Research Journal of Applied Sciences, Engineering and Technology 4(11): 1595-1603, 2012
ISSN: 2040-7467

Submitted: January 21, 2012 Accepted: March 10, 2012 Published: June 01, 2012

A New Heuristic Method to Control Cooperating Robots

[1]Maliheh Roghani-Mamaghani and [2]Seyed Hossein Sadati
[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Mechanical Engineering, K.N. Toosi University of Technology, Tehran, Iran

Abstract: This study proposed a new method for control cooperating robots. Many researchers have touched the problem of controlling an array of mobile robots. These controllers have been applied to different kinds of mobile robots. These robots are highly capable in industry due to their low cost and simplicity. Through their simple geometry, they showed to be an appropriate choice for varieties of applications. However, the presence of non-holonomic constraints in their motion renders the control of this robot quite a challenging issue. Conditions for which mobile robots are designed for include many uncertainties since these robots are employed in environments unknown to the robot and hence the robot may be experiencing the workspace for the first time. In other words, the robot may not have been in a similar situation before. In addition, in actual applications, the robot normally suffers from noise and perturbations inflicted upon its control system, making it extremely important to design a control system which would be able to cope with such problems. The fuzzy logic methodology is known to be a proper solution due to its remarkable capabilities. A novel approach based on fuzzy logic has been presented in this study for the formation control of such robots in their concerted motion. Simulation results demonstrate the efficacy of the proposed method.

Key words: Formation control, fuzzy logic, mobile robot, non-holonomic constraint

INTRODUCTION

Formation control: The notion of position control, relative configuration of a collection of robots and their total guidance in harmony for a concerted motion is collectively known as the *formation control*. The concept of a robot formation control (Bazoula and Maaref, 2007; Bazoula *et al*., 2008), is discussed within the realm of biology. Researchers in the natural environmental fields constantly make observations of behaviors of a formation type in many species. They found that many animals purposely form and move in groups so that they can exchange their findings more effectively, enhance their abilities to find foods more easily, and protect themselves against potential dangers, and more generally, to live up to their needs. It is evident that as much they try to live in harmony with their fellow species, they also continuously try to preserve and enforce their privacy and live in a definite distance from their neighbors, hence creating a formation type living environment for themselves.

Fuzzy controllers: Contrary to the past systems of logic (Sisto and Gu, 2006; Duan *et al*., 2008; Raimondi and Melluso, 2005), the well-known fuzzy logic upon which fuzzy control is based is inherently close to the human mentality and it is basically an effective method to express the inexact nature of the surrounding world. The major part of a Fuzzy Logic Controller (FLC) is a set of linguistic rules.

In fact, FLC generates an algorithm capable of converting a knowledge-based control method to an automatic control in an intelligent manner. In the control community, it has been established that the FLC method presents superior results compared with conventional control methods, particularly when the analysis of the conventional controller is qualitatively complex, or when the available data sources are inexact or uncertain. Therefore, fuzzy controller is a step toward restoring communication between typical mathematical control and human decision method. Presently, there is no systematic method to design FLC.

PROBLEM STATEMENT AND ROBOTS FORMULATION

Mathematical overview of the problem: In this section, the mathematical model in state space form for the motion of the robots is presented. Each robot of this study possesses two independent wheels to generate the driving force. There exists a non-holonomic constraint (Alexander

Corresponding Author: Maliheh Roghani-Mamaghani, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

and Maddocks, 1989; Luca *et al*., 2001; Desai *et al*., 2001) corresponding to the two wheels as:

$$\dot{x}\sin - \dot{y}\cos\varphi = 0 \tag{1}$$

$$A = [\sin\varphi \; -\cos\varphi \; 0] \tag{2}$$

λ = Langrange Multiplier

$$A^T\lambda = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix}\lambda \tag{3}$$

The kinetic energy Tand potential energy V of the system become:

$$T = \frac{1}{2}m\left(\dot{x}_G^2 + \dot{y}_G^2\right) + \frac{1}{2}I_G\dot{\varphi}^2 \tag{4}$$

$$V = 0 \tag{5}$$

Using the Lagrange's formulation [6] it can be written

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)^T - \frac{\partial L}{\partial q} = A^T\lambda + Q \tag{6}$$

where the Lagrangian L = T -V of the system is written as:

$$L = T - V = \frac{1}{2}m\left(\dot{x}_G^2 + \dot{y}_G^2\right) + \frac{1}{2}I_G\dot{\varphi}^2 \tag{7}$$

