

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Optimized Neuro-fuzzy Coordination for Multiple Four Wheeled Mobile Robots

H. Mehrjerdi, M. Saad, J. Ghommam and A. Zerigui
Department of Electrical Engineering, École de technologies supérieure,
Quebec University, 1100, rue Notre-Dame Ouest Montréal, Canada

Abstract: In this study, a neuro-fuzzy coordination for multiple mobile robots is presented. In coordination of a group of mobile robots in real environment, the formation and the method for communication of mobile robots are important tasks. For these problems, a method which implies two level controllers architecture based on a PID and an adaptive neuro-fuzzy is used to coordinate multi robots toward goals. A fuzzy logic system is designed with two central behaviors, path following and coordination. The fuzzy model has been identified by subtractive clustering algorithm and trained by adaptive neuro-fuzzy interface system. A learning algorithm based on neural network is developed to tune parameters of membership function, which reduce errors of coordination and path-following. Our coordination is based on a leader-followers conception which means one of the robots acts as a leader and related to its position, follower robots go along with it. Simulation and experimental results demonstrate the effectiveness of the proposed coordination.

Key words: Formation, intelligent control, nonholonomic mobile robots

INTRODUCTION

Multi-mobile robots can do a task faster and more structured together to increase the system effectiveness (Ishida *et al.*, 1994; Ghabcheloo *et al.*, 2006). The applications of multi-robot systems may involve different fields, e.g., industrial, military and service robotics (Wang, 1992; Barfoot and Clark, 2004; Chen and Luh, 1994).

Different formation control strategies (Balch and Arkin, 1998; Gerkey and Mataric, 2004; Cao *et al.*, 1995; Wang, 1992; Ghommam *et al.*, 2010) proposed on this topic. However, multi-robot coordination methods can be divided into three different categories: virtual structure, behaviour and leader-follower. Each of them has advantages and drawbacks.

The virtual structure approach treats the entire formation as a single virtual rigid structure (Do and Pan, 2007; Beard *et al.*, 1999). The main disadvantage of the virtual structure implementation is the centralization, which leads a single point of failure for the whole system.

By behaviour based approach, several desired behaviours are prescribed for each robot and desirable group behaviour emerges without an explicit model of the subsystems or the environment. The limitation of behaviour based approach is that it is difficult to analyze mathematically, therefore it is hard to guarantee a precise formation control (Balch and Arkin, 1998).

In the leader-follower approach (Wang, 1992), one of the robots is designated as the leader, with the rest robots as followers. The follower robots need to position themselves relative to the leader and to maintain a desired relative position with respect to the leader. In order, to prescribe a formation manoeuvre, we need only to specify the leader motion and the desired relative positions between the leader and the followers. When the motion of the leader is known, the desired positions (desired distance and orientation) of the followers relative to the leader can be achieved by local control law on each follower. This approach is characterized by simplicity, reliability and no need for global knowledge and computation. However, the disadvantage related to this approach is that there is no explicit feedback to the formation, that is, no explicit feedback from the followers to the leader in this case.

The problem of leader-follower robot navigation and coordination is broken down: place detection, localization, path planning and formation (Cao *et al.*, 1995). Coordination problem necessitates the knowledge of localization of leader robot and initial positions of follower robots. In leader-follower approach, leader robot should interpret the sensed information to obtain knowledge of its location to be capable of guiding the follower robots. Neuro-fuzzy based approaches are used to navigate mobile robots in known and unknown environments (Zhu and Yang, 2007; Li *et al.*, 2004; Antonelli *et al.*, 2007). In these studies a real-time fuzzy target tracking

control scheme for mobile robots by using sensors are designed. This study focuses its attention to solve the problem of coordination of a group of mobile robots with two level controllers and considering leader-follower conception. The low level controller is designed to adjust the linear and angular velocities of each robot to allow path following with the holding constraint kinematics. On the other hand a high level controller is presented to coordinate mobile robots in different formations. Leader- followers' conception is implied to coordinate among robots which means one of robots act as a leader and depends on its localization give commands to each independent follower robots to guide them on the desired path and formation. In this study we suppose robots move on the smooth paths. The neuro-fuzzy controller is trained with rules as; If robot which is leader is very far or far from followers, it will send commands to the followers robots to stop or decrease their linear velocities till the leader catch them in meanwhile the leader robot moves faster. If leader is ahead on the path, commands will send to the followers to go faster or leader robot will stop until the follower's robots can catch the leader. To reach this kind of formation, we used adaptive neuro-fuzzy controller to train and control robots.

The following problems in this study have been considered:

- Path planning and coordination
- Velocity and position control
- Wireless communication based on Application Programming Interface (API) and sensor network communication

To overcome the above problems for multiple mobile robots in a real environment, the following approaches will be presented:

- Develop an optimized neuro-fuzzy controller to coordinate multi robot systems while achieving different formations and control missions

- Design of a wireless communication with a Zigbee technology which give capability of network communication

In this study, we present a neuro-fuzzy control strategy to coordinate multiple robots by using localization data based on the robot which acts as leader. The proposed approach does not require the position of follower robots in training procedure, only it needs their initial positions. But positions of follower robots are measured to overcome uncertainty on their positions in path following and validate coordination algorithm. It does not require a path to be generated and tracked either for followers. The controller is able to decide about the motion direction at each time interval by processing the information obtained from the leader robot. The simulation and experimental results demonstrate the effectiveness of the developed algorithm.

All controller, path planning and coordination in this study are implemented on multiple mobile robots called EtsRos. These robots have four wheels which two front wheels are dc-motor powered.

MODELING AND KINEMATICS OF MOBILE ROBOT

Figure 1 shows the general model of system which is used to coordinate multiple mobile robots. As can be seen in Fig. 1 sensors mounted on DC motors of the leader robot can localize the leader robot and then its position can be estimated. Path following and coordination between leader and follower robots are given by a neuro-fuzzy algorithm.

Figure 2 shows a kinematics model of the mobile robot. The generalized coordinates are $q = (x,y,\theta,\varphi)$ where, (x,y) the Cartesian coordinates of the rear-axle midpoint, θ measures the orientation of body with respect to the x-axis and φ is the steering angle.

