Obstacle Avoidance Using Modified Hopfield Neural Network for Multiple Robots

Panrasee Ritthipravat[1] Thavida Maneewarn[1] Djitt Laowattana[1] and Kenji Nakayama[2]
[1] Center of Operation for Field Robotics Development (FIBO),
King Mongkut's University of Technology Thonburi. 91 Pracha-Uthit Road, Bangkok 10140, Thailand
Tel. +66-2-470-9339, Fax. +66-2-470-9691, e-mail: pan@fibo.kmutt.ac.th
[2] Dept. of Information and System Engineering, Faculty of Engineering,
Kanazawa University. 2-40-20, Kodatsuno, Kanazawa, 920, Japan.
Tel. +81-76-234-4896, Fax. +81-76-234-4900, e-mail: nakayama@t.kanazawa-u.ac.jp

Abstract: In this paper, dynamic path planning of two mobile robots using a modified Hopfield neural network is studied. An area which excludes obstacles and allows gradually changing of activation level of neurons is derived in each step. Next moving step can be determined by searching the next highest activated neuron. By learning repeatedly, the steps will be generated from starting to goal points. A path will be constructed from these steps. Simulation showed the constructed paths of two mobile robots, which are moving across each other to their goals.

1. Introduction

Obstacle avoidance is one of the most interesting issues in mobile robot field. It has been applied on either a single robot or multiple robot system. Many researchers proposed their algorithms to solve obstacle avoidance problem either static or dynamic environment.

Y. Arai, T. Fujii, H. Asama, Y. Kataoka [1] proposed collision avoidance in multiple robot system. Their robots are equipped with LOCISS. Each robot will recognize whether the approaching object is another robot or obstacle. The rule matrix for collision avoidance is predetermined. Reinforcement learning is applied to acquire adaptive behavior. Mochida, Ishiguro, Aoki and Uchikawa [2] presented a behavior control method inspired by living organisms which immune and emotion systems were introduced to cope with dynamic changing environment. The idea of emotional system was used to model frustration function, which would be used to determine robot behavior.

Ishiguro, Watanabe and Uchikawa [3] introduced using an immune system concept to cope with dynamic changing environment. Antibody and antigen matching were used to determine the robot behavior. Various kinds of robot behavior were represented as antibodies. The environment information was also set as antigens. The immune network system selected an antibody, which was most suitable for current situation or an antigen.

R. Mehrotra and D. M. Krause [4] presented an obstacle free path planning by using a quadtree coding method. M. G. Lagoudakis [5] introduced dynamic path planning and obstacle avoidance by using the Hopfield neural network. The idea of receptive field is presented. The external input is added in a goal and obstacles. Such the target is highest activated neuron. The activation levels of neurons are spread decreasingly around the target as wave propagation. However the Hopfield neural network cannot guarantee the monotonically decreasing from the goal because it employs symmetrical connection weights. Activation level can propagate both directions. This easily causes local peaks of activation level.

P. Ritthipravat and K. Nakayama [6] proposed a modified Hopfield neural network for static obstacle avoidance. By using asymmetric weight matrix, a distinguishable area of activation levels from the starting point to the goal can be constructed. The path can be generated easily by considering highest activated neurons in this area.

In this paper, dynamic path planning for multiple robots is studied. Next section, the Hopfield neural network will be introduced. Weight matrix determination and a step selection are described. Simulation results of dynamic path planning for either a robot or two robots are presented and discussed.

2. Hopfield Neural Network

The Hopfield neural network was proposed in1982, by physicist John J. Hopfield [7]. It can be used to solve information retrieval or optimization problems. It has a recurrent feature. Output of all another neurons, v_j, are fed back, weighted and summed to be the input to a neuron u_i. Only one neuron, i, is selected randomly and updated its state in each time.

$$u_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij} v_j + \theta_i \,. \qquad (1)$$

where u_i is input potential of neuron i, v_j is output of neuron j, w_{ij} is connection weight from neuron j to i and θ_i is bias of neuron i

According to P. Ritthipravat and K. Nakayama [6] external bias are added in the goal and obstacle units as:

$$\theta_i(t) = \begin{cases} +\infty & target-unit \\ -\infty & obstacle-unit \\ 0 & otherwise \end{cases} \,. \qquad (2)$$

The effect of adding bias in Eq(2) is that the goal unit will be maximally activated. The obstacle unit is deactivated independently of other units.