The parameter Q in Eq. (6) denotes the generalized forces. In this case, these forces are written based on the applied torques on the left and right wheels of the robot. Hence, the generalized forces along the x, y, and φ directions here become:

$$Q = \begin{pmatrix} Q_x \\ Q_y \\ Q_\varphi \end{pmatrix} = \begin{pmatrix} \frac{\tau_L + \tau_R}{r}Cos\varphi \\ \frac{\tau_L + \tau_R}{r}sSin\varphi \\ \frac{b}{2}\frac{\tau_L + \tau_R}{r} \end{pmatrix} \tag{8}$$

where τ_L and τ_R are the left and right applied torques to the left and right wheels, respectively,r is the wheel radius, and b is the baseline distance between the two wheels. From Eq. (6), the component equations become:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_G}\right) - \frac{\partial L}{\partial x_G} &= -\lambda\sin\varphi + Q_x \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}_G}\right) - \frac{\partial L}{\partial y_G} &= \lambda\cos\varphi + Q_y \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}}\right) - \frac{\partial L}{\partial \varphi} &= Q_z \end{aligned} \tag{9}$$

According to the above formulas, the following equations can be written:

$$\left(\frac{\partial L}{\partial \dot{x}_G}\right) = m\dot{x}_G; \left(\frac{\partial L}{\partial \dot{x}_G}\right) = m\dot{y}_G; \left(\frac{\partial L}{\partial \dot{\varphi}}\right) = I_G\dot{\varphi} \tag{10}$$

Combining these three and finding their time derivatives results in:

$$\frac{d}{dt}\begin{pmatrix} \frac{\partial L}{\partial \dot{x}_G} \\ \frac{\partial L}{\partial \dot{y}_G} \\ \frac{\partial L}{\partial \dot{\varphi}} \end{pmatrix} = \begin{pmatrix} m\ddot{x}_G \\ m\ddot{y}_G \\ I_G\ddot{\varphi} \end{pmatrix} \tag{11}$$

$$\begin{cases} m\ddot{x}_G = \frac{\tau_L + \tau_R}{r}\cos\varphi - \lambda\sin\varphi \\ m\ddot{y}_G = \frac{\tau_L + \tau_R}{r}\sin\varphi + \cos\lambda\varphi \\ I_G\ddot{\varphi} = \frac{b}{2}\frac{\tau_L - \tau_R}{r} \\ \dot{x}_G\sin\varphi - \dot{y}_G\cos\varphi = 0 \end{cases} \tag{12}$$

These equations can be written in matrix form as:

$$\begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_G \end{pmatrix}\begin{pmatrix} \ddot{x}_G \\ \ddot{y}_G \\ \ddot{\varphi} \end{pmatrix} + \begin{pmatrix} \lambda\sin\varphi \\ -\lambda\cos\varphi \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\tau_L + \tau_R}{r}\cos\varphi \\ \frac{\tau_L + \tau_R}{r}\sin\varphi \\ \frac{b}{2}\frac{\tau_L + \tau_R}{r} \end{pmatrix} \tag{13}$$

Dropping the subscript G for simplicity and defining the constants k_1 and k_2 as:

$$k_1 = \frac{\tau_L + \tau_R}{r} \tag{14}$$

$$k_2 = \frac{b}{2}\frac{\tau_L - \tau_R}{r} \tag{15}$$

Eq. (12) are rewritten as:

$$\begin{cases} m\ddot{x} = k_1\cos\varphi - \lambda\sin\varphi \\ m\ddot{y} = k_1\sin\varphi + \lambda\cos\varphi \\ I\ddot{\varphi} = k_2 \\ \dot{x}\sin\varphi - \dot{y}\cos\varphi = 0 \end{cases} \quad (16)$$

State space equations: The equations of motion derived in the previous section can be written in state space form as follows. Defining the six state variables x_1, x_2, x_3, x_4, x_5 and x_6 as:

$$\begin{aligned} x_1 &= x \\ x_2 &= \dot{x} \\ x_3 &= y \\ x_4 &= \dot{y} \\ x_5 &= \varphi \\ x_6 &\quad \dot{\varphi} \end{aligned} \quad (17)$$

The complete equations of motion in state space form can be written as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{m}(k_1\cos\varphi - \lambda\sin\varphi) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{1}{m}(k_1\cos\varphi - \lambda\sin\varphi) \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{k_2}{I} \end{aligned} \quad (18)$$

To eliminate the Lagrange multiplier λfrom these equations, the time derivative of the constraint Eq. (1) is obtained as:

$$-\ddot{x}\sin\varphi + \ddot{y}\cos\varphi = (\dot{x}\cos\varphi + \dot{y}\sin\varphi)\dot{\varphi} \quad (19)$$