For this mobile robot, the general kinematics model is (Alexander and Maddocks, 1987; Campion *et al.*, 1996):

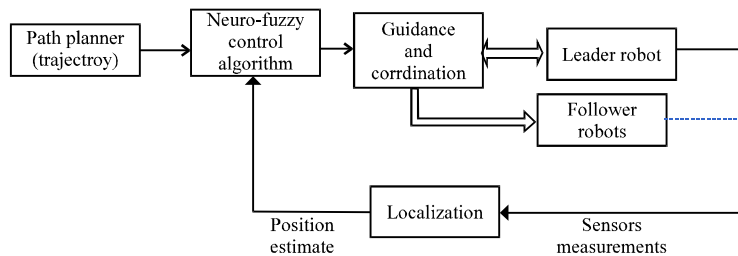


Fig. 1: Infrastructure of multi robot coordination

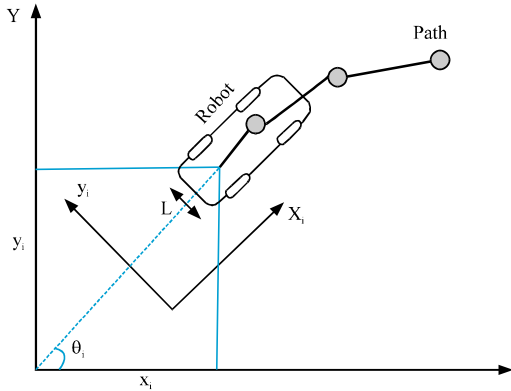


Fig. 2: Model of mobile robot

$$\begin{cases} \dot{x}_i = v_i \cos(\theta_i) \\ \dot{y}_i = v_i \sin(\theta_i) \\ \dot{\theta}_i = \omega_i \end{cases} \quad (1)$$

where, $\eta_i = [x_i, y_i, \theta_i]$ denote the position and orientation vector of the i th robot of the group and v_i is linear velocity and ω_i is angular velocity. The system is subject to nonholonomic constraints such as:

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (2)$$

Linear velocity $v_i(t)$ and the orientation angle $\theta(t)$ with respect to t can be calculated as:

$$v_i(t) = \sqrt{\dot{x}_i^2(t) + \dot{y}_i^2(t)} \quad (3)$$

$$\theta_i(t) = \text{atan2}(\dot{y}_i(t), \dot{x}_i(t)) \quad (4)$$

And angular velocity $\omega_i(t)$ is calculated by deriving of orientation with respect to t as:

$$\omega_i(t) = \frac{1}{1 + (\dot{y}_i(t)/\dot{x}_i(t))^2} \frac{\ddot{y}_i(t)\dot{x}_i(t) - \dot{x}_i(t)\ddot{y}_i(t)}{\dot{x}_i^2(t)} \quad (5)$$

GENERATION OF TRAINING DATA

Here, the generation of training data which are based on the position and trajectory of leader robot and initial positions of follower is explained in detail. In order to plan a path following and coordination for a group of mobile robots, the principal idea is to construct a mapping between the posture of the leader robot and linear and angular velocities of the leader and followers robots. We consider sine paths which is smooth and feasible path such that the robots can follow. The first step is the generation of an appropriate training data set. The input-output data pairs have been obtained by adopting

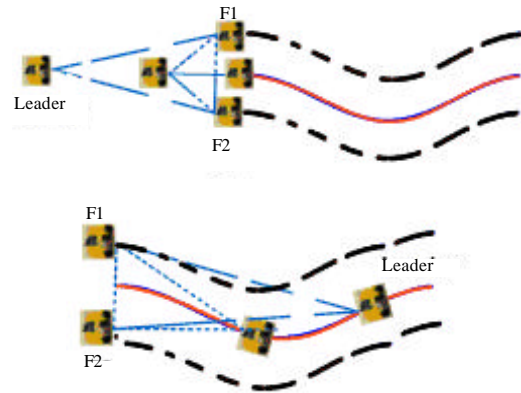


Fig. 3: Different formations of mobile robots

a sine wave path and discretizing it to 400 sampling points. The offline generation of the adopted path is based on knowing path and the initial position of robots. The training data set has been generated for the five different formation corresponding to position of leader and followers mobile robots. These five formation can be considered as follow: (1) leader robot is very far back compares to follower robots, in this situation, leader robot move very fast to reach the follower as follower robots move in desired speed, (2) leader robot is far back, in this situation, leader robot moves fast to reach the follower as follower in a desired speed, (3) leader robot is in the same position (x -reference) as followers robots, in this situation, leader robot moves in the same speed of follower on the path, (4) leader robot is far ahead, in this situation, leader robot stops and follower robots move until they catch leader and (5) Leader is very far ahead, again in this situation, leader robot stops and follower robots move until they catch leader. Figure 3 shows these different formations which are considered for training. In Fig. 3, F_1 and F_2 show the follower robots. Dashed lines show the desired trajectories for leader robots and red line represent desired trajectory for leader robot. Coordination between leader and follower robots is considered in training procedure.

For compatibility with the real Lab environment, a plane of 7 meters in length and 4 meters in width is considered for training which can be used in the simulation and experimental tests. Figure 4 shows the position of leader robot in x , y and θ coordination considering the training with fifth different formations. Figure 5a and b show the obtained linear and angular velocities used for the leader and follower robots in training.

