Obstacle Avoidance of mobile robot using PSO based Neuro Fuzzy Technique

Sourav Dutta

Department of Electrical Engineering
Mallabhum Institute Of Technology
Bishnupur ,West Bengal -722122,India

Abstract— **Navigation and obstacle avoidance are very important issues for the successful use of an autonomous mobile robot. To allow the robot to move between its current and final configurations without any collision within the surrounding environment, motion planning needs much treatment. Thus to generate collision free path it should have proper motion planning as well as obstacle avoidance scheme. This work mainly deals with the obstacle avoidance of a wheeled mobile robot in structured environment by using PSO based neuro-fuzzy approach. Here three layer neural network with PSO is used as learning algorithm to determine the optimal collision-free path.**

Keywords- Obstacle Avoidance of Mobile Robot, PSO based Neuro Fuzzy approach.

I. INTRODUCTION

Current research in robotics aims to build an autonomous and intelligent robot, which can plan its motion in a dynamic environment. A successful use of an autonomous mobile robot depends on its controller. Controlling of a car-like robot is difficult as they are subjected to nonholonomic (nonintegrable) kinematic constraints involving the time derivatives of configuration variables [2, 3, 4, 5, 6] and dynamic constraints. The path of the robot is also constrained by the partially-unknown movement of the moving obstacles [7], known as uncluttered environment. Thus, to generate collision-free path of a car-like robot during its navigation among several moving obstacles, it should have proper motion planning as well as obstacle avoidance schemes. Both analytical like potential field method as well as graph-based techniques have been used to solve the navigation problems of robots involving static obstacles. But, all such methods may not be suitable for on-line implementations due to their inherent computational complexity and limitations. Soft computing includes fuzzy logic, genetic algorithm, particle swarm optimization, neural network and their different combinations and it can solve such complex real world problems within a reasonable accuracy. Since artificial neural networks (ANN) have the ability to learn the situations, many investigators have successfully applied the feed-forward neural network to develop the model related to the navigation problem of a car-like robot. In this paper an PSO algorithm is used for learning purpose of the neuro-fuzzy system. Sensors are used to get the the distance between the robot and the obstacle. Inputs of the neuro-fuzzy system are based on the output data obtained from sensors.

II. NEURO-FUZZY APPROACH

In order to steer the mobile robot, a neuro-fuzzy technique can be applied to control so that the performance can automatically be improved. The fuzzy rules are generated by the trajectories provided by a human. The operator would provide some of the trajectories, which avoid the obstacles and the neuro-fuzzy system should be able to extract the corresponding fuzzy rules and membership functions. In the subsequent sections, the structure along with the learning algorithm of the neuro-fuzzy system is presented.

A. Structure of the Neuro-Fuzzy system

The neuro-fuzzy system could be seen as a three-layer network. The nodes of the network are cascaded together in layers. A diagram of the neuro-fuzzy system is shown in Figure 1.

The first layer or input layer comprises several nodes, each one consisting of a radial basis neuron. The inputs to the radial basis neuron are the inputs to the neuro-fuzzy system, while the outputs of the nodes are as follows:

$$p_{ij} = \exp\left(-\frac{(U_i - m_{ij})^2}{\sigma_{ij}^2}\right),$$

where $i = 1, 2, ..., N_{1,}\ j = 1, 2, ..., N_2$ (1)

m_{ij} = center of the membership function corresponding to the *i*th input and the *j*th neuron of the first layer.

U_i = *i*th input to the neuro-fuzzy system obtained from sensors.

σ_{ij} = width of the membership function corresponding to the *i*th input and the jth neuron of the first layer.

p_{ij} = output of the radial basis neuron (or degree of membership for the *i*th input corresponding to *j*th neuron).

N_1 = number of neuro-fuzzy system inputs.

N_2 = number of nodes at the hidden layer.

The output layer could be considered as a linear neuron layer, where the weight connections between the hidden layer and the output layer are the estimated values of the outputs.