Using the definition of the state variables of Eq. (17) in (19), the above equation becomes:

$$\frac{1}{m}[-\sin\varphi(k_1\cos\varphi - \lambda\sin\varphi) + \cos\varphi(k_1\sin\varphi + \lambda\cos\varphi)] = x_6(x_2\cos x_5 + x_4\sin x_5) \quad (20)$$

, from which we obtain the following for the Lagrange multiplier λ:

$$\lambda = mx_6(x_2\cos x_5 + x_4\sin x_5) \quad (21)$$

Using this result, the Lagrange multiplier λcan now be eliminated from the state space Eq. (18). Hence, by eliminating λ, the state space Eq. (18) are then rewritten in the final form as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_6\sin x_5(x_2\cos x_5 + x_4\sin x_5) + \frac{k_1}{m}\cos x_5 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= x_6\sin x_5(x_2\cos x_5 + x_4\sin x_5) + \frac{k_1}{m}\sin x_5 \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{k_2}{I} \end{aligned} \quad (22)$$

PRESENTATION OF THE METHODOLOGY

The control method employed here for the motion of the robots in a formation is the "leader/follower" (Wang, 1991) methodology. In this method, one of the robots in the group will move as the leader and all the other ones follow this leader robot. In this way, the problem of guiding and keeping the formation in the motion of the robots is divided into two smaller and simpler problems. To design a formation maneuver, it imply suffices to determine the motion of the leader and the position of the follower robot relative to the leader. When the desired trajectory of the leader is set, using the inverse kinematics, it is possible to determine the required linear and angular velocities of the robot in order that the leader robot moves along the desired trajectory. Once the motion of the leader is set, using local control laws, one can determine the position of all the follower robots for a given formation to take place. Therefore, the problem of robot formation control can be regarded as a special case of the more general robotic trajectory planning.

If we assume a desired position for the leader robot, the follower robot is then able to measure and calculate the distance l between its real position and the position of the leader, and the angle φ between the horizontal and line of sight connecting the leader and follower robots (Fig. 1) (Amoozgar *et al*., 2010). Using these two quantities, it is possible to calculate the relative position of each robot to its desired position as follows:

$$k = \sqrt{(l.\cos\varphi - l_d.\cos(\varphi d))^2 + (l.\sin\varphi - l_d.\sin(\varphi d))^2} \quad (23)$$

$$\alpha = \tan^{-1}\left(\frac{l.\sin\varphi - l_d.\sin(\varphi_d)}{l.\cos\varphi - l_d.\cos(\varphi_d)}\right) \quad (24)$$

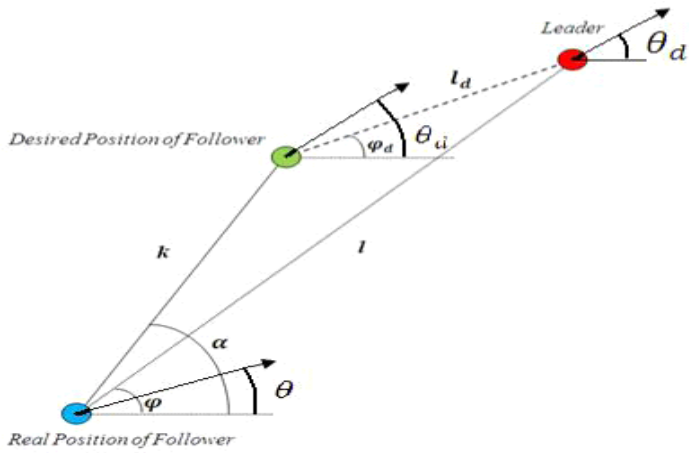


Fig. 1: Robots geometric model (Amoozgar *et al*., 2010).

Fuzzy controller: Based on Eq. (18) and (19), it is inferred that the robot desired formation will occur if the following two conditions are simultaneously satisfied. The system error is defined as below:

$$\begin{cases} k = 0 \\ \theta = \theta_d \end{cases} \tag{25}$$

So if k goes to 0 and θ goes to θd , the total system error will be 0. What is important here is, however, to note that for the follower robot to be on the reference trajectory, it is necessary first for the k to go to zero, after which the decision can be made to move the e_2 to zero. In the other words, by no means can one make the attempt to make the e_2 error go to zero and expect to decrease the distance k from the robot to the reference point, for if e_2 becomes zero, the robot will move in parallel to the reference trajectory. Therefore, the following two general solution strategies can be set forth.