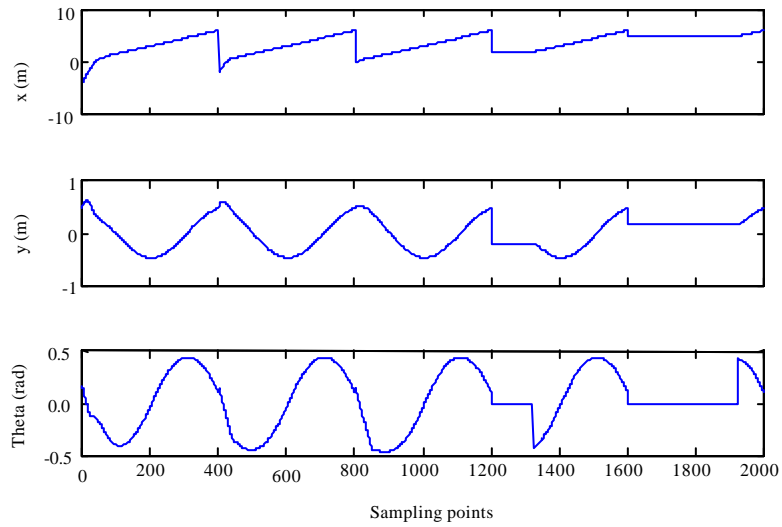


Fig. 4: Position of leader robot in training

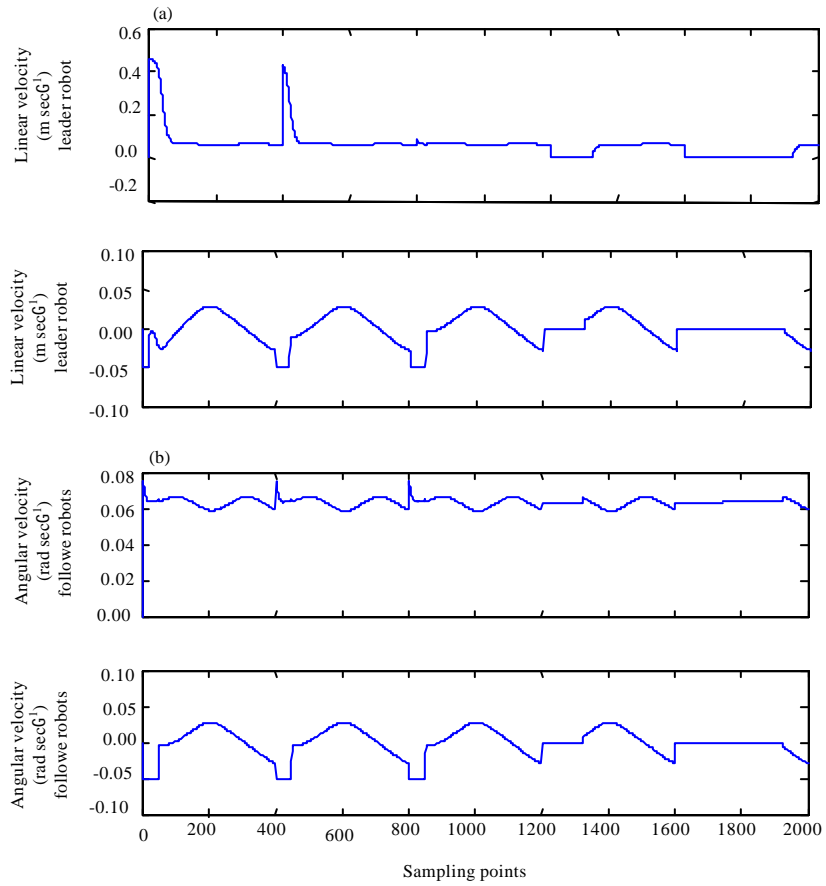


Fig. 5: Linear and Angular velocities of robots. (a) Leader robot and (b) follower robots

Each formation in training has been sampled to 400 points and five different formations are considered. Consequently the training data is a matrix with dimension of 2000×3 as an input and four matrixes which dimension of each is 2000×1 as outputs. The four output data are the linear and angular velocities for the leader and followers robots. We consider all follower robots as a unit and then we imply the same linear and angular velocities to them. With these input and outputs sets the subtractive clustering with Sugeno-type are made to train input-outputs.

NEORO-FUZZY CONTROLLER DESIGN

An effective solution to the online coordination and path following for a multiple mobile robots relies on the fuzzy logic based approach. The position of leader robot gives all necessary information to follow a desired path and coordinate between robots. Fuzzy networks can produce linear and angular velocities for leader and follower robots which result proper steering for robots. We have used the following type of first order Sugeno rules which is:

$$\begin{aligned}
 &R_i: \text{IF } x_i \text{ is } A_{i1} \text{ and } y_i \text{ is } A_{i2} \text{ and } \theta_i \text{ is } A_{i3} \\
 &\text{then} \\
 &v_{li} = P_{0i} + P_{1i} \times x_1 + P_{2i} \times y_1 + P_{3i} \times \theta_1 \\
 &\omega_{li} = Q_{0i} + Q_{1i} \times x_1 + Q_{2i} \times y_1 + Q_{3i} \times \theta_1 \\
 &v_{fi} = R_{0i} + R_{1i} \times x_1 + R_{2i} \times y_1 + R_{3i} \times \theta_1 \\
 &\omega_{fi} = S_{0i} + S_{1i} \times x_1 + S_{2i} \times y_1 + S_{3i} \times \theta_1 \\
 &\text{for } i = 1, \dots, M
 \end{aligned} \tag{6}$$

where, M is number of rules. The inputs to the fuzzy network controller are x_i , y_i and θ_i which present the position of leader robot in each point. These positions are obtained by the incremental encoder sensors mounted on the right and left motors on wheels of leader robot. v_{li} , ω_{li} and v_{fi} , ω_{fi} denote outputs of fuzzy networks which are the linear and angular velocities for leader and follower mobile robots. These fuzzy controllers provide the path following and formation to robots. Depends on the position of leader robot, the linear and angler velocities (v_i , ω_i) are calculated and sent to leader robot and in the same time the linear and angler velocities (v_f , ω_f) are calculated and sent as a command to follower robots. Four different 3-input 1-output training data has been prepared (from the different formations). Fuzzy rule antecedent membership functions have been identified by using the subtractive clustering algorithm with enumerative parametric search. The consequent parameters of the rules are optimized by Least Square Estimation (LSE). To find the appropriate number of rules and minimum Least Square Error (LSE), an optimization is done with different number of rules and the related LSE are obtained. Figure 6a and b show the result obtained for these 4 different fuzzy training sets.

Table 1: Fuzzy clustering data

Factors	Fuzzy network			
	1	2	3	4
Rule No.	8	8	8	8
Trading data error	7.99710e-003	2.01118e-003	8.28771e-004	2.75756e-003
Radius	0.9	0.9	0.5	0.9
Squash factor	0.5	0.5	0.1	0.5
Accept ratio	0.5	0.5	0.9	0.5
Reject ratio	0.1	0.1	0.7	0.3

Figure 6a and b present the error of fuzzy clustering in reference to number of rules for leader and follower robots.

In view of the Number of Rules and LSE's, eight rule numbers are considered for all four fuzzy training networks. Table 1 shows the fuzzy clustering Fig. 6: Training data errors and rule numbers. (a) Leader robot and (b) follower robots date for these fuzzy networks. These data consist of Radius, Squash factor, LSE, accept and reject ratios.

Figures 7a and b show the training results for subtractive clustering for leader and follower robots when inputs are exactly the same as training data.

The structure of the fuzzy inference system with 8 rules is depicted in Fig. 8. In Fig. 8, the inputs x_i , y_i and θ_i are measured by position of leader robot and outputs are the linear and angular velocities.

Figures 9a and b show the result obtained for output velocities after training is done. In this test, the initial position of leader robot is considered as $[x_l(t_0), y_l(t_0), \theta_l(t_0)]^T = [-1, .5, 0]^T$ and the initial position of follower robots are $[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)]^T = [0, 1, .5, 0]^T$ and $[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)]^T = [0, -.5, 0]^T$.

Here, Adaptive Neuro-Fuzzy Inference System (ANFIS) which constructs a Fuzzy Inference System (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone or in combination with a least squares type of method is explained. The parameters associated with the membership functions changes through the learning process. To overcome overfitting the training data set by model and to process of model validation, a set of input output which sufficiently distinct from the training data is considered. In principle, the model error for the checking data set tends to decrease as the training takes place up to the point that overfitting begins and then the model error for the checking data suddenly increases. Figure 10a and b show the set of inputs and outputs used to validate ANFIS. A different formation from training is considered for checking validation. In this situation the initial position of leader robot is considered as $[x_l(t_0), y_l(t_0), \theta_l(t_0)]^T = [-1, .5, 0]^T$ and the initial position of follower robots are $[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)]^T = [0, 1, .5, 0]^T$ and $[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)]^T = [0, -.5, 0]^T$.

The membership functions of the input variables identified by ANFIS are shown in Fig. 11.