The original Hopfield network assumes zero selfcoupling, $w_{ii}=0$, and symmetrical weights, $w_{ij}=w_{ji}$. This symmetrical property guarantees that the system energy function always converges to the basin of attraction or the equilibrium state. This means no state changed anymore.

Many forms of activation functions can be used to classify an output level of a neuron accordance with its input. In the discrete Hopfield network, we can use a sign function to classify output to be either 1 or 0 as:

$$sign(u_i)=\begin{cases}1 & u_i\geq 0\\ 0 & u_i<0\end{cases}. \quad (3)$$

Such output is discrete with value $\{1,0\}$. The output is 1 means the neuron is activated or fired. While the output is 0 means the neuron is deactivated or quenched. For the analog Hopfield network, a sigmoid function or a hyperbolic function can be used. For the sigmoid function, the output is analog with value [0,1] as:

$$\phi(u_i)=\frac{1}{1+e^{-u_i}}. \quad (4)$$

For the hyperbolic function, the output is analog with value [-1,1] as:

$$\phi(u_i)=\frac{1-e^{-u_i}}{1+e^{-u_i}}. \quad (5)$$

In this paper, system state is set with value [0,1]. By using the hyperbolic function, the activation function can be presented as:

$$\phi(u_i)=\begin{cases}\dfrac{1-e^{-u_i}}{1+e^{-u_i}} & u_i\geq 0\\ 0 & u_i<0\end{cases}. \quad (6)$$

Fig. 1 shows activation functions in each form. The activation function used in this paper is shown with a thickest line in the figure.

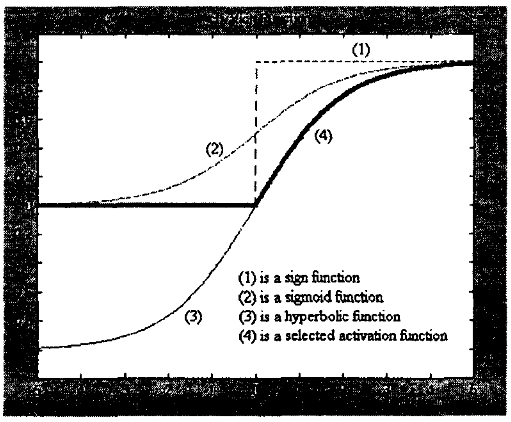

Figure 1. shows activation functions

3. Weight Matrix Determination

Either M. G. Lagoudakis [5] or P. Ritthipravat and K. Nakayama [6] mapped the Configuration Space, C into neuron space. Each neuron corresponds to subset C_i of C. Such all neurons can be represented as overall Configuration Space. Each neuron has a receptive field, RF_i which is subset of units of its neighborhood. Each unit i connects to the units in its RF_i. There are no connections outside the receptive field. Weight is determined as a decreasing function, which depends on Euclidean distance $\rho(i,j)$ between the units. Weight between unit i and unit j can be represented as:

$$w_{ij}=f(\rho(i,j)), \quad (7)$$

The function can take various forms as:

$$f(\rho)=e^{-\gamma\rho^{\alpha}}. \quad (8)$$

where γ is a positive real number. α indicates the width of a decreasing function. It may be 2,.4 or more.

