SETIT 2012 IEEE
Sciences of Electronics, Technologies of Information and Telecommunications
Sousse 21-24 March 2012-Tunisia

Path Following Behavior for an Autonomous Mobile Robot using Neuro-Fuzzy Controller

Lakhmissi CHERROUN* and Mohamed BOUMEHRAZ**

*University of Djelfa – Algeria

cherroun_lakh@yahoo.fr

** University of Biskra - Algeria

medboumehraz@netcourrier.com

Abstract: This paper presents a navigation method for an autonomous mobile robot. In order to equip the robot by capability of autonomy and intelligence in its environment, the control system must perform many complex information processing tasks in real time and it is well suited to use the soft-computing techniques. The objective of this paper is to elaborate an intelligent control system for the path following behavior by mobile robot using a neuro-fuzzy controller. The hybrid approach refers to the way of applying learning techniques offered by neural networks for fuzzy systems parameter identification. The proposed controller is used for pursing a moving target. Simulation results show the effectiveness of the designed controller.

Key words: mobile robot, path following, neuro-fuzzy controller, hybrid learning, moving target pursing.

INTRODUCTION

The evolvement of soft-computing paradigms have provided a powerful tool to deal with mobile robot navigation process, which exhibits incomplete and uncertain knowledge due to the inaccuracy and imprecision inherent from the sensory system. Among all the soft-computing methods fuzzy logic based decision-making and neural networks have been found to be the most attractive techniques that can be used for this purpose [TZA 97][FUL 95].

Fuzzy system is tolerant to noise and error in the informations coming from the sensory system, and most importantly; it is a factual reflection of the behavior of human expertise. A fuzzy controller is commonly defined as a system that emulates a human expert. The knowledge of the operator would be presenting in the form of a set of fuzzy linguistic rules [PAS 98]. These rules produce an approximate decision in the same manner as an expert would do. Ever since the fuzzy systems were applied in industrial applications, developers know that the construction of a well performing fuzzy system is not always easy.

The problem of finding appropriate membership functions and fuzzy rules is often a tiring process of trial and error. However, the design of fuzzy logic rules is often reliant on heuristic experience and it lacks systematic methodology, therefore these rules might not be correct and consistent, do not possess a complete domain knowledge, and/or could have a proportion of redundant rules. Furthermore, these fuzzy logic rules cannot be adjusted or tuned on real-time operation, and the off-line adjustment of their parameters is a time consuming process. Another problem could be raised when better precision is needed which is the huge expansion in the fuzzy rule-based system [JAN 98]. In general, there are two approaches to the application of fuzzy logic in mobile robot navigation, namely, behavior-based approach [BRO 86] and classical fuzzy rule-based approach [SAF 97].

Techniques based on the use of Artificial Neural Networks (ANN) have a great interest in control and robotic domains [GAU 99]. The fastness of treatment and their capacity of approximating complex nonlinear functions motivate their use for mobile robot control [JAN 04]. The learning parameters of neural networks made them a prime target for a given task. Learning allows autonomous robots to acquire knowledge by interacting with the environment. This kind of behavior learning methods can be used to solve control problems that robots encounter in real world environment. Artificial neural networks are considered to be simplified mathematical models of brain-like systems [FUL 95]. A neural network (NN) is a processor of information which can be represented in its simplest form by a set of connected and layered processing elements (*PEs*). Each *PE* is able of receiving an *n*-dimensional input vector from either external sources or *PEs* at previous layers, and processing the data to deliver a scalar output, which is the function of a present input. They are generally trained by means of training-data, and due their property of generalization, they can learn new associations, new functional dependencies and new patterns. Due to these properties, they have been widely used for control.

