Robot Motion Planning Using Neural Networks: A Modified Theory

Subhrajit Bhattacharya , Siddharth Talapatra
Department of Mechanical Engineering, IIT Kharagpur

Abstract -- **A model based on competitive learning has been developed for robot motion planning. The configuration of a robot with *n* degrees of freedom is described by a point in an *n*-dimensional variable Space. The variable space has been discretized into a finite number of cells each of which is associated with a neural activity. A modified version of a shunting equation describing the dynamics of the neurons has been presented. A new algorithm for updating the neural activities has been suggested. The model developed here has been compared to a previous model with the aid of simulations. This comparative study clearly highlights how the modified model enhances the target reaching capability of the robot.**

I. Introduction

One of the key issues in mobile robot systems is real time path planning. Autonomous navigation involves obstacle avoidance along with moving towards a target. There have been several approaches towards this issue, encompassing both the static and dynamic environment problem.

One approach to autonomous navigation is the wall-following method [1, 2]. Here the robot navigation is based on moving alongside walls at a predefined distance, while considering the obstacles as just another wall. Though computationally less demanding, this approach has only specific applications (a floor cleaning robot in a long hallway). An improvement upon this method is the edge detection approach [3, 4], where the positions of the vertical edges of the obstacle are determined and the robot is steered around either edge. However this approach is heavily dependent upon sensor accuracy.

A powerful scheme for autonomous navigation is the Potential Fields Method, originally suggested by Khatib [5]. Here obstacles are considered as centers of repulsion and targets as centers of attraction (with global effect). The robot traverses the path of least potential gradient. A shortcoming of this method is that it assumes a known and prescribed world model of obstacles. Later a model called the Virtual Force Field Model [6] was developed which employed the integration of the above mentioned concept and the concept of Certainty Grid.

This paper deals with the use of Neural Networks in autonomous navigation. Neural Network approaches have been used in quite a few path planning algorithms [7], where a Hopfield type of network with continuous neurons was employed. In this paper we have used a model loosely based on a Competitive Learning model for real-time obstacle free path traversal to track down a target moving in an arbitrary fashion. Unlike most approaches, this method does not require the global picture, each neuron in the neural network having only local connections. The constantly updating neural activities generate an activity landscape which guides the robot towards its goal. All this happens in real time due to which a fast moving target can be easily tracked down.

II. The Model

Our model is essentially based on the biologically inspired neural networks model developed by Yang and Meng [8]. A brief discussion on the original model will be first presented. It will be followed by the modifications made as part of our present work.

Based on the physical problem, a Neural Network architecture [8] is decided upon. The N-Dimensional Neural Network architecture corresponds to an N-Dimensional robot configuration space Z . The corresponding N-dimensional Neural activity landscape basically represents the dynamically varying environment of the configuration space.

The neurons are spatially arranged in the discretized N dimensional landscape representing the configuration space Z . With the i^{th} neuron is attached a quantity called Neural Activity x_i and an external input I_i .

The shunting equation [9, 10] determining the dynamic neural activity of the i^{th} neuron is:

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[x_j]^+ \right) - (D + x_i)[I_i]^- \quad (1)$$

Parameters *A*, *B* and *D* indicate the passive decay rate, the upper and lower bounds of the neural activity, respectively.

Variable x_i is the neural activity of the i^{th} neuron, a continuous variable $\in [-D, B]$. The input I_i = E if the ith neuron corresponds to a target, I_i = -E if it is an obstacle, otherwise I_i = 0. E generally is a large positive quantity. $[I_i]^-$ is the inhibitory input caused by the obstacles, while

$\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[x_j]^+ \right)$ is the excitatory input resulting from the target and the local lateral connection among the neurons.

The nonlinear function $[a]^+$ and $[a]^-$ are defined as, $[a]^+ = \max\{a, 0\}$, and $[a]^- = \max\{-a, 0\}$.

q_i and q_j represent the position vectors to the i^{th} and j^{th} neurons respectively.

The connection weight w_{ij} from the *j*th to the *i*th neuron is defined as $w_{ij} = f(|d_{ij}|)$, where $d_{ij} = |q_i - q_j|$ is the distance between the *i*th and q^{th} neurons in Z . $f(a)$ is a monotonically decreasing function. The function used by Yang and Meng [8] was,

$$f(a) = \begin{cases} \dfrac{m}{a} & \text{for } 0 < a < r_0 \\ 0 & \text{for all other } a \end{cases} \qquad (2)$$

m and a are positive. This functional form ensures that a neuron has local connections in a small region around it of radius r_0 neurons.

