Intelligent Neuro-Controller for Navigation of Mobile Robot

Mukesh Kumar Singh
Department of Mechanical Engineering,
Government Engineering College Bilaspur,
(Chhattisgarh)-495009, India, +919861260064
Mukesh3003@yahoo.co.in

Dayal R. Parhi
Department of Mechanical Engineering,
National Institute of Technology Rourkela, (Orissa)-769008, India, +916612464509
dayalparhi@yahoo.com

ABSTRACT

This paper deals with the reactive control of an autonomous robot which move safely in a crowded real world unknown environment and to reach specified target by avoiding static as well as dynamic obstacle. The inputs to the proposed neurocontroller consist of left, right, and front obstacle distance to its locations and target angle between a robot and a specified target being acquired by an array of sensors. A four layer neural networks is used to design and develop the neurocontroller to solve the path and time optimization problem of mobile robots which deals the with cognitive tasks such as learning, adaptation, generalization and optimization. Back propagation method is used to trained the network. This paper analyzes the kinematical modeling of mobile robots as well as the design of control systems for the autonomous motion of the robot. The training of the nets and the control performances analysis have been done in a real experimental setup. The simulation results are compared with experimental results which are satisfactory and shows a very good agreement.

Categories and Subject Descriptors

I.2.9 Robotics: *Autonomous vehicles, Kinematics.* F.1.1 Models of Computation: *Self-modifying machines (e.g., neural networks)*

General Terms: Design.

Keywords: Evolutionary robotics, Artificial neural network, Mobile robot, Behavioral robotics.

1. INTRODUCTION

One of the most important issues in the design and development of intelligent mobile system is the navigation problem. This consists of the ability of a mobile robot to plan and execute collision free motions within its environment. However, this environment may be imprecise, vast, dynamical and either partially or non-structured. Robots must be able to understand the structure of this environment[5, 9, 11, 12, 15]. To reach their targets without collisions, the robots must be endowed with perception, data processing, recognition, learning, reasoning, interpreting, and decision-making and action capacities.

Service robotics today requires synthesizing robust automatic systems able to cope with a complex and dynamic environment [4]. To demonstrate this kind of autonomy Muniz et al. [8] have introduced a neural controller for a mobile robot that learns both forward and inverse odometry of a differential drive robot through an unsupervised learning by doing cycle. They introduce an obstacle avoidance module that is integrated into the neural controller. However, generally, the evolved neural controllers could be fragile in un experienced environments, especially in real worlds, because the evolutionary optimization processes would be executed in idealized simulators. This is known as the gap problem between the simulated and real worlds. To overcome this, Kondo [7] has focused on evolving an on-line learning ability instead of weight parameters in a simulated environment. Based on this, a neuromodulatory neural network model was proposed by them and it utilized as a mobile robot controller. Corradini et al. [2] have used a neural networks approach to the solution of the tracking problem for mobile robots. Recz et al. [14] have presented a neural network based approach to a mobile robot localization in front of a certain local object. Yang et al. [16] have proposed a biologically inspired neural network approach to real-time collision-free motion planning of mobile robots or robot manipulators in a non stationary environment. Braganza et al. [1] have described a controller for continuum robots, which utilizes a neural network feedforward component to compensate for dynamic uncertainties. Research in autonomous multi-robot systems often focuses on mechanisms to enhance the efficiency of the group through some form of cooperation among the individual agents. Moreover, the versatility of a multi-robot system can provide the heterogeneity of structures and functions required to undertake different missions in unknown environmental conditions [9, 11-13].

This paper proposed a neural network based approach to the solution of the path and time optimization problem for mobile robots. A biologically inspired neural network is used to real-time collision-free motion planning of mobile robots in an unknown environment. Four layer perceptron neural network has been used to design the controller the first layer is used to input layer which directly read from the arrays of sensor of the robot network consisting with two hidden layer which adjusted the weight of neuron and a output layer which provide hading angle of the robot. Back propagation method is used to minimize the error and optimize the path and time of mobile robot to reach the target.