- The follower robot is far from the reference trajectory. Then, how can it be reverted back to the vicinity of the reference trajectory?
- The follower robot is in the vicinity of the reference trajectory. Then, how can it be made to stay in that the vicinity?

The problem will now be considered with the above in mind. As for the control issues, the main question is how to determine the direction of the robot motion at every moment relative to the desired position of the follower. Is the direction of the robot to be aligned with the angle θ_d, or is it to be some value between zero and this angle. The answer is that the proper direction depends on the situation at the moment in question. To obtain the navigation angle θ_{nav}, use will be made of a two-stage fuzzy controller system. The general solution strategy used here is that the required navigation angle at each moment can be determined based on the l, v, α and θ_d parameters. To obtain the best orientation angle for the robot, linguistic rules can be used as follows:

- The greater the distance of the robot from the reference, the closer the robot orientation angle should be to the line connecting the desired and real positions of the follower.
- The closer the robot is to the reference, the closer the desired angle for the robot orientation must be to the robot angle along the reference direction, θ_d.
- The larger the robot velocity is along the reference, the closer the desired angle for the robot orientation must be to the robot angle along the reference direction, θ_d.
- The smaller the robot velocity is along the reference, the closer the desired angle for the robot orientation must be to the angle α.

As mentioned above, to determine the robot proper orientation angle, the desired response will be obtained in two stages. In the first stage, the orientation angle is calculated for two different cases. In the second stage, considering the robot distance with the reference, a proper decision will be made between the outputs of the two previous stages. The magnitude of the navigation angle at the first stage is calculated based on the following two assumptions of symmetry.

First case: The robot is located in a distance far from the reference point.
Second case: The robot is very close to the reference point.
In the first case, the orientation angle is as follows:

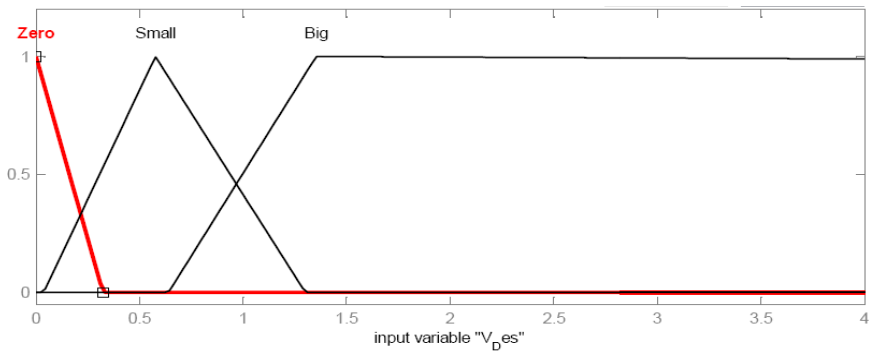


Fig. 2: Fuzzy set to determine the Co_{nav} coefficient

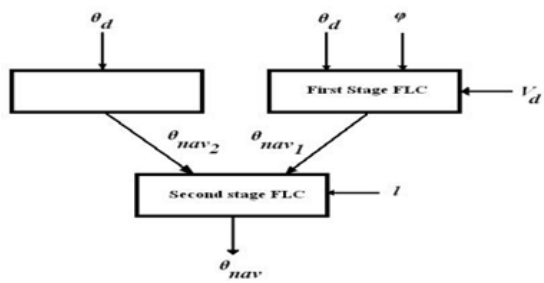


Fig. 3: Schematic view of the two-stage fuzzy controller to determine the navigation angle

$$\theta_{nav1} = Co_{nav}\theta_d + (1 - Co_{nav})\alpha \qquad (26)$$

in which the navigation coefficient varies between 0 and 1, determined by using a fuzzy controller. Three rules are used for the determination of Co_{nav}. If the velocity of the virtual robot along the reference trajectory is large, then

Co_{nav} will increase. If the velocity of the virtual robot along the reference trajectory is small, then Co_{nav} will decrease. If the velocity of the virtual robot along the reference trajectory is zero, then Co_{nav} will be zero as well. The overall fuzzy rules are depicted in Fig. 2.