The learning parameters of neural networks made them a prime target for a combination with a fuzzy system in order to automate or support the process of developing a fuzzy system for a given task. Recently the role of neural networks has been found to be very useful and effective when integrated with fuzzy control systems to produce what is called neuro-fuzzy systems [FUL 95]. These hybrid systems provide an urgent synergy can be found between the two paradigms, specifically the capability to mimic human experts in fuzzy logic, and learning from previous experience capability in neural networks. Generally, neuro-fuzzy systems can be classified into two categories, adaptive neuro-fuzzy inference system (ANFIS) [JAN 93] and hybrid neuro-fuzzy systems [FUL 95]. The first category is the most widely used, and they are designed to combine the learning capabilities of neural networks and reasoning properties of fuzzy logic. The main function of neural network is to learn about the fuzzy inference system (FIS) behavior and uses this knowledge to adaptively modify its parameters. The adaptability of the fuzzy inference system can be achieved by either rule base modification and/or membership functions modifications. Rules can be generated, modified, and/or eliminated, while membership functions of the input variables can adjusted and tuned by scaling mechanism [JAN 93]. The basic idea behind the use of the second category is to replace all or parts of the basic modules that builds a FIS [KHA 09]. The only advantage that can be gained from such arrangements is the high processing speed, presuming that a hardware implementation of such neural networks exists.

In this paper an approach to design a simple neuro-fuzzy controller is described for the path following task. The control system generates the appropriate action that will drive the mobile robot straight on the path to reach the final destination. This paper is organized as follows: section 2 gives the necessary background of ANFIS model. In section 3, we will describe the path following behavior. The proposed controller is introduced and explained in section 4. Section 5 shows simulation results for examples of the path following. In section 6, we present a moving target pursing by a mobile robot and the section 7 concludes this paper.

1. Adaptive Neuro-Fuzzy Inference System

(*ANFIS based controller design*)

1.1. ANFIS structure

In this section we introduce the basic of ANFIS network architecture and its hybrid learning rule. Inspired by the idea of basing the fuzzy inference procedure on a feed forward network structure, Jang [JAN 93] proposed a fuzzy neural network model (Adaptive Neural-based Fuzzy Inference System) whose architecture is shown in Figure1. He reported that the ANFIS architecture can be employed to model nonlinear functions, identify nonlinear components on-line in a control system, and predict a chaotic time series. It is a hybrid neuro-fuzzy technique that brings learning capabilities of neural networks to fuzzy inference system. The learning algorithm tunes the membership functions of a sugeno-type fuzzy inference system using the training input-output data. ANFIS consists of five layers; the adaptive nodes of the neural network are the nodes in layers 1 and 4. The depicted model defines a controller with two inputs and one output. Each input has two membership functions. We assume that the rule base contains two fuzzy if-then rules of a Takagi and Sugeno's type:

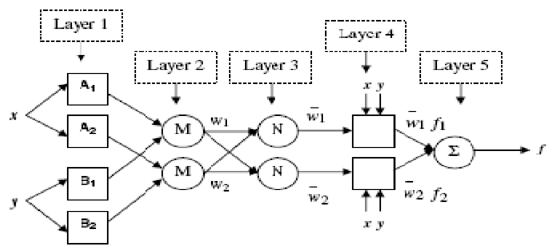


Figure 1. ANFIS structure for TS system with 2 inputs-one output.

R_1 : If x *is* A_1 *and* y *is* B_1 *Then* $y = f_1(x, y) = p_1 x + q_1 y + r_1$

R_2 :If x *is* A_2 *and* y *is* B_2 *Then* $y = f_2(x, y) = p_2 x + q_2 y + r_2$

The output of the nodes in layer 1 is the membership values of the premise part:

$$O_i^1 = \mu_{A_i}(x) \text{ , } i = 1,2 \quad (1)$$

Every node in layer 2 is a fixed node labeled *M*, which multiplies the incoming signals:

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \text{ , } i = 1,2 \quad (2)$$

Every node in layer 3 is fixed node labeled *N* for normalization. it calculates the ration of the *i-th* rule`s firing strength to the sum of all rules firing strengths:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2} \text{ , } i = 1,2 \quad (3)$$

In layer 4, every node is an adaptive node while the node function is*:*

$$O_i^4 = \overline{w}_i \times f_i = \overline{w}_i (p_i x + q_i y + r_i) \text{ . , } i = 1,2 \quad (4)$$

Where $\overline{w}_i$ is the output of layer 3 and p_i, q_i, r_i are the parameters for the first order Sugeno rule.