This paper organized into five sections following the introduction, the kinematics behavior of mobile robot is described in section 2. Analysis of navigation method using neural network architecture explained in section 3. The simulation results are discussed and to demonstrate the superiority of the proposed methodology comparison has been made with other methods [12]. In section 4. Finally conclusions are discussed in section 5.

2. KINAMATICS OF MOBILE ROBOT

The kinematics approaches to controlling mobile robots are posture stabilization. Posture stabilization is to stabilize the robot to a reference point. The kinematics analysis of the khepra-III mobile robot is analyzed in this section. The kinematics model of mobile robots has been shown in Figure 1.

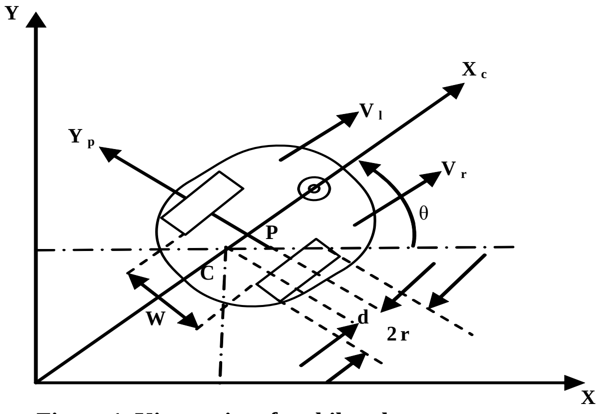

Figure 1. Kinematics of mobile robot

It consists of a vehicle chassis with two driving wheels mounted on the same axis and a front point sliding support. The two driving wheels are independently driven by two actuators to achieve the motion and orientation. Both wheels have the same diameter denoted by ‘2r’. The two driving wheels are separated by distance ‘W’. The center of gravity (COG) of the mobile robot is located at point ‘C’. The point ‘P’ is located in the intersection of a straight line passing through the middle of the vehicle and a line passing through the axis of the two wheels. The distance between points P and C is ‘d’. A motion controller based on neural network technique is proposed for navigation of the mobile robot. The main component in the motion controller is the low level inverse neural controller, which controls the dynamics of the mobile robot. The kinematics of the differential drive mobile robot is based on the assumption of pure rolling and there is no slip between the wheel and surface.

$$v_t = \frac{1}{2}\left[v_r + v_l\right] \tag{1}$$

$$\omega_t = \frac{1}{W}\left[v_r - v_l\right] \tag{2}$$

$$v_r = r\,\omega_r \text{ and } v_l = r\,\omega_l \tag{3}$$

Where v is linear and ω is angular velocity of the vehicle. Suffix r, l and t stand for right, left wheel and tangential (with respect to its center of gravity point ‘C’ measured in a right wheel) respectively.

The position of the robot in the global coordinate frame [O X Y] is represented by the vector notation as,

$$q = \left[X_c \; Y_p \; \theta\right]^T \tag{4}$$

Where X_c and Y_p are the coordinates of the point P in the global coordinate frame [Figure 1]. The variable θ is the orientation of the local coordination of the local coordinate frame [P X_c Yp] attached on the robot platform measured from the horizontal axis. Three generalized coordinates can describe the configuration of the robot as equation (4). The mobile robot system considered here is a rigid body and the wheels are pure rolling and no slippage. This states that the robot can only move in the direction normal to the axis of the driving wheels. Therefore, the component of the velocity of the contact point with the ground, orthogonal to the plane of the wheel is zero[3, 17] i.e.

$$[\dot{y}_p \cos\theta - \dot{x}_c \sin\theta - d\dot{\theta}] = 0 \tag{5}$$

All kinematics constraints are independent of time, and can be expressed as

$$A^T(q)\dot{q} = 0 \tag{6}$$

Where A(q) is the input transformation matrix associated with the constraints.

$$C^T A(q) = 0 \tag{7}$$

Where C(q) is the full rank matrix formed by a set of smooth and linearly independent vector fields spanning the null space of $A^T(q)$.

