20

Neuro-Fuzzy Navigation Technique for Control of Mobile Robots

Dr. Dayal R. Parhi
Department of Mechanical Engineering, N.I.T.
India

Abstarct

Navigation of multiple mobile robots using neuro-fuzzy controller has been discussed in this paper. In neuro-fuzzy controller the output from the neural network is fed as an input to fuzzy controller and the final outputs from the fuzzy controller are used for motion control of robots. The inputs to the neural network are obtained from the robot sensors (such as left, front, right obstacle distances and the target angle). The neural network used consists of four layers and the back propagation algorithm is used to train the neural network. The output from the neural network is initial-steering-angle. Inputs to the fuzzy-controller are initial-steering-angle (the output from neural network) and left, front, right obstacle distances. The outputs from the fuzzy controller are the crisps values of left and right wheel velocity. From the left and right wheel velocity final-steering-angle of a robot is calculated. The neuro-fuzzy controller is used to avoid various shaped obstacles and to reach target. A Petri-net model has been developed and is used to take care of inter-robot-collision during multiple mobile robot navigation. A piece of software has been developed under windows environment to implement the neuro-fuzzy controller for robot navigation (appendix-1). Six real mobile robots are built in the laboratory for navigational purpose (appendix-2) in reality. By using the above algorithm it is visualised that, multiple mobile robots (up-to one thousand) can navigate successfully avoiding obstacles placed in the environment.

1. Introduction

Developing navigation technique for mobile robot remains on of the frontier research field since last two decades. Many researchers are focusing their thoughts in scientific manner to find out a good navigation technique for mobile robot. By increasing the number of mobile robots (i.e. multiple mobile robot navigation) the criticality of the problem is increased by many folds.

Beaufrere et al.[1,2] have discussed about navigation planning through an unknown obstacle field for mobile robot. They have used a two dimensional array to rapidly model the local free environment for the mobile robot navigation. The algorithm composed of three modules whose function were to avoid obstacles, to reach the target point and to manage direction changing of the mobile robot during path planning. For their approach they have used a method based on fuzzy reasoning and have also tested the approach by simulation. By taking Gaussian function as an activation function, a fuzzy-Gaussian neural network

(FGNN) controller for mobile robot has bin described by Watanabe et al.[3]. The effectiveness of their proposed method has been illustrated by performing the simulation of a circular or square trajectory tracking control. The paper by Racz et al.[4] has presented a neural network based approach to mobile robot localisation in front of a certain local object.
Ishikawa [5] in his paper has described about a navigation method using fuzzy control. His purpose is to construct an expert knowledge for efficient and better piloting of autonomous mobile robot. He has used fuzzy control to select suitable rules i.e. tracing a path/ avoiding obstacles according to a situation, which was derived from sensor information by using fuzzy control. He has established his theory by means of simulation.
Martinez et al. [6] have considered a problem which consists of achieving sensor based motion control of mobile robot among obstacles in structured and/or unstructured environments with collision-free motion as the priority. For this they have taken fuzzy logic based intelligent control strategy, to computationally implement the approximate reasoning necessary for handling the uncertainty inherent in the collision avoidance problem. Sensor-based navigation method, which utilised fuzzy logic and reinforcement learning for navigation of mobile robot in uncertain environment, has been proposed by Boem et al. [7]. Their proposed navigator has consisted of avoidance behaviour and goal-seeking behaviour. They have designed the two behaviours independently at the design stage and then combined together by a behaviour selector at the running stage.
Kam et al.[8] have discussed about the fuzzy techniques of using sensors in robot navigation. They have also discussed about the problem in machine intelligence, including Kalmann filtering, rule-based techniques and behavior based algorithms. Wang [9] has used fuzzy logic for navigation of mobile robot. Tschicholdgurman [10] has described about fuzzy rule-net for the mobile robot navigation. Using the rule-net he has also shown the simulation result for mobile robot. Benreguieg et al.[11] have discussed about navigation of mobile robot using fuzzy logic.
Kodaira et al.[12] have described an intelligent travel control algorithm for mobile robot vehicle using neural networks. They have proposed a method that realises path planning and generation of motion command simultaneously. They have confirmed the validity of the proposed travel by computer simulation. Aoshima et al.[13] have described a simplified dynamic model of a small tunneling robot. They have constructed a dynamic model for directional correction and determined its parameter by least square method. They have used a neural network to automatically obtain four feedback gains for the directional control of both pitching and yawing. Tani et al. [14,15,16,17] have presented a novel scheme for sensory-based navigation of a mobile robot. They have shown that their scheme constructs a correct mapping from sensory inputs sequences to the manoeuvring outputs through neural adaptation, such that a hypothetical vector field that achieves the goal can be generated. Their simulation results has shown that robot can learn task of homing and sequential routing successfully in the work space of a certain geometrical complexity.
Neural network approach for navigation of indoors mobile robot has been discussed by Dubrawski [18]. His algorithm allows for an efficient search a decision space and also for a concurrent validation of the learning algorithm performance on a given data. Fiero et al.[19,20] have discussed about the navigation of mobile robot using neural network. Burgess et al.[21] have used neural network technique of navigation of miniature mobile robot. By using the sensory data and the algorithm, the robot is able to find out current heading direction. Masek et al.[22] have discussed about the mobile robot navigation using sonar data. There robot

navigation task is performed by simulating a simple back-propagation neural network. Chang et al.[23] in their paper, have presented an environment predictor that provides an estimate of future environment configuration by fusing multi-sensor data in real time. They have implemented the predictor by an artificial neural network (ANN) using a relative-error-back-propagation (REBP). Their REBP algorithm enables the artificial neural network to provide output data with a minimum relative error. They have verified their result by computer simulation and navigation experiment.

In the above literature survey, it is seen that many researcher have focused their idea in finding out navigational technique for mobile robot. Still a systematic approach is needed in finding out a proper navigation technique of multiple mobile robots navigating in a highly cluttered unknown environment.

Keeping the above objectives in mind navigation technique has been developed systematically for navigation of multiple mobile robots in a highly cluttered unknown environment. A neuro-fuzzy controller has been developed for avoidance of the obstacle and for target seeking behaviour. The inputs for the neural network robots left obstacle distance, front obstacle distance, right obstacle distance and target angle(i.e. angle made by the robot with respect to. the target). The output from the neural network is initial-steering-angle. The output from the neural network along with the left-obstacle, front-obstacle and right obstacle distance is input to the fuzzy controller. The outputs from the fuzzy controller are the crisp values of left-wheel-velocity and right-wheel-velocity of the robot. From the left-wheel-velocity and right-wheel-velocity the final-steering-angle of a robot is calculated. Inter robot collision avoidance are achieved by using the Petri-net model, by which robots are prioritised. Six mobile robots are also built up in the laboratory for navigation purpose. Windows based software has been developed where the above technique has been implemented. Results achieved by the use of above technique for the navigation of multiple mobile robot shows the authenticity of the proposed technique. Navigation of multiple mobile robots in a highly cluttered unknown environment has got many application such as, automation in industry, working in hazardous conditions (where human being can not reach), space mission, or any other mass activity where there is a need of multiple mobile robots.

2. Analysis of navigation method

Navigation of multiple mobile robots in a highly cluttered environment, using neuro-fuzzy controller, has been analysed systematically in the following section. For the neuro-fuzzy controller the inputs to the neural network are left-obstacle distance, front-obstacle distance, right-obstacle distance and target angle (angle of robot with respect to target). The output from the neural network is initial-steering-angle. Again the output from the neural network along with the left-obstacle distance, front-obstacle distance and right-obstacle distance are the inputs to the fuzzy controller. The outputs from the fuzzy controller are left-wheel velocity and right-wheel velocity, which decides the final-steering-angle (Figure 1).

2.1 Analysis of neural network used in neuro-fuzzy controller

The neural network used is a four-layer perceptron. This number of layers has been found empirically to facilitate training. The input layer has four neurons, three for receiving the values of the distances from obstacles in front and to the left and right of the robot and one for the target bearing. If no target is detected, the input to the fourth neuron is set to 0. The

output layer has a single neuron, which produces the steering angle to control the direction of movement of the robot. The first hidden layer has 10 neurons and the second hidden layer has 3 neurons. These numbers of hidden layer have also been found empirically. Figure 1 depicts the neural network with its input and output signals.

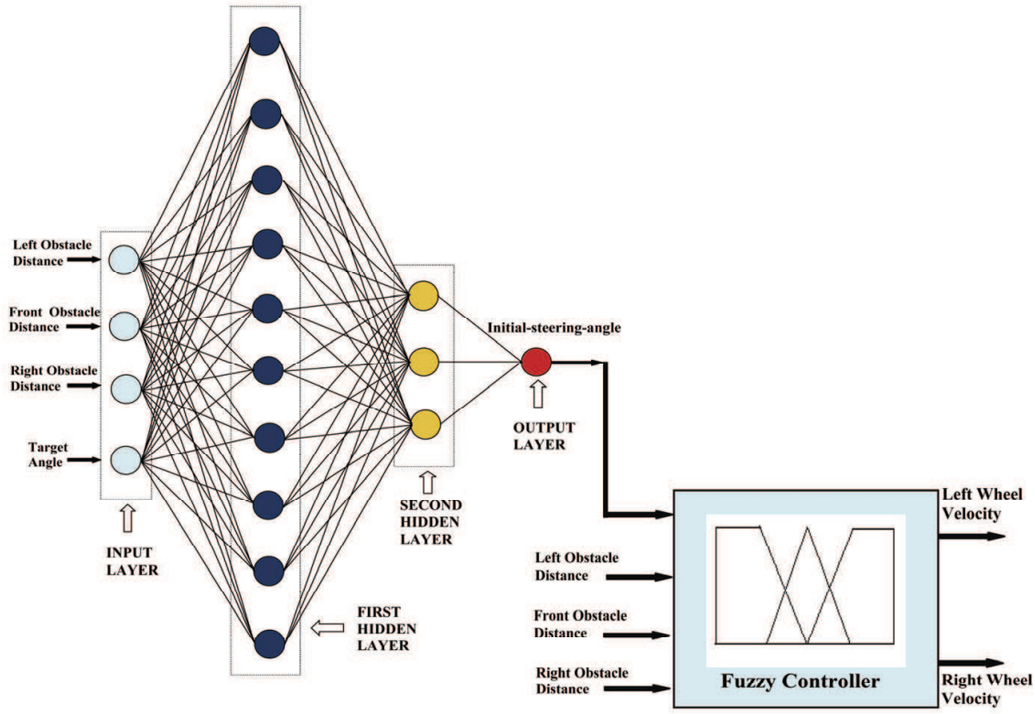


Figure 1. Neuro-fuzzy controller for mobile robots navigation

The neural network is trained to navigate by presenting it with patterns representing typical scenarios, some of which are depicted in Figure 2. For example, Figure 2a shows a robot advancing towards an obstacle, another obstacle being on its right hand side. There are no obstacles to the left of the robot and no target within sight. The neural network is trained to output a command to the robot to steer towards its left side.

