

## **Intelligent Car Parking Using Fuzzy-Neural Networks**

Antonio Moran Cardenas, Ph.D.  
amoran@upc.edu.pe

Electrical Engineering Department  
Universidad Peruana de Ciencias Aplicadas UPC  
Av. Prolongacion Primavera 2390, Monterrico, Surco, Lima, Peru

### **Abstract**

This paper analyzes the performance and practical implementation of fuzzy-neural networks for the autonomous motion of mobile robots. The designed fuzzy-neural controller is a refined version of a conventional fuzzy controller, and was trained to optimize a given cost function minimizing positioning error. It was found that the mobile robot with fuzzy-neural controller presents good positioning and tracking performance for different types of desired trajectories. It was verified by computer simulation as well as experimentally using a laboratory-scale car-like robot model.

### **Key Words**

Fuzzy-Neural Control, Neural Network Training, Dynamic Back Propagation, Mobile Robot

### **1. Introduction**

There is a significant interest in autonomous mobile robots which can be defined as vehicles that are capable of intelligent motion without requiring neither a guide to follow nor teleoperator (remote) control, but autonomously plan and control their own motion. Many are the potential applications of autonomous mobile robots including reconnaissance/exploratory robots for land, air and undersea environments, remote repair and maintenance in inaccessible places, material handling systems, etc.

This paper analyzes the kinematical modeling of mobile robots as well as the design of control systems for the autonomous motion of the robot using fuzzy-neural networks. Several control systems have been designed on the basis of fuzzy logic and neural networks which are two widely used concepts in the fields of artificial intelligence and nonlinear processing. Fuzzy-neural networks integrate the human knowledge processing abilities of fuzzy logic with the learning and adaptive capabilities of neural networks. Fuzzy-neural networks implement the process of fuzzy reasoning through a neural network structure so that they behave as fuzzy systems with learning capabilities.

### **2. Problem Definition**

The autonomous mobile robot control problem consists in designing a controller capable to move the robot from an arbitrary initial position to a goal position avoiding collision with objects that could be present. The problem can be decomposed in two sub-problems: path planning and motion control. By

solving the path planning problem, the trajectory to be followed by the robot is determined. By solving the motion control problem, the control actions required to achieve the desired trajectory are determined.