P. Ritthipravat and K. Nakayama [6] proposed the modified Hopfield network by using asymmetric weight matrix as follows:

Weight matrix is determined by using the distance between the goal and each neuron. The strongest weight will be set on a neuron, which is nearest to the goal direction. The weight will be decreased spread out of the strongest weight as shown in Fig. 2

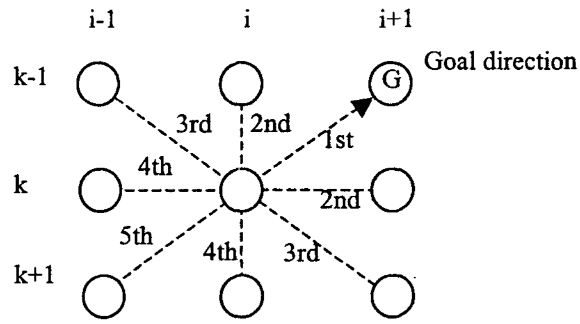

Figure 2. shows relations of weight in each unit

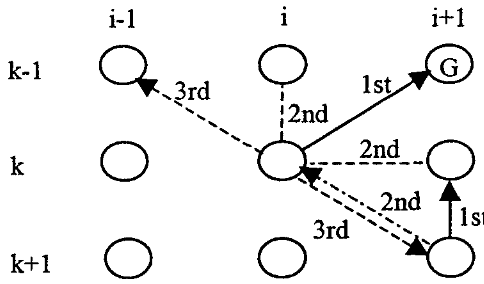

Figure 3. shows asymmetric weight feature

As shown in Fig. 2, the largest weight is the weight from a neuron (k,i) to a neuron (k-1,i+1), which locates toward the goal direction. The connection weights from the neuron (k,i) to (k-1,i) and (k,i+1) have the 2^{nd} largest value. By setting the weight in this manner, neurons, which close to the goal direction, have higher activation level and are activated easily. Such the level of activating can be distinguished obviously. However the symmetrical property could not be conserved as shown in Fig.3. The weight from a neuron (k+1,i+1) to (k,i) is stronger than weight from the neuron (k,i) to (k+1,i+1).

As expressed in Eq (8), the connection weigh can be set accordance with the nearness to the goal by varying γ in each direction.

4. Step Selection

As discussed above, the level of activated neuron can be separated noticeably. Next moving step will be generated from selecting the best-activated neuron. The 'best' means it has the highest average neighboring activated neurons. The criteria for selecting the best-activated neuron is same as discussed in [6] as follows:

Two highest activated neurons are searched from its closet neurons. The level of activating of their adjacent units, excluding the current neuron and its neighborhood, are compared and selected the highest one. Not only the next step of its neighboring units is concerned, but prevention of moving back to the previous step should also be concerned in order to avoid loop constructing.

In order to compare the level of activation among adjacent units, a diagram of current neuron (k,i) and its vicinity are presented in Fig. 4.

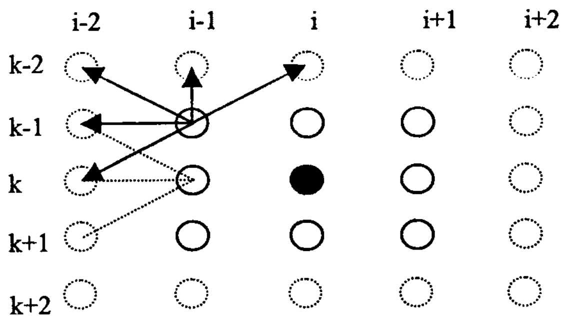

Figure 4. shows a neuron (k,i) and its vicinity

There are 8 neighboring neurons around the neuron (k,i). Each neighboring unit has its nearby units. Neuron (k-1,i-1) has 5 next closest units, i.e., (k,i-2), (k-1,i-2), (k-2,i-2), (k-2,i-1), (k-2,i). Neuron (k,i-1) has 3 next neighboring units, i.e., (k+1,i-2), (k,i-2), (k-1,i-2). By considering in this way, the adjoining units in each direction can be obtained. Three highest units in each direction are averaged. Two of them corresponding to two highest neighboring units are used for next step selection. Such a flow chart of path planning can be shows in Fig. 5.