The overall output of the network can be defined as:

$$O_i^5 = f = \sum_i \overline{w}_i \times f_i \quad (5)$$

1.2 Hybrid learning techniques

Using a given input-output data set, constructs a fuzzy inference system whose membership function parameters are tuned (adjusted) using either a back-propagation algorithm alone or in combination with a least squares method [JAN 93][BOT 05]. This adjustment allows the fuzzy systems to learn from the data they are modeling. If we suppose that the S_1 is the set of premise parameters and S_2 is consequent parameters set.

$S_1 = ((a_{11}, b_{11}, c_{11}), (a_{12}, b_{12}, c_{12}), \ldots, (a_{1p}, b_{1p}, c_{1p}), \ldots, (a_{np}, b_{np}, c_{np}))$
$S_2 = (p_1, p_2, p_3 \ldots, q_1, q_2, q_3 \ldots, r_1, r_2, r_3 \ldots)$

Jang proposed that the learning task is done in two passes using a hybrid learning algorithm as shown at table 1. In the forward pass the first set is fixed and S_2 is optimized by the least square estimate (*LSE*). In the backward pass S_1 is tuning by the back-propagation algorithm [JAN 93].

TABLE 1
Two Passes in the Hybrid Learning procedure for ANFIS

	Forward Pass	*Backward Pass*
Premise Parameters	Fixed	Back-propagation
Consequent Parameters	Least Squares Estimate	fixed

After a learning phase, the controller is able to generate the appropriate actions for the desiring task.

2. Path following for a mobile robot

2.1. Mobile robot model

The mobile robot used in this study is a tricycle robot with non-holonomic property that restricts its mobility in the sideways direction and with limitation of angle. It has two rear driving wheels and a passive front wheel. The inputs of this kinematic system are the steering angle of the front wheel α and the velocity v_r. The outputs are: (x_r, y_r, θ_r) (see Figure 2). In perfect adhesion conditions (movement without sliding), this kinematic model can be described by the following equations [GUE 05]:

$$\begin{aligned} \dot{x}_r &= v_r \cos(\theta_r) \\ \dot{y}_r &= v_r \sin(\theta_r) \\ \dot{\theta}_r &= \frac{v_r}{l} tg(\alpha) \end{aligned} \tag{6}$$

Where Δt is the sample time and l the robot long.

2.2 Path following behavior

The path following is one of the basic missions of a mobile robot navigation. It is a significant task that must have the robot, because it permits this machine to execute its path with a minimum error [GUE 05]. It consists to direct the robot to follow a trajectory at the best possible precision, and must not be necessary that the robot passes exactly through the points on the trajectory, but at least passes in their proximity and arrives to a final destination. Generally, the mobile robot executes its movement with a constant velocity and estimates the trajectory position with its own odometric sensors. The trajectory to follow is stored in the memory in the form of a three elements vector (x_p, y_p, θ_p). In the present work we will only consider a vector with two elements (x_p, y_p) generated by a module named (path generation module). The third parameter will be calculated by the robot during displacement. In its actual position, the robot (Figure 2) calculates the desired orientation θ_d which allows it to go ahead to the desired point of path [GAU 99][GUE 05][BEN 07].