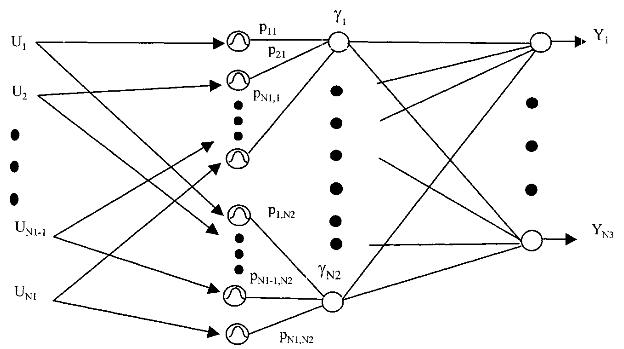

Fig 1. Diagram of the Neuro-Fuzzy system

the node outputs corresponding to the hidden layer are calculated as:

$$\gamma_j = \min[p_{1j}, p_{2j}, ..., p_{ij}, ..., p_{N_1 j}]$$
$$j = 1, 2, ..., N_2 \tag{2}$$

Where

γ_j = output of the *j*th node at the hidden layer.

The outputs of these nodes are calculated by this expression:

$$Y_k = \frac{\sum_j sv_{jk}\gamma_j}{\sum_j \gamma_j},$$
$$j = 1, 2, ..., N_2, k = 1, 2, ...N_3 \tag{3}$$

Where

Y_k = *k*th output of the neuro-fuzzy system.

sv_{jk} =estimated value of the *k*th output provided by the *j*th node at the hidden layer.

N_3 =number of outputs of the neuro-fuzzy system.

It could be said that the output layer carries out the defuzzification process, providing the outputs of the neuro-fuzzy system. To sum up, the structure of the neuro-fuzzy system could be seen as a typical radial basis network, where an additional layer has been inserted between the radial basis layer (the input layer) and the linear layer (the output layer).

The neurons of this additional layer calculate the degrees of membership corresponding to the different rules, that is, they apply the min fuzzy operator, being N_2 the total number of fuzzy rules. Once these calculations have been carried out, the output layer applies a defuzzification process in order to obtain numeric values for the outputs of the system.

In the neuro-fuzzy system there are some parameters which determine the relation between the inputs and outputs. In this case, the behavior of the neuro-fuzzy system depends on the value of the following parameters: the membership function centers, the widths of the membership functions, and the estimated output values. In order to determine these parameters a learning algorithm has to be designed.

B. *Learning Algorithm*

The subsequent to the development of ANFIS approach, a number of methods have been proposed for learning rules and for obtaining an optimal set of rules.

In this paper PSO algorithm is used for learning of the ANFIS structure.

B.1 Evolution of PSO

Let *N* denotes the swarm numbers. In general, there are three attributes, current position a_{ij}, current velocity v_{ij} and past best position Pb_{ij}, for particles in the search space to present their features. Each particle in the swarm is iteratively updated according to the aforementioned attributes assuming that the objective function *f* is to be minimized so that the dimension consists of *n* particles and the new velocity of every particle is updated by (4).

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + c_1 r_{1,i}(t)[Pb_{ij}(t) - a_{ij}(t)] \\ &+ c_2 r_{2,i}(t)[Gb_i(t) - a_{ij}(t)] \end{aligned} \tag{4}$$

where v_{ij} is the velocity of the *j*-th particle of the *i*-th swarm for all $i \in 1...N, w$ is the inertia weight of velocity, c_1 and c_2 denote the *acceleration coefficients*, r_1 and r_2 are two uniform random values falling in the range between (0, 1), and *t* is the number of generations. The new position of the *i*-th particle is calculated as follows:

$$a_{ij}(t+1) = a_{ij}(t) + v_{ij}(t+1) \tag{5}$$

The past best solution of each particle is updated by:

$$Pb_i(t+1) = \begin{cases} Pb_i(t), \; f(a_i(t+1) \geq f(Pb_i(t)) \\ a_i(t+1), Otherwise \end{cases} \tag{6}$$

The global best solution Gb will be found from all of particles during previous three steps are defined as:

$$Gb(t+1) = \arg\min_{Pb_i} f(Pb_i(t+1)), 1 \leq i \leq n \tag{7}$$

B.2 Learning by PSO

In this section, the way PSO employed for updating the ANFIS parameters is explained. The Gaussian type membership functions are used here and their parameters are $\{m_{ij}, \sigma_{ij}\}$. The center of the membership function m_{ij} is replaced by particle position and the rate of change i.e $\partial E / \partial m$ of the ANFIS is replaced with particle velocity expression. In this learning scheme sv_{jk} is replaced by $a_{N_1+k,j}$ where $j = 1, 2, ..., N_2$ and $k = 1, 2, ..., N_3$

$$E = \frac{1}{2}\sum_{k=1}^{N_3}\left(Y_k - \hat{y}_k\right) \tag{8}$$

Y_k = kth neuro-fuzzy system output.