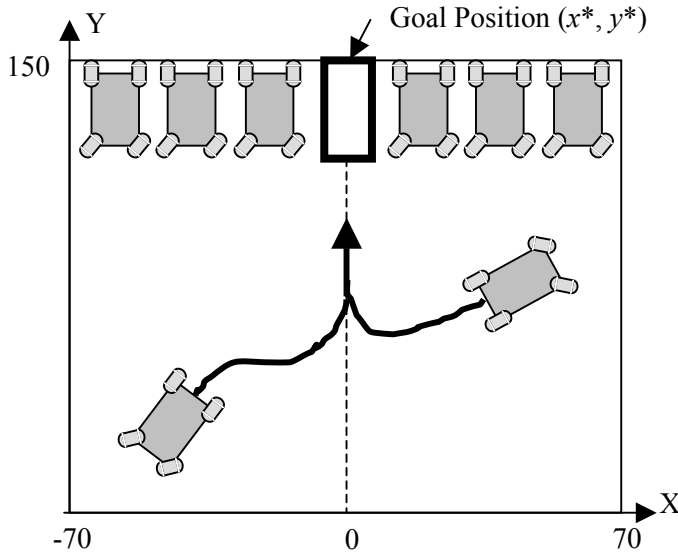


Fig. 1. Mobile robot control problem

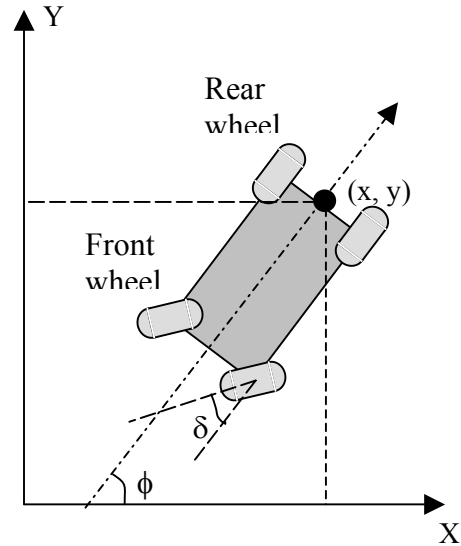


Fig. 2 Mobile robot

The objective of this paper is to design an autonomous control system for mobile robots moving backwards toward the desired position or to follow a desired trajectory. As it is shown in Fig. 1, there is a mobile robot inside a working area represented by the plane X-Y. An autonomous fuzzy-neural controller will be designed to drive the robot from arbitrary initial positions to the goal position given by coordinates  $(x^*, y^*)$ . It is imposed the constraint of no-collision with objects placed at both sides of the goal position.

Figure 2 shows the robot model and the state variables  $x, y$  and  $\phi$  which exactly determine the position of the mobile robot. The coordinate pair  $(x, y)$  specifies the position of the rear center of the robot in the horizontal plane X-Y, and the angle  $\phi$  specifies the inclination of the robot respect to the X-axis. The steer angle at the front wheels  $\delta$  is the control input to be determined given the present coordinate  $x$  and inclination  $\phi$ .

### 3. Robot Model and Control Strategy

#### 3.1 Robot kinematical model

Considering only backward motion and assuming the robot moves a fixed distance  $r$  at every stage, the discrete- time equations describing the planar motion of the robot are:

$$x_{k+1} = x_k + \Delta x_k \dots\dots\dots (1) \quad , \quad y_{k+1} = y_k + \Delta y_k \dots\dots\dots (2) \quad , \quad \phi_{k+1} = \phi_k + \Delta \phi_k \dots\dots\dots (3)$$

$$\text{where } \Delta x_k = r \sin(\alpha_k) \cos(\phi_k - \alpha_k) / \alpha_k \dots\dots\dots (4) \quad , \quad \Delta y_k = r \sin(\alpha_k) \sin(\phi_k - \alpha_k) / \alpha_k \dots\dots\dots (5)$$

$$\Delta \phi_k = 2 \alpha_k \dots\dots\dots (6) \quad \text{and} \quad \alpha_k = r \tan(\delta_k) / 2L \dots\dots\dots (7)$$

$L$  represents the length of the robot. If  $r \ll L$ , then  $\alpha_k = 0$ , and Eqs.4 and 5 can be simplified to:

$$\Delta x_k = r \cos(\phi_k) \dots\dots\dots (8)$$

$$\Delta y_k = r \sin(\phi_k) \dots\dots\dots (9)$$

Equations (4) to (7) have been derived considering only the kinematics of the longitudinal motion of the robot at two consecutive steps. Assuming the robot moves at low speed and without slipping or skidding, the dynamics of the motion of the robot has been neglected.

The state variables  $x$ ,  $y$ , and  $\phi$ , and the steer angle  $\delta$  are limited to the following ranges:

$$-70 < x < 70 \text{ (units)}, \quad 0 < y < 150 \text{ (units)}, \quad -90 < \phi < 270 \text{ (degrees)}, \quad -30 < \delta < 30 \text{ (degrees)}$$

### 3.2. Control Strategy

The control objective is to place the mobile robot at the position with coordinates  $x = x^* = 0$ ,  $y = y^* = 150$  and with vertical inclination  $\phi = \phi^* = 90^\circ$ . Since the robot must not collide with objects fixed around the goal position, the controller will be designed so that the robot firstly moves to the middle of the working area ( $x = 0$ ) with vertical inclination ( $\phi = 90^\circ$ ). After this, the robot will move straightforward to the goal position ( $\delta = 0^\circ$ ). This control strategy does not require the measurement (or estimation) of coordinate  $y$  and can be accomplished if there is enough space between the initial and goal positions.

## 4. Experimental Mobile Robot

The structure of the experimental mobile robot and its control system is shown in Fig.3. The robot is a four-wheels vehicle with front (steering) and rear (traction) wheels connected to two independently controlled stepping motors. The front wheels are properly turned to guide the motion of the robot and the rear wheels are driven to provide the tractive forces to move the robot a fixed distance  $r$  at every stage of control.

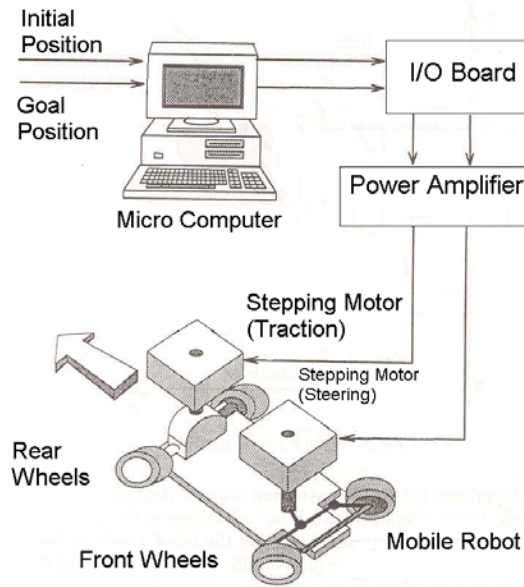


Fig. 3. Experimental mobile robot and control system

The steer angle at the front wheels  $\delta$  is the control input to be determined given the present coordinate  $x$  and inclination  $\phi$  both of which require to be measured or estimated. In order to simplify the structure of the control system, the coordinated  $x$  and inclination  $\phi$  have not been measured but estimated using the mathematical model of the robot kinematics given by Eqs. 1 to 7. Using the robot mathematical model and the algorithm of the fuzzy-neural controller, a microcomputer computes the control signal (steering angle  $\delta$ ) which is sent to a power amplifier through an I/O board to control the angle of rotation of the stepping motor connected to the steering wheels. Since hard sensors are not used to monitor the motion of the robot, it is clear that the robot is controlled in an open-loop configuration. The stepping motor connected

to the rear wheels has been set to rotate a small angle at every stage of control to move the robot a fixed distance  $r$ . This motor rotates slowly enough to avoid slipping of the traction wheels.

## 5. Design of Fuzzy-Neural Controller

Several autonomous control systems have been designed on the basis of fuzzy logic and neural networks which are two widely used concepts in the fields of artificial intelligence and nonlinear processing. In this study a fuzzy-neural controller was designed which is a refined version of a conventional fuzzy controller and was trained to minimize a given cost function defined in terms of the positioning error.

Differing from conventional neural networks whose internal structure is unclear, fuzzy-neural networks have a particular structure which represents the process of fuzzy reasoning. Figure 4 shows the structure of a fuzzy-neural network with inputs  $x_1$  and  $x_2$  and output  $u$ . Although these variables do not correspond to the mobile robot control problem analyzed in this study, they will be used to describe in a simple way the basic structure of fuzzy-neural networks. Similarly as the IF-THEN rules of fuzzy logic, fuzzy-neural networks have *premise* and *consequence* parts connected by a *relation* function which relates normalized membership functions of input and output variables. The first layer of a fuzzy-neural network is composed of neurons each of one representing a particular division of the input variables ( $x_1$  with 3 divisions and  $x_2$  with 2 divisions in Fig. 4). The nonlinear functions of these neurons are selected to be the membership function of the respective division. Neurons in the premise and consequence parts are properly connected to represent the process of fuzzy reasoning, and the nonlinear function of neurons in intermediate layers are selected to mathematically represent the AND and OR operations of fuzzy logic.