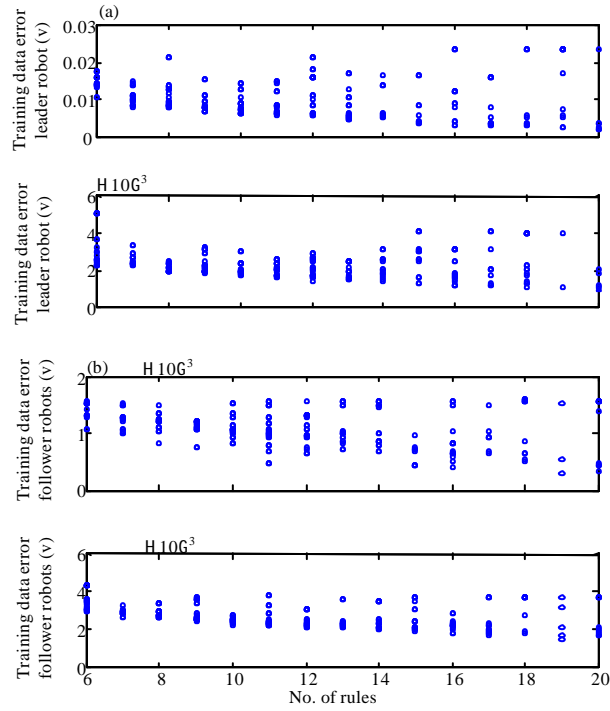


Fig. 6: Training data errors and rule numbers. (a) Leader robot and (b) follower robots

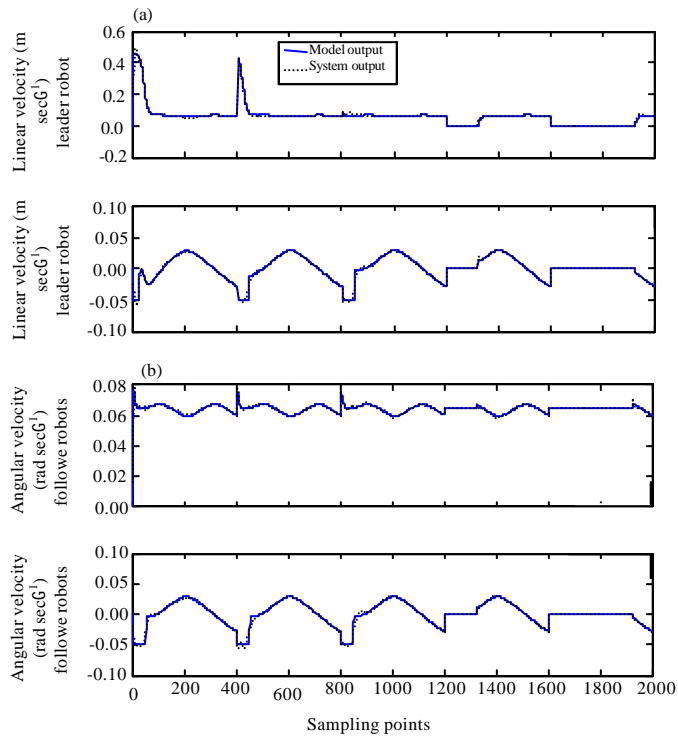


Fig. 7: Linear and angular velocities after a subtractive clustering. (a) Leader robot (b) follower robots

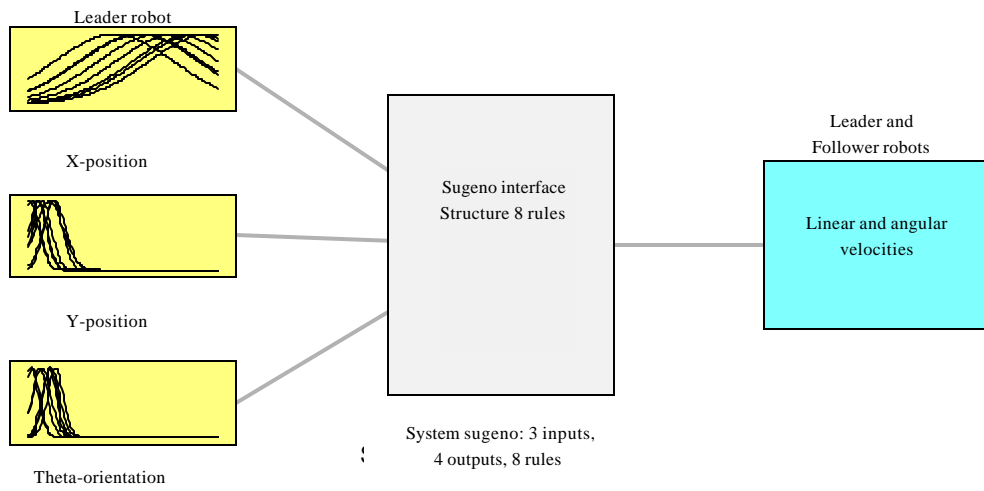


Fig. 8: Fuzzy inference structure

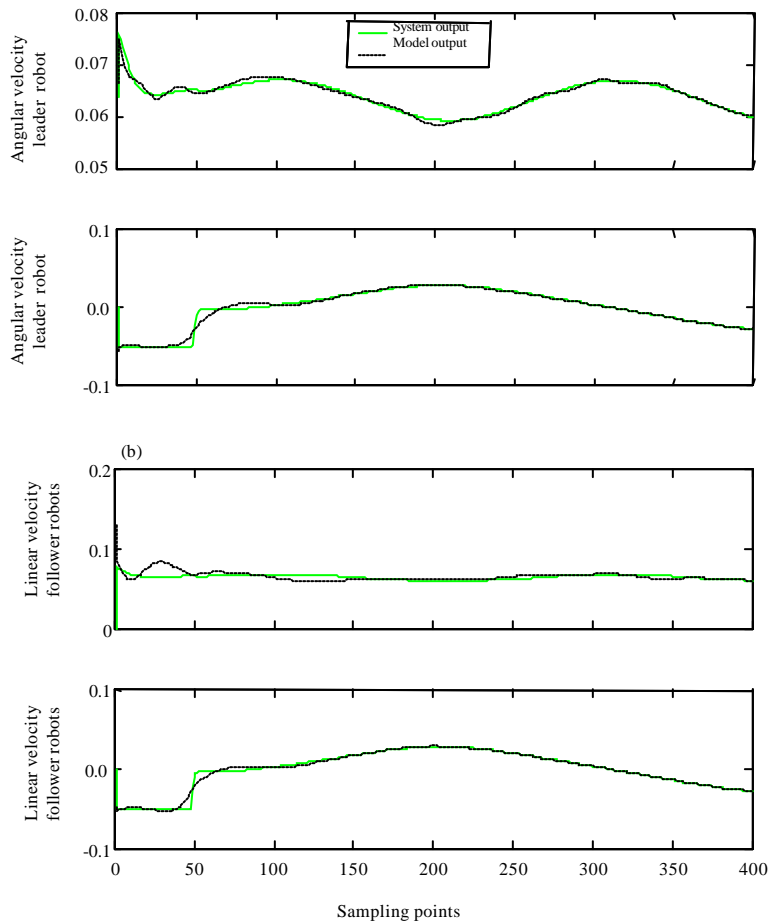


Fig. 9: Linear and angular velocities after subtractive clustering. (a) Leader robot (b) Follower robots