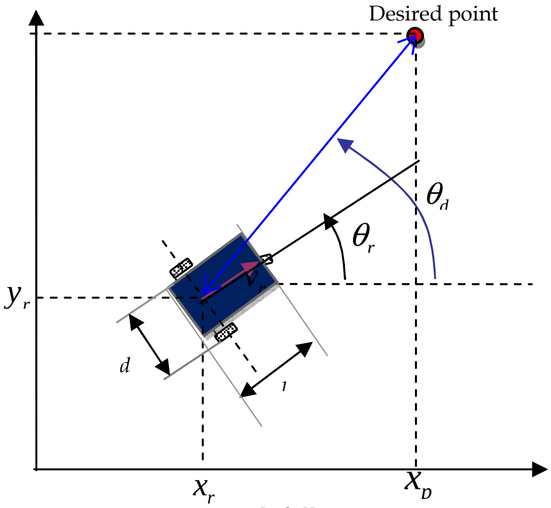


Figure 2. Path following strategy

3. Proposed controller

In this section, the developed approach is discussed. It is based on a simple design method for a path following behavior by a mobile robot. The robot controller is a neuro-fuzzy model. It is a one order Takagi-Sugeno model trained by a data-base for this task. To guide the studied robot to follow the path, at each sample, the movement represented by the angle error *e* is given to the controller to let the robot to reach the next reference. This controller should generate the appropriate action α allowing executing the right movement. Figure 3 presents the proposed neuro-fuzzy controller. The network structure is similar to that of a neural network, which maps inputs through output membership functions and associated parameters, and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map.

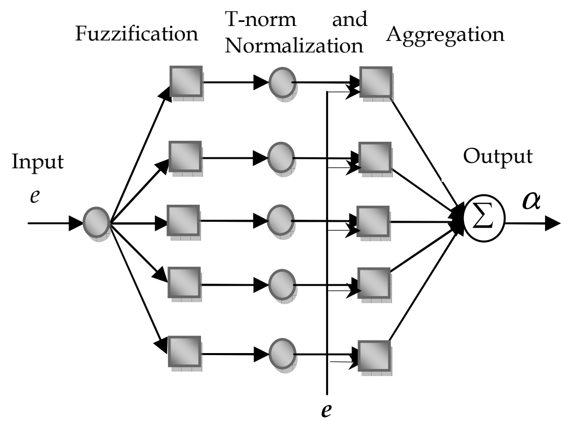


Figure 3. The proposed ANFIS structure

3.1 Learning phase

This phase consists to adjust the fuzzy controller parameters (premise part and conclusion part) using the training data for the path following task. This data-base contains two vectors of the angle error e and its appropriate action α. Figure 4 shows the using of the ANFIS editor to elaborating the robot controller.

3.1.2 Loading training data

Firstly, we begin by loading a training data set that contains the desired input-output pairs of the studied task (Figure 4-a).

3.1.2 Generating the Initial FIS Structure

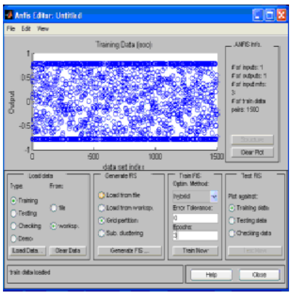
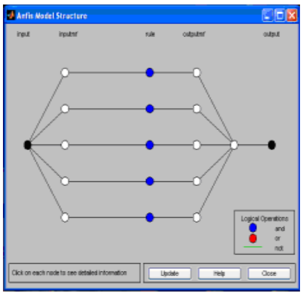
It consists to specify an initial FIS model structure. The generation of this model can be done by the generation of a single-output Sugeno-type FIS using grid partitioning on the data or subtractive clustering one. The graphical representation of the initial FIS model is depicted in figure 4-b.

3.1.3 Training the FIS

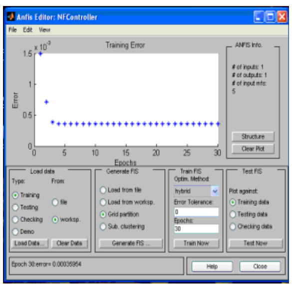
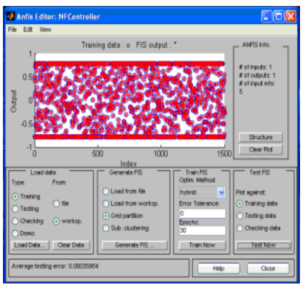
It consists to tuning the FIS parameters. The optimization method is a combination of least-squares and back-propagation gradient descent method in order to emulate the training data. The training error is depicted in (Figure 4-c) to illustrate the learning process. The error decreases and the training stops whenever the maximum epoch number is reached or the training error goal is achieved.

3.1.4 Testing the trained FIS

The figure 4-d compares the FIS outputs with the training data set. We observe the similarity between the two responses. So the controller is able to generate the appropriate action for the robot motion.