In the second stage, where the robot is close to reference point, the navigation angle must be chosen equal to the angular direction along the reference trajectory, since the ultimate goal is to track the reference trajectory. Hence:

$$\theta_{nav2} = \theta_d \qquad (27)$$

In the second case, based on the distance the robot has with the reference point, the outputs of the first stage are given a weight so as to find the robot final navigation angle. In Eq. (27), the Co_1 coefficient is variable between 0 and 1, determined by a fuzzy controller. There are five rules to determine Co_1 as:

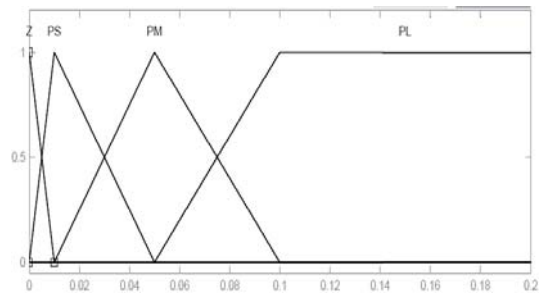
If (K is Zero), then (Co_1 is ONE)
If (K is PS), then (Co_1 is PL)
If (K is PM), then (Co_1 is PM)
If (K is PL), then (Co_1 is PS)
If (K is PVL), then (Co_1 is PVS)

Figure 3 shows a schematic view of the two-stage fuzzy controller.

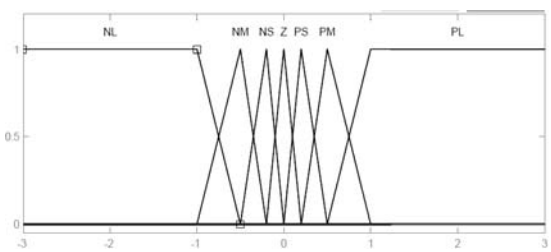
Linear system motion controller: The controller system consists of a prime controller and a correcting controller. In the prime controller, based on the distance to the reference point, an appropriate controller signal is issued. This controller signal is then improved in the second stage, based on the robot orientation. The overall logic is that it is not possible to determine the linear velocity controller simply based on the distance the robot has with the reference point. In fact, with the application of such a controller, it is possible in some situations to incur an adverse effect by causing the robot distance to the reference to increase. As an example, if the robot is a distance far from the reference, and the orientation of the robot is very different from the navigation angle, a large increase in the linear velocity of the robot will simply result in an increase in the distance. Instead, it is best if the robot is set along the right direction before its motion starts. The two inputs to the prime fuzzy controller include the distance to the reference point and its rate. The output is then the applied velocity to the system.

Table 1: The fuzzy rules used for the linear velocity determination

	$\dot{k}$						
k	NL	NM	NS	ZE	PS	PM	PL
ZE	...	...	...	ZE	PS	PM	PL
PS	...	...	ZE	PS	PM	PL	PL
PM	...	ZE	PS	PM	PL	PL	PL
PL	ZE	PM	PL	PL	PL	PL	PL



(a)



(b)

Fig. 4: The prime fuzzy controller set for the linear velocity determination

The controller includes 22 rules in the general form given as:

If k is X and k is Y, then V is Z

These rules are given in Table 1 and the fuzzy rules for the determination of the linear velocity are depicted in Fig. 4.

The corrector controller is obtained by improving the output of the prime controller. This improvement is done by multiplying the prime controller output by a coefficient between 0 and 1. The input of the correcting controller includes the error in the actual and desired orientation angles and the robot distance to the desired point, while its output is the correcting coefficient described above.

The rules used in the controller are given below:

If (k is large) and (error is ZE) ,then (Co_v is One)
If (k is large) and (error is small) ,then (Co_v is PL)
If (k is large) and (error is Big) ,then (Co_v is PS)
If (k is Small) and (error is ZE) ,then (Co_v is One)

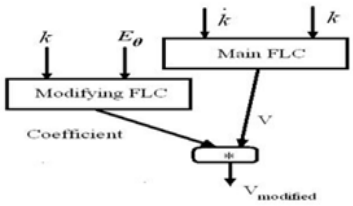


Fig. 5: Two-stage controller for the determination of navigation angle

Table 2: The fuzzy rules used for the angular velocity

	$\dot{e}$						
e	NL	NL	NL	NL	NM	NS	ZE
NL	NL	NL	NL	NL	NM	NS	ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	NL	NS	ZE	PS	PM
ZE	NL	NM	NS	ZE	PS	PM	PL
PS	NM	NS	ZE	PS	PM	PL	PL
PM	NS	ZE	PS	PM	PL	PL	PL
PL	ZE	PS	PM	PL	PL	PL	PL

If (k is Small) and (error is Small) ,then (Co_v is PM)
If (k is Small) and (error is Big) ,then (Co_v is PVS)
If (k is Small) and (error is ZE) ,then (Co_v is One)
If (k is ZE) and (error is Small) ,then (Co_v is PVS)
If (k is ZE) and (error is Big) ,then (Co_v is PVS)

In which the error is defined as:

$$\text{Error} = |\theta_{nav} - \theta| \quad (28)$$

The schematic view of the two-stage controller for the determination of navigation angle is shown in Fig. 5.