$\hat{y}_k$ = kth desired output.

Width of the membership function calculated as follows

$$\frac{\partial E}{\partial \sigma_{ij}} = -2\left[\sum_{k=1}^{N_3}\left(y_k - \hat{y}_k\right)\frac{sv_{jk}\sum_{j=1}^{N_2} p_{ij} - \sum_{j=1}^{N_2} sv_{jk}\, p_{ij}}{(\sum_{j=1}^{N_2} p_{ij})^2}\right]$$

$$\times p_{ij}\left[\frac{\left(u_i - m_{ij}\right)^2}{\sigma_{ij}^2}\right] \tag{9}$$

and it adapted as

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \eta\frac{dE}{d\sigma_{ij}} \tag{10}$$

Where η is the learning rate.

III. Simulation Results

The neuro-fuzzy system along with the learning algorithm has been used to steer a mobile robot, so that the robot could avoid the obstacles found in its trajectory. The results have been obtained by simulation. PSO parameters c_1 and c_2 are chosen as 2, r_1 and r_2 chosen as random number between 0 and 1,w is assigned as random value between 0.4 to 0.9.

The following PSO parameters are chosen as follows

Number of particles = 25;

Maximum particle velocity = 5;

Training is done off-line with the help of PSO algorithm. The computer simulation is carried out by considering more than one obstacle.

IV. Conclusion

Figure 2,3 and 4 shows schematic diagrams of the PSO based neuro-fuzzy approach for obstacle avoidance problem. In this paper, a PSO-neuro-fuzzy strategy has been proposed to drive a mobile robot. This approach is able to extract automatically the fuzzy rules and the membership functions in order to guide a wheeled mobile robot. The proposed neuro-fuzzy strategy consists of a three-layer neural network along with an evolutionary (PSO based) learning algorithm. This system has been implemented in simulation obtaining satisfactory results.

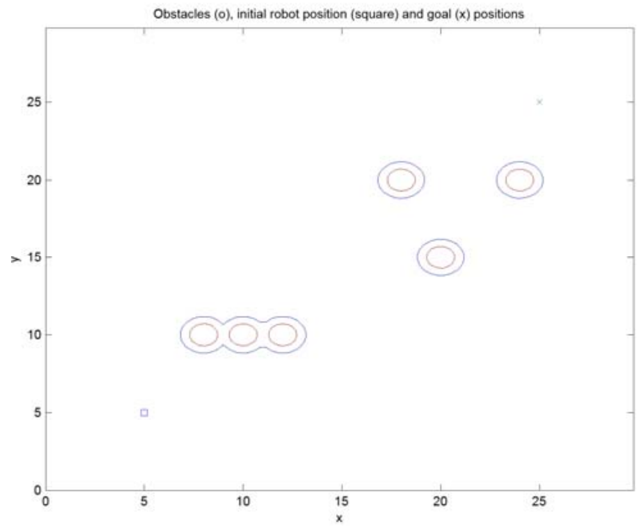

Fig 2. Obstacle (o), Initial robot position (square), goal (x)