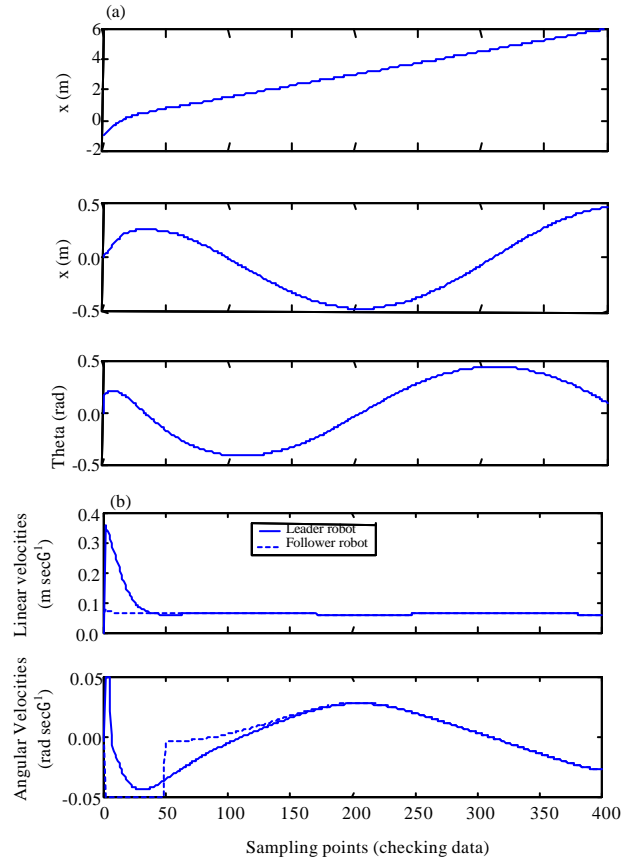


Fig. 10: Checking data. (a) Input data (b) output data

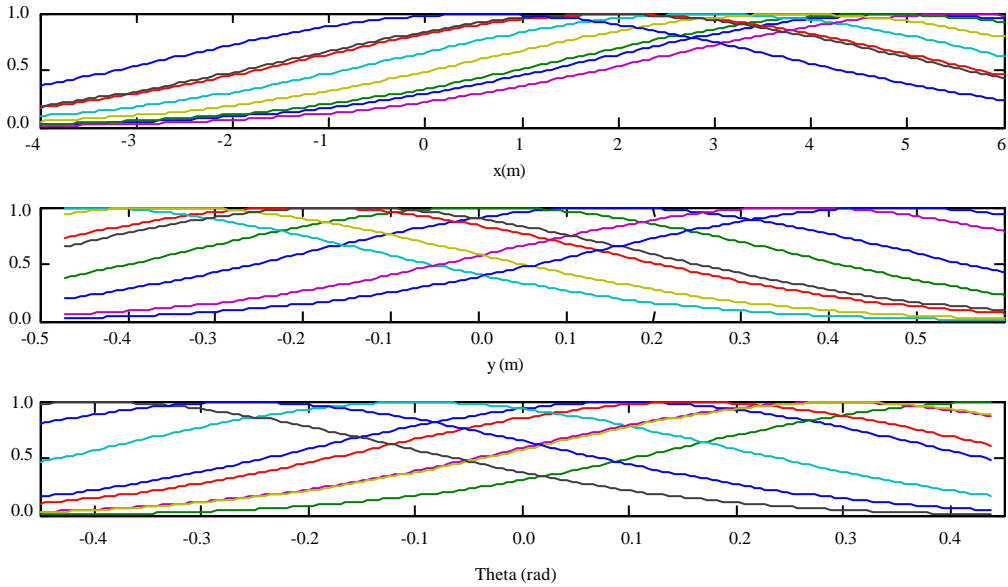


Fig. 11: Antecedent membership functions for the derived fuzzy model (Fuzzy network 1)

Table 2: Antecedent parameters (leader robot)

Rule	Fuzzy network 1 (linear velocity)			Fuzzy network 2 (angular velocity)		
	Input 1 (c,σ)	Input 2 (c,σ)	Input 3 (c,σ)	Input 1 (c,σ)	Input 2 (c,σ)	Input 3 (c,σ)
1	[3.159 5.006]	[0.4799 0.2307]	[0.3132 -0.01601]	[3.201 4.999]	[0.3777 0.2068]	[0.2781 0.08375]
2	[3.177 4.703]	[0.3243 -0.01734]	[0.2728 0.4576]	[3.197 4.666]	[0.3385 -0.009894]	[0.2881 0.4361]
3	[3.251 1.979]	[0.2905 -0.1226]	[0.2502 0.2665]	[3.23 1.992]	[0.3569 -0.2098]	[0.234 0.1455]
4	[3.195 2.906]	[0.335 -0.4667]	[0.2654 -0.08668]	[3.19 5.451]	[0.3604 0.3317]	[0.2451 0.3057]
5	[3.172 5.569]	[0.3231 0.3682]	[0.2276 0.2465]	[3.197 3.704]	[0.3419 -0.3881]	[0.3357 0.2373]
6	[3.194 3.825]	[0.2902 -0.3674]	[0.298 0.2993]	[3.195 2.731]	[0.2556 -0.3129]	[0.3605 -0.2413]
7	[3.201 1.852]	[0.2909 -0.1657]	[0.2697 -0.4027]	[3.19 1.913]	[0.3058 -0.214]	[0.342 -0.3712]
8	[3.184 0.5505]	[0.3902 0.4227]	[0.2462 -0.3412]	[3.182 0.3796]	[0.3895 0.3296]	[0.1725 -0.06248]

Table 3: Antecedent parameters (Follower Robots)

Rule	Fuzzy network 3 (linear velocity)			Fuzzy network 4 (angular velocity)		
	Input 1 (c,σ)	Input 2 (c,σ)	Input 3 (c,σ)	Input 1 (c,σ)	Input 2 (c,σ)	Input 3 (c,σ)
1	[1.729 5.006]	[0.1954 0.1562]	[0.1396 0.06158]	[3.187 4.996]	[0.2632 0.186]	[0.4019 0.2934]
2	[1.748 5.385]	[0.1837 0.2869]	[0.1796 0.3569]	[3.203 4.665]	[0.3505 0.00267]	[0.2626 0.02391]
3	[1.75 5.236]	[0.1473 0.2297]	[0.1429 0.3935]	[3.202 1.999]	[0.2913 -0.2202]	[0.2786 0.4366]
4	[1.733 5.527]	[0.213 0.3657]	[0.162 0.2967]	[3.209 5.452]	[0.3476 0.3457]	[0.2398 0.1336]
5	[1.8 5.063]	[0.1577 0.1604]	[0.1494 0.419]	[3.197 3.707]	[0.3123 -0.3925]	[0.2434 0.3304]
6	[1.762 5.753]	[0.2323 0.4292]	[0.1573 0.2126]	[3.192 2.716]	[0.2893 -0.3877]	[0.303 0.2727]
7	[1.891 4.786]	[0.1576 0.1062]	[0.08245 0.3861]	[3.2 1.835]	[0.4023 -0.1443]	[0.3278 -0.2268]
8	[1.768 4.489]	[0.1777 -0.098]	[0.1538 0.4303]	[3.192 0.5071]	[0.4019 0.2934]	[0.3149 -0.424]