(a) (b)

(c) (d)

Figure 4. GUIs for training the ANFIS controller

The input variable of the neuro-fuzzy controller is fuzzified by the five following labels: **PS:** Positive Small **PB:** Positive Big **ZR:** Zero **NB:** Negative Big **NS:** Negative Small.

3.2 Control phase

The bloc diagram of this control system is shown in Figure 5. At each sample, the calculation module compares the actual robot coordinates with the coordinates of the path and computes the desired angle noted θ_d. This value is compared with the orientation of the robot measured by the odometry module in order to compute the error of angle e. The trained neuro-fuzzy controller uses this angle to generate the appropriate action conducting the robot to reach the next reference point of the path.

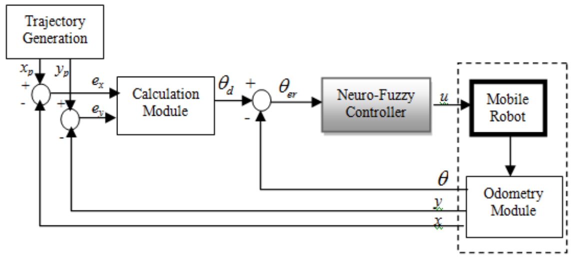
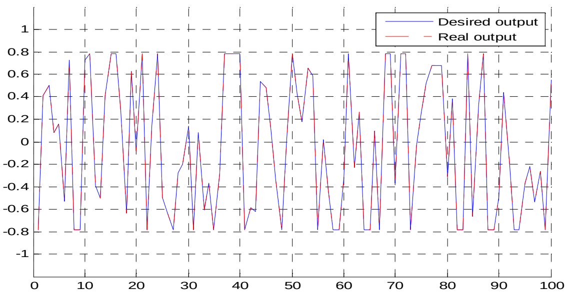


Figure 5. Neuro-Fuzzy controller

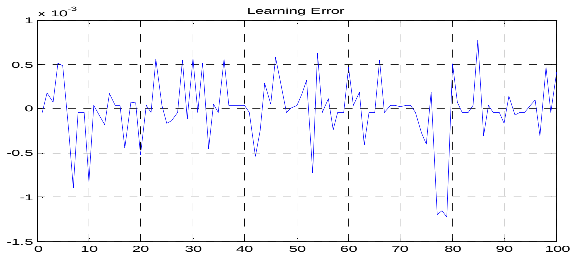
4. Simulation results

To compare the desired control output by the ANFIS one, Figure 6.a presents the two responses. We observe a big similarity between them. As depicted in Figure 6.b, the error is minimized to zeros illustrating the best learning of the used data-base.

To illustrate the effectiveness of the designed controller, we have simulated the robot behavior with different reference paths to provide several direction changes and types of curves. The paths are composed of several segments (discontinuous curve which does not respect the kinematic constraints of the robot). These paths lead the robot to the final destination pn the path. The simulation results for a path with one segment are given in figure 7 (a-b). The path following is good and the robot can follow the path with a minimum error. For different straight paths with different slopes following, the results are shown in Figure 8. In this case, the robot is initially at (10,10) point with a null orientation. As depicted, the proposed controller can behave correctly in all cases and this behavior is realized correctly. The robot tends to overlapping the trajectory by few errors at the first turning.

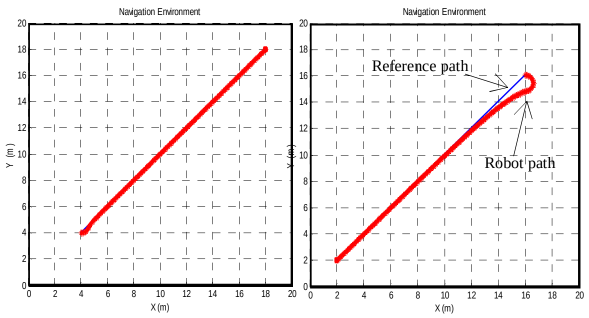


(a)



(b)

Figure 6 (a) the desired control with the ANFIS output (b) the learning error.



