IEEE Transactions on 8, no. 1 (2000): 46-66.

[17] Li, Tzuu-Hseng S., Shih-Jie Chang, and Wei Tong. "Fuzzy target tracking control of autonomous mobile robots by using infrared sensors." Fuzzy Systems, IEEE Transactions on 12, no. 4 (2004): 491-501.

[18] Maalouf, Elie, Maarouf Saad, and HamadouSaliah. "A higher level path tracking controller for a four-wheel differentially steered mobile robot."Robotics and Autonomous Systems 54, no. 1 (2006): 23-33.

[19] Antonelli, Gianluca, Stefano Chiaverini, and Giuseppe Fusco. "A fuzzy-logic-based approach for mobile robot path tracking." IEEE Transactions on Fuzzy Systems 15, no. 2 (2007): 211-221.

[20] Mbede, Jean Bosco, Xinhan Huang, and Min Wang. "Robust neuro-fuzzy sensor-based motion control among dynamic obstacles for robot manipulators." Fuzzy Systems, IEEE Transactions on 11, no. 2 (2003): 249-261.

[21] Zhu, Anmin, and Simon X. Yang. "Neurofuzzy-based approach to mobile robot navigation in unknown environments." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37, no. 4 (2007): 610-621.