Table 4: Consequent parameters (leader robot)

Rule	Fuzzy network 1 (linear velocity)				Fuzzy network 2 (angular velocity)			
	P _{1i}	P _{2i}	P _{3i}	P _{4i}	Q _{1i}	Q _{2i}	Q _{3i}	Q _{4i}
1	-0.01651	0.0319	-0.0223	0.0771	-0.2398	-5.147	-6.918	3.265
2	0.4022	-0.02227	-0.7788	-1.318	0.221	0.9656	2.945	-3.645
3	-0.1351	-1.20	1.519	1.13	-0.9262	-6.303	7.207	-1.14
4	-0.1884	0.2303	0.3745	0.53	-0.1634	-3.077	-3.415	4.804
5	-0.833	-0.3675	2.264	2.431	0.4762	2.905	0.868	-1.297
6	-0.0098	-1.203	-0.109	0.7862	0.249	3.408	3.207	4.10
7	1.476	-2.942	-1.121	-6.367	-0.2407	0.8054	-3.621	-2.523
8	0.00814	-0.03152	0.05042	0.00193	0.004623	0.01106	-0.189	-0.02402

Table 5: Consequent parameters (follower robots)

Rule	Fuzzy network 3 leader robot (linear velocity)				Fuzzy network 4 leader robot (linear velocity)			
	R _{1i}	R _{2i}	R _{3i}	R _{4i}	S _{1i}	S _{2i}	S _{3i}	S _{4i}
1	1.504	32.85	34.58	-20.4	-0.01683	-6.836	-1.102	3.607
2	1.57	-8.551	-2.619	-15.97	0.04864	-0.4284	0.0635	0.3286
3	5.219	62.63	-27.95	5.955	-0.5693	-6.271	2.212	-2.352
4	-0.573	3.701	2.056	8.091	-0.329	0.7379	-2.326	2.722
5	0.6148	7.805	15.09	-19.11	-0.7134	-1.494	3.19	-0.6362
6	-2.372	-2.137	4.933	16.81	1.704	3.518	2.952	-0.3323
7	4.199	9.684	8.577	0.4569	-1.407	-2.424	-2.939	0.3328
8	0.3871	-5.17	2.612	5.771	0.04004	-0.3503	0.2926	0.3574

The premise parameters which are identified by the cluster center *c* and the standard deviations *σ* of the extracted Gaussian membership function parameters are shown in Table 2 and 3 for all four fuzzy networks.

The consequent parameters as mentioned in Eq. 6 and have been optimized by LSE are given in Table 4 and 5.

Figure 12 shows a general model for adaptive neuro-fuzzy used for modeling of our system. The inputs are position of leader robot and outputs are the velocities of leader and follower robots.

Figure 13a and b show the result obtained for output velocities after our input-output data sets are trained by ANFIS. In this situation the initial position of leader robot is considered as:

$$[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [-1, 5, 0]^T$$

and the initial position of follower robots are

$$[x_{r1}(t_0), y_{r1}(t_0), \theta_{r1}(t_0)]^T = [0, 1, 5, 0]^T$$

and

$$[x_{r2}(t_0), y_{r2}(t_0), \theta_{r2}(t_0)]^T = [0, -5, 0]^T.$$

Figure 14a and b show the LSE's obtained by checking input-output set and ANFIS training with 60 epochs.

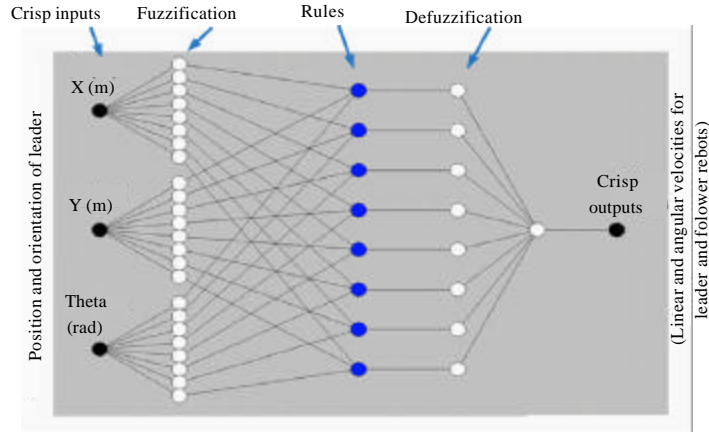


Fig. 12: Adaptive neuro-fuzzy model (fuzzy network 1)

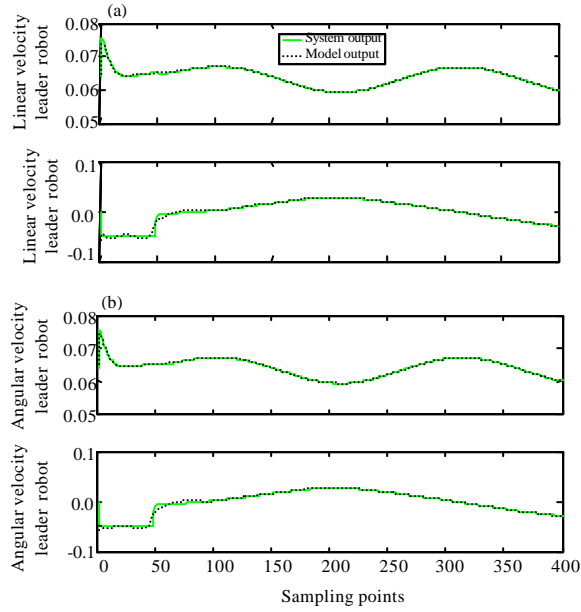


Fig. 13: Linear and angular velocities after ANFIS training. (a) Leader robot and (b) follower robots

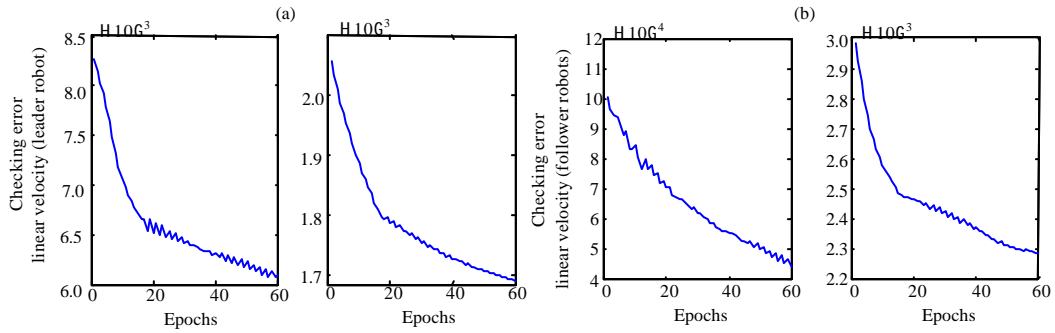


Fig. 14: Checking error versus epoch number. (a) Leader robot and (b) follower robots

EXPERIMENTAL RESULTS

Figure 15 shows the EtsRo mobile robot used for the experimental set up. EtsRo is a four-wheel mobile robot, front wheels are equipped with two DC motors 7.5 Volt, 175 rpm which are installed on the right and left front wheels. The incremental Encoders are mounted on the motors counting with resolution $6000 \text{ pulses turn}^{-1}$. The wheels have radius of $r = 4.5 \text{ cm}$, the length, width and height of EtsRo are 23,20,11 cm, respectively. The total weight of robot is around 2.3 kg. Maximum linear velocity is 1.12 m sec^{-1} and maximum angular velocity is 574 rad sec^{-1} . Figure 16 shows a general view of the experimental set up.