From equation (6) and (7) it is possible to find an auxiliary vector time function V(t) for all time ‘t’

$$\dot{q} = C(q)V(t) \tag{8}$$

The constraint matrix in (6) for a mobile robot is given by

$$A^T(q) = [-\sin\theta \quad \cos\theta \quad -d] \tag{9}$$

And C(q) matrix is given by

$$C(q) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \tag{10}$$

And

$$V(t) = [v \quad \omega]^T \tag{11}$$

Where v is the linear velocity of the point ‘p’ along the robot axis and ω is the angular velocity.

Therefore, the kinematics equation in (8) can be described as

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_p \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{12}$$

Equation (12) is called the steering system of the vehicle. The control problem is to find a suitable control law so that the system can track desired reference trajectories. The control laws are designed to produce suitable left and right wheel velocities for driving the mobile robot to follow required path trajectories.

3. ANALYSIS OF NEURO-CONTROLLER

Artificial neural networks consist of a set of simple, densely interconnected processing units. These units transform signals in a non-linear way. Neural networks are non-parametric estimators which can fit smooth functions based on input-output examples. The neural network used is a four-layer perceptron [6]. The chosen number of layers was 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.

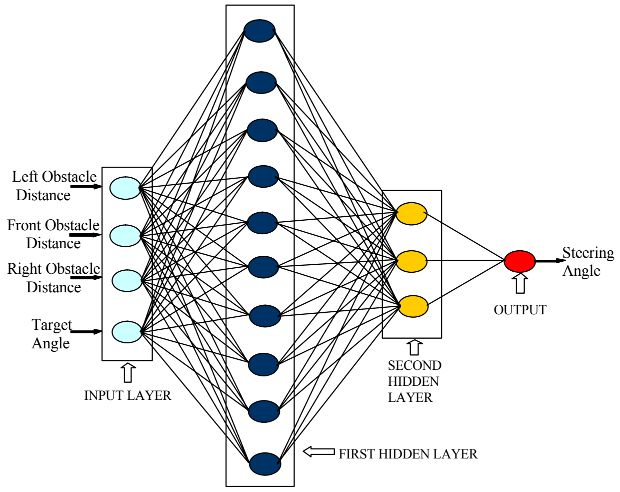

Figure 2.Four-layer neural network for robot navigation

These numbers of hidden neurons were also found empirically. Figure 2 depicts the neural network with its input and output signals.

The neural network is trained to navigate by presenting it with 200 patterns representing typical scenarios, some of which are depicted in Figure 3. For example, Figure 3a 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.

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

$$y_1^{\{1\}} = \text{Left obstacle distance from the robot} \tag{13a}$$

$$y_2^{\{1\}} = \text{Front obstacle distance from the robot} \tag{13b}$$

$$y_3^{\{1\}} = \text{Right obstacle distance from the robot} \tag{13c}$$

$$y_4^{\{1\}} = \text{Target bearing} \tag{13d}$$

These input values are distributed to the hidden neurons which generate outputs given by [6]:

$$y_j^{\{lay\}} = f\left(V_j^{\{lay\}}\right) \tag{14}$$

where

$$V_j^{\{lay\}} = \sum_i W_{ji}^{\{lay\}} . y_i^{\{lay-1\}} \tag{15}$$

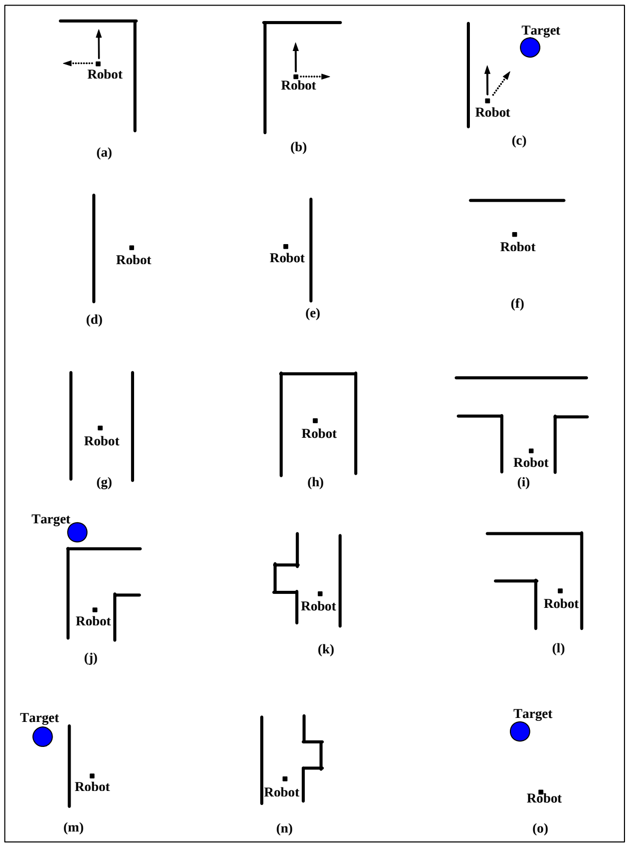

Figure 3. Example training patterns

lay = layer number (2 or 3)

j = label for jth neuron in hidden layer 'lay'

i = label for ith 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(.) = activation function, chosen in this work as the hyperbolic tangent function :

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{16}$$

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 set of presented training patterns:

$$Err = \frac{1}{2} \sum_{\substack{\text{all training}\\ \text{patterns}}} \left(\theta_{desired} - \theta_{actual}\right)^2 \tag{17}$$

The error back propagation method is employed to train the network [6]. This method requires the computation of local error gradients in order to determine appropriate weight corrections to reduce Err. For the output layer, the error gradient $\delta^{\{4\}}$ is:

$$\delta^{\{4\}} = f'\left(V_1^{\{4\}}\right)\left(\theta_{desired} - \theta_{actual}\right) \tag{18}$$

The local gradient for neurons in hidden layer {lay} is given by:

$$\delta_j^{\{lay\}} = f'\left(V_j^{\{lay\}}\right)\left(\sum_k \delta_k^{\{lay+1\}} W_{kj}^{\{lay+1\}}\right) \tag{19}$$

The synaptic weights are updated according to the following expressions:

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t+1) \tag{20}$$

and $\Delta W_{ji}(t+1) = \alpha \Delta W_{ji}(t) + \eta \delta_j^{\{lay\}} y_i^{\{lay-1\}}$ (21)

where

δ= 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:

$$\theta_{actual} = f\left(V_1^{\{4\}}\right) \tag{22}$$

where

$$V_1^{\{4\}} = \sum_i W_{li}^{\{4\}} y_i^{\{3\}} \tag{23}$$

It should be noted learning can take place continuously even during normal target seeking behaviour. This enable the neural controller to adopt the changes in the robot's path while moving towards target. The proposed neural controller and kinematics gives steering angle from wheel velocities based on the environmental conditions.

4. SIMULATION RESULTS

The simulations were conducted with the ROBNAV software being developed in the laboratory using C++ [10]. Figure 4, show a typical screen of the software. It can be noted that, in addition to the neural network based navigation, the software also allows other navigation control. To demonstrate the effectiveness and the robustness of the proposed method, simulation results on mobile robot navigation in various environments are exhibited.