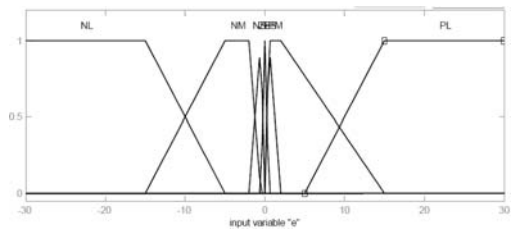
System controller for the angular motion: A second fuzzy controller is also devised to determine the proper angular velocity of the robot. The inputs to this controller are the error between the robot desired and actual sense as well as its time rate.

The controller includes 49 rules in the general form given as:

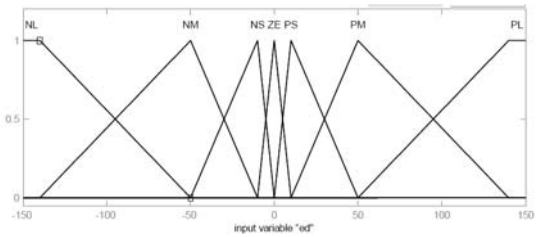
If e is X and $\dot{e}$ is Y, then ω is Z

These rules are given in Table 2 and the fuzzy rules for the angular velocity are depicted in Fig. 6.

Acceleration controller system: In this section, the acceleration required for the robot motion is determined. In order to determine the acceleration of the robot, a fuzzy controller with two inputs is devised. The inputs are the robot distance to the reference point as well as the robot velocity.



(a)



(b)

Fig. 6: The prime fuzzy controller set for the angularvelocity to determine the e, e

Table 3: The fuzzy rules used for the acceleration

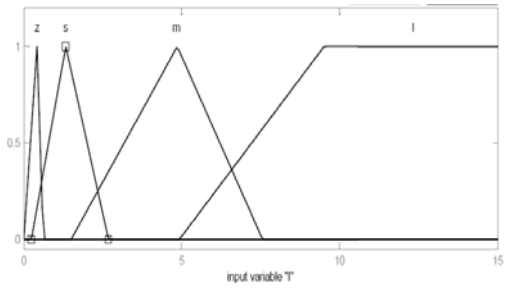
	V			
1	Z	S	M	L
Z	Z	NS	NM	NL
S	S	Z	NS	NM
M	S	S	Z	NS
L	M	M	S	Z

The controller includes 16 rules in the general form given as

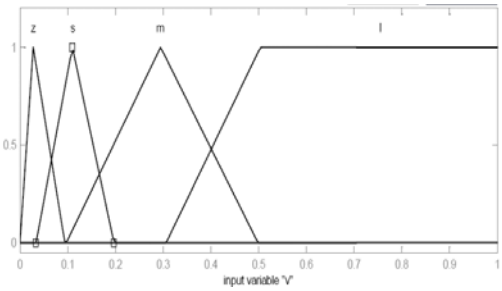
If l is X and V is Y then a is z

These rules are given in Table 3 and the fuzzy rules for the acceleration are depicted in Fig. 7.

Controller system: Figure 8 shows a schematic view of the overall controller system used.



(a)



(b)

Fig. 7: The prime fuzzy controller set for the acceleration to determine the l and v

As shown in Fig. 8, the component fuzzy controllers generate their control signals by receiving their input signals relative to the desired position, as well as the controller feedback from the current position and robot velocity. The CTM box in Fig. 8 generates appropriate signals for the robot actuators to drive the system. In other words, by generating the required signals to follow the motion, the fuzzy controller calls for the proper control signal from the CTM controller. As such, the appropriate torque signals determined by the CTM controller are handed over to the robot in the next box in order to drive the robot wheels. The CTM torque control formulation is given bellow as:

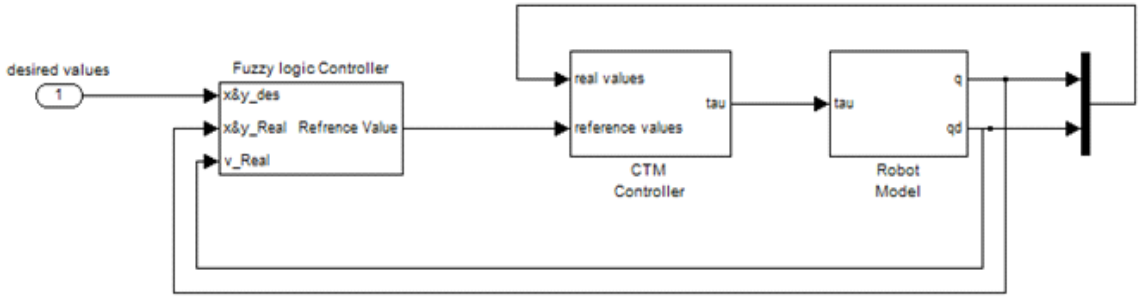


Fig. 8: Overall schematic view of the fuzzy controller system used

$$\zeta = EM^{-1}(\ddot{q} + K_d(\dot{q}_{des} - \dot{q})) - A^T \lambda \quad (29)$$

where M, E, A and λ are defined as:

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \quad (30)$$

$$E = \begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ -b2 & b/2 \end{bmatrix} \quad (31)$$

$$A = [-\sin(\theta) \quad \cos(\theta) \; 0] \quad (32)$$

$$\lambda = m\,\theta_{des}\,(x_{des}\cos(\theta) + (y_{des}\sin(\theta)) \quad (33)$$

To drive the robot, the torque output signals generated in the CTM controller are transmitted to the robot wheels. Hence:

$$\ddot{q} = M^{-1}(A^T \lambda + E\,\lambda)$$

SIMULATION RESULTS

In order to assess the overall controller designed for driving the robot, the controller was simulated in which a group of three robots was to move a given desired robot configuration along the trajectory. These routes are shown below in Fig. 9.

In the Fig. 10, The robots move along a straight path in a given configuration.

x = 1.5 T

y = 1.5 T

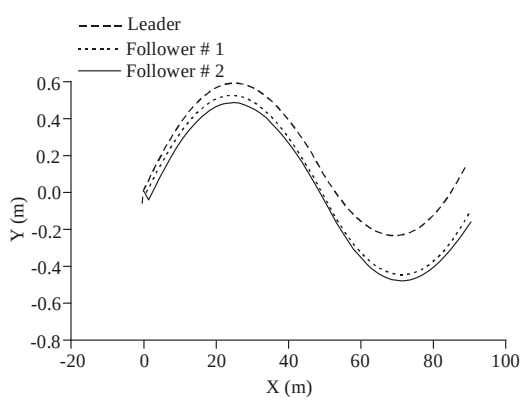


Fig. 9: Motion along sine wave trajectories with low altitude and frequency

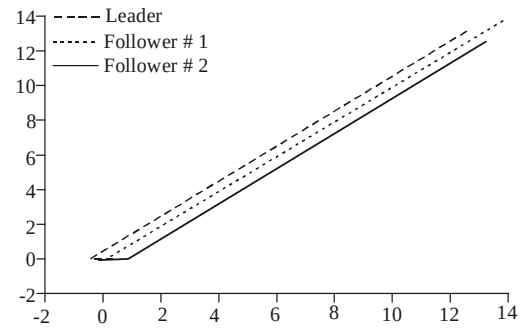


Fig. 10: Motion in a straight line

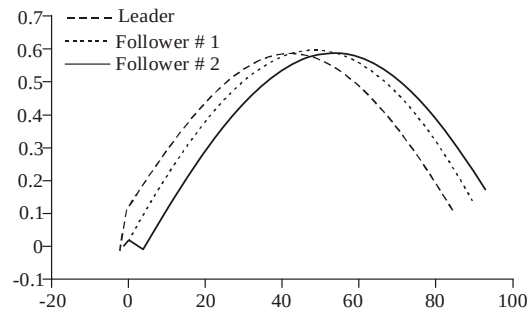


Fig. 11: Motion along a half-sine wave

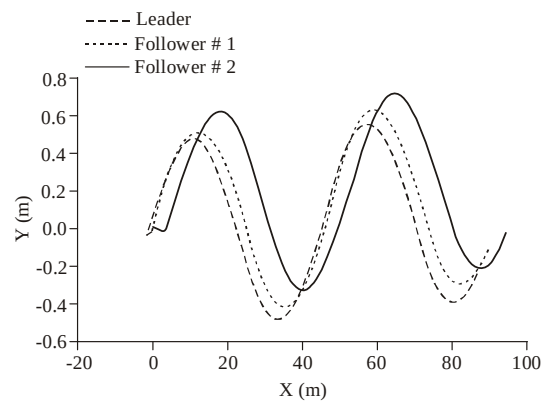


Fig. 12: Motion along sine wave trajectories with low altitude and high frequency

In Fig. 11, the robots follow a half-sine wave given by the parametric equations:

X = 1.5 T, y = 0.5 sin (0.05 T)

The motion of the robots in sine waves is shown in Fig. 12. The corresponding parametric equations are:

x = 1.5 * T, Y = 0.5 * sin (0.2 * T)

Samples of controllers' outputs are shown in Fig. 13 and 14.