The shunting equation is such that the target has global impact as the neural activity of a target propagates while that of an obstacle doesn't. So effect of obstacles is only local.

Robot motion is decided by choosing a winning neuron based on the neural activity landscape.

For a given *present location* in S, denoted by *qp*, the next location *qn* (also called "command location") is obtained by,

$$q_n \Leftarrow x_{q_n} = \max(x_i, i = 1,2,\cdots,k) \qquad (3)$$

The neighboring neurons of the neuron corresponding to the present location of the robot in the configuration space are considered, and the robot moves to the position of one of the neighboring neurons (*including the neuron corresponding to the current position*) having the maximum neural activity.

The values chosen by Yang and Meng [8] for the different parameters in their simulations were, A=10, B=D=1, μ=1, r_0=2 and E=100.

Before discussing the achievements of the present work, some observations on equations (1), (2) and (3) ought to be made, which formed the basis of our development.

i. According to (3), a neuron with higher value of xi has greater potential for attracting the robot, while the one with a lower value somewhat repels it.

ii. The first term in (1) always tries to bring the value of x_i closer to 0. For the second term, if xi becomes greater than B, the term tries to reduce the value of xi, else it tries to increase it. And as far as the third term is concerned, it tries to decrease the value of xi till xi is greater than –D. That's how B and –D form the upper and lower bounds of xi.

iii. The second term uses the $[\,]^+$ operator, hence taking into account the influence of the positive values (i.e. potential targets or attractors) of Ii and xj. While the third term using the $[\,]^-$ operator takes into account the influence of the negative values (i.e. potential obstacles or repellers) of Ii.

iv. It is only in the second term that the influence of the nearby neurons is considered. It is of interest to note that while taking into account the influence of the neighboring neurons, the quantity $[x_j]^+$ has been chosen. This ensures that nearby neurons with negative activity have no influence on the activity of the ith neuron. Thus it implies that although there is a way in which information about a target may propagate from one neuron to its neighbor, the information about an obstacle can't propagate in similar fashion if we use the shunting equation (1). This somewhat give preference of being attracted towards a target over being repelled from an obstacle. In fact, when placed in a complex maze with a distant target, the robot often stumbled over an obstacle or cut through it! Such examples demonstrated the mentioned observation.

v. Though the summation in (1) was mentioned to be done over all the neighboring neurons, *including itself*, from (2) we get w_{ii} ? 8, since d_{ii} = 0. With the present form of (2) there are scopes of divergence. Hence (2) can be modified.

vi. According to (3), as the new position of the robot is to be determined by finding the neighboring neuron with highest activity, including itself, it is highly probable that the robot may trace back its path when it encounters a complex maze ahead. Or in the worst case, the robot may keep on oscillating within a few positions. Hence it is logical to give lower preference to those neurons whose positions have already been traversed by the robot.

Keeping in view the above mentioned observations, the model was modified as follows:

- In light of observation iv., (1) was modified as,

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)\left([I_i]^+ + a\sum_{j=1}^{k} w_{ij}[x_j]^+ \right) - (D + x_i)\left([I_i]^- + b\sum_{j=1}^{k} w_{ij}[x_j]^- \right) \qquad (4)$$

where, 0=*a* =1 and 0=*ß*=1.

Note that *a*=1, *ß*=0 gives back the original Yang and Meng's model. But through experiments it was observed that a small positive value of ß makes the model more efficient in complex mazes.

- In light of observation v., (2) was modified as,

$$f(a)=\begin{cases}\dfrac{m}{a+1} & \text{for } 0<a<r_0 \\ 0 & \text{for all other } a\end{cases} \qquad (5)$$

- To solve the problem mentioned in observation vi., an innovative technique was devised. In order to insure that the robot has a lower tendency to go back to a position it has recently visited, the external inputs Ii of those neurons are temporarily given a small negative value. For example (as used in our simulations), the position visited in the last time step is given a value –E/8, the one visited two time steps before is enforced with a value of –E/16 and the one visited three time steps before is enforced with a value of -E/32. All still previously visited positions are returned to their original values of Ii.

And it was also made sure that the robot doesn't get stuck in its own position by making the comparison stated in (3) among the neighboring neurons except that of the presently occupied position.

Hence the robot is forced to make a move, preferably to a new position, in every time step.