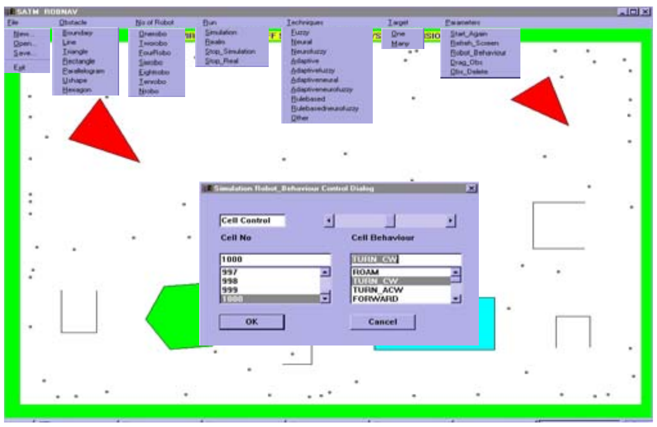
SATM ROBNAV
File Obstacle No of Robot Run Techniques Target Parameters
Simulation Robot_Behaviour Control Dialog
Cell Control
Cell No
Cell Behaviour
OK
Cancel

Figure 4. Robot navigation software package (ROBNAV).

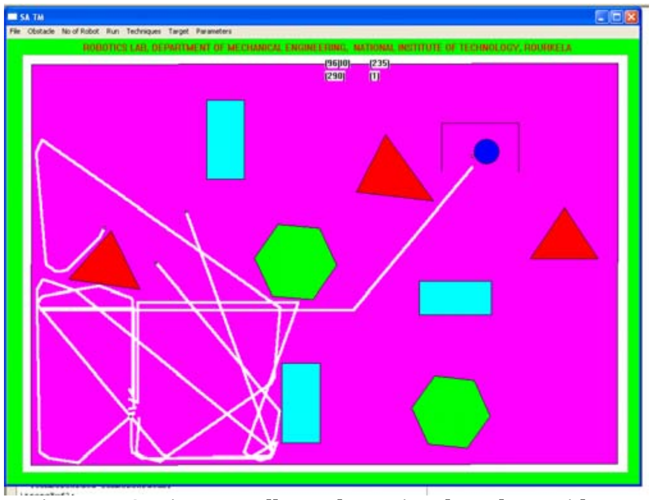
SA TM
File Obstacle No of Robot Run Techniques Target Parameters
ROBOTICS LAB, DEPARTMENT OF MECHANICAL ENGINEERING, NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

Figure 5. Static as well as dynamic obstacle avoidance behavior

The obstacle avoidance behavior is activated when the reading from any sensors are less than the minimum threshold values. This is how the robot determines if an object is close enough for a collision. When an object is detected too close to the robot, it avoids a collision by moving away from it in the opposite direction. Collision avoidance has the highest priority and therefore, it can override other behaviors, in this case, its main reactive behavior is decelerating for static as well as dynamic obstacle avoidance as shown in Figure 5.

The wall following behavior mode will be adopted when the mobile robot detects an obstacle in the front while it is moving towards target along the left or right side of the wall, the mobile robot may turn left or right because presence of obstacle in the front. Another special condition appears as the mobile robot detects an obstacle in the front while the target tracking control mode is on operation. In this case, the fixed wall following behavior should be firstly performed, that is, the mobile robot must rotate clockwise or counterclockwise such that it can align and move along the wall (Figure 6). In absence of wall following behavior the robot is incapable of reaching the goal position when its en count U shaped or dead end obstacles on their path. In such a situation the robot should keep on heading towards the goal position. But when it moves towards the goal position, the robot also comes closer to the obstacles. Any obstacle-avoidance