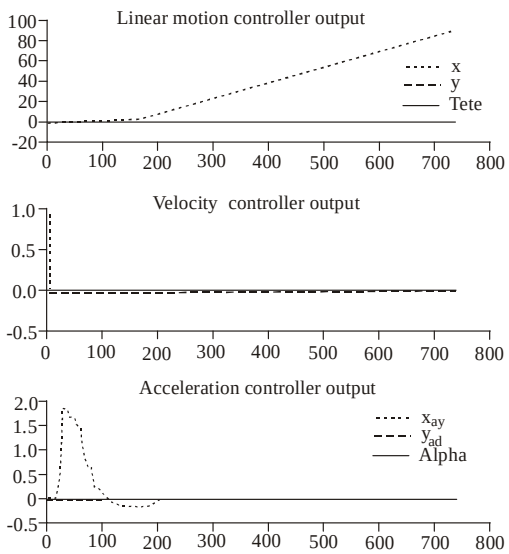


Fig. 13: The fuzzy controller outputs for the leader robot

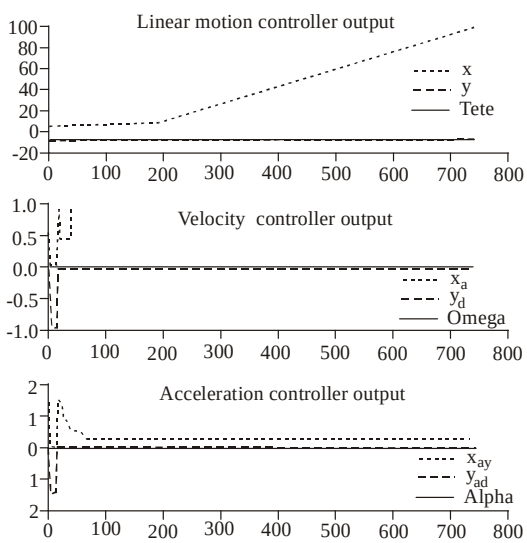


Fig. 14: The fuzzy controller outputs for the follower robot no. 1

The simulation results demonstrate that the performance of the proposed control system for the formation control of a group of robots has been satisfactory. This guarantees the preservation of the required robot formation using the proposed fuzzy controllers. In this configuration, it is required that every follower robot to have a certain robot assigned as its leader.

CONCLUSION

The design and implementation of the proposed fuzzy controller algorithms can be regarded as an effective means for the control and improvement of the motion of a group of robots moving in formation. In addition, this control algorithm can also be considered for further research activities in the robotics community, in particular in the areas of collision avoidance.

REFERENCES

Alexander, J.C. and J.H. Maddocks, 1989. On the kinematics of wheeled mobile robots. Int. J. Robot., 8(5): 15-27.

Amoozgar, M.H., K. Alipour and S.H. Sadati, 2010. A fuzzy logic based formation controller for wheeled mobile robots. J. Indus. Robot., 38(3): 269-281.

Bazoula, A. and H. Maaref, 2007. Fuzzy separation bearing control for mobile robots formation. Int. J. Mech. Syst. Sci. Eng., 1(1): 14-19.

Bazoula, A., M.S. Djouadi and H. Maaref, 2008. Formation control of multi-robots via fuzzy logic technique. Proceedintg of ICCCC.

Desai, J.P., J.P. Ostrowski and V. Kumar, 2001. Modeling and control of formations of nonholonomic mobile robots. IEEE Trans. Robot. Autom., 17(6): 905-908.

Duan, Y., B. Cui and H. Yang, 2008. Robot Navigation based on fuzzy RL Algorithm. Springer.

Luca, A., G. Oriolo and M. Vendittelli, 2001. Control of Wheeled Mobile Robots: An Experimental Overview. Springer-Verlag.

Raimondi, F.M. and M. Melluso, 2005. A new fuzzy dynamic controller for autonomous vehicles with nonholonomic constraints. J. Robot. Autonomous Syst., 52(2-3): 115-131.

Sisto, M. and D. Gu, 2006. A fuzzy leader-follower approach to formation control of multiple mobile robots. IEEE Proceeding International Conference on Intelligent Robots and System.

Wang, P.K.C., 1991. Navigation strategies for multiple autonomous robots moving in formation. J. Robotic Syst., 8: 177-195.