III. RESULTS AND COMPARISONS

In this section a comparative study will be presented on the performances of the original model of Yang and Meng and our modified model, under various cases of complex mazes and target location. The parameters used with the original model (Yang and Meng) have already been mentioned. The corresponding parameters used by us in the modified model are:

E = 100, A = 10, B = 5, D = 1, r0 = 1.41, alpha = 0.9 and beta = 0.2. Following are some simulation results, where the dotted line shows the path t raversed by the robot.

The problem presently under study consists of two dimensional mazes and uses a neural network activity landscape with same topology as physical space.

Maze I

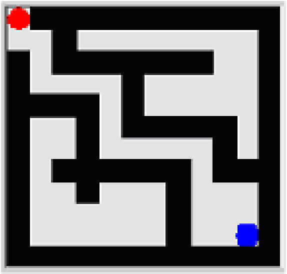


Fig.1a. The maze

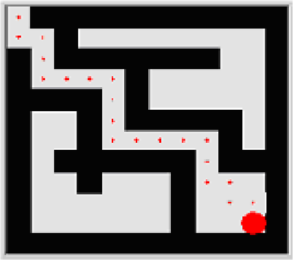


Fig.1b. Yang-Meng model model

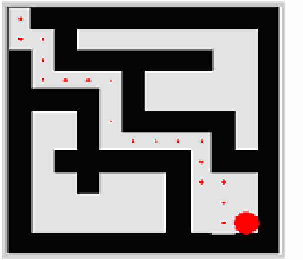


Fig.1c. The modified

Maze II

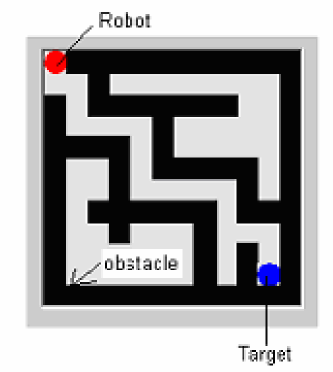
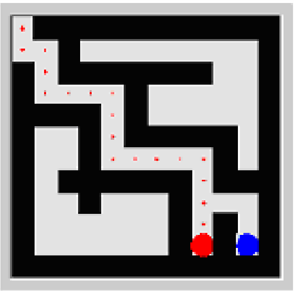


Fig.2a. The maze



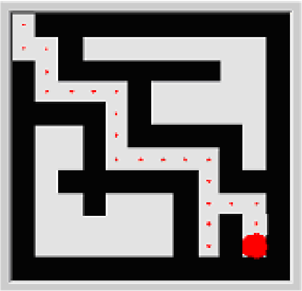


Fig.2b. Yang-Meng model Fig.2c. The modified
model

Maze III

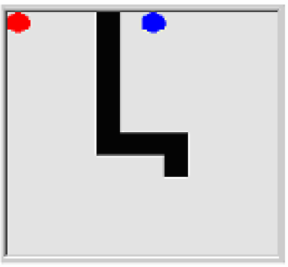


Fig.3a. The maze

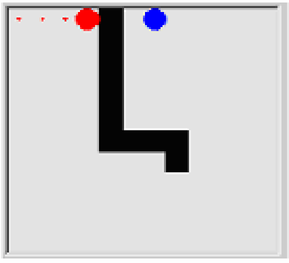


Fig.3b Yang-Meng model

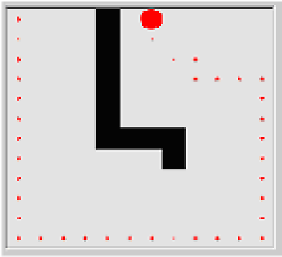


Fig.3c. The modified
model

Maze IV

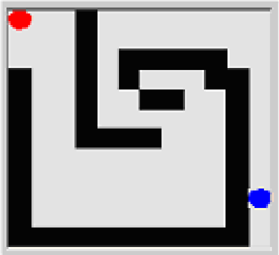


Fig.4a. The maze

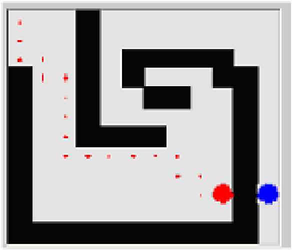


Fig.4b. Yang-Meng model

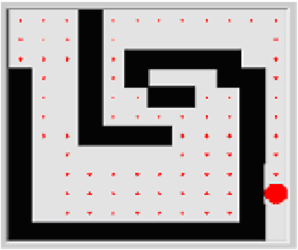


Fig. 4c. The modified
model

In the first illustration, we have considered a maze which has a well guided path leading to the target, the sort of examples that were considered in [8]. Here we see that both models perform equally well.

In the second case, the maze considered previously has been slightly modified so as to make the target more inaccessible. The original model guides the robot to a point separated from the target by a single cell of the obstacle. The robot is trapped and hence unsuccessful in its mission. The modified model however manages to guide the robot out this trap. This can be attributed to the combined effect of two modifications: excluding the current neuron as a possible candidate for the winner neuron, and making the robot reluctant to traverse a path already covered. The last behavior is due to the third modification shown in the last section.

The third maze is a simple case which highlights the positive effects of the modifications made. In the original model the robot stumbles over an obstacle due to the total lack of propagation of obstacle information. The small non-zero beta in the new term introduced in this model (see equation (4)) removes this deficiency. Hence we find that in our model, the robot manages to find its way around the obstacle and reach its goal. The fourth maze is the most challenging of the lot. Once again the model proposed by Yang and Meng fails and gets stuck due to its inability to effectively represent obstacles in the neural activity map. In the modified model, the robot is strongly repelled by the obstacles and after considering different paths, it manages to find the correct path.

IV. Conclusion

In this paper a model for autonomous robotic motion guidance for obstacle avoidance has been developed. It is loosely based on a competitive learning model. It was seen that in the neural activity map, each neuron had only local connections and the global picture was not required during the path planning. The neural activities are not allowed to attain values outside the prescribed range. The stability and convergence of the original shunting equation has been proved rigorously using the Lyapunov stability theory (Grossberg, 1988). Hence the system is stable.

The model developed in this paper has been shown to perform better in complicated mazes, especially when the path to the target is not well guided. Simulations have been made depicting the relative performances of the two models in four different mazes. It has been clearly shown how the modifications incorporated by us have affected the performance of the robot. Unlike the previous model, propagation of neural activity due to obstacles takes place, characterized by the parameter *ß*. Also the robot is reluctant to trace a path already traversed due to a different neural activity updating algorithm. Our algorithm forces the robot to move on to a different cell in every iteration. All this prevents the robot from getting stuck at one place and since the robot has a greater tendency to move to new places, its ability to reach its goal is enhanced.

However there is a shortcoming to this algorithm. In certain cases, especially in those where the Yang-Meng model fails, the robot takes sub optimal routes. A judicious choice of the parameters can fix this problem, and this can be achieved by optimizing the parameter values for shortest possible path.

Acknowledgment

We would like to thank Prof. S.K. Barai, Deptt. of Civil Engg., IIT Kharagpur for his invaluable guidance and suggestions which helped us in completing the present work.

References

[1] Giralt, G., "Mobile Robots." NATO ASI Series, Vol. F11, Robotics and Artificial Intelligence, Springer-Verlag, 1984, pp. 365-393.

[2] Iijima, J., Yuta, S., and Kanayama, Y., "Elementary Functions of a Self-Contained Robot "YAMABICO 3.1”." Proc. of the 11th Int. Symp. on Industrial Robots, Tokyo, 1983, pp. 211-218.

[3] Borenstein, J., "The Nursing Robot System." Ph. D. Thesis, Technion, Haifa, Israel, 1987.

[4] Borenstein, J. and Koren, Y., "Obstacle Avoidance With Ultrasonic Sensors." IEEE Journal of Robotics and Automation., Vol. RA-4, No. 2, 1988, pp. 213-218.

[5] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." 1985 IEEE International Conference on Robotics and Automation, March 25-28, 1985, St. Louis, pp. 500-505.

[6] Borenstein, J. and Koren, Y., 1989, "Real-time Obstacle Avoidance for Fast Mobile Robots." IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept./Oct., pp. 1179-1187.

[7] J.J.Hopfield, “Neural networks and physical systems with emergent collective computational abilities”, Proceedings of the National Academy of Sciences of the United States of America, 79:2554, 1982.

[8] Simon X. Yang, Max Meng, “An efficient neural network approach to dynamic robot motion planning”, Neural Networks, 13 (2000), 143-148.

[9] Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology (London)*, *117*, 500–544.

[10] Grossberg, S. (1988). Nonlinear neural networks: principles, mechanisms, and architecture. *Neural Networks*, *1*, 17–61.