(a) (b)

Figure 7. Line following using Neuro-fuzzy controller

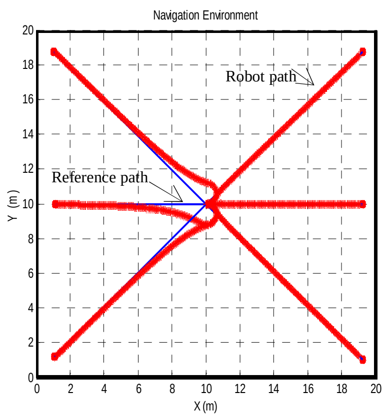
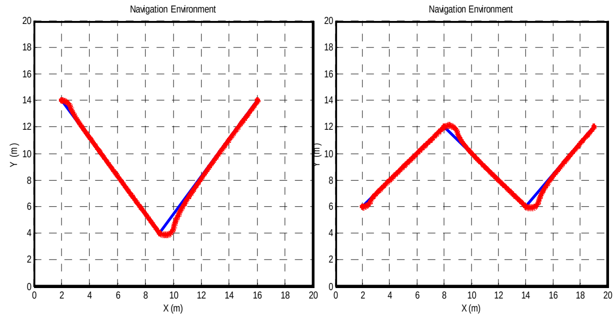


Figure 8. Straight paths following using ANFIS controller

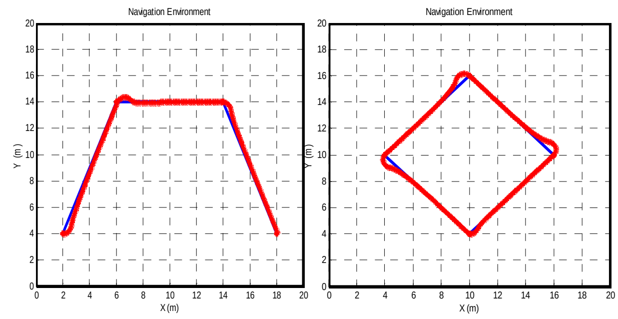
The application of this controller for a V and N shaped references trajectories composed of segments and presenting attenuated angles (not respecting the kinematic constraints of the robot and steering limitation) leads to the robot trajectory depicted at figure 9. The behavior in this case is good; the tracking error exists at changed points of the trajectory. This is due to the sudden direction change and especially to the limitation of the steering angle (control value).

For a trapezoidal and a polygon composed of several segments; the results are shown in figure 10. The path following is satisfactory and the robot tends to overlapping the trajectory by few errors at the turning steps. This is due to the curvature discontinuity which leads to abrupt change of α at segments beginning.



(a) (b)

Figure 9. Robot trajectory using neuro-fuzzy controller, (a) V shaped reference trajectory, (b) N shaped form,



(a) (b)

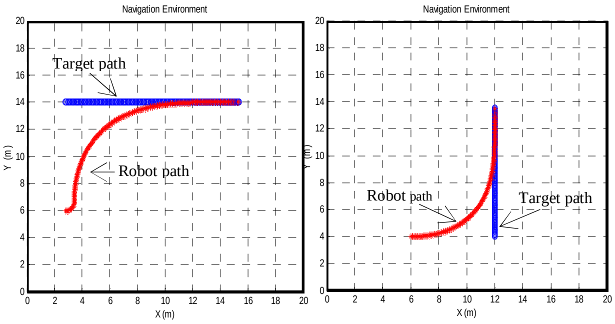
Figure 10. Robot path using ANFIS controller for a trapezoidal path and a polygon path form.

5. Moving target pursing

If we consider a moving target in the robot environment. The mission is to purse this target; while there coordinates are known at any time. The mobile robot purses the target by the calculation of the steering angle that conducts the robot ahead to the target.

