Hierarchical Fuzzy Cooperative Control and Path Following for a Team of Mobile Robots

Hasan Mehrjerdi, *Student Member, IEEE*, Maarouf Saad, *Senior Member, IEEE*, and Jawhar Ghommam

***Abstract*—In this paper, an intelligent cooperative control and path-following algorithm is proposed and tested for a group of mobile robots. The core of this algorithm uses a fuzzy model, which mimics human thought to control the robot's velocity, movement, and group behavior. The designed fuzzy model employs two behaviors: path following and group cooperation. Hierarchical controllers have also been developed based on fuzzy and proportional integral derivative to instruct the robots to move in a group formation and follow specific paths. As the robots move along individual predetermined paths, the designed algorithm adjusts their velocities so that the group arrives at their target points within the same time duration regardless of the length of each individual path. The fuzzy rules applied to the robots are defined by the kinematics limitation, which is bounded by both linear and angular velocities and the length and curvature of the individual paths. The experimental results of three mobile robots traveling on different paths are presented to show the accuracy of obtaining control and cooperation by using the fuzzy algorithm.**

***Index Terms*—Cooperation, fuzzy control, path following, unicycle wheeled mobile robots.**

I. Introduction

MULTIROBOTS have been studied as having many possible applications from grounded mobile robots [15], [16], to underwater autonomous vehicles [20], [21], aerial vehicles [17]–[19], and fleets of marines [22], [23]. There are many advantages of deploying more than one robot to solve a task and the motivation of using multimobile robots (MMR) is to improve task allocation, performance, and time [6]–[9].

The ability to control a group of mobile robots has long been an intriguing idea in robotics and artificial intelligence. The challenge has been to create a model that will exhibit cooperative behavior amongst multiple robots to move along designated paths and stay in a group formation. Although seemingly easy for human beings, the concepts of real-time path following and synchronized behavior are difficult tasks for robots. When formulating how to make "human thought processes" a reality in multirobotic modeling, several key concepts are considered necessary. The ability of a robot to track its path as well as transmitting its location data will therefore allow a trajectory to be planned between the robot's initial starting point and its target point. This model must also be able to control all the robots within the group to reach their target points both individually and in formation. To achieve this processing for a MMR group, it is necessary to have localized feedback sensors for each robot in the group. There have been researches using differing sensors and algorithms to localize a robot in its environment [32]–[34].

In order to design an algorithm that synchronizes multiple robot control, three common models were studied to evaluate their application and effectiveness. They are: 1) the behavior-based model [3], [5], [13]; 2) the leader–follower method [11]; and 3) the virtual structure model [12], [30]. There are differences in the control section or the implementation phases [14], [24]–[27], [35].

The "behavior"-based model employs several behaviors for each robot, but there is lack of modeling for the subtasks (task decomposition) or robot surroundings. This method is mathematically difficult to analyze, and therefore, it is hard to guarantee a precise formation control.

In the "leader–follower method," we have one robot nominated as the leader, while the other robots are nominated as followers. This method is both simple and reliable as the follower robots have only two tasks, that is, to keep a certain distance and relative position from the leader. However, there is no explicit feedback from the followers to the leader and that is the disadvantage of this method.

The "virtual structure" method treats the entire robot formation as a single virtual rigid structure. As the mobile robots are behaving as a single entity, this model exhibits a relative simplicity in the coordination of the whole group.

However, as this model treats the entire formation as a single virtual rigid structure, it can also be a common point of failure for the whole system.

Besides this, the authors considered this to be the best platform due to its simplicity, and therefore, it was decided to keep the "virtual structure" concept. However, it was considered that this model could be improved by incorporating a control architecture that was more streamlined.

Earlier study that explored algorithms for use with cooperative multiple robot control often used complex mathematical and nonlinear methods, therefore, making them hard to implement. The breakthrough came when it was decided to use "fuzzy logic" to simplify the computations used by the algorithm controller. Fuzzy logic mimics the way the human brain makes decisions

Manuscript received February 4, 2010; revised April 21, 2010; accepted June 12, 2010. Date of publication July 23, 2010; date of current version August 30, 2011. Recommended by Technical Editor P. X. Liu.

H. Mehrjerdi and M. Saad are with the Department of Electrical Engineering, École de Technologie Supérieure, Quebec University, Montreal, QC H2X 3X2, Canada (e-mail: hasan.mehrjerdi.1@ens.etsmtl.ca; maarouf.saad@etsmtl.ca).

J. Ghommam was with the Research Unit on Intelligent Control and Optimization of Complex Systems, Ecole Nationale d'Ingénieurs de Sfax (ENIS), 3038 Sfax, Tunisia. He is now with the Institut Nationale des Sciences Appliquées et de Technologies, 1080 Tunis Cedex, Tunisia (e-mail: jghommam@gmail.com).

Digital Object Identifier 10.1109/TMECH.2010.2054101

by grouping similar objects together, and therefore, creates fast and accurate response times in the decision making process. This has distinct advantages in MMR modeling, where multiple robots are moving along designated paths and simultaneously being directed with rapid velocity changes.

The model described in this paper uses a cooperative virtual model upgraded with a fuzzy-logic controller. Artificial-intelligence approaches are well cited for path following of mobile robot. Research evaluating both neural network approach or fuzzy logic controllers has been developed for path following in "known" and "unknown" environments [2], [4], [28], [29]. There are some researches with combinations of neural network and fuzzy, termed "neurofuzzy" applications [1], [10]. Though single robots have been the primary reference for much of the research that employs artificial intelligence, these methods can also be used to solve cooperative problems for multiple robots. Cooperation between robots is an issue, where robots can move on different paths with different lengths to shape special formations.

It was decided that the virtual fuzzy model to be used would consist of a two-level hierarchical structure in which "fuzzy logic" is the prime controller, and therefore, performs the tasks of path following, localization, and group cooperation. The secondary controller is a proportional integral derivative (PID) controller that ensures that the motor velocity to the robot's left and right front wheels is always accurate. The direction angle of each robot is determined by the desired paths and the formations of the MMR group. This cooperative method, derived from the fuzzy logic and PID, empowers the robots to move, follow, and coordinate their speed and direction on paths with differing lengths.

Experimental tests are performed with robots in various formations to verify the reliability of the fuzzy control and the cooperative algorithm. The control and path planning in this paper are implemented on multiple robots called ETsRo. These robots have four wheels with the front two wheels being dc-powered.

The paper is organized as follows. In Section II, we present model of unicycle-type mobile robot. Section III describes the fuzzy-based control and the cooperation algorithm. Section IV demonstrates the experimental results. Finally, we conclude paper in Section V.

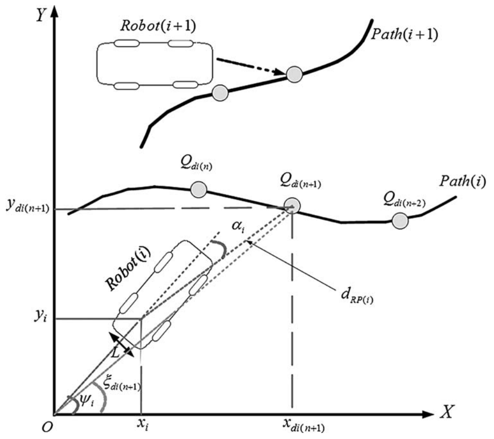

Fig. 1. Kinematic model of unicycle-type mobile robot.

II. Modeling of Unicycle-Type Mobile Robot

Fig. 1 shows the general kinematic model of a unicycle-type mobile robot and the discretized paths.

In Fig. 1, $i = 1, \ldots, k$ represents the number of individual mobile robots, $P_i = [x_i, y_i, \psi_i]^T$ denotes the position and orientation vector of the ith robot of the group.