Fig. 15: The mobile robot (EtsRo)

EtsRo has two-level control architecture as can be seen in Fig. 17. Low-level control algorithm is written in C language and run with a sampling time of $T_s = 10 \text{ m sec}$.

There are two second-order filters designed to eliminate sparks on the velocities and two PID controllers are designed to reach the best accuracy for the right and left motors. High-level controller which is a neuro-fuzzy controller is designed in real-time simulink (Matlab) with a sampling time of $T_s = 50 \text{ m sec}$. The PC communicates through a serial port with a modem and with API mode with robots. The speed of ZigBee modem is set on $9600 \text{ bits sec}^{-1}$. In this project ATmega32 microcontroller is used which generates a PWM signal to control the motor speed. The STK500 development board is designed to be programmed from a PC computer and is supported by many software tools, such as the GCC compiler and development environment AVRStudio.

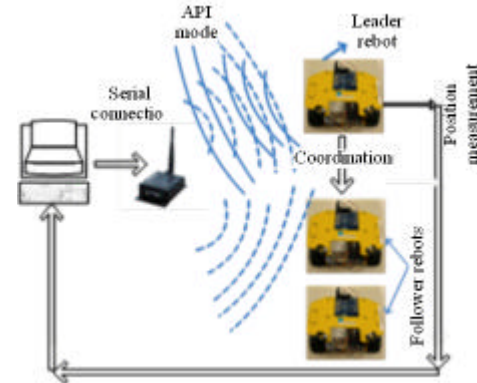


Fig. 16: The general view of experimental set up

ZigBee USB-RF Modem is the communication device which used to transmit the data. The range of modem is about 100 m which make it a high-quality Modem for indoor applications. We need to communicate with the three independent entities from a single transmitter. In

addition, all communication should be bidirectional. To do this, a personal network is created to communicate with nodes (robots).

An address is assigned to each node and API mode is used to address the packets. In API mode, each node

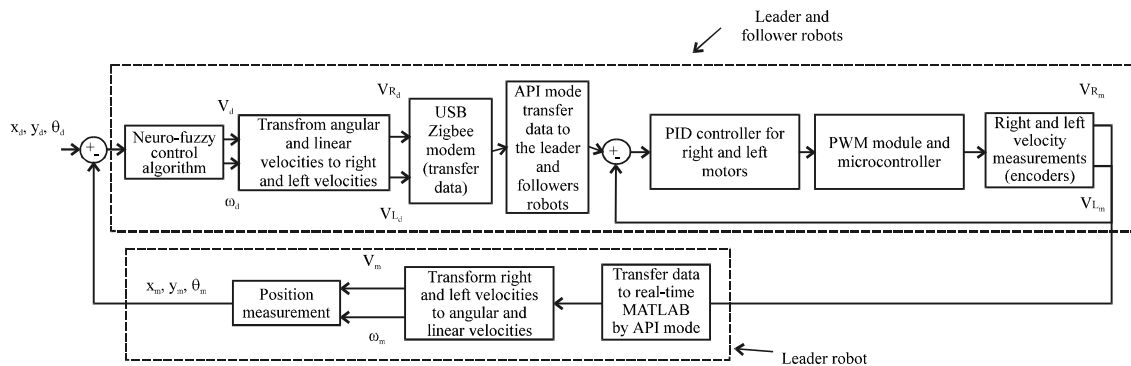


Fig. 17: Intelligent coordination and path-following architecture for multiple mobile robots

sending its data preceded by a frame which contains the destination address. The frame must be completed by a check sum which allows the receiver to confirm receipt of intact data. Leader robot send information of its position to the PC, PC receives the posture information, processes and referrals commands to each individual robot.

In each time, n data bytes will be send from PC to the robot, the size of data is 2×8 bits. Each robot sends back its position to the microcontroller and from there, left and right speeds are calculated and second-order filters remove spikes. Using the following equation we can find the right and left motor velocities:

$$v_R(t) = v + (L \omega/2) , v_L(t) = v - (L \omega/2) \quad (7)$$

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

And then with using of the following equation, we obtain the linear and angular velocities:

$$v = \frac{v_R(t) + v_L(t)}{2} , \omega = \frac{v_R(t) - v_L(t)}{L} \quad (8)$$

And Cartesian coordination of mobile robots with:

$$x = v \cos(\theta) , y = v \sin(\theta) , \theta = \int \omega dt \quad (9)$$

In the first experimental test, the leader robot is placed far back of the follower robots with initial position of $[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [-1, 5, 0]^T$. And the initial position of follower robots are $[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)]^T = [0, 1.5, 0]^T$ and $[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)]^T = [0, -5, 0]^T$ accordingly. Figure 18 shows the results obtained in x,y coordinate which presents the position of mobile robot in path. As we can see in Fig. 18, the leader moves fast to catch the follower robots as it sends commands to followers to move in the slower speed. As soon as leader robots catch the followers, all robots continue to move on the sine path with the same speed.

Figure 19a and b show the reference and measured linear and angular velocities obtained by fuzzy networks and given to leader and follower robots.

In the second experimental test, the leader robot is placed far ahead of the follower robots with initial position of $[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [-1, 5, 0]^T$. And the initial position of follower robots are $[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)]^T = [0, 1.5, 0]^T$ and $[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)]^T = [0, -5, 0]^T$ accordingly. Figure 20 shows the results obtained in x,y coordinate. As we can see in Fig. 20, the leader robot stops and sends commands to follower robots to move till all robots they arrive in the same point (x-reference). As soon follower robots catch the leader, all robots continue to move on the sine path.