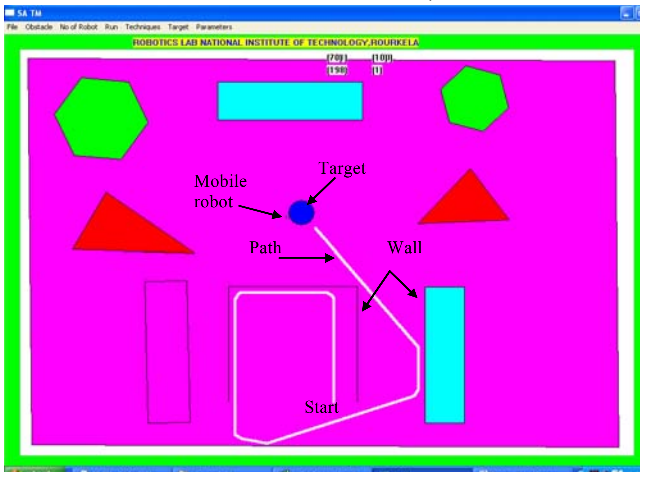
SA TM
File Obstacle No of Robot Run Techniques Target Parameters
ROBOTICS LAB NATIONAL INSTITUTE OF TECHNOLOGY,ROURKELA
Mobile robot
Target
Path
Wall
Start

Figure 6. Robot escaping from U shaped wall

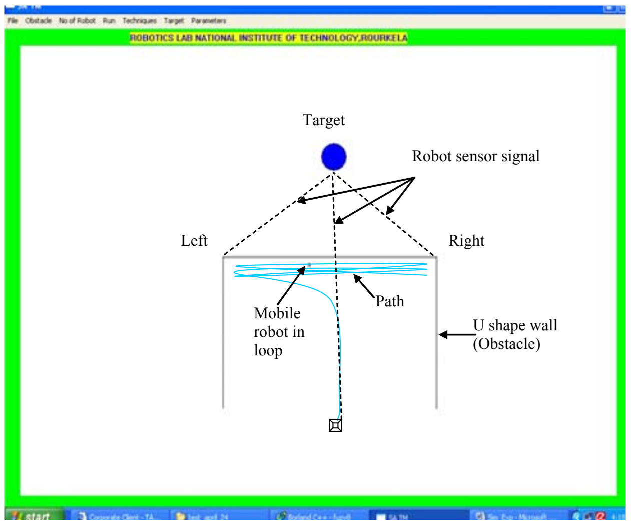

Figure 7. Robot in concave loop

behavior except wall following behavior would make the robot divert from its goal position. When robot move in large U shape obstacle, at the initial stage, the robot runs directly toward the target, since the obstacles sensed are far away from the robot. Then, the robot makes a turn to left, in order to avoid the obstacles at the direct front. Since the target is located to the right side of the robot, the behavior of approaching target tries to make the robot turn to the right. Contrarily, the obstacle-avoidance behavior makes the robot move away from the obstacles. As a result, the robot moves into the right, and the target orientation is increasing gradually. Consequently, the robot travels along the indefinitely loop in this concave trap as shown in Figure 7. To avoid this loop robot must have wall following behavior, when the robot is moving to a specified target through a U shaped obstacle or narrow channel, it must reflect following edge behavior so that robot may locate, find and reach the specified target as shown in Figure 6 and escape from dead end shown in Figure 8.

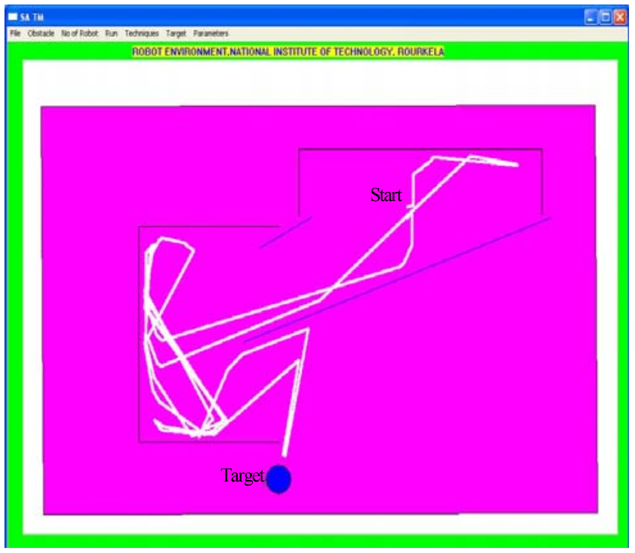
SA TM
ROBOT ENVIRONMENT,NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
Start
Target