During training and during normal operation, input patterns fed to the neural network comprise the following components.

$$y_1^{[l]} = \text{Left obstacle distance from the robot} \tag{1a}$$

$$y_2^{[l]} = \text{Front obstacle distance from the robot} \tag{1b}$$

$$y_3^{[l]} = \text{Right obstacle distance from the robot} \tag{1c}$$

$$y_4^{[l]} = \text{Target bearing} \tag{1d}$$

These input values are distributed to the hidden neurons which generate outputs given by:

$$y_j^{[lay]} = f\left(V_j^{[lay]}\right) \tag{2}$$

where

$$V_j^{[lay]} = \sum_i W_{ji}^{[lay]} . y_i^{[lay-1]} \quad (3)$$

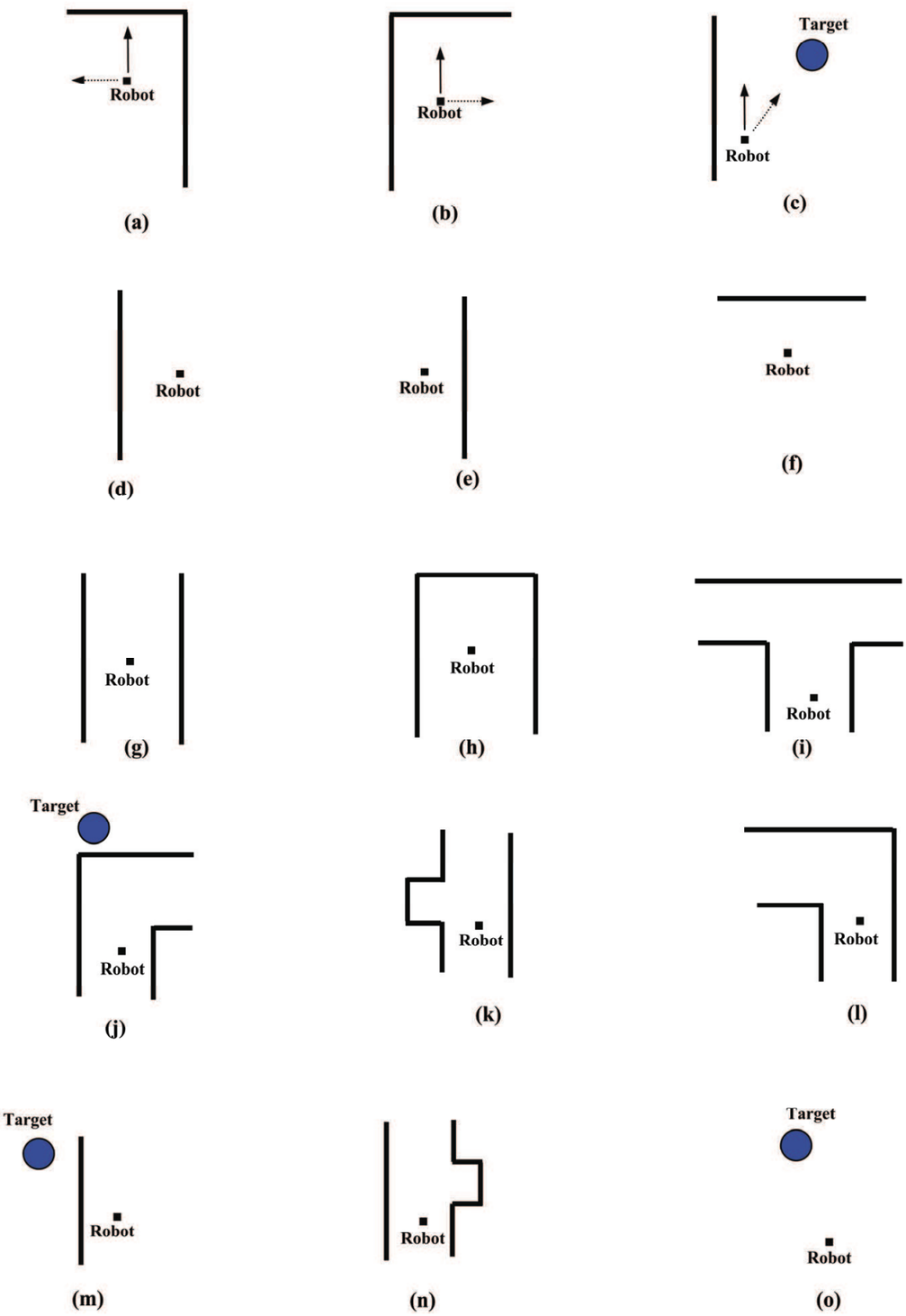


Figure 2. Example training patterns

lay = 2, 3
j = label for j^{th} neuron in hidden layer 'lay'
i = label for i^{th} neuron in hidden layer 'lay-1'
$W_{ji}^{[lay]}$ = weight of the connection from neuron i in layer 'lay-1' to neuron j in layer 'lay'
f(.) = an activation function chosen as the sigmoid function (Figure 3):

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4}$$

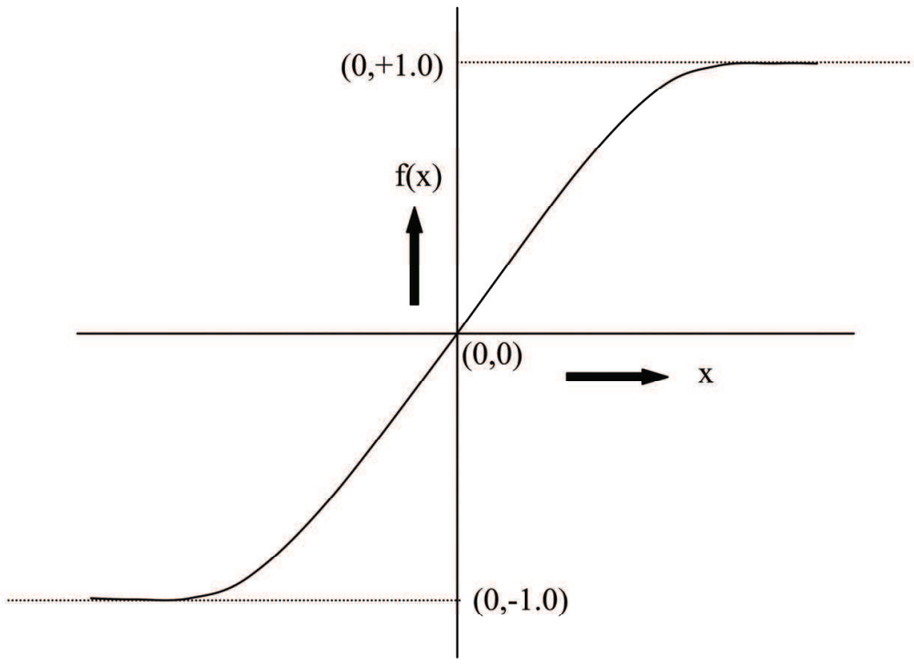


Figure 3. Hyperbolic tangent function

During training, the network output θ_{actual} may differ from the desired output $\theta_{desired}$ as specified in the training pattern presented to the network. A measure of the performance of the network is the instantaneous sum squared difference between $\theta_{desired}$ and θ_{actual} for the presented training patterns:

$$Err = \frac{1}{2} \sum_{\substack{\text{all training} \\ \text{patterns}}} (\theta_{desired} - \theta_{actual})^2 \tag{5}$$

As mentioned previously, the error back propagation method is employed to train the network [7]. This method requires the computations of local error gradients in order to determine appropriate weight corrections to reduce Err. For the output layer, the error gradient $\delta^{[4]}$ is:

$$\delta^{[4]} = f'\left(V_1^{[4]}\right)\left(\theta_{desired} - \theta_{actual}\right) \tag{6}$$

The local gradient for neurons in hidden layer [lay] is given by:

$$\delta_j^{[lay]} = f'\left(V_j^{[lay]}\right)\left(\sum_k \delta_k^{[lay+1]} W_{kj}^{[lay+1]}\right) \tag{7}$$

The synaptic weights are updated according to the following expressions:

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t+1) \tag{8}$$

and

$$\Delta W_{ji}(t+1) = \alpha \Delta W_{ji}(t) + \eta \delta_j^{[lay]} y_i^{[lay-1]} \tag{9}$$

where
α = momentum coefficient (chosen empirically as 0.2 in this work)
η = learning rate (chosen empirically as 0.35 in this work)
t = iteration number, each iteration consisting of the presentation of a training pattern and correction of the weights.
The final output from the neural network is:

$$\theta_{actual} = f\left(V_1^{[4]}\right) \tag{10}$$

where

$$V_1^4 = \sum_i W_{1i}^{[4]} y_i^3 \tag{11}$$

2.2 Analysis of fuzzy logic used in neuro-fuzzy controller

The fuzzy logic used in the neuro-fuzzy controller has been discussed in the following section.
Fuzzy logic was 1st introduced in 1965, by Lotif Zadeh [24]. A fuzzy set A on the universe X is a set defined by a membership function μ_A representing a mapping

$$\mu_A : X \rightarrow \{0,1\}. \tag{12}$$

Where the value of $\mu_A(x)$ for the fuzzy set A is called the membership value or the grade of membership of $x \in X$. The membership value represents the degree of x belonging to the fuzzy set A [25]. Fuzzy-fied inputs are inferred to a fuzzy rule base. This rule base is used to characterise the relationship between fuzzy inputs and fuzzy outputs. Let us consider an example in which a simple fuzzy control rule relating the input n to the output p may be expressed in the condition-action form as follows. If x is A then z is C. Where A and C is fuzzy values defined in the universe x and z, respectively. The inference mechanism provides a set of control action according to fuzzy-fied inputs. As the outputs are in fuzzified sense, a defuzzification method is required to transform fuzzy outputs into a crisp

output value, which can be applied in real sense. For this a well known defuzzification method i.e. centriod method i.e.:

$$z_0 = \frac{\int \mu_c(z)\, z\, dz}{\int \mu_c(z)\ dz} \tag{13}$$

Where $\int$ means ordinary integral . z_0 is a crisp output value of the fuzzy controller.

2.2.1. Inputs and outputs from the fuzzy controller

The inputs and outputs from the fuzzy controller are analysed in the following section.
The inputs to the fuzzy controller are left_obs (left obstacle distance), right_obs (right obstacle distance) and front_obs(front obstacle distance) and initial-steering-angle (out put from the neural network controller). Terms such as near, medium and far are used for left_obs, right_obs and front_obs (Figure 4). Terms such as pos(Positive), zero and neg(Negative) are defined for initial-steering-angle(Figure 4). The out-put from the the fuzzy controller are left_velo and right_velo. Terms such as fast, medium and slow, are defined for left_velo(left velocity) and right_velo(right velocity). The member ship functions described above are shown in Figure 4. All these membership function are triangular or trapezoidal which can be determined by three inputs, the parameters are listed in the Table-1.