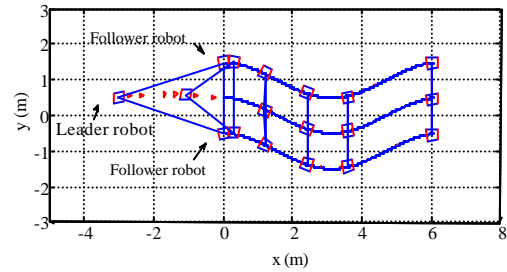


Fig. 18: Coordination and path-following for multiple robots

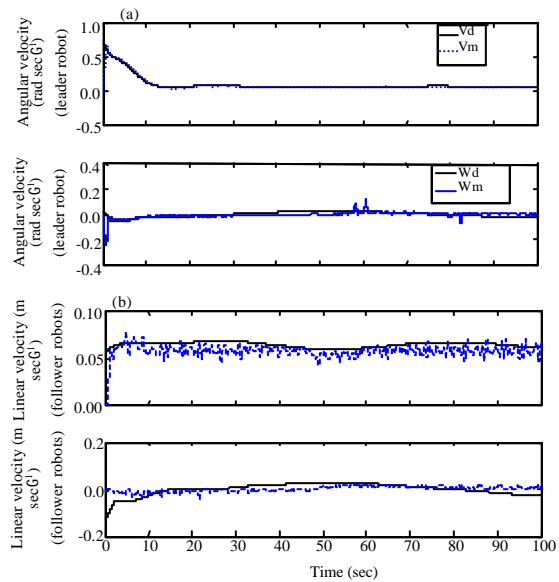


Fig. 19: Linear and Angular velocities for robots. (a) Leader robot (b) Follower robots

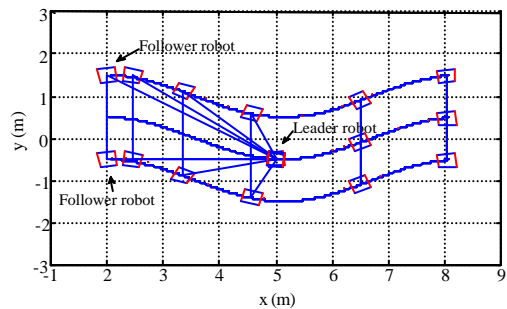


Fig. 20: Coordination and path-following for multiple robots

Figure 21a and b illustrate the reference and measured linear and angular velocities obtained by fuzzy networks and given to leader and follower robots.

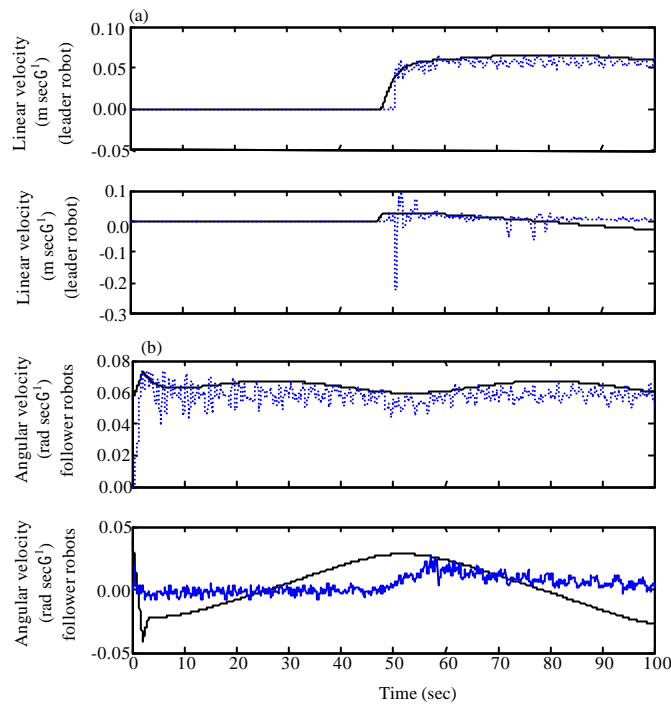


Fig. 21: Linear and angular velocities for robots. (a) Leader robot and (b) follower robots

CONCLUSION

Path following and coordination for a group of mobile robots have been proposed in this study. The approach that has been developed is based on a leader-follower coordination conception and on-line path following. To reach the best performance and minimize the path following and coordination errors, a two level controllers are designed. To control the right and left motors mounted on the right and left wheels, a low level controller (PID) is considered and a high level controller based on a neuro-fuzzy interface is proposed which enable robots to follow the desired path and coordinate with minimum errors. Communication was designed with Zigbee technologies which enable to communicate with multiple mobile robots.

REFERENCES

Alexander, J.C. and J.H. Maddocks, 1987. On the kinematics of wheeled mobile robots. *Intl. J. Robotics Res.*, 8: 15-27.
 Antonelli, G., S. Chiaverini and G. Fusco, 2007. A fuzzy-logic-based approach for mobile robot path tracking. *IEEE Trans. Fuzzy Syst.*, 15: 211-221.

Balch, T. and R.C. Arkin, 1998. Behavior-based formation control for multirobot teams. *IEEE Trans. Robotics Automation*, 14: 926-939.
 Barfoot, T.D. and C.M. Clark, 2004. Motion planning for formations of mobile robots. *Robotics Autonomous Syst.*, 46: 65-78.
 Beard, R.W., J. Lawton, F.Y. Hadaegh, 1999. A coordination architecture for spacecraft formation control. *IEEE Trans. Control Syst. Technol.*, 9: 777-790.
 Campion, G., G. Bastin and B. D'Andrea-Novet, 1996. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. Robotics Automation*, 2: 47-62.
 Cao, Y.U., A.S. Fukunaga, A.B. Kahng and F. Meng, 1995. Cooperative mobile robotics: Antecedents and directions. *Proceedings of the International Conference on Intelligent Robots and Systems*, Aug. 05-09, Pittsburgh, Pennsylvania, USA., pp: 226-234.
 Chen, Q. and J.Y.H. Luh, 1994. Coordination and control of a group of small mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, May 8-13, San Diego, CA, USA., pp: 2315-2320.

- Do, K.D. and J. Pan, 2007. Nonlinear formation tracking control of unicycle-type mobile robots. *Robotics Automation Syst.*, 55: 191-204.
- Gerkey, B.P. and M.J. Mataric, 2004. A formal framework for the study of task allocation in multi-robot systems. *Int. J. Robotics Res.*, 23: 939-954.
- Ghabcheloo, R., A. Pascoal, C. Silvestre, I. Kaminer, 2006. Coordinated path following control of multiple wheeled robots using linearization techniques. *Int. J. Syst. Sci.*, 37: 399-414.
- Ghommam, J., H. Mehrjerdi, M. Saad and F. Mnif, 2010. Formation path following control of unicycle-type mobile robots. *Robotics Autonomous Syst.*, 58: 727-736.
- Ishida, Y., H. Asama, S. Tomita, K. Ozaki, A. Matsumoto and I. Endo, 1994. Functional complement by cooperation of multiple autonomous robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, May 8-13, San Diego, CA, USA, pp: 2476-2481.
- Li, T.H.S., S.J. Chang and W. Tong, 2004. Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. *IEEE Trans. Fuzzy Syst.*, 12: 491-501.
- Wang, P.K.C., 1992. Navigation strategies for multiple autonomous robots moving in formation. *J. Robotic Syst.*, 8: 177-195.
- Zhu, Z. and S.X. Yang, 2007. Neurofuzzy-based approach to mobile robot navigation in unknown environments. *IEEE Trans. Syst. Man Cybernetics Part C: Appl. Rev.*, 37: 610-621.