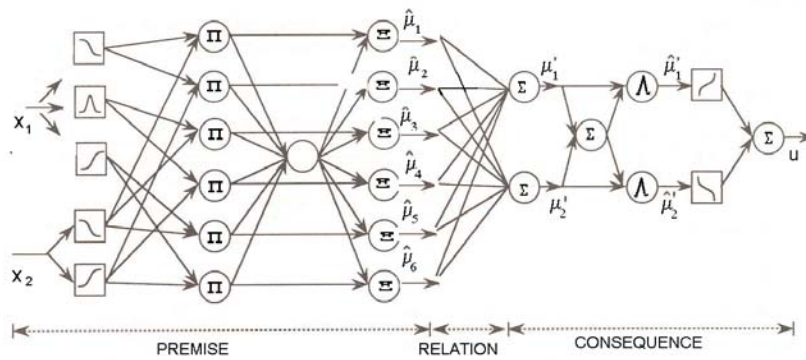


Fig. 4. Structure of fuzzy-neural network

The process to design a fuzzy-neural network is as follows: (a) Represent the fuzzy controller through a fuzzy-neural network. A roughly designed fuzzy controller is enough in the most of cases. (b) Determine approximate initial values for the connection weights  $\omega$  of the relation part of the fuzzy-neural network. For this, linear equations may be formulated by equating the outputs of the fuzzy controller and fuzzy-neural controller for a given set of inputs. (c) Define a cost function to be optimized. (d) Optimize the cost function by updating the connection weights  $\omega$ . Any gradient-based algorithm may be used to update the connection weights.

Given the symmetry of the robot kinematics, the inputs of the fuzzy-neural controller are  $x$  and  $(\phi-90^\circ)$ . Since the control strategy is firstly to move the robot to the closest place with coordinate  $x = 0$  and with inclination  $\phi = 90^\circ$ , the following cost function  $J$  has been defined:

$$J = 0.5 (x_N^2 + \rho (\phi_N - 90^\circ)^2) \quad \dots\dots\dots (10)$$

$x_N$  and  $\phi_N$  are the coordinate and inclination of the robot when it reaches the limits of the working area,  $\rho$  is a positive weighting coefficient determined in order to achieve the performance specifications. The

parameters to be adjusted during network training are the connection weights  $\omega$  of the relation part of the network. The training of the fuzzy-neural controller can be performed using any gradient based optimization algorithm such as the dynamic back propagation algorithm which takes into account the dynamics of the system being controlled. The connection weights  $\omega$  of the neural controller are iteratively updated according to:

$$\omega_{k+1} = \omega_k + \Delta\omega_k \quad \dots\dots\dots (11)$$

where  $k$  is the iteration counter and  $\Delta\omega_k$  is calculated from the following equation:

$$\Delta\omega_k = \eta \partial J / \partial \omega + \beta \Delta\omega_{k-1} \quad \dots\dots\dots (12)$$

$\eta$  is the learning rate and  $\beta$  is the moment coefficient. The total partial derivate  $\partial J / \partial \omega$  is calculated considering the kinematics of the robot motion as given by Eqs.1 to 7. Deriving Eq.12,  $\partial J / \partial \omega$  can be expanded as:

$$\partial J / \partial \omega = x_N \partial x / \partial \omega + (\phi_N - 90) \partial \phi / \partial \omega \quad \dots\dots\dots (13)$$

Defining the state vector  $\mathbf{z} = [x \quad \phi]^T$ , the total partial derivative  $\partial \mathbf{z} / \partial \omega$  and  $\partial \phi / \partial \omega$  in Eq. 13 can be calculated while the robot is running using the following recurrent equation:

$$[\partial \mathbf{z} / \partial \omega]_k = [\partial \mathbf{z}_k / \partial \mathbf{z}_{k-1} + \partial \mathbf{z}_k / \partial \delta_{k-1} \partial \delta_{k-1} / \partial \mathbf{z}_{k-1}] [\partial \mathbf{z} / \partial \omega]_{k-1} + \partial \mathbf{z}_k / \partial \delta_{k-1} \partial \delta_{k-1} / \partial \omega \dots\dots\dots (14)$$

The Jacobian  $\partial \mathbf{z}_k / \partial \mathbf{z}_{k-1}$  and the partial derivative  $\partial \mathbf{z}_k / \partial \delta_{k-1}$  are calculated using the mathematical model of the robot. The partial derivatives  $\partial \delta_{k-1} / \partial \mathbf{z}_{k-1}$  and  $\partial \delta_{k-1} / \partial \omega$  are calculated using the fuzzy-neural controller with a proper modification of the back propagation algorithm.

Compared with conventional neural controllers, fuzzy-neural controllers present the following characteristics: (a) The fuzzy structure of fuzzy-neural controllers allows an easy and direct integration of the human processing abilities of fuzzy logic with the nonlinear an self-tuning capabilities of neural networks. (b) The number of connection weights is generally lesser for fuzzy-neural networks. The number of connection weights to be determined by training can be reduced even more if some desired properties of the neural network (such as symmetry) are taken into account. (c) Pre-training is not required for fuzzy-neural networks. Initial connection weights are determined by solving a set of linear equations. (d) The training time is shorter for fuzzy-neural networks. (f) The structure of fuzzy-neural networks allows a simple tractability of the training process and training objectives. Some desired properties such as symmetry, rotational invariance, etc., can be easily achieved with fuzzy-neural networks.

## 6. Results and Discussion

After the fuzzy-neural controller was trained, its positioning and tracking performance were evaluated. Figure 5 shows the trajectory of the mobile robot for different initial positions and desired position  $x^* = 0$ ,  $\phi^* = 90^\circ$ . It is noted that the robot is able to achieve the desired position without colliding with the obstacles placed around the desired position. It is also noted that the robot develops symmetrical trajectories for symmetrical initial positions. Figure 6 shows the trajectory for the desired position  $y^* = 75$ ,  $\phi = 0^\circ$ . Again, the robot achieves the desired position without collision. Finally, figure 7 shows that the robot is also able to follow linear and nonlinear trajectories.

These results verify the effectiveness of fuzzy-neural networks to be applied for positioning control and trajectory following of mobile robots.

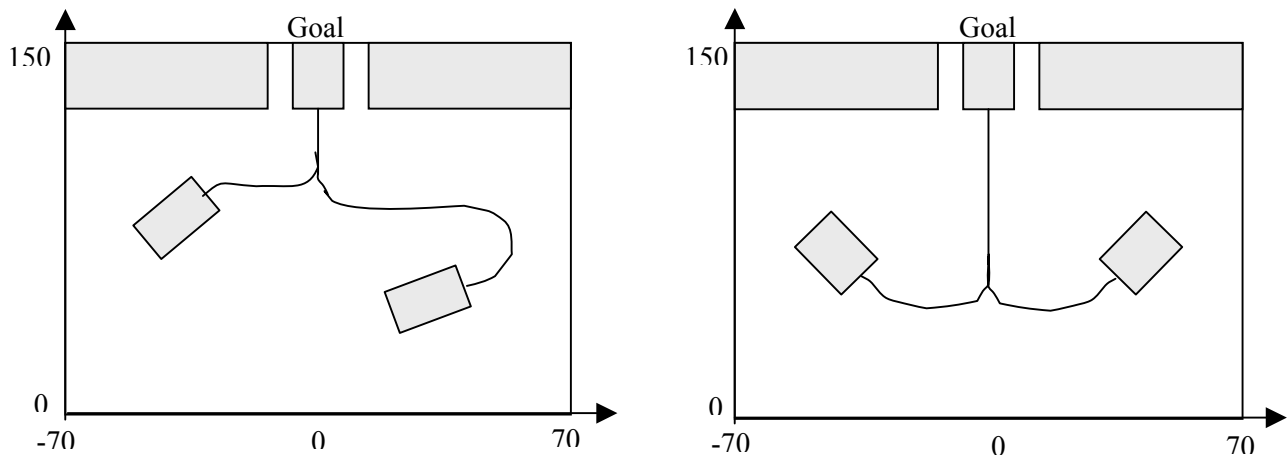


Fig. 5. Trajectories of mobile robot for desired position  $x^* = 0, \phi^* = 90^\circ$

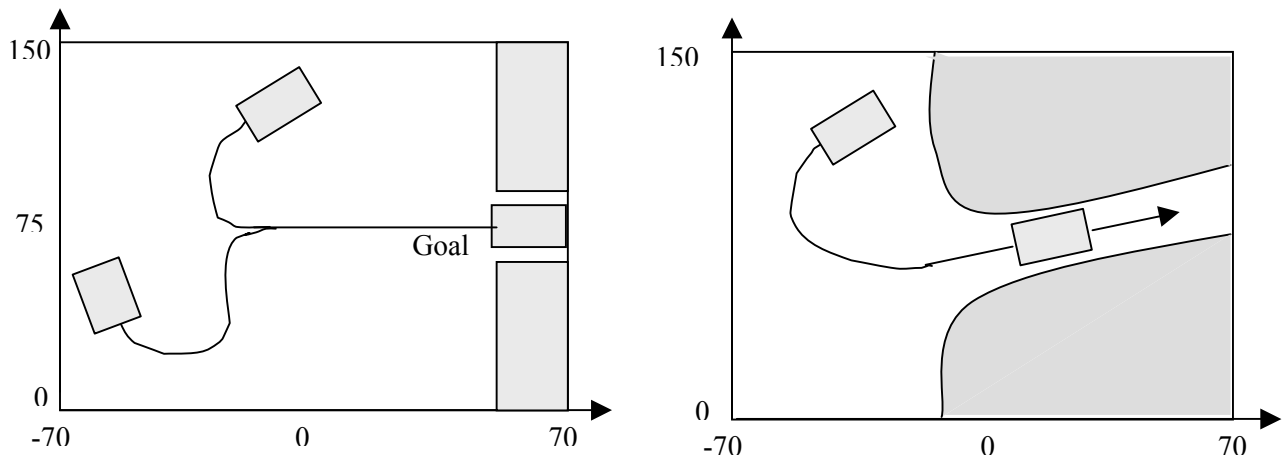


Fig.6. Trajectories of mobile robot for desired position  $y^* = 75, \phi^* = 0^\circ$

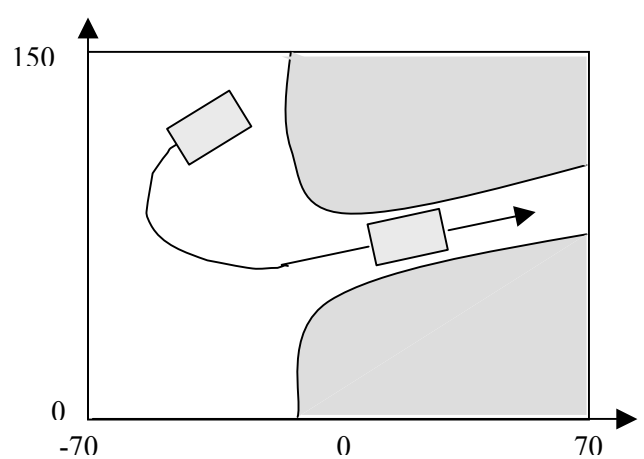


Fig.7. Trajectory of mobile robot for variable desired position

## 7. Conclusions

This paper has analyzed the autonomous motion of mobile robots with fuzzy-neural control. The obtained results show that the robot presents good positioning accuracy and tracking performance.

Fuzzy-neural networks whose internal structure represents the process of fuzzy reasoning, require less training time than conventional neural networks, and its fuzzy structure allows a simple tractability of the training process and a simple way to formulate a symmetrical network.

## 8. References

- (1) A. Moran, and M. Nagai, "Autonomous Parking of Vehicles with Intelligent Fuzzy-Neural Networks", Proc. Of AVEC'94, pp.270-275, Tsukuba, Japan.
- (2) J. Alexander, and J. Maddocks, "On the Kinematics of Mobile Wheeled Robots", Autonomous Robot Vehicles, Springer Verlag, New York, 1990.
- (3) A. Moran, and M. Nagai, "Efficient On-Line Training of Recurrent Networks for the Identification and Optimal Control of Dynamical Systems", Proc. Of IJCNN'93, Vol.2, pp.1789-1792, Nagoya, Japan.