Variables	Near (Meter)	Medium (Meter)	Far (Meter)
Left Obstacle (LD) and Right Obstacle (RD)	0.0	0.8	2.0
	0.8	2.0	3.2
	2.0	3.2	4.0

(a) Parameters for Left and Right Obstacle

Variables	Near (Meter)	Medium (Meter)	Far (Meter)
Front Obstacle (FD)	0.0	0.6	2.0
	0.6	2.0	3.4
	2.0	3.4	4.0

(b) Parameters for Front Obstacle

Variables	Negative (Degree)	Zero (Degree)	Positive (Degree)
Heading Angle (HA)	-180.0	-20.0	0.0
	-20.0	0.0	20.0
	0.0	20.0	180.0

(c) Parameters for Heading Angle

Variables	Slow (Meter/Sec)	Medium (Meter/Sec)	Fast (Meter/Sec)
Left Wheel Velocity (LV) and Right Wheel Velocity (RV)	-0.35	-0.1	0.0
	-0.1	0.0	0.1
	0.0	0.1	0.35

(d) Parameters for Left and Right Velocity

Table 1. Parameters of fuzzy membership functions

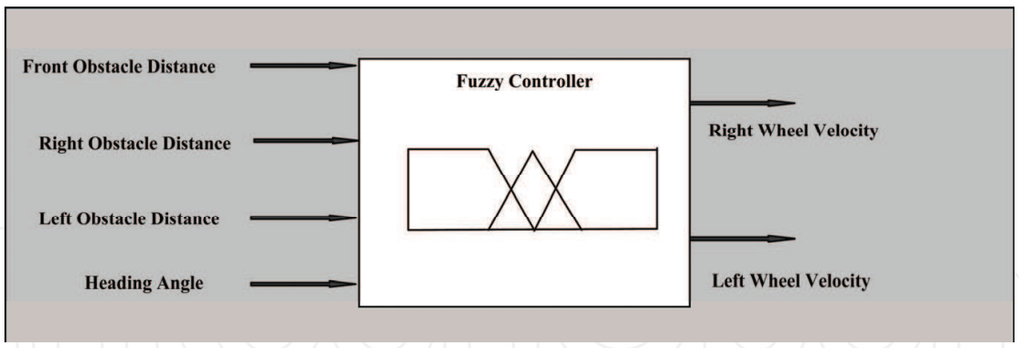


Figure 4a. Fuzzy controller for mobile robot navigation

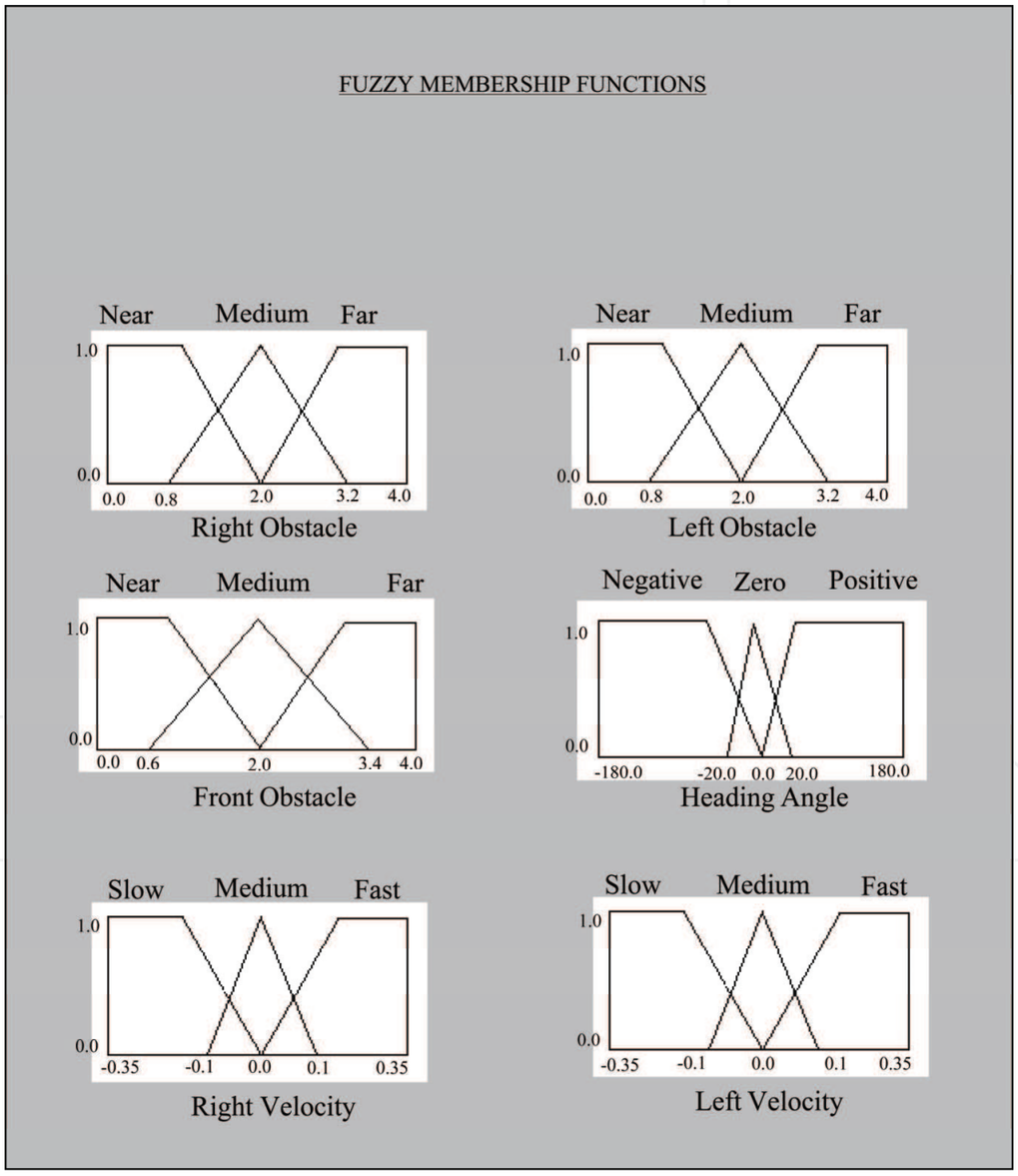


Figure 4b. Fuzzy membership functions

2.2.2 Obstacle avoidance

The fuzzy control rules for avoiding collision with obstacles are listed in Tables 2a and 2b. Rules 1-12 (Table 2a) are dealing with pure obstacle avoidance when the robot turns away from an obstacle as soon as possible. For example, if the obstacle to the left of the robot is far, the obstacles to the right and in front of the robot are near and the heading angle is negative, then the robot should immediately turn to the left. To achieve this, the left wheel velocity should be small and the right wheel velocity should be large (see Rule 3).

As with the fuzzy controller defined in Rules 13-18 (Table 2b) handle obstacle avoidance and wall following simultaneously. For instance, Rule 13 caters for the case where the left and front obstacles are far from the robot and the right obstacle is near. In this situation, the robot continues to move ahead, keeping the right obstacle (a wall) to its right hand side.

FuzzyRuleNo.	Action	Left_obs	Front_obs	Right_obs	Head_ang	Left_velo	Right_velo
1	OA	Far	Near	Med	Neg	Slow	Fast
2	OA	Far	Near	Far	Z	Fast	Slow
3	OA	Far	Near	Near	Neg	Slow	Fast
4	OA	Med	Med	Near	Neg	Slow	Fast
5	OA	Med	Near	Far	Pos	Fast	Slow
6	OA	Med	Near	Med	Neg	Slow	Fast
7	OA	Med	Near	Near	Neg	Slow	Fast
8	OA	Near	Med	Med	Z	Fast	Slow
9	OA	Near	Med	Near	Z	Fast	Slow
10	OA	Near	Near	Far	Pos	Fast	Slow
11	OA	Near	Near	Med	Pos	Fast	Slow
12	OA	Near	Near	Near	Pos	Fast	Slow

Table 2a. Obstacle avoidance

FuzzyRuleNo.	Action	Left_obs	Front_obs	Right_obs	Head_ang	Left_velo	Right_velo
13	OA & WF	Far	Far	Near	Z	Med	Med
14	OA & WF	Far	Med	Near	Z	Med	Fast
15	OA & WF	Med	Far	Near	Z	Med	Med
16	OA & WF	Near	Far	Med	Z	Med	Med
17	OA & WF	Near	Far	Near	Z	Slow	Slow
18	OA & WF	Near	Med	Far	Z	Fast	Med

Table 2b. Obstacle avoidance and wall following

FuzzyRuleNo.	Action	Left_obs	Front_obs	Right_obs	Head_ang	Left_velo	Right_velo
19	TS	Far	Far	Far	Pos	Fast	Slow
20	TS	Far	Far	Far	Neg	Slow	Fast
21	TS	Far	Far	Near	Neg	Slow	Fast
22	TS	Near	Far	Far	Pos	Fast	Slow
23	TS	Far	Near	Near	Neg	Slow	Fast
24	TS	Near	Near	Far	Pos	Fast	Slow

Table 2c. Target finding

Table 2. List of fuzzy rules for multiple mobile robot navigation

Note: Left_obs – Left Obstacle, Front_obs – Front Obstacle, Right_obs – Right Obstacle, Head_ang – Heading Angle, Left_velo – Left Velocity, Right_velo – Right Velocity, OA – Obstacle Avoidance, WF – Wall Following, TS – Target Steering, Neg – Negative, Pos – Positive, Z – Zero, Med – Medium

2.2.3 Targets finding

If a mobile robot detects a target, it will directly move towards it unless there is an obstacle obstructing its path. In that case, the robot will move around the obstacle before proceeding towards the target. The target finding rules are listed in Table 2c. An example is Rule 19 which directs the robot to turn right (high left wheel velocity and low right wheel velocity) because that is where the target is located and there are no obstacles in the vicinity.

2.3. Inter robot collision avoidance

At start robots are placed in the cluttered unknown environment, without any prior knowledge of targets, obstacles and other robots in the environment. Each robot has a aim of finding the targets avoiding obstacles and inter robot collision (Task-1) as shown in Figure 5. Once the robots have received a command to start, they will navigate in search of targets by avoiding obstacles with the help of proper steering angle, which will be decided by neuro-fuzzy controller (Task-2).

During the navigation if path of a robot is obstructed by another robot, then conflicting situation between the robots is detected (Task-3). Conflicting robots will negotiate with each other and they will be prioritised. The lower priority robot will be treated as static obstacle and higher priority robot as moving robot (Task-4). As soon as the conflicting situation is solved between the above two robots, the free robots will co-ordinate with the other conflicting robots (Task-5) to see whether there is any other conflicting situation with them.

If a robot during navigation meet with other two conflicting robots (Task-6), than the priority of the last robot will be lowest and will be treated as a static obstacle till the conflicting situation is being resolved between the first two robots. As soon as conflicting robots resolve all conflicting situations, they will again start Task-2.