Figure 8. Robot escaping from dead end obstacle

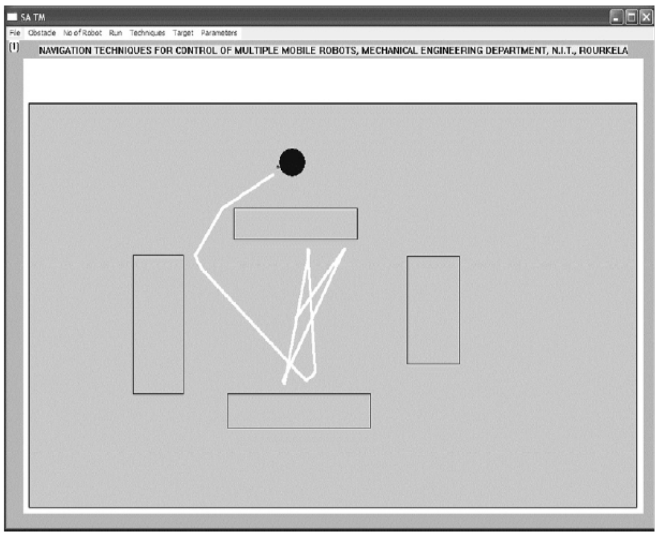
SA TM
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA

Figure 9 (a). Navigation path of mobile robot using fuzzy controller by Pradhan et al.

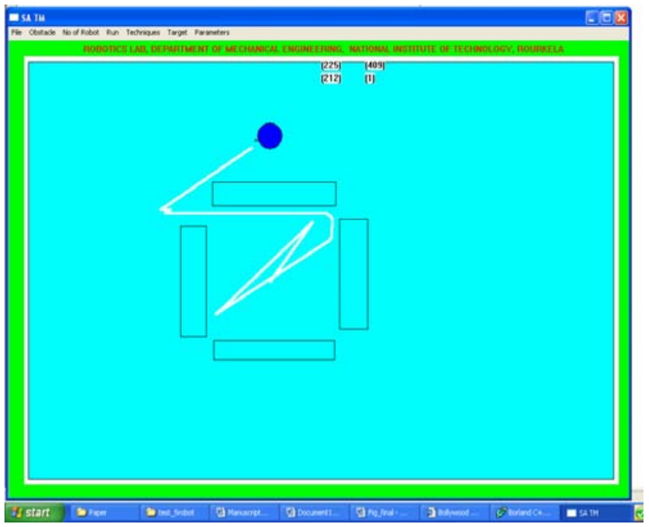
SA TM
ROBOTICS LAB, DEPARTMENT OF MECHANICAL ENGINEERING, NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

Figure 9 (b). Navigation path of mobile robot using proposed neural controller

Figure 9. Result comparisons with Pradhan et al.

When the acquired information from the sensors shows that there are no obstacles around robot, its main reactive behavior is target steer. Neural controller mainly adjusts robots motion direction and quickly moves it towards the target if there are no obstacles around the robot as shown in Figure 5. In the proposed control strategy, reactive behaviors are formulated and trained by neural network.

To verify the superiority of the controller the results from the proposed neural controller for mobile robot have been compared with the result from fuzzy controller by Pradhan et al.[12] has been compared from proposed neural controller of mobile robot (Figure 9). They shows a very good agreement.

5. CONCLUSIONS

The simulation results are compared with the results obtained from the other investigation and they are in very good agreement. Back-propagation neural network technique has been used for making the controller. Software has been developed using C++ to get the simulation results. The developed neural controller has got the following salient feature:

1. Avoid any static as well as dynamic obstacle along the path.
2. The robot rapidly recognizes its surroundings which provides sufficient information for path optimization during navigation.
3. The proposed Neural controller successfully be applied to dynamic as well as static environments.
4. The proposed Neural controller is simple but efficient tool for mobile robot navigation, especially in a dynamic environment.
5. Training patterns of each network can be generated by simulation rather than by experiment, saving considerable time and effort.