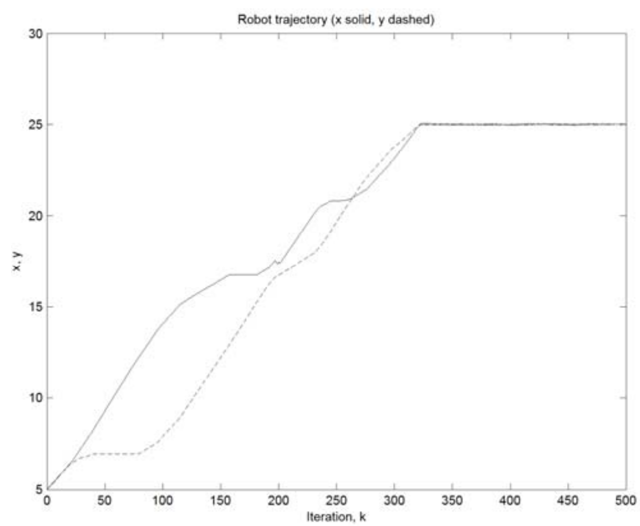

Fig 3. Robot trajectory (x solid, y dashed) , Iteration -500

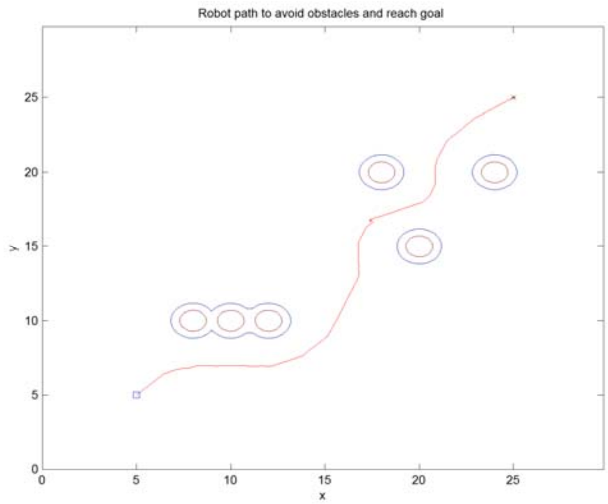

Fig 4. Robot path to avoid obstacle and reach goal

V. REFERENCES

[1] J.C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, Dordrecht, 1991.

[2] A. Bemporad, A.D. Luca, G. Oriolo, Local incremental planning for a car-like robot navigating among obstacles, Proceedings of IEEE Conf.on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 1205–1211.

[3] S. Hong, S. Kim, K. Park, J. Lee, Local motion planner for nonholonomic mobile robots in the presence of the unknown obstacles, Proc. IEEE Conf. on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 1212–1216.

[4] T. Fraichard, A. Scheuer, Car-like robots and moving obstacles, Proc. IEEE Conf. on Robotics and Automation 1996, pp. 64–69.

[5] A.D. Luca, G. Oriolo, Local incremental planning for non-holonomic mobile robots, 1994 IEEE Int. Conference on Robotics and Automation, San Diego, CA, 1994, pp. 104–110.

[6] D. Feng, H. Krogh, H. Bruce, Dynamic steering control of conventionally-steered mobile robots, Proc. IEEE Conf. Robotics Automation, 1990, pp. 390–395.

[7] D.K. Pratihar,W. Bibel, Near-optimal, collision-free path generation for multiple robots working in the same workspace using a genetic-fuzzy system, Machine Intell. Robot. Control 5 (2) (2003) 45–58.

[8]. Li, H. and Gupta, M. (eds): *Fuzzy Logic and Intelligent Systems*, Kluwer Academic, Dordrecht, 1995. MOBILE ROBOT CONTROL USING HUMAN OPERATOR EXPERIENCE 319

[9]. Maclin, R. and Shavlik, J. W.: Creating advice – taking reinforcement learners, *Machine Learning* **22** (1996), 251–281.

[10]. Maeda, M., Maeda, Y., and Murakami, S.: Fuzzy drive control of an autonomous mobile robot, *Fuzzy Sets and Systems* **39** (1991), 195–204.

[11]. Miller, W., Sutton, R., and Werbos, P. (eds): *Neural Networks for Control*, Mit Press, Cambridge, MA, 1990.

[12]. Mitchell, R. J. and Keating, D. A.: Neural network control of a simple mobile robot, in: L. J. Landau and J. G. Taylor (eds), *Concepts for Neural Networks: A Survey*, Springer, London.

[13]. Wang, T. and Zhang, B.: Time-varying potential field-based perception–action behaviors of mobile robot, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, May 1992, pp. 2549–2554.

[14] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization," *In: Proceedings of IEEE International Conference on Neural Network. Piscataway: IEEE Press,* 1995, pp. 1942-1948.

[15] T. L. Vincent, and W. J. Grantham, "Optimality in Parametric Systems," *Wiley: New York,* 1981.

[16] Y. H. Shi, and R.C. Eberhart, "Experimental study of particle swarm optimization," *In: Proceedings of SCI Conference. Piscataway: IEEE Press,* 1999, pp. 1945-1950.

[17] D. Y. Zhu, and H. Shi, Principle and application of artificial neural network, *Beijing: Science Press,* 2006.

Sourav Dutta received his M.Tech degree in Control System under Electrical Engineering department from National Institute of Technology, Kurukshetra, India. Currently he is a lecturer of Electrical Engineering department at Mallabhum Institute of Technology, West Bengal, India. His research interests are intelligent control, mobile robot and soft computing. He has two international research publications.