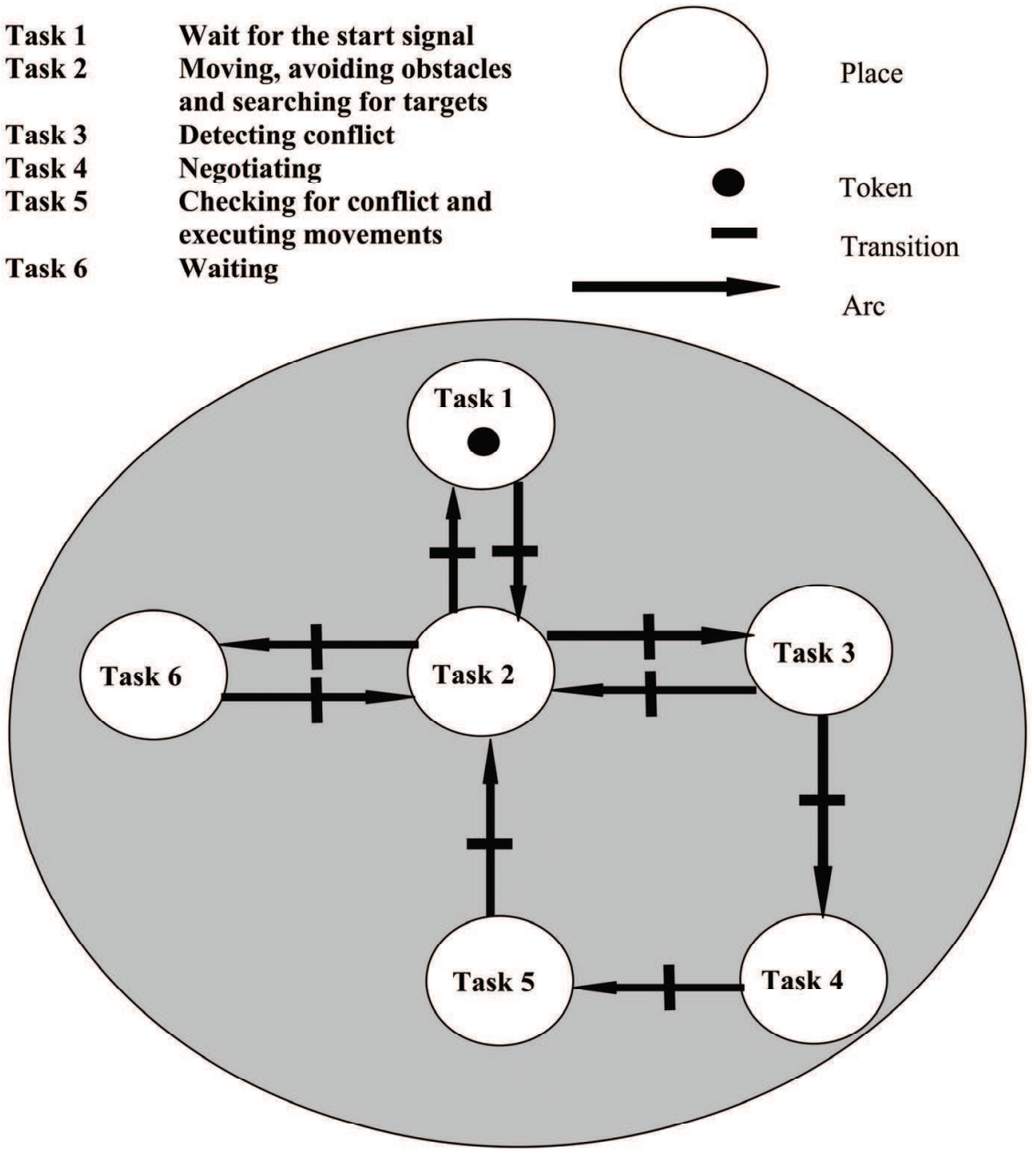


Figure 5. Petri Net Model for avoiding inter-robot collision

3. Demonstrations

3.1 Simulation Results

- Obstacle Avoidance and Target Seeking

This exercise shows that the mobile robots can navigate without hitting obstacles and can find targets in a cluttered environment. Nine robots are involved together with several obstacles including two U-shaped containers housing one target each. Figure 6a depicts the initial state when the nine robots are arranged into two groups positioned at two locations in an enclosure. From Figure 6b, it can be seen that the robots are able to locate the targets while successfully avoiding collision against the obstacles.

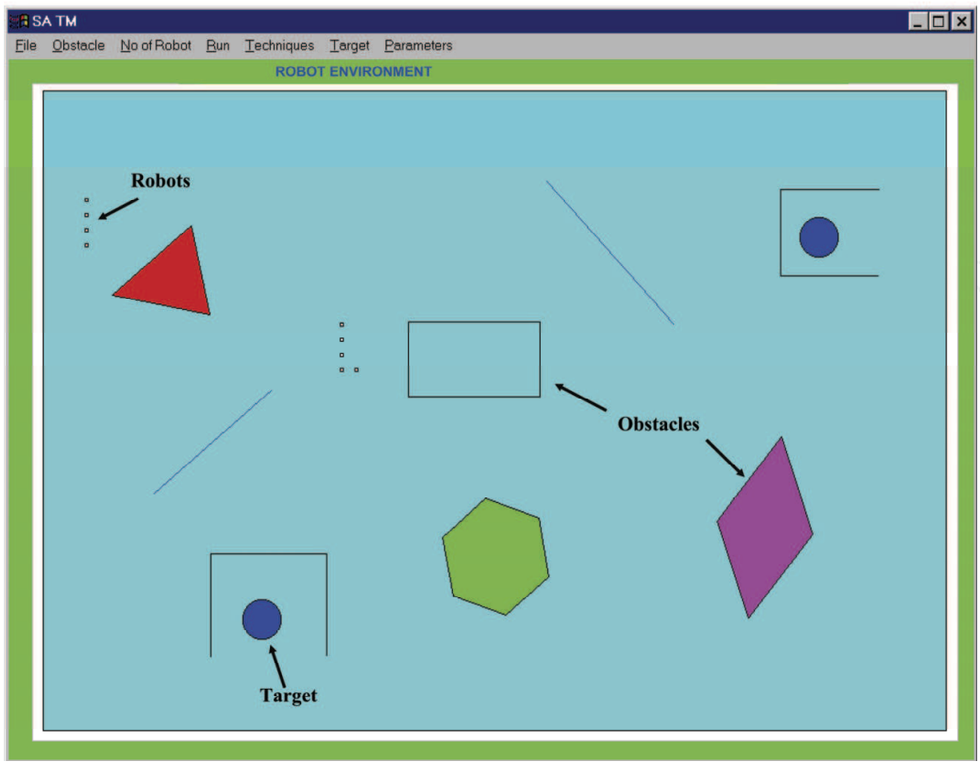


Figure 6a. Obstacle avoidance and target seeking (initial state)

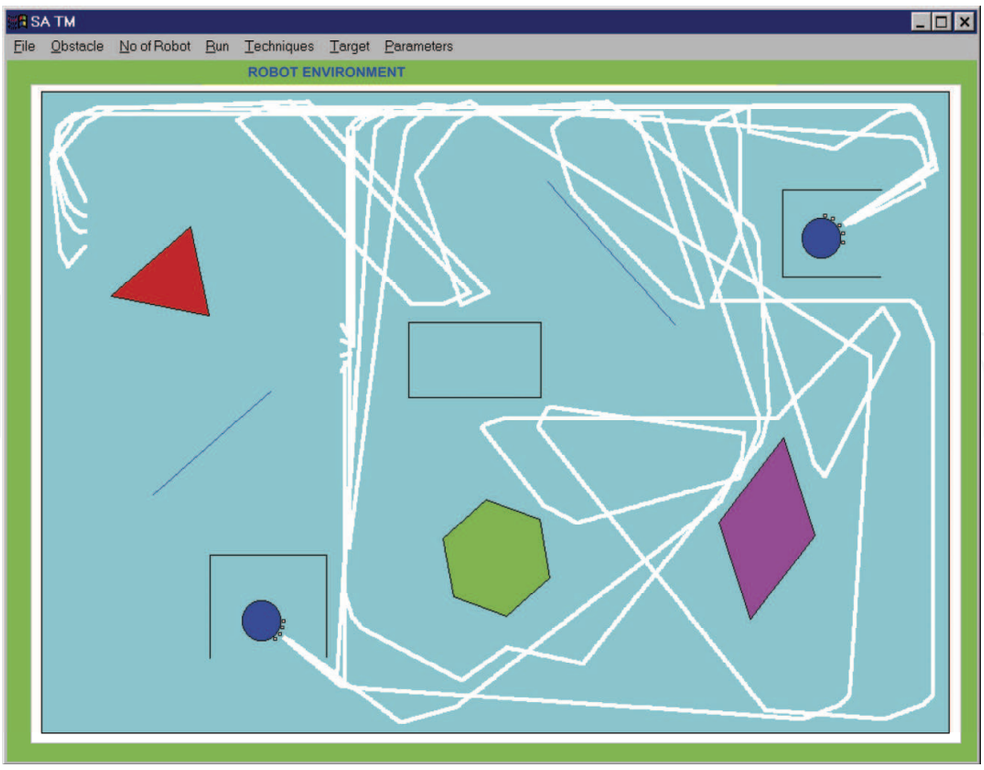


Figure 6b. Obstacle avoidance and target seeking (final state)

- Escape from Dead Ends

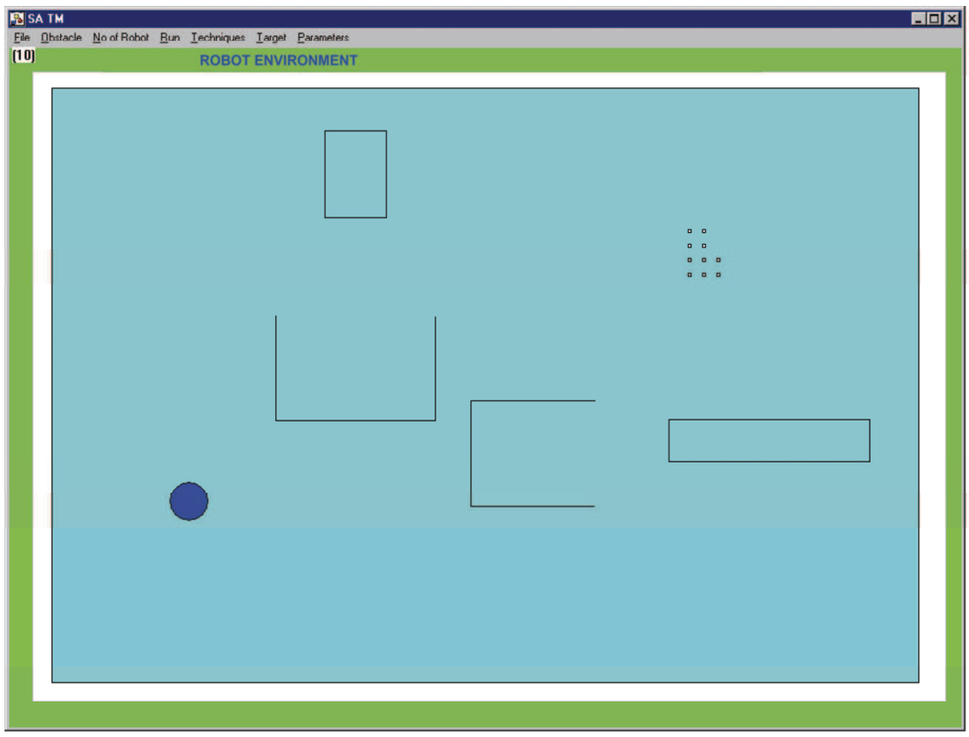


Figure7a. Escape from dead ends (initial state)