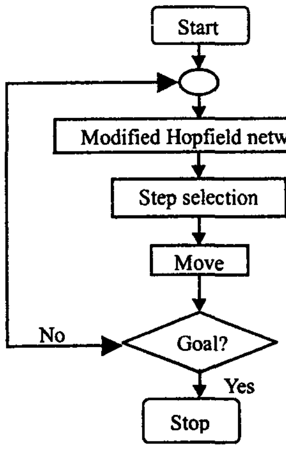

Figure 5. shows system model

As seen in Fig.5, the step will be selected and set as a sub-goal of a robot. The robot will move to the sub-goal and begin to generate next sub-goal by using the previous neuron states. If there is an obstacle, the external bias will be added according to Eq(2). Doing this, dynamic path planning will be generated. In order to apply this algorithm to multiple robots, other robots will be realized as obstacles. So deactivated neurons will be set in the other robot positions.

5. Simulation Results

The simulation results are composed of 2 parts, i.e., comparison to a static path planning generated by [6] and applying to two robots, which are moving across each other. A network composes of 10x10 neurons. The starting, goal and obstacle units are marked by 'S', 'G' and 'O', respectively. The starting and goal states are fixed to be maximally activated neurons. The obstacle states are set to be deactivated units. Initial state of another neurons are selected randomly and scaled in a range of [0,0.5]. For each step generation, different learning time will be used. The 150 epochs will be set for first 5 steps for an equilibrium state approaching purpose. After that, the step can be selected by using 20 epochs per step. So number of epochs can be calculated by (150*5)+((number of step - 5)*20). After the step was selected, the small negative bias was added to the previous state in order to prevent moving back.

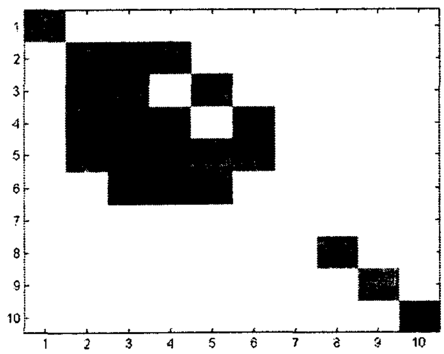

Figure 6. shows static path planning for application 1

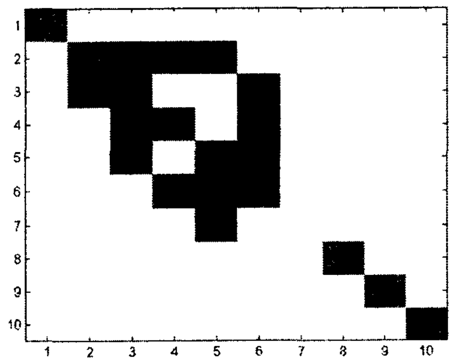

Figure 7. shows dynamic path planning for application 1

By comparison to a static path planning which is simulated for 1500 epochs, the simulation results are shown in Figs. 6-7. The other dynamic path planning for different application is presented in Fig. 8. Fig. 9 shows simulation result for two mobile robots, which are moving across each other.