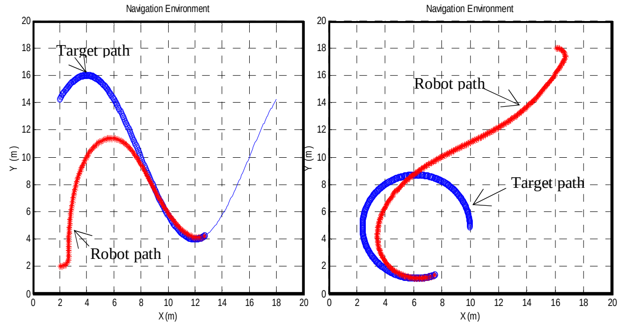
In the first example (Figure 11.a), the mobile robot is initially located at point (3,6) with a null orientation. The target is located at (3,14) and will start moving along a straight line parallel to the abscissa axis with a linear velocity. Until the target starts moving the mobile robot begins pursing it. The pursing will be halted when the robot catches the target. Another example is shown in Figure 11.b, which illustrate the efficiency of the robot control. It is also shown that the proposed controller can guide the robot toward the target.

In figures 12.(a-b), examples of a moving target pursing will be presented. In each example, the target has a different path (sinusoidal and a circle). The robot can purse the target effectively.



(a) (b)

Figure 11. Moving target pursing using ANFIS controller



(a) (b)

Figure 12. Pursing a target with sinusoidal and circular paths

6. Conclusion

In this paper, we have presented a control system based on a neuro-fuzzy model for an autonomous mobile robot movement. ANFIS controller is well suited for controlling a mobile robot because it capable of making inference ever uncertainty with a learning capacity. The elaborated controller is used to realize the path following behavior and for moving target pursing. Different reference trajectories with different curves are simulated. The simulation results show the efficiency of the proposed controller for the robot control and permitting to equip the mobile robot with a certain degree of intelligence. The advantage of the proposed controller is the simplicity and the efficiency for the robot control.

As prospects, it is interesting to improve the robot behavior and apply this strategy on a real mobile robot. The interest of this type of approach will be given to the development of a complete navigation system including other behaviors.

REFERENCES

[TZA 97] S. G. Tzafestas, K. D. Blekas, " Hybrid Soft Computing Systems: A Critical Survey with Engineering Applications", *National Technical University of Athens*, Greece, 1997.

[FUL 95] R. Fulles," *Neuro Fuzzy Systems"*, Abo Academy University. Abo 1995.

[PAS 98] K. M. Passino and S. Yurkovich,*"Fuzzy Control"*, Menlo Park: Addison Wesley, 1998.

[JAN 98] Jan Jantzen," Design of Fuzzy Controllers". *Technical University of Denmark, Department of Automation. Tech. report*, N° 98-E 864 (design), 19 Aug 1998.

[BRO 86] R. Brooks,"A Robust Layered Control System for a Mobile Robot", *Robotics and Automation, IEEE Journal* 2(1):14-23, 1986.

[SAF 97] A. Saffiotti," The Uses of Fuzzy Logic for Autonomous Robot Navigation", *Soft Computing, 1*(4): 180-197, 1997.

[GAU 99] E. Gauthi*er,"Utilisation des Réseaux de Neurones Artificiel pour la Commande d'un Véhicule Autonome*". Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1999.

[JAN 04] D. Janglová," Neural Networks in Mobile Robot Motion", *International Journal of Advanced Robotic Systems*, Vol. 1 N° 1, pp. 15-22, 2004.

[JAN 93] J.-S.R Jang,"ANFIS: Adaptive-Network-based Fuzzy Inference System", *IEEE Trans. on Systems, Man, and Cybernetics*, 23:665-685, 1993.

[KHA 09] Khaldoun K. Tahboub, Munaf S. N. Al-Din, "A Neuro-Fuzzy Reasoning System for Mobile Robot Navigation", *Jordan Journal of Mechanicaland Industrial Engineering*, Vol. 3, N° 1, pp. 77-88, 2009.

[BOT 05] M. Botros," *Application of Fuzzy Logic in Mobile Robots Control*", StudFuzz 163-193. Springer-Verlag Berlin Heidelberg 2005.

[GUE 05] F. Cuesta, A. Ollero, *Intelligent Mobile Robot Navigation*, Springer-Verlag, Berlin Heidelberg, 2005.

[BEN 07] A. Benmakhlouf, D. Djarah and A. Louchene," A Novel Approach for intelligent Control Systems: Application to a path following By a Mobile Robot", *International Conference on Electrical Engineering Design and Technology, Hammamet* Tunisia, 2007.