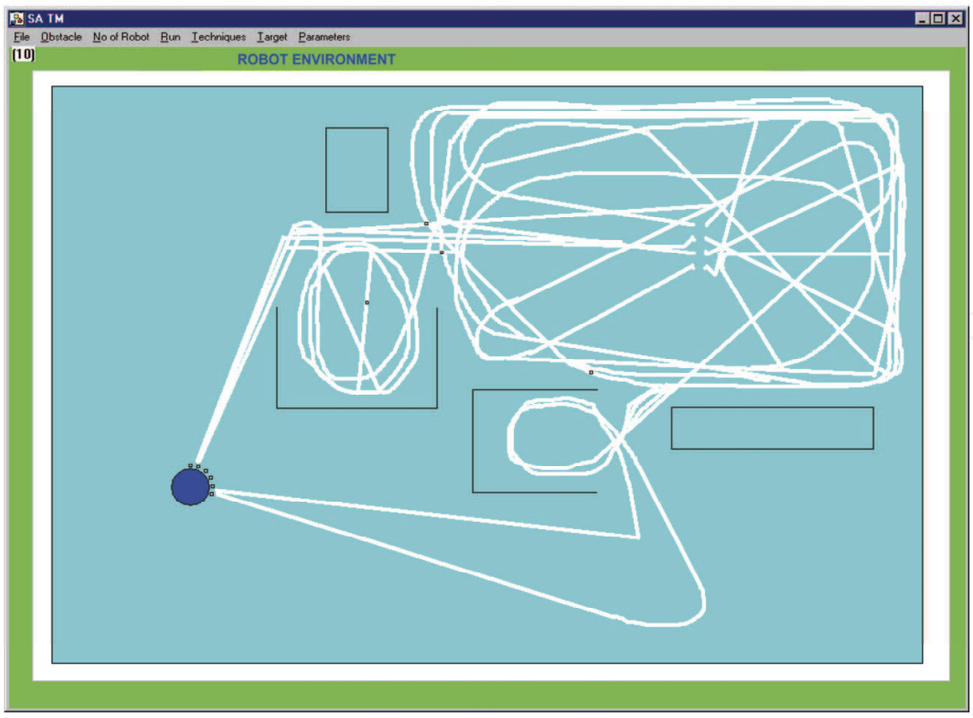


Figure 7b. Escape from dead ends (intermediate state)

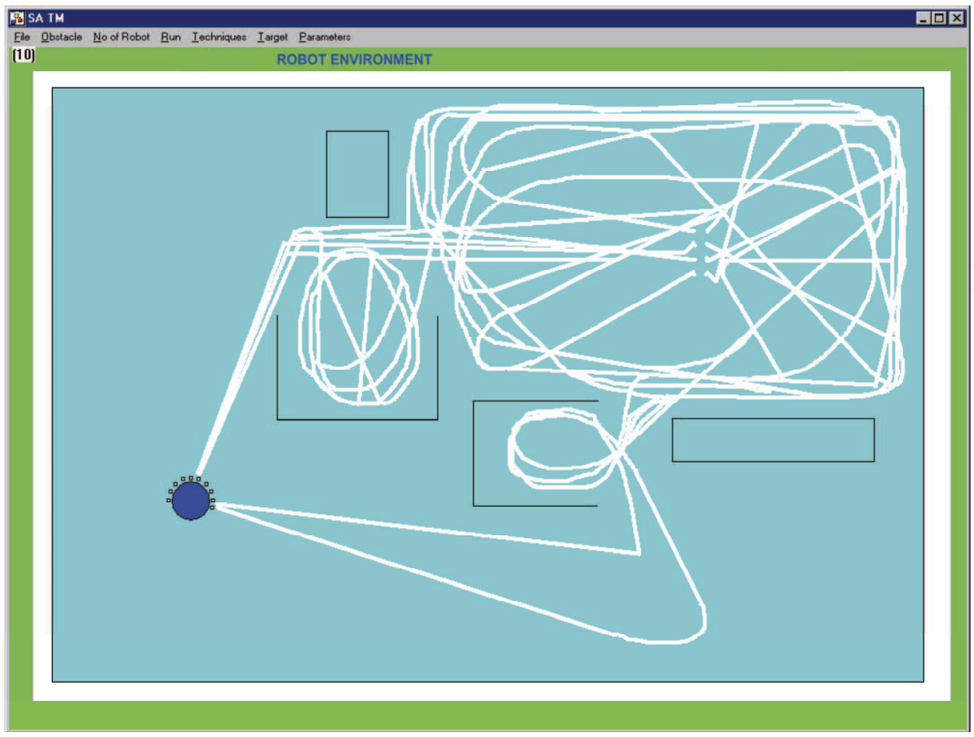


Figure 7c. Escape from dead ends (final state)

This exercise is similar to that discussed in the previous chapter (see Figure 7a) where U-shaped containers are used to simulate dead ends from which trapped robots should escape. There are 10 robots in an enclosure comprising two U-shaped containers and other obstacles. Figure 7b depicts an intermediate state. It can be seen that most of the robots are outside the containers. All of those that have strayed inside have escaped. Figure 7c shows that the other trapped robots have also escaped and all the targets have been located.

- Wall Following

Two robots are involved in this exercise. Figure 8 shows that the robots are able to follow the walls of a corridor and reach the targets successfully.

- Navigation of a Large Number of Robots

One thousand robots are involved in this exercise. Figure 9 is a snap shot of what happens. It can be noted that all the robots stay well away from the obstacles.

- Inter-robot Collision Avoidance

This exercise demonstrates that the robots do not collide with one another even in a cluttered environment. For ease of visualisation, only a small number of robots are employed. Figure 10a and 10b depict the trajectories of the robots for the neuro-fuzzy controller. It can be seen that the robots are able to resolve conflict and avoid one another.

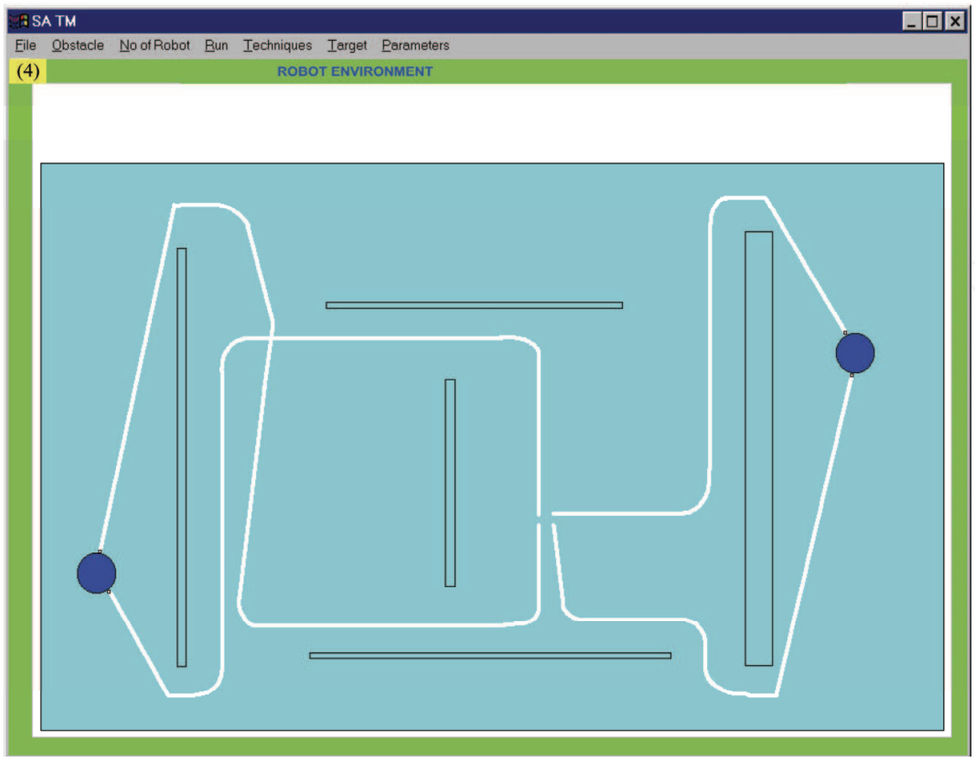


Figure 8. Wall following (final state)