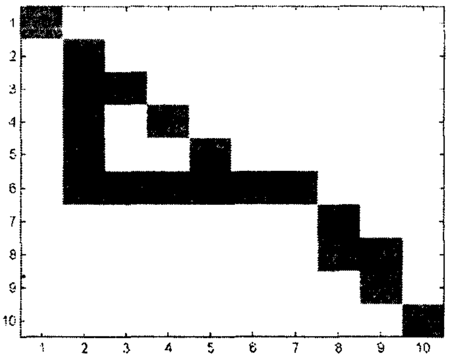

Figure 8. shows dynamic path planning for application 2

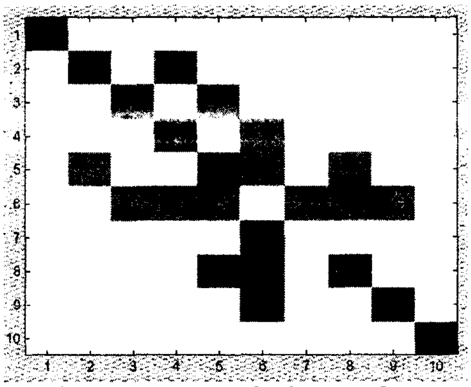

Figure 9. shows dynamic path planning for two robots

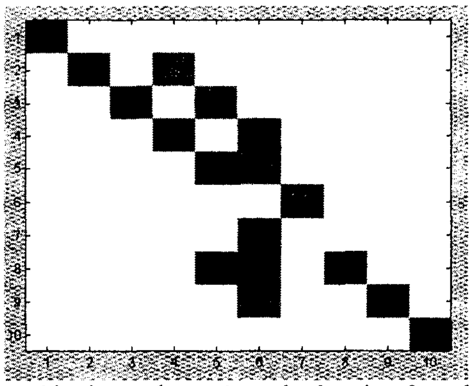

Figure 10. shows dynamic path planning for robot1

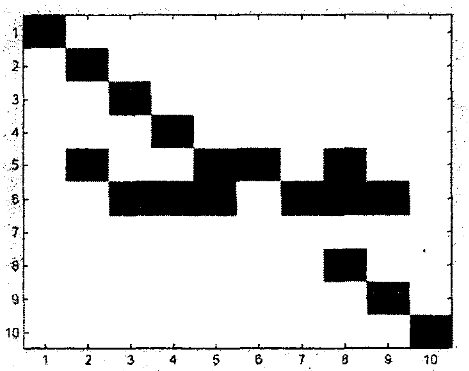

Figure 11. shows dynamic path planning for robot2

6. Discussion

By comparing to the static path planning which was done until equilibrium state approaching, the simulation result showed shorter time simulation and a shorter path because the starting state had been changed according to the step selection. The starting bias was also changed to the next step. However it couldn't move closely to obstacles because the deactivated neuron and the small negative bias in the previous step affects. Fig. 8 showed these two affects obviously.

For the case of two robots that are moving across each other, robot1 avoided robot2 when it came closely as seen in Fig2 10,11. The robot1 selected the forth step in a deviate direction for avoiding the robot2. This showed the effective path for the robot path planning.

According to the learning must be done in each step it couldn't work for highly dynamic environment. Longer learning time lead to more effective path generation. However [6] had been showed that the learning time should not take too long. It can be determined from the minimum state energy as discussed in [6].

7. Conclusion

In this research, obstacle avoidance for multiple robots is studied. Modified Hopfield network using asymmetric weight matrix has been explored. The moving step can be generated from the system model, which are composed of the Modified Hopfield network and the best step selection. The simulation results showed effective path generations. However the learning still takes time, such it couldn't apply to highly dynamic environment.

8. Reference

[1] Y. Arai, T. Fujii, H. Asama, Y. Kataoka, 'Adaptive Behavior Acquisition of Collision Avoidance among Multiple Autonomous Mobile Robots', *Proceeding of IROS*, 1997

[2] Mochida, T.; Ishiguro, A.; Aoki, T.; Uchikawa, Y., 'Behavior arbitration for autonomous mobile robots using emotion mechanisms', *International Conference on Intelligent Robots and Systems, IROS*, Pittsburgh, Pennsylvania, USA,1995

[3] Ishiguro, A., Watanabe, R., Uchikawa, Y., 'An immunological approach to dynamic behavior control for autonomous mobile robots', *International Conference on Intelligent Robots and Systems, IROS*, Pittsburgh, Pennsylvania, USA, 1995

[4] R. Mehrotra and D. M. Krause, 'Obstacle-free Path Planning for Mobile Robots'

[5] M. G. Lagoudakis, 'Hopfield Neural Network for Dynamic Path Planning and Obstacle Avoidance', Technical Paper, University of Southwestern Louisiana.

[6] P. Ritthipravat and K. Nakayama, 'Obstacle Avoidance by Using Modified Hopfield Neural Network', *International Conference of Artificial Intelligence 2002, ICAI 2002*, Las Vegus.

[7] S. Haykin, 'Neural Networks: A Comprehensive Foundation', Prentice Hall International, Inc., New Jersy