In the future analysis a hybrid controller can be formed for more efficient navigational the mobile robots.

6. REFERENCES

[1] Braganza, D., Dawson, D. M., Walker, I. D. and Nath, N. 2007. A Neural Network Controller for Continuum Robots. In Proceedings of the IEEE Trans. on robotics. (December 2007), 1270-1277.

[2] Corradini, M. L., Ippoliti G. and Longhi, S.2003. Neural Networks Based Control of Mobile Robots: Development and Experimental Validation. Journal of Robotic Systems. 20 no.10(2003), 587–600.

[3] Das, T. and Kar, I.N. 2006. Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. IEEE Trans. Cont. syst. Technology, 14 no.3(2006), 501-510.

[4] Folgheraiter, M., Gini, G., Nava, A. and Mottola, N. 2006. A BioInspired Neural Controller For a Mobile Robot. In Proceedings of the IEEE Int. Conf. Robotics and Biomimetics, (Kunming, China, December 17 - 20, 2006). 1646-1651.

[5] Gregor, K. and Igor, S. 2007. Tracking-error model-based predictive control for mobile robots in real time. Robotics and Autonomous Systems. 55(2007), 460–469.

[6] Haykin, S. 2006. Neural Networks a Comprehensive Foundation, 2nd Ed., Pearson prentice hall India, 2006.

[7] Kondo, T. 2007. Evolutionary design and behavior analysis of neuromodulatory neural networks for mobile robots control. Applied Soft Computing. 7(2007), 189–202.

[8] Muñiz, F., Zalama, E., Gaudiano, P. and López Coronado, J., 1995. Neural controller for a mobile robot in a non stationary environment, In Proceedings of the Second IFAC Conference on Intelligent Autonomous (Vehicles, Helsinki, Jun 12-14, 1995).

[9] Nelson, A.L., Grant, E. and Henderson, T.C. 2004. Evolution of neural controllers for competitive game playing with teams of mobile robots. Robotics and Autonomous Systems. 46 (2004), pp. 135–150.

[10] Parhi D.R., 2000.Navigation of multiple mobile robots in an unknown environment, Doctoral Thesis, Cardiff School of Engineering, University of Wales, UK.

[11] Parhi, D.R. Pradhan, S.K. Panda, A.K. and Behra, R.K. 2008. “The stable and precise motion control for multiple mobile robots”, Applied Soft Computing, In Press, Available online 22 June 2008.

[12] Pradhan, S.K. Parhi, D.R. Panda, A.K. and Behra, R.K. 2008. “Fuzzy logic techniques for navigation of several mobile robots”, Applied Soft Computing, In Press, Available online 22 April 2008.

[13] Pallottino, L., Scordio,V. Bicchi, G. A. and Frazzoli, E. 2007. Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems. In Proceedings of the IEEE Trans. Robotics. 23 no. 6 (Dec. 2007), 1170-1183.

[14] Racz, J. and Dubrawski, A. 1995. Artificial neural network for mobile robot topological Localization. Robotics and Autonomous Systems, 16(1995), 73-80.

[15] Wolf, D. F. and Sukhatme, G.S. 2007. Semantic Mapping Using Mobile Robots. IEEE Trans. Robotics. 24(2), (April 2008). pp.245-258.

[16] Yang, S.X. and Meng, M. 2000. An efficient neural network approach to dynamic robot motion planning. Neural Networks, 13(2000), 143–148.

[17] Zhang, Q., Shippen, J. and Jones, B. 1999. Robust backstepping and neural network control of a low quality nonholonomic mobile robot. Int. J. Mach. Tools Manufact. 39(1999), 1117–1134.