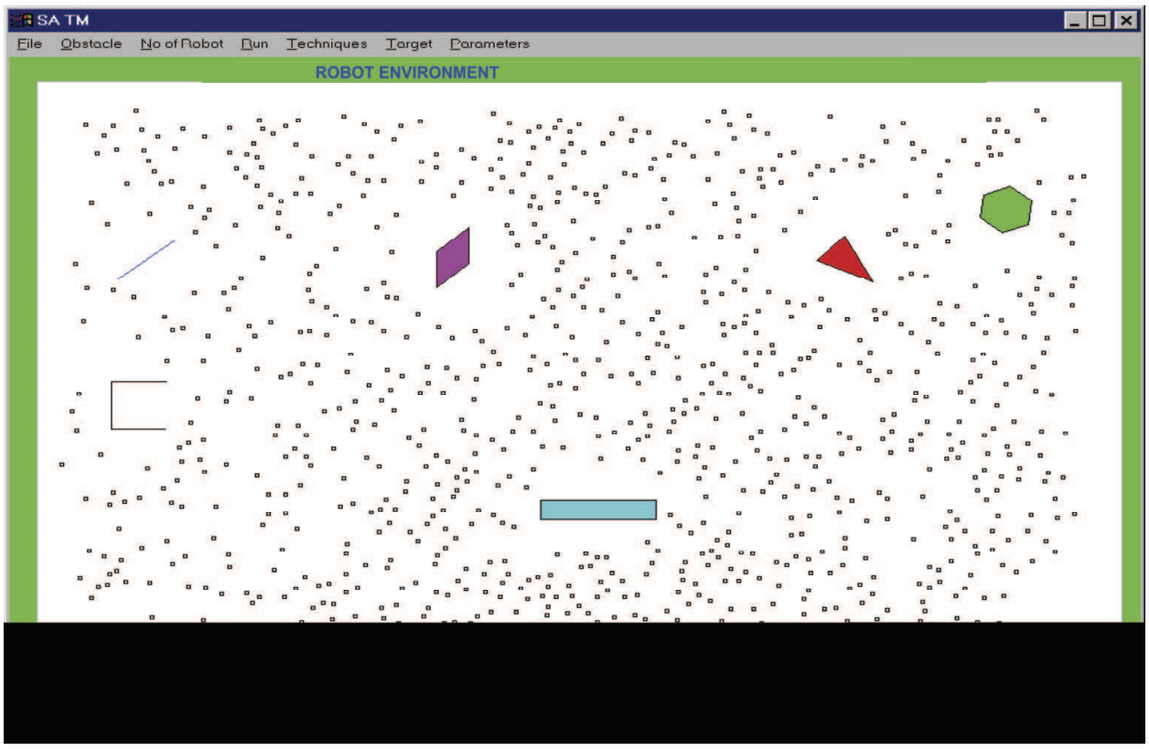


Figure 9. Navigation of a large number of robots

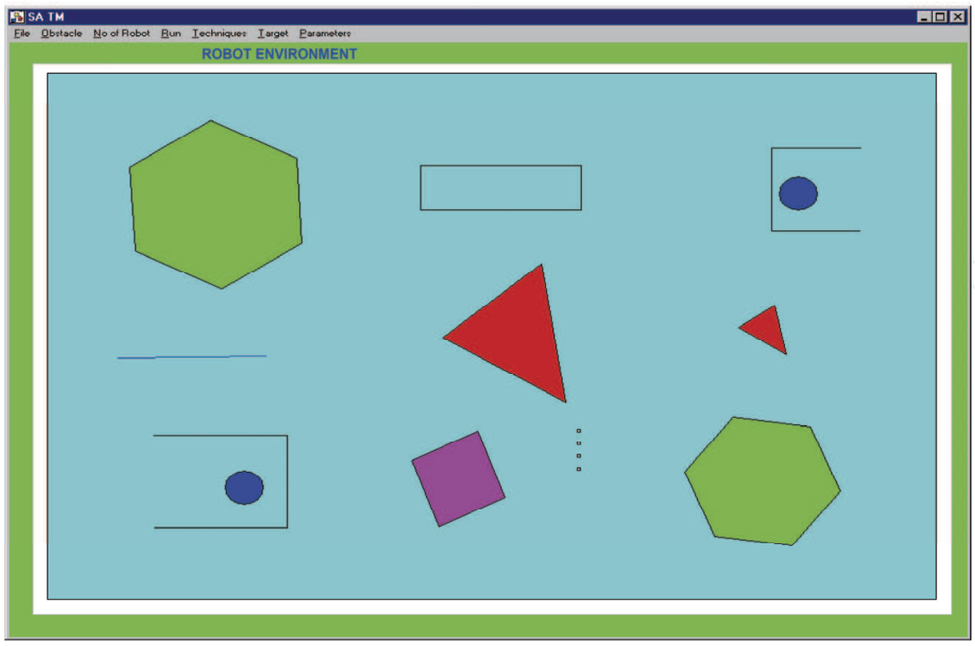


Figure 10a. Collision free movements (initial state)

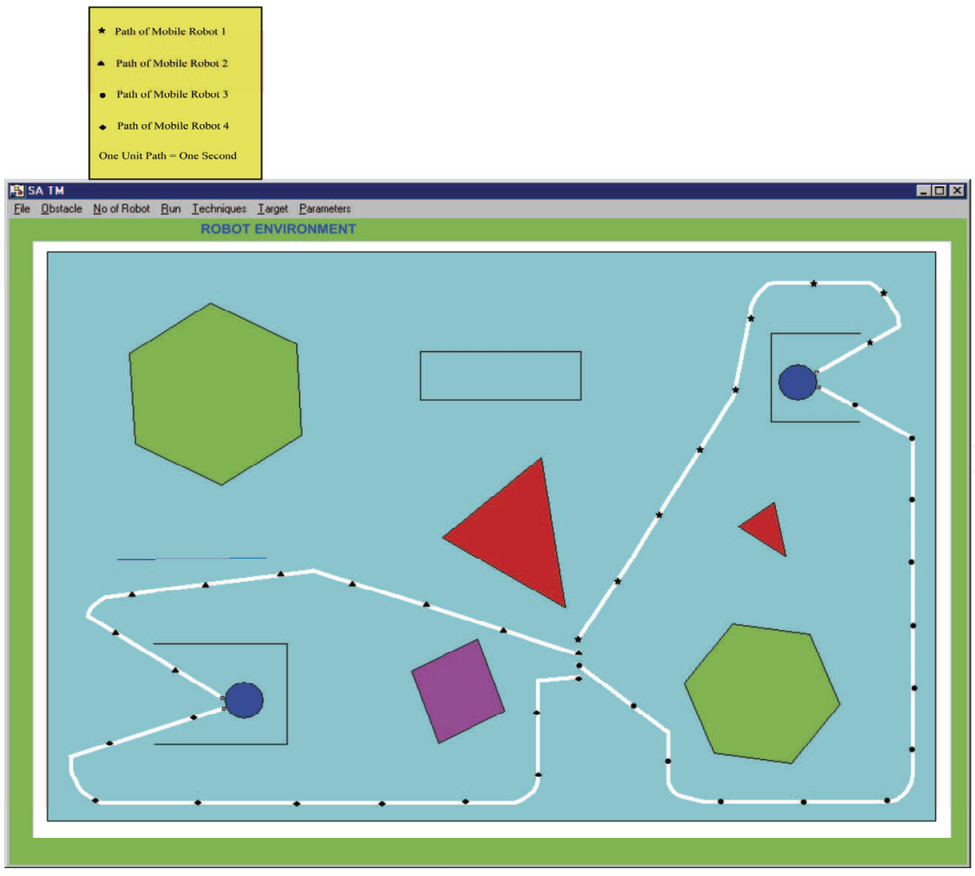


Figure 10b. Collision free movements (final state)

3.2 Experimental Results

Figures 11a, 11b, 12a and 12c show the experimental results obtained. The results for the neural and neuro-fuzzy controllers for a single robot are presented in Figure 11a and 11b. In Figures 12a and 12b, the results for the neural and neuro-fuzzy controllers for four mobile robots are shown respectively. The experimental paths drawn follow closely those traced by the robots during simulation. It can be seen that the robots are able to avoid obstacles and reach the targets. Table 3 shows a comparison between the average time taken by the robots in simulation and the practical tests for obstacle avoidance and target seeking

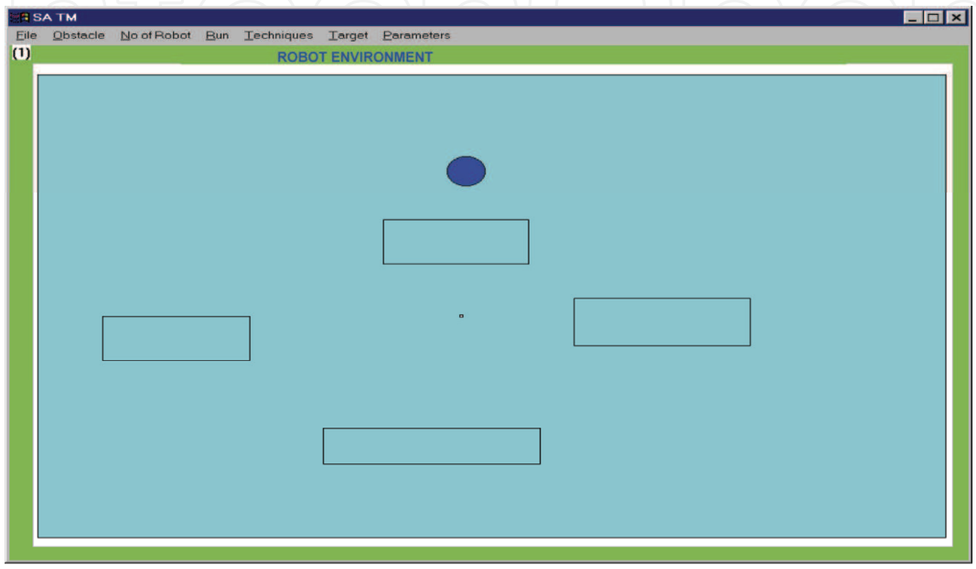


Figure 11a. Work space environment of one mobile robot (initial state)

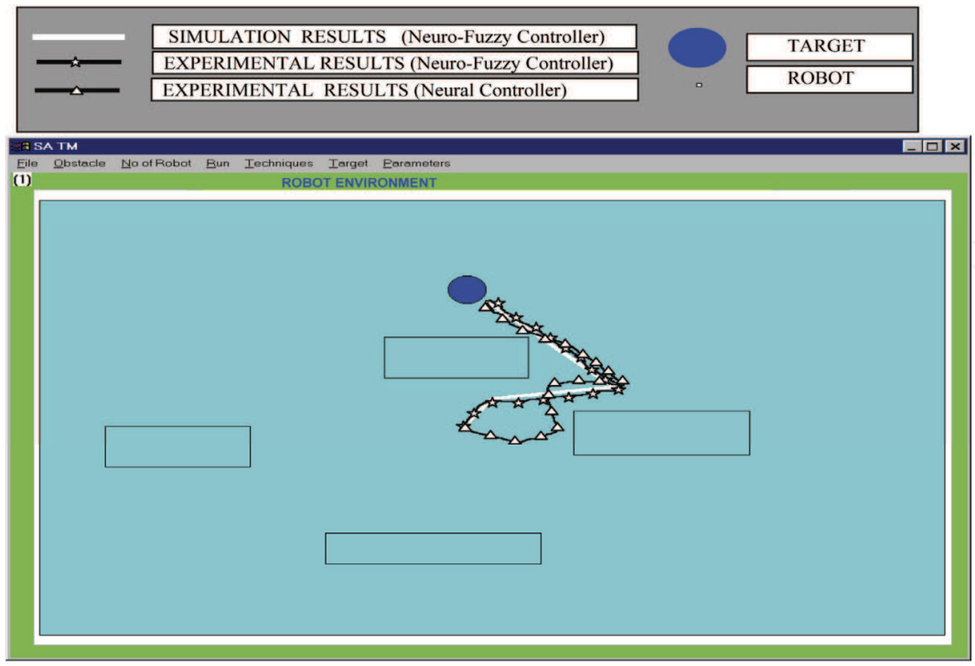


Figure 11b. Experimental results for one robot

Number of robots	Time taken (Seconds) using Neural technique	Time taken (Seconds) using Neuro-Fuzzy technique
4	11.14	10.39
8	23.37	20.25
10	22.46	19.43
16	34.42	30.49
24	54.31	51.05
40	78.12	71.05
56	123.57	111.45
70	259.01	228.54