$Q_{di(n)} = [x_{di(n)}, y_{di(n)}, \xi_{di(n)}]^T$ represents the coordination of nth sample point on the ith path where $n = 0, \ldots, f$. The path is described by a set of discrete node positions $Q_{di(0)}$ to $Q_{di(f)}$ linked to each other starting from the initial position to the final desired position. $Q_{di(n)}$ is nth discretized sample on the path. Kinematic equation of unicycle-type can be explained as [31]

$$\begin{cases} \dot{x}_i(t) = v_i(t)\cos\psi_i(t) \\ \dot{y}_i(t) = v_i(t)\sin\psi_i(t) \\ \dot{\psi}_i(t) = \omega_i(t) \end{cases} \tag{1}$$

where v_i and ω_i are the linear and angular velocities of each robot, respectively. The system is subject to nonholonomic constraints such as

$$\dot{x}_i(t)\sin\psi_i(t) - \dot{y}_i(t)\cos\psi_i(t) = 0. \tag{2}$$

Linear and angular velocities of mobile robots are subject to the following bounds:

$$\begin{array}{ll} |v_i(t)| \le v_{i\max} & |\omega_i(t)| \le \omega_{i\max} \\ |\dot{v}_i(t)| \le \dot{v}_{i\max} & |\dot{\omega}_i(t)| \le \dot{\omega}_{i\max}. \end{array} \tag{3}$$

In discrete time, (1) can be explained as follows:

$$\begin{cases} x_i(t_{n+1}) = x_i(t_n) + Tv_i(t_n)\cos\psi_i(t_n) \\ y_i(t_{n+1}) = y_i(t_n) + Tv_i(t_n)\sin\psi_i(t_n) \\ \psi_i(t_{n+1}) = \psi_i(t_n) + T\omega_i(t_n) \end{cases} \tag{4}$$

where T is sampling time and t_n express time in nth sample point. Desired robot position and orientation in sample time t_n can be defined as follows:

$$P_i(t_n) = [x_i(t_n), y_i(t_n), \psi_i(t_n)]^T. \tag{5}$$

III. Fuzzy Control and Cooperative Algorithm

Fig. 2 illustrates the block diagram used to control and synchronize a group of mobile robots. There are two hierarchical levels of controller being low and high, respectively. The low-level controller is designed to adjust each robot's right- and left-wheel velocities. The high-level controller, which is a fuzzy controller, is designed to coordinate the robots' group formation as well as following the generated paths. Posture sensors are also used to localize the robots. A designed algorithm has been used to localize and synchronize the robots into a group formation and direct them to their desired paths.

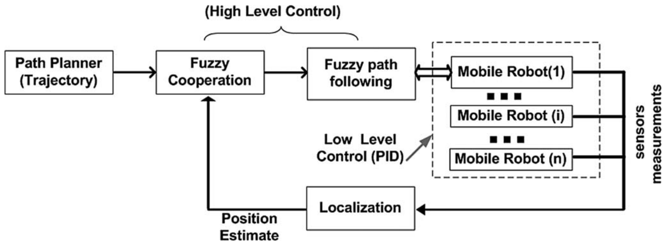

Fig. 2. Infrastructure of multirobot control and cooperation.

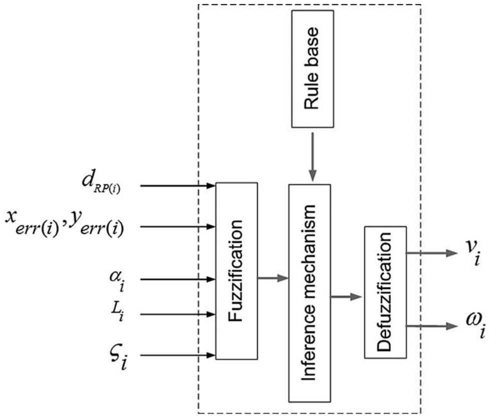

Fig. 3. Fuzzy-control structure.

Fig. 3 shows fuzzy system structure with desired inputs and outputs.

Inputs of the fuzzy controller are $d_{\mathrm{RP}(i)}, \alpha_i, L_i, \varsigma_i\ x_{\mathrm{err}(i)}$, and $y_{\mathrm{err}(i)}$, where $d_{\mathrm{RP}(i)}$is the distance from the actual position of ith robot to the next desired position, $\alpha_i$is the difference between the line joining the current position to the next desired position and the actual heading of the robot. L_i is the length of paths and ς_i denotes the path parameter being used synchronize the individual robot within the group formation. $x_{\mathrm{err}(i)}$, $y_{\mathrm{err}(i)}$, and α_i are calculated by the following equations in robot reference as [31]:

$$\begin{bmatrix} x_{\mathrm{err}(i)} \\ y_{\mathrm{err}(i)} \\ \alpha_i \end{bmatrix} = \begin{bmatrix} \cos\psi_i & \sin\psi_i & 0 \\ -\sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{di} - x_i \\ y_{di} - y_i \\ \zeta_{di} - \psi_i \end{bmatrix}. \quad (6)$$

The output of the fuzzy controller determines the linear and angular velocities of individual robots.

To solve the problem of propelling a robot along a continuous desired path, the key idea is to analyze the path in discrete segments and consider the robot moving between discontinuous sampling points. The paths are modeled by a fifth-order polynomial and divided into segments for analysis with the same sampling point's numbers regardless of the shape, curvature, or path length.

A. Path Following and the Cooperation Problem

The task of the fuzzy path-tracking controller is to command the robots to follow the paths in a smooth and continuous manner with the best possible precision. In order to achieve this, it is not necessary for the robots to pass exactly through specific sampling points on their paths, but they must at least pass within a proximity to them while reaching their final destination. For our purposes, path following is categorized in two different groups.

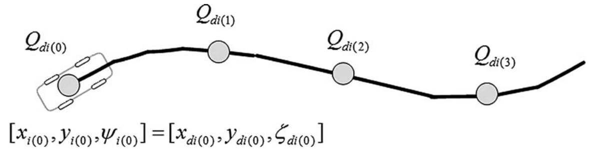
$Q_{di(0)}$
$Q_{di(1)}$
$Q_{di(2)}$
$Q_{di(3)}$
$[x_{i(0)}, y_{i(0)}, \psi_{i(0)}] = [x_{di(0)}, y_{di(0)}, \zeta_{di(0)}]$

Fig. 4. Robot is on the desired path.

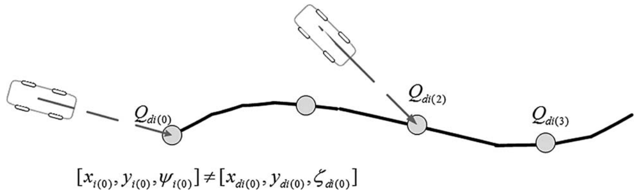
$Q_{di(0)}$
$Q_{di(2)}$
$Q_{di(3)}$
$[x_{i(0)}, y_{i(0)}, \psi_{i(0)}] \neq [x_{di(0)}, y_{di(0)}, \zeta_{di(0)}]$

Fig. 5. Robot is not on the desired path.

1) The robot is placed on its predefined path. In this scenario, which is shown in Fig. 4, the robot tries to follow and stay on its desired path.
2) In the second modeling, the robot is not placed on its predefined path. In this scenario, which is shown in Fig. 5, the robot tries to move forward toward the path to reach and follow its desired path.

We consider P_i and Q_{di} to represent the position of robot i and desired path discretized. We let v_{di} and ω_{di} denote the desired linear and angular velocities assignment for robot i. The purpose of path following is to make the robot's velocity $u_i = [v_i, \omega_i]^T$ track a desired velocity reference $u_{di} = [v_{di}, \omega_{di}]$, which means $\|u_i - u_{di}\| \to 0$ and also $\|P_i - Q_{di}\| \to 0$.

Now, consider a group of mobile robots each with a local controller for path following. In order to have cooperation across the whole robot group, they are required to move along their paths while maintaining a desired interrobot formation pattern as well as reaching their final goals at the same time regardless of their path lengths or their initial positions.

B. Fuzzy Path Following and Cooperative Controller

The form of the control-law equation for path following and cooperation is as follows:

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} f_1(d_{\mathrm{RP}(i)}, \alpha_i, x_{\mathrm{err}(i)}, y_{\mathrm{err}(i)}, L_i, \varsigma_i) \\ f_2(d_{\mathrm{RP}(i)}, \alpha_i, x_{\mathrm{err}(i)}, y_{\mathrm{err}(i)}, L_i, \varsigma_i) \end{bmatrix}. \quad (7)$$

The functions f_1 and f_2 are the control laws of a Sugeno-type fuzzy controller. Sugeno controllers take in fuzzy inputs and outputs. The task of the path-following program is to make the robots pass in proximity of the sampling points in a continuous and smooth manner. The behavior of the controller is such that if the discrete points are close to each other then a higher precision is achieved but with the robot moving at a lower speed due to the high number of sampling points. If less precision is required,

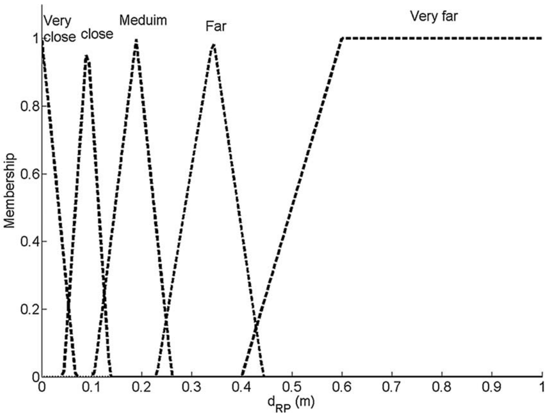

Fig. 6. Membership function of $d_{\mathrm{RP}(i)}$.

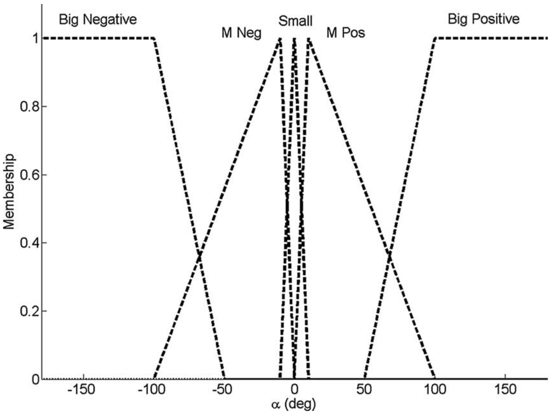

Fig. 7. Membership function of α_i.

the discrete points can be selected further apart and the robot will therefore move at higher speed.

The membership functions of inputs $d_{\mathrm{RP}(i)}$ and α_i are shown in Figs. 6 and 7.

Linear velocity obtained by fuzzy controller related to inputs $d_{\mathrm{RP}(i)}$ and α_i is shown in Fig. 8. As can be seen in this figure, when the robot is far from the path, or the distance between the actual position of robot and the next ahead sampling point is large, then its linear velocity will be increased. When the robot is close to the path, or the distance between actual position of robot and the next ahead sampling point is small, the velocity will be decreased and the robot will move slower.

The angular velocity obtained by the fuzzy controller related to its inputs $d_{\mathrm{RP}(i)}$ and α_i is shown in Fig. 9. As can be seen in this figure, when the angle between the robot and the path is large, the robot will have more angular velocity to decrease this error.

The reason for using $x_{\mathrm{err}(i)}$ as an input to the fuzzy system in path following can be described as follows: if we suppose that a robot should travel from $Q_{di(n)}$ to $Q_{di(n+1)}$, this means that it has to catch the targeting point $Q_{di(n+1)}$. However, if the robot passes points $Q_{di(n+1)}$ and then $Q_{di(n+2)}$ in the robot reference, then when moving to the next step, the targeting point $Q_{di(n+1)}$ falls behind the actual position of the robot, which means ($x_i > x_{di} \Rightarrow x_{\mathrm{err}(i)} < 0$).

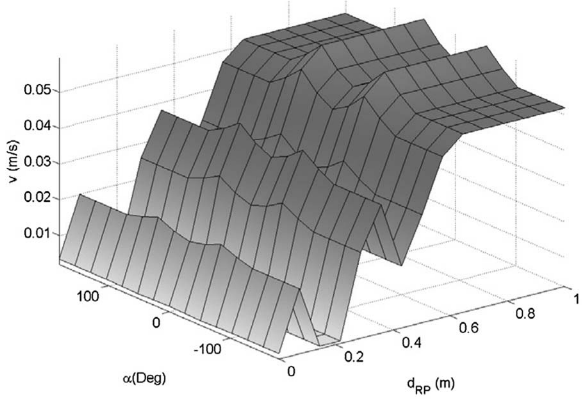

Fig. 8. Linear velocity obtained by fuzzy controller.

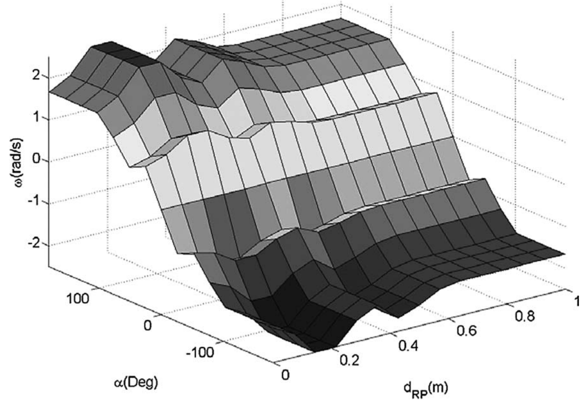

Fig. 9. Angular velocity obtained by fuzzy controller.

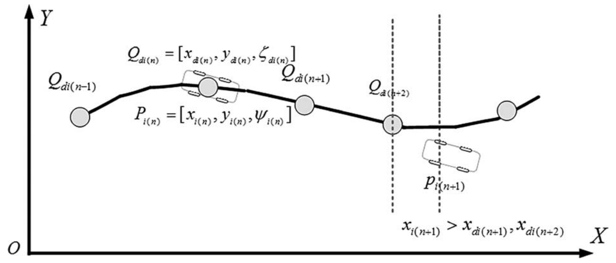
Y
$Q_{di(n)} = [x_{di(n)}, y_{di(n)}, \zeta_{di(n)}]$
$Q_{di(n-1)}$
$Q_{di(n+1)}$
$Q_{di(n+2)}$
$P_{i(n)} = [x_{i(n)}, y_{i(n)}, \psi_{i(n)}]$
$P_{i(n+1)}$
$x_{i(n+1)} > x_{di(n+1)}, x_{di(n+2)}$
O
X

Fig. 10. Robot passes targeting point ahead.

To solve this issue, we add the following rule to the fuzzy controller.

1) *If $x_i > x_{di}$, then robot will stop, until condition $x_i \leq x_{di}$ is fulfilled.*

Fig. 10 depicts this issue. If robot is not on the path or the vertical position of robot in *x–y* coordination (y_i) is different from the next vertical sampling point on the path, then $y_{\mathrm{err}(i)}$ will be used as an extra input to the fuzzy controller. This input is used to help the robot turn toward the paths and catch it. Fig. 11 shows the necessity of using the $y_{\mathrm{err}(i)}$ as an input to the fuzzy controller. If the robot is not on the path and there is an error between the heading angle of the robot and the path, the robot will turn to minimize this error. However, when this error is zero

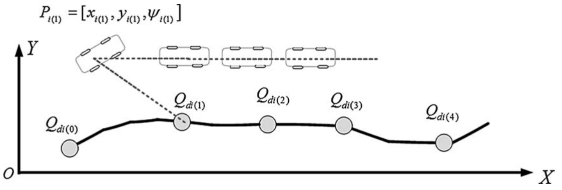

Fig. 11. Robot is not on the same y-coordination as path.

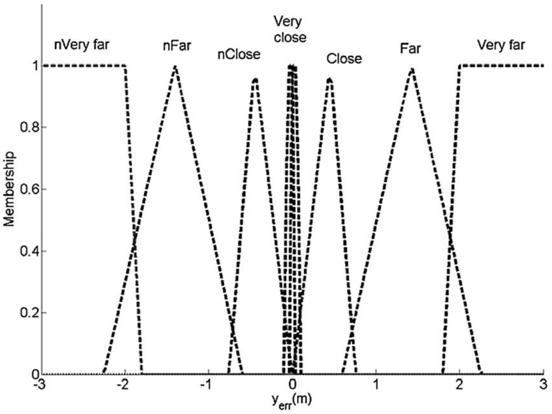

Fig. 12. Membership function of $y_{\mathrm{err}(i)}$.

($\alpha_i = 0$), the robot will move to a path parallel to the actual path, and therefore, will never reach it.

To solve this problem, an angle is added to α_i, which is λ_i. Using the y-coordination, the further the robots are from the path, the larger this angle becomes, and will only reduce as the robots move closer to the path. Once the robot catches the path, λ_i will be finally zero.

$$\text{if } \left|y_{\mathrm{err}(i)}\right| > 0 \Rightarrow \alpha_i(\text{new}) = \lambda_i + \alpha_i. \qquad (8)$$

The membership functions of this input is shown in the Fig. 12 and Fig. 13 shows the change of λ_i related to change of $y_{\mathrm{err}(i)}$. To obtain a group cooperation between the robots, each path is parameterized in terms of parameter ς_i. The robots will keep in cooperation, if $\varsigma_i = \varsigma_j$ for all i, j. This parameter is defined as $\varsigma_i = s_i/L_i$, where s_i is signed curvilinear abscissa of sampling points $Q_{di(n)}$ along the path. For group cooperation, all paths are discretized to the same number of sampling points. As the robot group need to reach their destination at the same time, if a path is longer, then the distance between the sampling points will increase, thus forcing the individual robot to move faster in order to complete the same number of sampling points within the allocated time. However, if the path length is smaller, then the distance between the sampling points will decrease and the robot move slower with respect to the other robots in the group. To improve the cooperation between the robots traveling in formation, the following rules are applied when they are ahead or behind of their desired path position:

1) *If the robot is ahead of the desired path* ($x_{\mathrm{err}(i)} < 0$)*, then it will stop until the condition* $\varsigma_i = \varsigma_j$ *is fulfilled for all robots.*
2) *If the robot is behind of the desired path* ($x_{\mathrm{err}(i)} \geq 0$)*, then the robot will go faster to catch the path and other robots.*

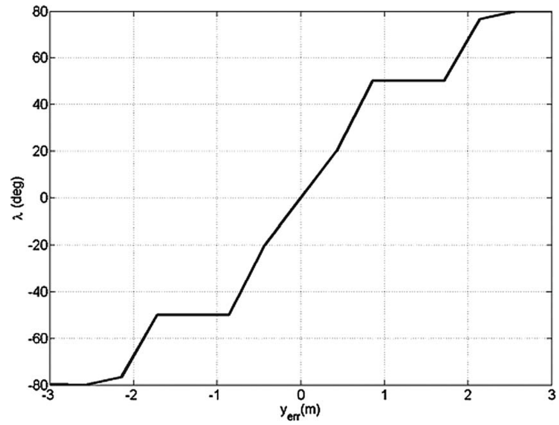

Fig. 13. λ_i obtained by fuzzy controller and $y_{\mathrm{err}(i)}$.

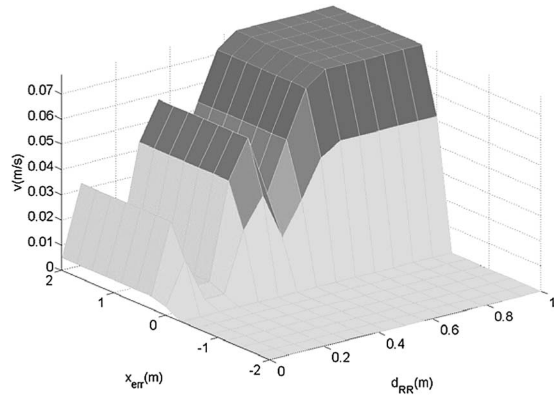

Fig. 14. Linear velocity obtained by the fuzzy controller and inputs $d_{\mathrm{RP}(i)}, x_{\mathrm{err}(i)}$.

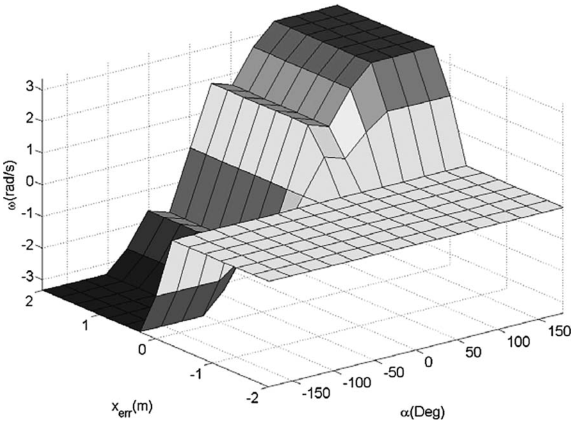

Fig. 15. Angular velocity obtained by the fuzzy controller and inputs $\alpha, x_{\mathrm{err}(i)}$.

Figs. 14 and 15 show the linear and angular velocities obtained by the inputs $d_{\mathrm{RP}(i)}, x_{\mathrm{err}(i)}$. As can be seen in these figures, if $x_{\mathrm{err}(i)} < 0$, then the linear and angular velocities will be zero and the robot will stop.

TABLE I
RULE BASE FOR PATH FOLLOWING

Rule1	if (*input1 is Very close*) and (*input 2 is Nomatterwhat*) and *(input3 is Poserr)* then (*output 1 is veryvery slow*) (*output2 is zero*)
Rule2	if *(input1 is close)* and (*input 2 is Nomatterwhat*) and (*(input3 is Poserr)* then (*output 1 is very slow*) (*output2 is zero*)
Rule3	if (*input1 is Medium*) and (*input 2 is Nomatterwhat*) and *(input3 is Poserr)* then (*output 1 is slow*) (*output2 is zero*)
Rule4	if (input1 is Far) and (*input 2 is Nomatterwhat*) and *(input3 is Poserr)* then (*output 1 is Fast*) (*output2 is zero*)
Rule5	if (*input1 is Very far*) and (*input 2 is Nomatterwhat*) and (*(input3 is Poserr)* then (*output 1 is Very fast*) (*output2 is zero*)
Rule6	if (*input1 is Nomatterwhat*) and *(input 2 is Big negative)* and (*input3 is Poserr*) then (*output 1 is veryvery slow*) (*output2 is Big negative*)
Rule7	if (*input1 is Nomatterwhat*) and (*input 2 is Negative*) and *(input3 is Poserr)* then (*output 1 is veryvery slow*) (*output2 is Negative*)
Rule8	if (*input1 is Nomatterwhat*) and (*input 2 is Straight*) and (*(input3 is Poserr*) then (*output 1 is veryvery slow*) (*output2 is zero*)
Rule9	if (*input1 is Nomatterwhat*) and (*input 2 is Positive*) and (*(input3 is Poserr*) then (*output 1 is veryvery slow*) (*output2 is Positive*)
Rule10	if (*input1 is Nomatterwhat*) and (*input 2 is Big positive*) and (*(input3 is Poserr*) then (*output 1 is veryvery slow*) (*output2 Big positive*)
Rule11	if (*input1 is Nomatterwhat*) and (*input 2 is Nomatterwhat*) and (*input3 is Negerr*) then (*output 1 is Zero*) (*output2 Zero*)

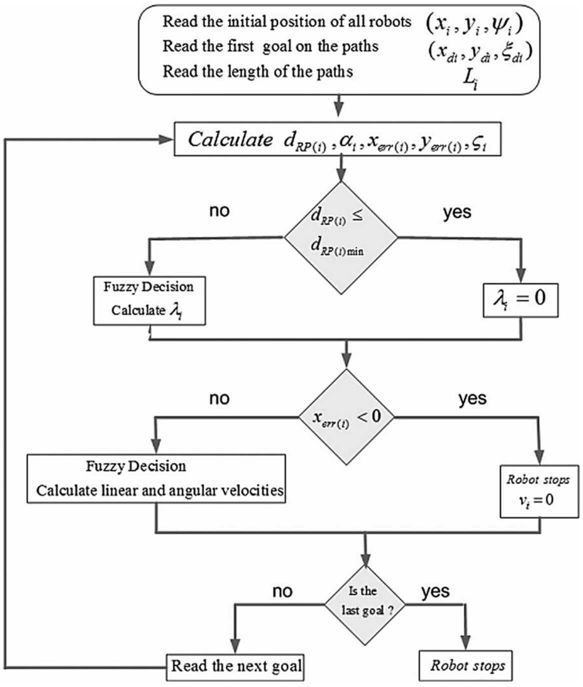
Read the initial position of all robots (x_i, y_i, ψ_i)
Read the first goal on the paths $(x_{di}, y_{di}, \xi_{di})$
Read the length of the paths L_i
Calculate $d_{RP(i)}, \alpha_i, x_{err(i)}, y_{err(i)}, \varsigma_i$
no
$d_{RP(i)} \le d_{RP(i)\min}$
yes
Fuzzy Decision
Calculate λ_i
$\lambda_i = 0$
no
$x_{err(i)} < 0$
yes
Fuzzy Decision
Calculate linear and angular velocities
Robot stops
$v_i = 0$
no
Is the last goal ?
yes
Read the next goal
Robot stops

Fig. 16. Intelligent flowchart of path following and cooperation.

Rule bases for path following and cooperation are shown in Table I, where the inputs 1, 2, and 3 are $d_{\mathrm{RP}(i)}, \alpha_i$, and $x_{\mathrm{err}(i)}$, and the outputs 1 and 2 are v_i and ω_i, respectively.

Block in Fig. 16 represents the path following and cooperation flow chart algorithm.

C. Stability Proof of Path-Following Algorithm

Taking the derivative of (6) along the kinematics equation (1) gives the following velocity error:

$$\begin{aligned} \dot{x}_{\mathrm{err}(i)} &= -v_i + v_{di}\cos\alpha_i + y_{\mathrm{err}(i)}\,\psi_i \\ \dot{y}_{\mathrm{err}(i)} &= v_{di}\sin\alpha_i - x_{\mathrm{err}(i)}\,\psi_i \\ \dot{\alpha}_i &= \dot{\zeta}_{di} - \dot{\psi}_i. \end{aligned} \tag{9}$$

The available controllers to steer the robot to its path are v_i and ω_i. $d_{\mathrm{RP}(i)}$ is defined as $d_{\mathrm{RP}(i)} = \sqrt{x^2_{\mathrm{err}(i)} + y^2_{\mathrm{err}(i)}}$, where the derivation of $d_{\mathrm{RP}(i)}$ along (9) gives the following:

$$\dot{d}_{\mathrm{RP}(i)} = \frac{-v_i x_{\mathrm{err}(i)}}{d_{\mathrm{RP}(i)}} + v_{di}\cos(\alpha_i + \gamma_i) \tag{10}$$

where

$$\gamma_i = -\tan^{-1}(y_{\mathrm{err}(i)}/x_{\mathrm{err}(i)}).$$

The state error can be written as

$$\dot{X}_i = f_i(d_{\mathrm{RP}(i)}, \alpha_i, x_{\mathrm{err}(i)}, y_{\mathrm{err}(i)}, \delta_i) \text{ where } \delta_i = [v_i, \omega_i].$$

The idea of the fuzzy controller is to linearize $\dot{X}_i$about a number of operating points depending on the linguistic rules that are defined. First, we need to separate path following from the group synchronization. Since in path following the robots are required to move tangentially on their desired paths with a given velocity, we redefine the desired velocity to be tracked as $v_{di} = \bar{v}_{di}\,\dot{\varsigma}_i$, where ς_i is a path parameter characterizing a desired path. While robots move along their predefined paths, to ensure their cooperation, the path parameters should be synchronized with the center of the virtual structure that defines the rigid formation. That is $\varsigma_i - \varsigma_0 \to 0\ \forall i = 1, \ldots, n$. To do so, let

$$\dot{\varsigma}_i = \dot{\tilde{\varsigma}}_i + \bar{\omega}_i(d_{\mathrm{RP}(i)}). \tag{11}$$

For path following, we assume first that $\dot{\tilde{\varsigma}}_i = 0$; then, we disregard this assumption when we deal with the group cooperation. Linearizing (9) about an operating point using Taylor series would give the Takagi–Sugent fuzzy model for M^j linguistic rule as [36]

$$\text{if } X_i \text{ is } (X_{\mathrm{ieq}}, \delta_{\mathrm{ieq}}), \text{ then } \dot{\tilde{X}}_i = A_i^j \tilde{X}_i + B_i^j \tilde{\delta}_i \tag{12}$$

where $\tilde{X}_i = X_i - X_{\mathrm{ieq}}$ and $\tilde{\delta}_{\mathrm{ieq}} = \delta_i - \delta_{\mathrm{ieq}}$. The matrices A_i^j and B_i^j can be found as:

$$A_i^j = \frac{\partial f_i}{\partial X_i^j}\bigg| X_{\mathrm{ieq}}, \delta_{\mathrm{ieq}} \qquad B_i^j = \frac{\partial f_i}{\partial \delta_i^j}\bigg| X_{\mathrm{ieq}}, \delta_{\mathrm{ieq}}.$$

This definition gives the following:

$$A_i^j = \begin{bmatrix} -\bar{v}_{di}(\bar{\omega}_i^{\mathrm{eq}})'\cos(\alpha_i^{\mathrm{eq}} + \gamma_i^{\mathrm{eq}}) + \bar{v}_{di}\bar{\omega}_i^{\mathrm{eq}}\sin(\alpha_i^{\mathrm{eq}} + (\gamma_i^{\mathrm{eq}})') & \bar{v}_{di}\bar{\omega}_i\sin(\alpha_i^{\mathrm{eq}} + (\gamma_i^{\mathrm{eq}})') \\ 0 & 0 \end{bmatrix} \quad B_i^j = \begin{bmatrix} \cos\gamma_i^{\mathrm{eq}} & 0 \\ 0 & 1 \end{bmatrix} \tag{13}$$

where $(\bullet)' = \partial/\partial d_{\mathrm{RP}(i)}$. Let $\mu_j$be the membership function of the inferred system set corresponding to each operating point;

therefore, the linearized system would be written as

$$\dot{\tilde{X}}_i = \frac{\sum_{j=1}^{r}\left(\mu_j \tilde{X}_i + B_i^j \tilde{\delta}_i^j\right)}{\sum_{j=1}^{r}\mu_j} \tag{14}$$

where $(j = 1, \ldots, r)$ and r is the number of rules. Then for each model, we select a fuzzy-state feedback controller of the form $\tilde{\delta}_i^j = -K_i^j\left(X_i - X_{\text{ieq}}\right)$.

The structure of the inferred controller is then as follows:

$$\tilde{\delta}_i = \frac{\sum_{j=1}^{r}\mu_j K_i^j \tilde{X}_i^j}{\sum_{j=1}^{r}\mu_j}. \tag{15}$$

The inferred closed-loop fuzzy system has the following form:

$$\dot{X}_i = \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j\left(A_i^j - B_i^j K_i^j\right)X_i}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j}. \tag{16}$$

The matrix K_i^j is chosen such that the matrix $(A_i^j - B_i^j K_i^j)$ is Hurwitz. Stability for the equilibrium points of the fuzzy system (16) are known and reduce to find an appropriate Lyapunov function, such that there exists a common positive matrix P_i verifying

$$(A_i^j - B_i^j K_i^j)^T P_i + P_i(A_i^j - B_i^j K_i^j) < 0 \qquad \forall j = 1, \ldots, r$$

and

$$Q_i^{jKT} P_i + P_i Q_i^{jKT} < 0$$

where

$$Q_i^{jk} = 0.5((A_i^j - B_i^j K_i^j) + (A_i^k - B_i^k K_i^k)).$$

From this, we can show that the equilibrium of the fuzzy system (16) is globally asymptotically stable.

D. Stability Proof of the Cooperative Algorithm

To consider the cooperation between the robots, we will make use of the virtual structure approach. We will no longer suppose that $\dot{\tilde{\varsigma}}_i = 0$, since it will be a corrective signal to ensure synchronization of the robot along with the center of the virtual structure of the rigid formation. The fuzzy system (16) considering the corrective term $\dot{\tilde{s}}_i$ is rewritten as follows:

$$\dot{X}_i = \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j\left(A_i^j - B_i^j K_i^j\right)X_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j} + \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j}\dot{\tilde{\varsigma}}_i \tag{17}$$

where H_i^j is defined as

$$H_i^j = \begin{bmatrix} -\bar{v}_{di}\sin(\alpha_i^{\text{eq}} + (\gamma_i^{\text{eq}})') & \bar{v}_{di}\sin(\alpha_i^{\text{eq}} + (\gamma_i^{\text{eq}})') \\ 0 & 0 \end{bmatrix}. \tag{18}$$

Define the Lyapunov function for this system as

$$V = 0.5\sum_{i=1}^{n}\tilde{X}_i^T P_i \tilde{X} + (\varsigma_i - \varsigma_0)^2 . \tag{19}$$

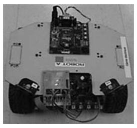

Fig. 17. Mobile robot (EtsRo).

The differentiation of this lyapunov function gives the following:

$$\dot{V} = -\sum_{i=1}^{n}\tilde{X}_i^T Q_i \tilde{X}_i + (\varsigma_i - \varsigma_0)\left(\dot{\tilde{\varsigma}}_i + \bar{\omega}_i(d_{\text{RP}(i)}) - \dot{\varsigma}_0\right) + \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j}\dot{\tilde{\varsigma}}_i. \tag{20}$$

Choose

$$\bar{\omega}_i(d_{\text{RP}(i)}) = \dot{\varsigma}_0 = \bar{\omega}_i(t)(1 - k_1 e^{k_2(t-t_0)})(1 - \tanh(d_{\text{RP}}^T d_{\text{RP}}))$$

where $d_{\text{RP}} = \left[d_{\text{RP}(1)}, d_{\text{RP}(2)}, \ldots, d_{\text{RP}(n)}\right]^T$; this yields

$$\dot{V} = -\sum_{i=1}^{n}\tilde{X}_i^T Q_i \tilde{X}_i + \left(\frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j} + (\varsigma_i - \varsigma_0)\right)\dot{\tilde{\varsigma}}_i. \tag{21}$$

If we choose

$$\dot{\tilde{\varsigma}}_i = -\delta_i\left(\frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j} + (\varsigma_i - \varsigma_0)\right) = -\delta_i\eta_i$$

then from (21), we will have

$$\dot{V} = -\sum_{i=1}^{n}\tilde{X}_i^T Q_i \tilde{X}_i - \delta_i\eta_i^2 \tag{22}$$

which is negative definite. This shows by Barbalet's lemma that $\tilde{X}_i \to 0$ and $\eta_i \to 0$. It is easy to see then that by construction, we have $\varsigma_i - \varsigma_0 \to 0$.

IV. Experimental Results

A. Experimental Setup

Fig. 17 shows the EtsRo mobile robot used for the experimental setup. EtsRo is a unicycle-type mobile robot, with the front wheels equipped with two dc motors 7.5 V and 175 r/min, which installed on the right and left front wheels. The incremental Encoders are mounted on the motors counting a resolution of 6000 Pulses/Turn. The wheels have radius of $r = 4.5$ cm, the length, width, and height of EtsRo are $23, 20,$ and 11 cm, respectively. The total weight of robot is around 2.3 kg. The maximum linear velocity is 1.12 m/s and maximum angular velocity is 5.74 rad/s. The linear and angular velocities of individual robots are obtained by [31] the following:

$$v = \frac{v_R(t) + v_L(t)}{2}, \; \omega = \frac{v_R(t) - v_L(t)}{L} \tag{23}$$

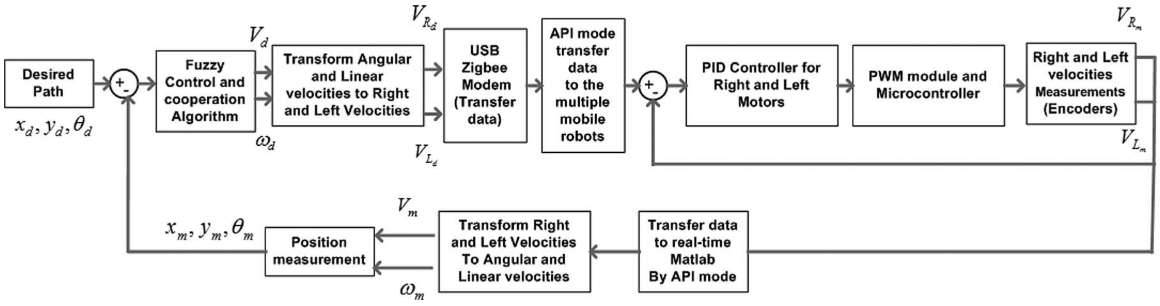
Desired Path
x_d, y_d, θ_d
+
-
V_d
Fuzzy Control and cooperation Algorithm
ω_d
Transform Angular and Linear velocities to Right and Left Velocities
V_{R_d}
V_{L_d}
USB Zigbee Modem (Transfer data)
API mode transfer data to the multiple mobile robots
+
-
PID Controller for Right and Left Motors
PWM module and Microcontroller
Right and Left velocities Measurements (Encoders)
V_{R_m}
V_{L_m}
x_m, y_m, θ_m
Position measurement
V_m
ω_m
Transform Right and Left Velocities To Angular and Linear velocities
Transfer data to real-time Matlab By API mode

Fig. 18. Control architecture of EtsRo.

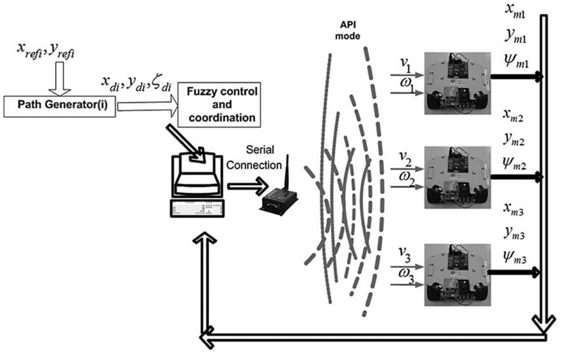

Fig. 19. General view of experimental setup.

where $v_R(t)$ and $v_L(t)$ denote the right and left velocities and L shows the distance between the two actuated wheels.

EtsRo has two-level control architecture. A low-level control algorithm is written in C language and run with a sampling time of $T_s = 10$ ms. This is a PID controller and designed to give the best accuracy for the right and left motors. The high-level controller, which is a fuzzy controller, is designed in real-time simulink (MATLAB) with a sampling time of $T_s = 50$ ms. Fig. 18 demonstrates the control and cooperation architecture for the mobile robots.

Fig. 19 shows the structural design of the control, path planning, and cooperative behavior for the group of mobile robots being used in the experimental tests. A PC communicates through a serial port and modem using application programming interface mode with robots.

In this project, ATmega32 microcontroller is used, which generates a PWM signal to control the motor speed. ZigBee USB-RF modem is the communication device, which is used to transmit the data. The range of modem is about 100 m making it a high-quality modem for indoor applications. The speed of ZigBee modem is set for 9600 bit/s. To illustrate the performance of the proposed cooperation and control scheme, some tests were done on different paths with three mobile robots.

B. Experimental Results

In this section, we discuss the results of different formations of robots. These formations are as follows:

1) *where robots travel on paths with different lengths*,
2) *where robots travel on paths with the same lengths but the robots are not placed on the paths*.

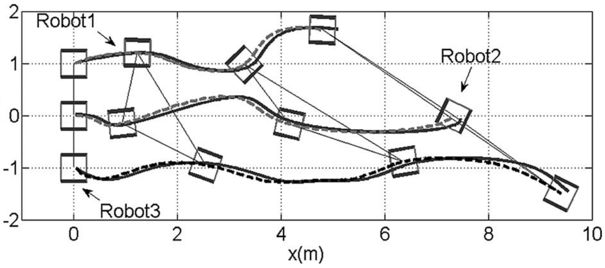

Fig. 20. Reference paths and real robots trajectories.

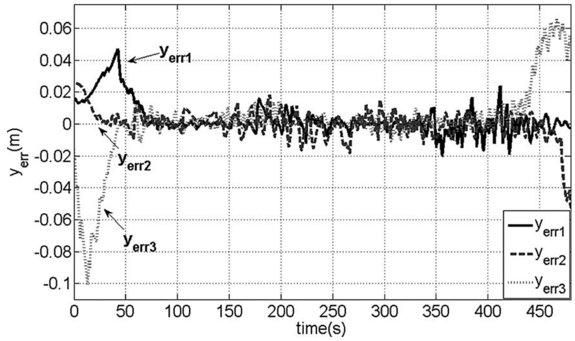

Fig. 21. Path-following errors $y_{\mathrm{err}(i)}$.

In the first test, paths that have different lengths are considered. In this scenario, robots are placed on a common vertical line with initial position and lengths as

$$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [0, 1, 0]^T \qquad L_1 = 5.41 \text{ m}$$
$$[x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [0, 0, 0]^T \qquad L_2 = 7.75 \text{ m}$$
$$[x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [0, -1, 0]^T \qquad L_3 = 9.83 \text{ m.}$$

Fig. 20 shows the reference and the actual robots trajectories in the first scenario. The path tracking error y_{err} is shown in Fig. 21. As can be seen in these figures, the robots travel along their paths with negligible errors and the formation is experimentally successful.

Linear and angular velocities are plotted in Figs. 22 and 23, respectively. These figures show that the robots travel with different velocities relative to the length of the path on which they travel. We observe that robot 3 has the highest velocity (largest path) and robot 1 the lowest velocity (shortest length).

In the second test, paths with same lengths are considered, but robots are not placed on the paths. As can be seen in Fig. 24,

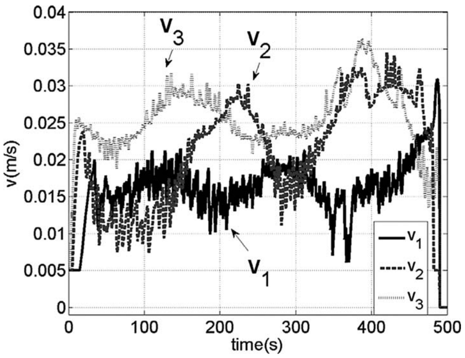

Fig. 22. Linear velocity of robots.

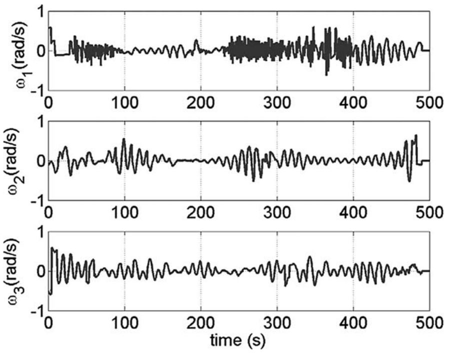

Fig. 23. Angular velocity of robots.

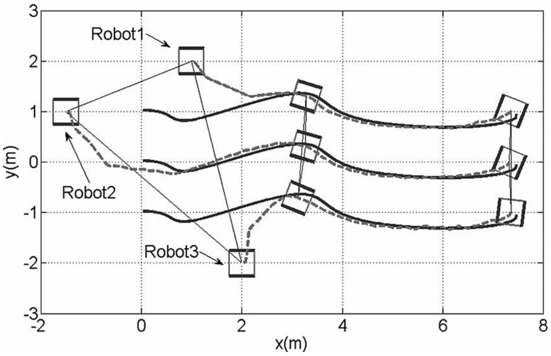

Fig. 24. Reference paths and real robots trajectories.

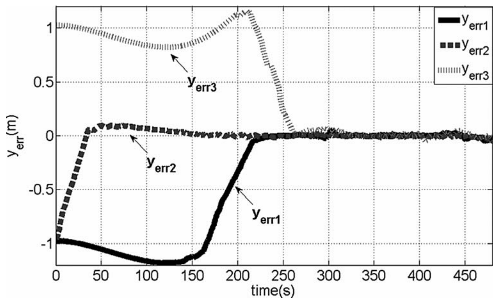

Fig. 25. Path-following errors $y_{\mathrm{err}(i)}$.

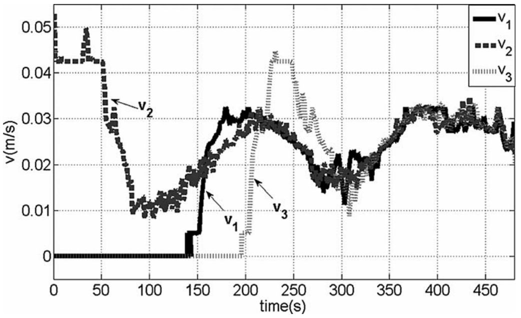

Fig. 26. Linear velocity of robots.

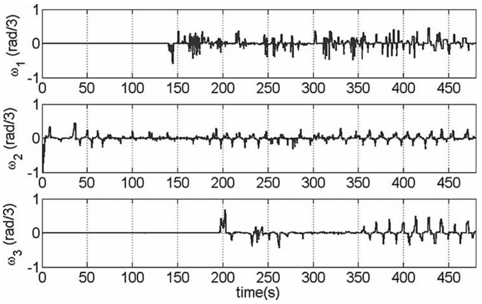

Fig. 27. Angular velocity of robots.

robots 1 and 3 are ahead of their paths and robot 2 is behind its path. In this scenario, all paths have the same length as $L_1 = 7.76$ m and the initial positions of robots are defined by

$$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [1, 2, 0]^T$$
$$[x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [-1.5, 1, 0]^T$$
$$[x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [2, -2, 0]^T.$$

Fig. 24 shows both the reference and the actual robots trajectories in this scenario. This figure shows that to keep the formation and cooperation between the MMR group, robots 1 and 3 come to a stop and allow robot 2 to travel faster to reach them. As soon as robot 2 arrives at the same vertical point as robots 1 and 3, they adjust their velocities so that all three robots arrive at the desired sampling points at the same time.

The path-tracking error y_{err} is shown in Fig. 25. As can be seen in this figure, the errors reduce to zero as the robots catch their predefined paths.

The linear and angular velocities are plotted in Figs. 26 and 27, respectively. These figures show that robots 1 and 3 stop and robot 2 moves faster to catch the path.

C. Results Comparison of Both the Fuzzy and the Nonlinear Method

In this section, the results obtained by the proposed fuzzy algorithm and a nonlinear control method proposed in [30] are compared. Tables II and III show the results obtained by these methods with the path following errors compared for tests 1, 2, and 3 in which $x_{\mathrm{err-total}}$, $y_{\mathrm{err-total}}$, α_{total}, and mean square

TABLE II
COMPARISON BETWEEN NONLINEAR AND FUZZY CONTROLLER FOR TEST 1

	Nonlinear Control				Fuzzy Control			
Total errors	x_{err}	y_{err}	α	*MSE*	x_{err}	y_{err}	α	*MSE*
Robot1	**-.0801**	**.00013**	**-.0002**	**.0891**	**.0216**	**.0015**	**-3.5e-005**	**.0325**
Robot2	**-.1160**	**-.0032**	**.0057**	**.1345**	**.2408**	**-.0016**	**-.0054**	**.0493**
Robot3	**-.1506**	**-.0025**	**.0055**	**.1589**	**.2842**	**.0018**	**.27e-005**	**.0056**

TABLE III
COMPARISON BETWEEN NONLINEAR AND FUZZY CONTROLLER FOR TEST 2

	Nonlinear Control				Fuzzy Control			
Total errors	x_{err}	y_{err}	α	*MSE*	x_{err}	y_{err}	α	*MSE*
Robot1	**-.0993**	**.2920**	**-.2728**	**.4981**	**0.040**	**-0.531**	**0.060**	**0.6858**
Robot2	**-.1979**	**.0334**	**-.0527**	**.2309**	**.110**	**-0.201**	**.091**	**0.2619**
Robot3	**-.0084**	**.0749**	**-.1106**	**.2962**	**-0.437**	**0.592**	**-0.138**	**.8985**

error (MSE) can be calculated by the following:

$$x_{\text{err}-\text{total}} = \frac{\sum_{n=0}^{f} x_{\text{err}(n)}}{f} \qquad y_{\text{err}-\text{total}} = \frac{\sum_{n=0}^{f} y_{\text{err}(n)}}{f}$$

$$\alpha_{\text{total}} = \frac{\sum_{n=0}^{f} \alpha_{(n)}}{f}$$

$$\text{MSE} = \frac{\sum_{n=0}^{f} \sqrt{x_{\text{err}(n)}^2 + y_{\text{err}(n)}^2 + \alpha_{(n)}^2}}{f}. \tag{24}$$

Fuzzy logic mimics the way the human brain makes decisions, thus, simplifying the computations needed by the algorithm controller. As these tables show, when the robots move on their desired paths, the fuzzy controller produces a smaller MSE compared to the nonlinear method, which means that the robots follow their paths more precisely. However, when the initial positions of the robots are not on the desired paths (test 2) the nonlinear control is faster at guiding them to their desired paths.

V. CONCLUSION

In this paper, real-time path following, control and cooperation for a group of mobile robots has been presented using a fuzzy logic controller. A two-level hierarchical controller was designed to facilitate the best performance control algorithm. Firstly, a low-level controller (PID) is used to adjust the left and right motors velocities, and secondly, a high-level fuzzy controller was used to perform the tasks of path following, localization and cooperation. This cooperation method, which was derived from the fuzzy logic and PID, empowers the robots to move, follow and coordinate paths in different formations. The outputs of the fuzzy module controller are the linear and angular velocities of the individual robots. The experimental results obtained using three different paths scenarios demonstrate the effectiveness of the proposed algorithm. Future work will consider the improvement of the fuzzy algorithm for more elaborated cases, such as crash avoidance among robots.

ACKNOWLEDGMENT

The authors would like to thank Neil Burton for his collaboration on presentation of this paper.

REFERENCES

[1] J. B. Mbede, X. Huang, and M. Wang, "Robust neuro-fuzzy sensor-based motion control among dynamic obstacles for robot manipulators," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 2, pp. 249–261, Apr. 2003.

[2] Y. C. Chang and B. S. Chen, "Robust tracking designs for both holonomic and nonholonomic constrained mechanical systems: Adaptive fuzzy approach," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 1, pp. 46–66, Feb. 2000.

[3] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[4] T. H. S. Li, S. J. Chang, and W. Tong, "Fuzzy target tracking control of autonomous mobile robots by using infrared sensors," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 491–501, Aug. 2004.

[5] H. Li and S. X. Yang, "A behavior-based mobile robot with a visual landmark-recognition system," *IEEE/ASME Trans. Mechatronics*, vol. 8, no. 3, pp. 390–400, Sep. 2003.

[6] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige, "Many robots make short work," *AI Mag.*, vol. 18, no. 1, pp. 55–64, 1997.

[7] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auton. Robots*, vol. 4, no. 1, pp. 7–27, 1997.

[8] M. Schneider-Fontan and M. Mataric, "Territorial multi-robot task division," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 815–822, Oct. 1998.

[9] A. Viguria and A. M. Howard, "An integrated approach for achieving multirobot task formations," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 176–186, Apr. 2009.

[10] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 610–621, Jul. 2007.

[11] J. Huang, S. M. Farritor, A. Qadi, and S. Goddard, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 205–215, Apr. 2006.

[12] 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.

[13] D. Gu and H. Hu, "Integration of coordination architecture and behavior fuzzy learning in quadruped walking robots," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 670–681, Jul. 2007.

[14] J. R. T. Lawton, R. W. Beard, and B. J. Young, "Decentralized approach to formation maneuvers," *IEEE Trans. Robot Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[15] N. Wu and M. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 1, pp. 50–57, Mar. 2004.

[16] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *Int. J. Robot. Res.*, vol. 25, no. 5, pp. 431–447, May 2006.

[17] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 6, pp. 777–790, Nov. 2001.

[18] B. Seanor, Y. Gu, M. R. Napolitano, G. Campa, S. Gururajan, and L. Rowe, "3-aircraft formation flight experiments," presented at the 14th Mediterr. Conf. Control Autom., Ancona, Italy, Jul. 2006.

[19] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed- wing miniature UAVs," *Proc. IEEE*, vol. 94, no. 7, pp. 1306–1324, Jul. 2006.

[20] D. J. Stilwell and B. E. Bishop, "Platoons of underwater vehicles," *IEEE Control Syst. Mag.*, vol. 20, no. 6, pp. 45–52, Dec. 2000.

[21] E. Yang and D. Gu, "Nonlinear formation-keeping and mooring control of multiple autonomous underwater vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 2, pp. 164–178, Apr. 2007.

[22] P. Encarna and A. Pascoal, "Combined trajectory tracking and path following for marine craft," in *Proc. 9th Mediterr. Conf. Control Autom.*, 2001, pp. 1–8.

[23] R. Skjetne, S. Moi, and T. I. Fossen, "Nonlinear formation control of marine craft," in *Proc. 41st IEEE Conf. Decis. Control*. Las Vegas, NV, 2002, pp. 1699–1704.

[24] C. Belta and V. K. Kumar, "Abstraction and control of groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, Oct. 2004.

[25] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[26] J. A. Marshall, T. Fung, M. E. Broucke, G. M. T. DEleuterio, and B. A. Francis, "Experiments in multirobot coordination," *Robot. Auton. Syst.*, vol. 54, no. 3, pp. 265–275, 2006.

[27] C. L. Hwang and N. W. Chang, "Fuzzy decentralized sliding-mode control of car-like mobile robots in a distributed sensor-network space," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 97–109, Feb. 2008.

[28] E. Maalouf, M. Saad, and H. Saliah, "A higher level path tracking controller for a four-wheel differentially steeres mobile robot," *Robot. Auton. Syst.*, vol. 54, pp. 23–33, 2006.

[29] G. Antonelli, S. Stefano, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 2, pp. 211–221, Apr. 2007.

[30] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 727–736, 2010.

[31] B. Siciliano and O. Khatib, *Handbook of Robotics*. New York: Springer-Verlag, 2008.

[32] D. M. Bevly and B. Parkinson, "Cascade Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 2, pp. 199–208, Mar. 2007.

[33] S. Han, H. S. Lim, and J. M. Lee, "An efficient localizations scheme for a differential-driving mobile robot based on RFID system," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3362–3369, Dec. 2007.

[34] K. B. Purvis, K. J. Astrom, and M. Mhammash, "Estimation and optimal configurations for localization using cooperative UAVs," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 5, pp. 947–958, Sep. 2008.

[35] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 163–175, Apr. 2009.

[36] H. O. Wang, K. Tanaka, and M. F. Griffin, "An approach to fuzzy control of nonlinear systems: Stability and design issues," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 1, pp. 14–23, Feb. 1996.

Hasan Mehrjerdi (S'09) was born in Tehran, Iran. He received the B.Sc. and M.Sc. degrees in electrical engineering from Ferdowsi University of Mashhad, Mashhad, Iran, and Tarbiat Modares University, Tehran, Iran, respectively. He is currently working toward the Ph.D. degree in electrical and robotics engineering as a member of the Power Electronics and Industrial Control Research Group (GREPCI), Quebec University (École de Technologie Supérieure), Montreal, QC, Canada.

His research interests include mobile robotics, artificial intelligence, mobile robots coordination in known and unknown environments, and nonlinear control.

Maarouf Saad (SM'08) received the Bachelor's and Master's degrees in electrical engineering from the École Polytechnique, Montreal, QC, Canada, in 1982 and 1984, respectively, and the Ph.D. degree in electrical engineering from McGill University, Montreal, in 1988.

He joined the École de Technologie Supérieure, Quebec University, Montreal, in 1987, where he is currently involved in teaching control theory and robotics courses. His research interests include nonlinear control and optimization applied to robotics and flight control systems.

Jawhar Ghommam was born in Tunis, Tunisia, in 1979. He received the B.Sc. degree from the Institut Nationale des Sciences Appliquées et de Technologies (INSAT), Tunisia, In 2003, the M.Sc. degree in control engineering from the Laboratoire d'Informatique, de Robotique et de Microelectronique (LIRMM), Montpellier, France, in 2004, and the Ph.D. in control engineering and industrial computers, in 2008, jointly from the University of Orléans, France, and the Ecole Nationale d'Ingénieurs de Sfax, Tunisia.

He is currently an Assistant Professor of control engineering at the INSAT, Tunisia. He is a member of the research unit on Intelligent Control and Optimization of Complex Systems. His research interests include nonlinear control of underactuated mechanical systems, adaptive control, guidance and control of underactuated ships, and cooperative motion of nonholonomic vehicles.