Table 3. Times taken to reach target using different techniques

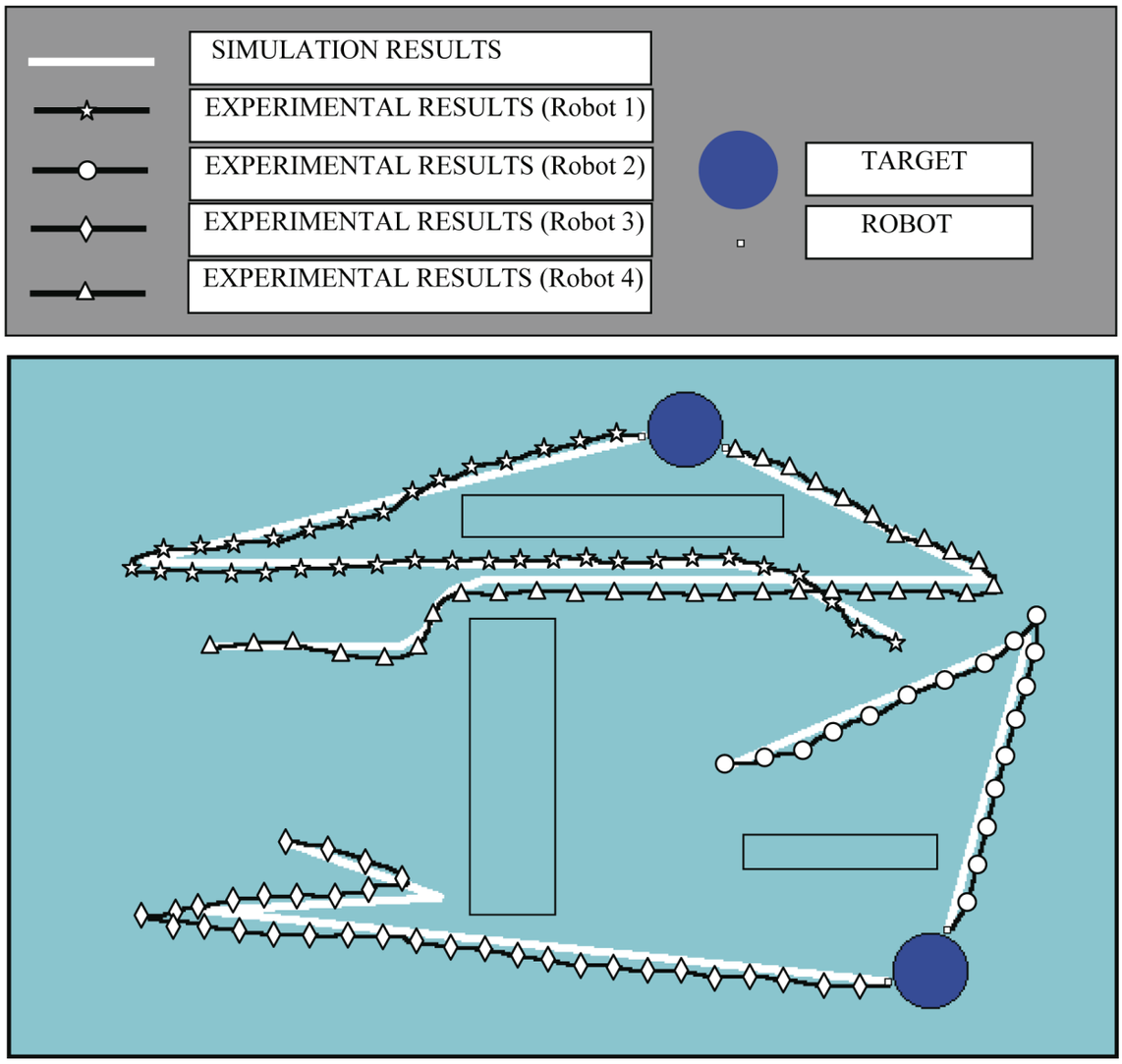


Figure 12a. Experimental results for four robots (neural controller)

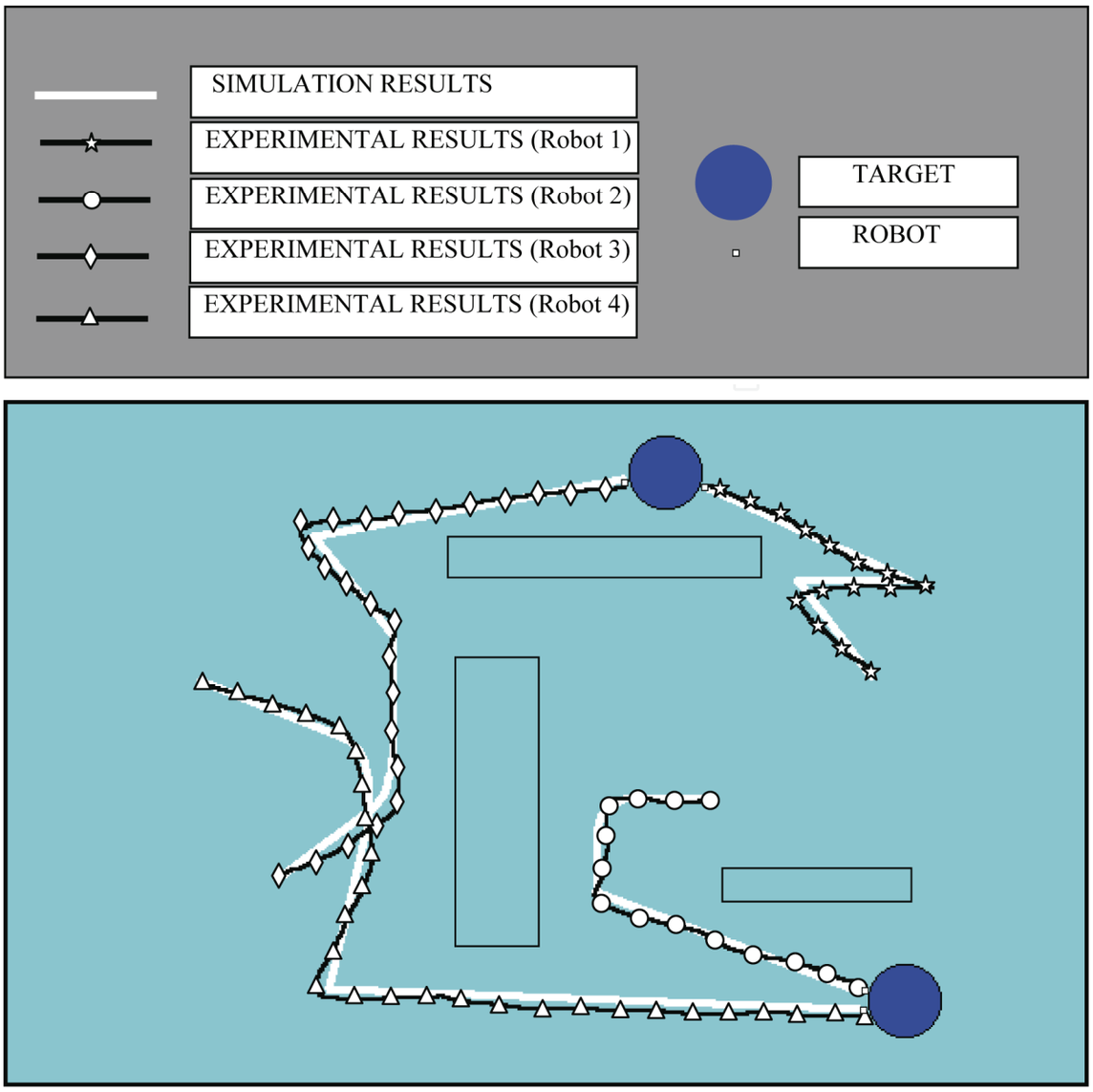


Figure 12b. Experimental results for four robots (neuro-fuzzy controller)

4. Conclusions:

Analysis and discussion on navigation of multiple mobile robots has been carried out in this paper using neuro-fuzzy technique. The conclusions drawn from the above analysis have been depicted below.

Using neuro-fuzzy technique mobile robots are able to navigate in a cluttered unknown environment. Mobile robots are able to escape from U-shaped objects(dead end obstacles) and can get the targets successfully using neuro-fuzzy technique (Figure 7.). When the robots are not able to see any target in one side of wall, they will continue at the edge of the wall assuming that the targets are lacated in other side of the wall(wall following action). The wall following action has been achieved successfully using neuro-fuzzy technique

(Figure 8.). In the present navigation method robots are able to avoid inter robot collision, which can be visualised from (Figures 10). As many as one thousand mobile robots can navigate successfully using neuro-fuzzy controller (Figure 9.).
The present research has got a tremendous application such as automation of industry, space mission, agricultural activity and mass activity where many robots are required at a time. This technique can again revised by fusing some other technique to neuro-fuzzy technique such as adaptive technique.

5. References:

Beaufrere, B. and Zeghloul, S. A mobile robot navigation method for using a fuzzy based method simulation and experimental aspects. *International journal of robotics and automation*. 1995, vol-10, no-3, 106-113.

Beaufrere, B. and Zeghloul, S. A mobile robot navigation method using a fuzzy-logic approach. *Robotica*. 1995, vol-13, no-pt5, 437-448.

Watanabe,K., Jang, J., Nakamura, M., Koga, S. and Fukuda, T., A fuzzy-gaussian neural-network and its application to mobile robot control, *IEEE transactions on contro systems technology*, Vol.4(2)(1996) 193-199.

Racz, J. and Dubrawski, A., Artificial neural-network for mobile robot topological localisation, *Robotics and autonomous systems*, Vol. 16(1)(1995) 73-80.

Ishikawa, S. A method of autonomous mobile robot navigation by using fuzzy control. *Advanced robotics*. 1995, vol-9, no-1, 29-52

Martinez, A; Tunstel, E; Jamshidi, M. Fuzzy-logic based collision-avoidance for a mobile robot. *Robotica*. 1994, vol-12, no-pt6, 521-527.

Boem, H.R. and Cho, H.S. A sensor-based navigation for a mobile robot using fuzzy-logic and reinforcement learning. *IEEE transactions on systems man and cubernetics*. 1995, vol-25, no-3, 464-477.

Kam,M, Zhu, X.X. Kalata, P. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*. 1997, vol-85, no-1, 108-119.

Wang, L.X. Modelling and control of hierarchical systems with fuzzy systems. *Automatica*, 1997, vol-33, no-6, 1041-1053.

Tschicholdgurman, N. The neural-network model rulenet and its application to mobile robot navigation. *Fuzzy sets and systems*. 1997, vol-85, no-2, 287-303.

Benreguieg, M., Hoppenot, P., Maaref, H., Colle, E., and Barret, C. Fuzzy navigation strategy: application to two distinct autonomous mobile robots. *Robotica*, 1997, Vol-15, 609-615.

M. Kodaira, T. Ohtomo, A. Tanaka, M. Iwatsuki and T. Ochuchi, Obstacle avoidance travel control of robot vehicle using neural network, *Systems and computers in Japan*, Vol. 27(12)(1996) 102-112.

S. Aoshima, K. Takeda, K. Hanari, T. Yabuta and M. Shiraishi, Dynamic simplified model and autotuning of feedback gain for directional control using a neural network for a small tunneling robot. *JSME international journal series c-dynamics control robotics design and manufacturing*, Vol.40(2)(1997) 245-252.

J. Tani and N. Fukumura, Learning goal-directed sensory-based navigation of a mobile robot, *Neural networks*, Vol. 7(3)(1994) 553-563.

J. Tani and N. Fukumura, Self-organising internal representation in learning of navigation. A physical experiment by the mobile robot YAMABICO, *Neural networks*, Vol.10(1)(1997) 153-159.

J. Tani and N. Fukumura, Embedding task-based behavior into internal sensory-based attraction dynamics in navigation of a mobile robot. *Proc. of the IEEE internatinal conf. of intelligent robots and system-94*, (1994) 886-893.

J. Tani and N. Fukumura, Embedding a grammatical description in deterministic chaos, An experiment in recent neural learning, *Biolog Cyber.*, Vol.72(1995), 365-370.

A. Dubrawski, Stochastic validation for automated tuning of neural network's hyper-parameters, *Robotics and autonomous systems*. Vol. 21(1)(1997) 83-93.

R. Fierro, F.L. Lewis, Robust practical point stabilization of a nonholonomic mobile robot using neural networks, *Journal of intelligent & robotic systems*, Vol. 20(2-4)(1997) 295-317.

R. Fierro, F.L. Lewis, Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics, *Journal of Robotic Systems*, Vol.14(3)(1997)149-163.

N. Burgess, J.G. Donnett, K.J. Jeffery and J. Okeefe, Robotic and neural simulation of the hippocampus and rat navigation, *Philosophical transactions of the royal society of London series B-Biological sciences*, Vol-352(1360)(1997) 1535-1543.

Masek, V., Kajitani, M., Ming, A. and Viacic, L., Fast mobile robot obstacle detection using simultaneous firing of sonar ring sensors, *International Journal of the Japan Society for Precision Engineering*, Vol.32(3)(1998) 207-212.

C.C. Chang and K.T. Song, Environment prediction for a mobile robot in a dynamic environment, *IEEE transactions on robotics and automation*, Vol.13(6)(1997) 862-872.

Zadeh, L.A. Fuzzy SetsInformation and Control. 8, 1965, 338-353.

Tanaka, K, An introduction to fuzzy logic for practical application, 1991.

6. Appendix-1

The software ROBNAV has been developed in the laboratory under Microsoft's Windows environment.

By using the software users can have:

i. Various options and can create a navigational environment for multiple mobile robots.
ii. Many mobile robots inside the environment (multiple mobile robots).
iii. Different types of obstacles inside the environment (in order to create a cluttered environment for multiple mobile robots navigation).
iv. Many targets in the environment.

By the help of the software, neuro-fuzzy controller has been implemented for navigation of multiple mobile robots and the results obtained are shown in Figures 6 to 12. The overview outlay of the software has been shown in Figure 13.

7. Appendix-2

A prototype view of a mobile robot (out of several similar mobile robots that have been constructed in the laboratory for navigational purposes) is shown in Figure 14. Each mobile robot consists of:

i. Three wheels, of which two are driven by stepper motors and the third by caster wheel.
ii. PC-Mother board.

iii. Ultrasonic Card, six ultrasonic transmitters and receivers, for measuring the obstacle distances around the robot.
iv. Infrared card, six infrared transmitters and eight infrared receivers, used for detecting the targets.
v. Radio Modem Card, for remote transactions of commands with other robots and also with other computers.
vi. Hard-disk, for storing the software required for navigation of the mobile robot.
vii. Two touch sensors, one at the front end and one at the rear end of the robot.
viii. An onboard battery for power supply.

The components discussed above are shown in the Figure 14.

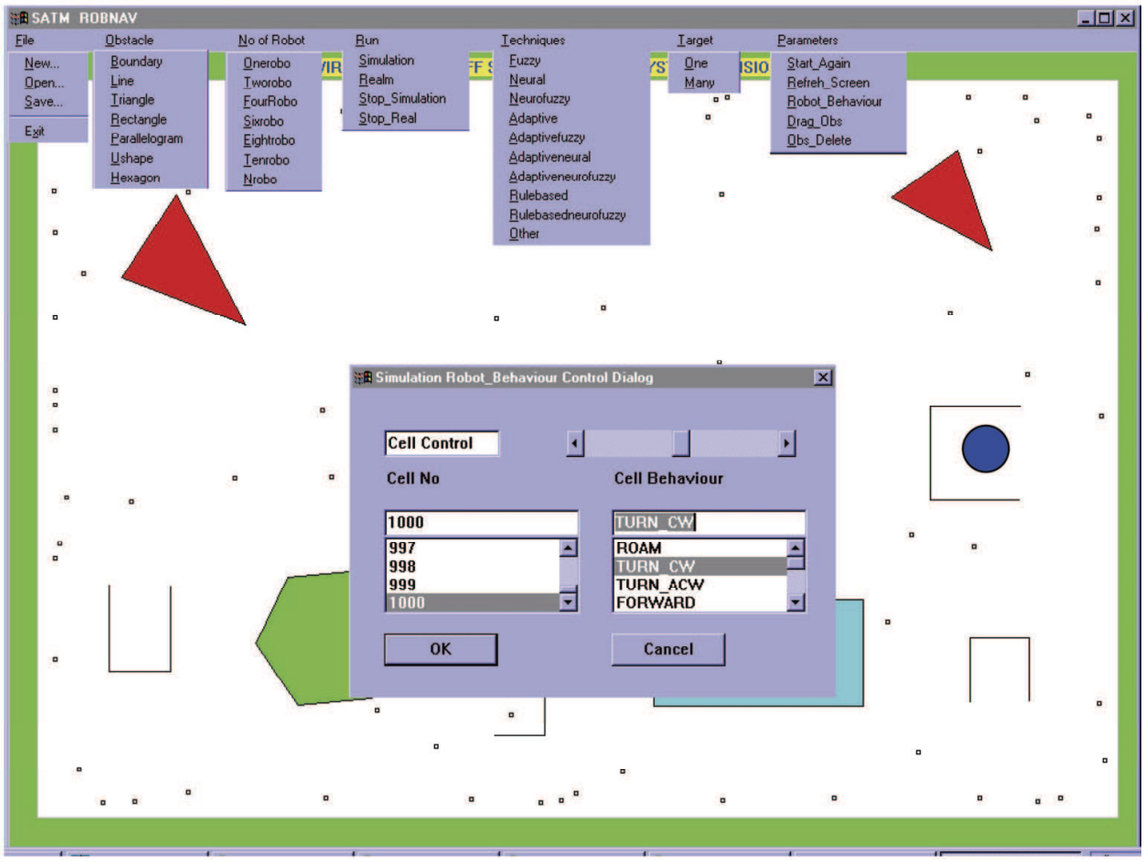
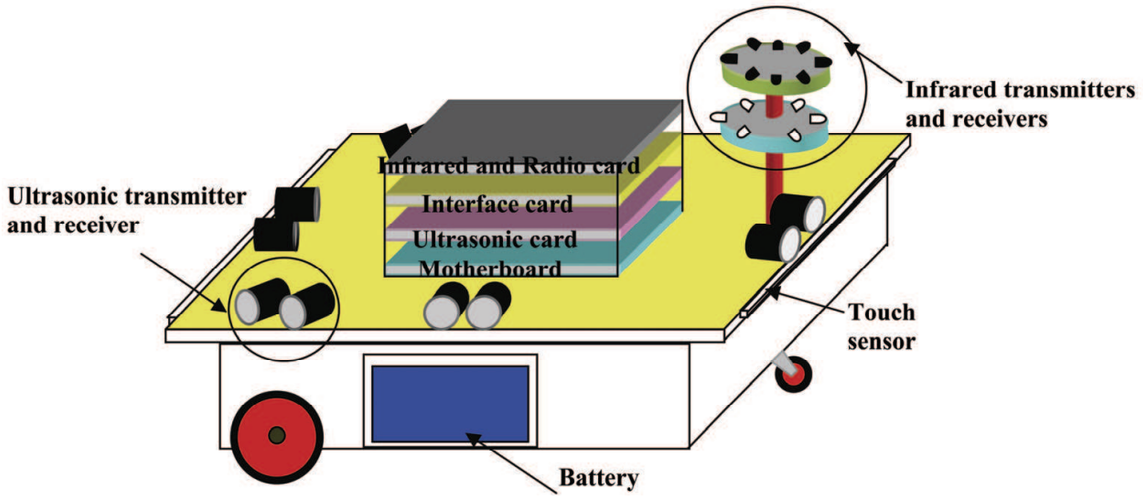
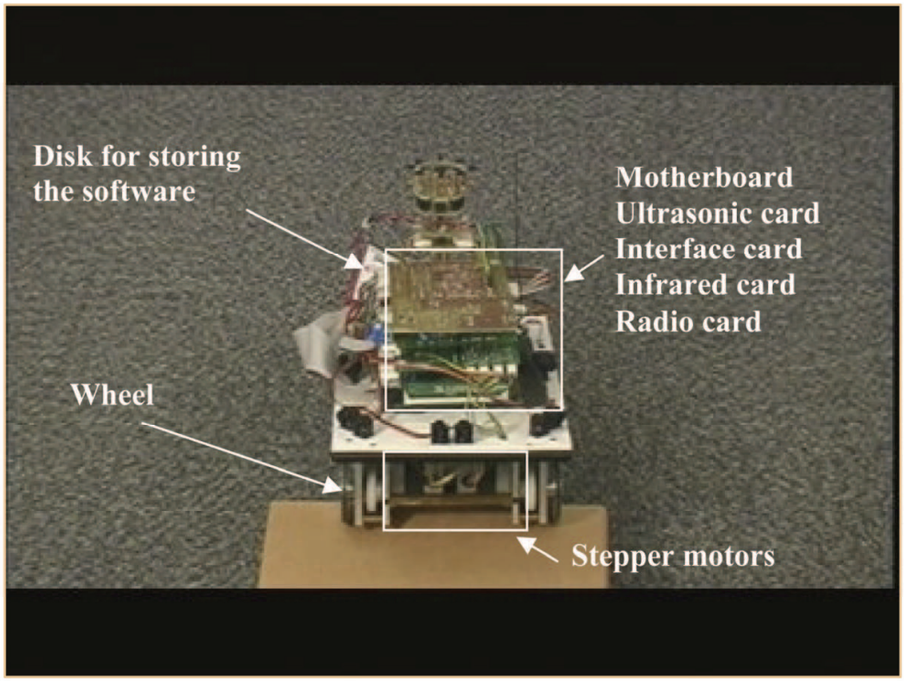


Figure 13. ROBNAV software package



(a) Schematic view of a robot



(b) Actual view of a robot

Figure 14. A mobile robot

Motion Planning

Edited by Xing-Jian Jing

ISBN 978-953-7619-01-5

Hard cover, 598 pages

Publisher InTech

Published online 01, June, 2008

Published in print edition June, 2008

In this book, new results or developments from different research backgrounds and application fields are put together to provide a wide and useful viewpoint on these headed research problems mentioned above, focused on the motion planning problem of mobile ro-bots. These results cover a large range of the problems that are frequently encountered in the motion planning of mobile robots both in theoretical methods and practical applications including obstacle avoidance methods, navigation and localization techniques, environmental modelling or map building methods, and vision signal processing etc. Different methods such as potential fields, reactive behaviours, neural-fuzzy based methods, motion control methods and so on are studied. Through this book and its references, the reader will definitely be able to get a thorough overview on the current research results for this specific topic in robotics. The book is intended for the readers who are interested and active in the field of robotics and especially for those who want to study and develop their own methods in motion/path planning or control for an intelligent robotic system.

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen