Proceedings of the 2008 IEEE/ASME
International Conference on Advanced Intelligent Mechatronics
July 2 - 5, 2008, Xi'an, China

A New Neural Network for Robot Path Planning*

Yongmin Zhong, Bijan Shirinzadeh and Yanling Tian
Robotics & Mechatronics Research Laboratory
Monash University
VIC 3800, Australia
{Yongmin.Zhong; Bijan.Shirinzadeh; Yanling.Tian}@eng.monash.edu.au

Abstract – **This paper presents a new methodology based on neural dynamics for robot path planning. The target activity is treated as an energy source injected into the neural system and is propagated through the local connectivity of neurons in the state space by neural dynamics. The elegant properties of harmonic functions are incorporated in the neural system by formulating the local connectivity of neurons as a harmonic function. An improved Hopfield-type neural network model is established for propagating the target activity among neurons in the manner of physical heat conduction, which guarantees that the target and obstacles remain at the peak and the bottom of the activity landscape of the neural network, respectively. The real-time collision-free robot motion is planned through the dynamic neural network activity without any prior knowledge of the dynamic environment, without explicitly searching over the global free workspace or searching collision paths, and without any learning procedures. Examples are presented to demonstrate the effectiveness and efficiency of the proposed methodology.**

Index Terms – mobile robots, path planning, Hopfield neural network, collision avoidance and analogy systems.

I. INTRODUCTION

There are a lot of studies on robot motion planning using various approaches. Some researchers use global methods [1] to search the possible paths in the workspace. However, the global methods can only deal with static environments, and are computationally expensive when the environment is complex. The searching methods [2] suffer from undesired local minima, i.e. the robots may be trapped in some cases such as with concave U-shaped barriers. Seshadri and Ghosh presented a path planning model by using an iterative approach [3]. However, this model is computationally complicated, especially in a complex workspace. Li and Bui proposed a fluid-based model for robot path planning, while only static workspaces are considered in this model [4]. Ong and Gilbert presented a method for robot path planning with penetration growth distance [5]. This method searches over collision paths instead of the free space, and can generate optimal and continuous robot paths in a static environment. In general, most of the above methods can only generate the accessible path with free collision in a static environment, while a moving object or introduction of new objects requires that the whole work environment be constructed dynamically. In addition, with the increase of obstacles, the complexity of the algorithms increases exponentially.

Neural network based methods have received considerable attention for generating real-time robot trajectories. Svestka and Overmars presented a probabilistic learning approach to mobile robot path planning [6]. This method uses a learning phase, a query phase and a local method for computation of feasible robot paths. Zalama et al presented a neural network model for mobile robot navigation [7, 8]. This model can generate dynamic paths with avoidance of obstacles through unsupervised learning. However, this model is computationally complicated since it incorporates the vector associative map model and the direction-to-rotation effector control transform model. Quoy et al. reported a dynamic neural network based method for robot path planning and control [9]. In this method, robot trajectories are generated by using a learning neural network to learn the planning map continuously. However, the learning process is computationally complicated with the complex work environment. In general, these learning based approaches have a time-consuming process, and the generated trajectories are not optimal, particularly during the initial learning phase [10].

To avoid the time-consuming learning process, Glasius et al proposed a Hopfield-type neural network model for real-time trajectory generation with obstacle avoidance in a non-stationary environment [11]. It is rigorously proven that the generated trajectory does not suffer from undesired local minima and is globally optimal in a stationary environment. However, this model requires that the robot dynamics be faster than the target and obstacle dynamics, and have difficulty in dealing with fast changing environments. Yang and Meng proposed a neural network approach to dynamic collision-free trajectory generation for robots in dynamic environments [12, 13]. This approach was extended to various robot systems [14, 15] and further a distance-propagating system was developed for robot path planning [10]. The method is more general and more powerful than the Glasius's model. However, this model has a set of parameters that depends on the grid solution or the dynamic scene itself, and there are no explicit rules for selection of these parameters. In addition, it is not clear whether the generated path is optimal or not, since the model generates different paths according to different parameter values. Further, the aimless path search can also be resulted in some cases.

This paper presents a new methodology based on neural dynamics for real-time collision-free robot path generation in an arbitrarily varying environment. The proposed methodology incorporates elegant properties of harmonic functions in a neural system to carry out real-time robot path planning. The target activity is treated as an energy source injected into the neural system, and is propagated in the state space by neural dynamics. By formulating the local connectivity of neurons as the local interaction of harmonic

* This work is supported by Australian Research Council

functions, an improved Hopfield-type neural network model is established to propagate the target activity among neurons in the manner of physical heat conduction, which guarantees the target and obstacles remain at the peak and the bottom of the activity landscape, respectively. The novelty of the proposed neural network model is that the local connectivity of neurons is harmonic rather than symmetric in the existing neural network models. The proposed neural network model cannot only generate real-time optimal and collision-free robot paths without learning procedures, prior knowledge of target or barrier movements, optimizations of any cost functions, and explicitly searching over the free work space or collision paths, but it can also easily respond to real-time changes in dynamic environments. Further, the proposed neural network model is parameter-independent and has an appropriate physical meaning.

II. MODEL ALGORITHM

A. Originality

In the recent twenty years, a great deal of attention has been dedicated to the Hopfield-type neural network, which was proposed by Hopfield [16] with an electrical circuit implementation. The dynamics of the Hopfield-type neural network are described by:

$$C_i \frac{dy_i}{dt} = -\frac{1}{R_i} y_i + \sum_{j=1}^{n} T_{ij} g_j(y_j) + I_i \tag{1}$$

where $i = 1, 2, \ldots, n$, and $n \geq 2$ is the number of neurons in the network. For the i^{th} neuron *e(i)*, y_i is the activity/potential, I_i is the input, and $C_i > 0$ and $R_i > 0$ are the capacitance and resistance, respectively. T_{ij} is the weight of the connection from neuron *e(j)* to neuron *e(i)*, which indicates the effect of neuron *e(j)* on neuron *e(i)*. Functions g_j ($j = 1, 2, \cdots, n$) are neuron activation functions, and are usually represented as sigmoid functions. The neural network is globally stable, if the connected weights are symmetric, i.e. $T_{ij} = T_{ji}$.

The Hopfield-type neural network has many applications in optimization problem solving, visual perception, sensory motor control, robot path planning and image processing. These applications heavily depend on the dynamic behaviours of the neural network.

B. Proposed Neural Network Model

The neural network architecture of the proposed model is a discrete topographically organized map. The location of the *i*th neuron *e(i)* at the grid in the finite dimensional state space represents a position in the work space or a configuration in the joint space. Each neuron has only local connectivity to its neighbouring neurons that constitute the neighbourhood of the neuron. The neuron responds only to the stimulus within its neighbourhood.

The dynamics of the proposed neural network are described by:

$$\frac{dy_i}{dt} = -\frac{1}{R_i} y_i + \sum_{j=1}^{n} T_{ij} y_j + I_i \tag{2}$$

where

$$I_i = \begin{cases} E & \text{if there is a target} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\begin{aligned} y_i = y_i(0) = F & \quad \text{if there is an obstacle} \\ y_i(0) = F & \quad \text{otherwise} \end{aligned} \tag{4}$$

where E is a positive constant, and F is a arbitrary constant bounded by the constant K.

$$T_{ij} = \begin{cases} A_{ij} & \text{if } e(j) \in N_r(i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where E is a positive constant. F is an arbitrary constant, and it can be simply set to zero. A_{ij} defines the local connectivity of neuron *e(i)*, and $N_r(i)$ is the connected neighbourhood of neuron *e(i)* within a radius r, which is defined as:

$$\begin{aligned} N_r(i) = \{ & e(j) \mid \max\{|\, j - i \,|\} \leq r, \\ & 1 \leq j \leq n, j \neq i, \ \ \overline{e(j)e(i)} = TRUE \} \end{aligned} \tag{6}$$

where r is a positive integer number, which represents the size of the neighbourhood defining the interaction range of neurons. Without loss of generality, we consider the case of the smallest neighbourhood, i.e. $r = 1$. $\overline{e(i)e(j)}$ is a Boolean variable, which indicates whether there is a connection between neurons *e(j)* and e*(i)* or not. If $\overline{e(i)e(j)}$ is *TRUE*, it means there is a connection between neurons *e(i)* and *e(j)*; otherwise, there is no connection between these two neurons. Therefore, $N_r(i)$ describes all the neurons connected to neuron *e(i)* in the small region with a radius r. The neuron *e(i)* has only local connections in the small neighbourhood. All the neurons having a connection to neuron *e(i)* are defined as the neighbouring neurons of neuron *e(i)*.

Therefore, the dynamics of the proposed neural network can be further written as:

$$\frac{dy_i}{dt} = -\frac{1}{R_i} y_i + \sum_{e(j) \in N_r(i)} A_{ij} y_j + I_i \tag{7}$$

III. MODEL CONSTRUCTION

A. Formulation for Local Connectivity of Neurons

Due to the elegant properties of harmonic functions, such as harmonic functions completely eliminate local minima by nature as they satisfy a min-max principle [17], the local connectivity of neurons is defined as a harmonic form. Since Laplace operator is a typical harmonic function, it is used as the formulation of the neural connections.

To formulate the local connections, Laplace operator has to be discretized on a grid map. The discretization of Laplace operator on a grid map can be easily obtained by a finite difference scheme [18] or a finite volume scheme

[19], and thus the local connectivity of neurons can be subsequently obtained. For example, for the point $\mathbf{P}_{i,j}$ shown in Fig. 1, Laplace operator discretized at point $\mathbf{P}_{i,j}$ by using a finite difference scheme is shown in Eq. (8).

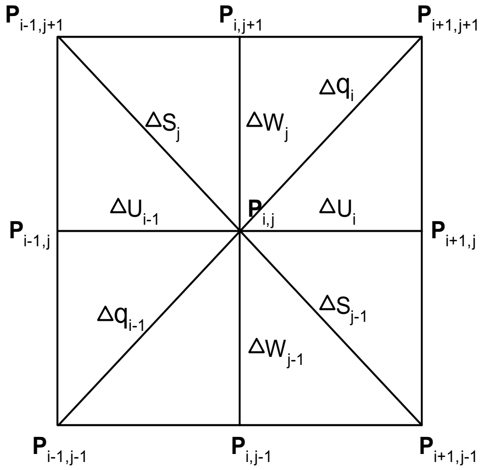


Figure 1: A eight-connected grid map

$$
\begin{aligned}
(\nabla y)_{\mathbf{P}_{i,j}} &= \frac{2y_{i+1,j}}{\Delta u_i(\Delta u_{i-1}+\Delta u_i)} + \frac{2y_{i-1,j}}{\Delta u_{i-1}(\Delta u_{i-1}+\Delta u_i)} \\
&+ \frac{2y_{i,j+1}}{\Delta w_j(\Delta w_{j-1}+\Delta w_j)} + \frac{2y_{i,j-1}}{\Delta w_{j-1}(\Delta w_{j-1}+\Delta w_j)} \\
&+ \frac{2y_{i+1,j+1}}{\Delta q_i(\Delta q_{i-1}+\Delta q_i)} + \frac{2y_{i-1,j-1}}{\Delta q_{i-1}(\Delta q_{i-1}+\Delta q_i)} + \\
&\frac{2y_{i-1,j+1}}{\Delta s_j(\Delta s_{j-1}+\Delta s_j)} + \frac{2y_{i+1,j-1}}{\Delta s_{j-1}(\Delta s_{j-1}+\Delta s_j)} \\
&- \frac{2y_{i,j}}{\Delta v_{i-1}\Delta v_i} - \frac{2y_{i,j}}{\Delta w_{j-1}\Delta w_j} - \frac{2y_{ij}}{\Delta q_{i-1}\Delta q_i} - \frac{2y_{ij}}{\Delta s_{j-1}\Delta s_j}
\end{aligned} \tag{8}
$$

where $\Delta u_{i-1} = \left\|\overrightarrow{\mathbf{P}_{i-1,j}\mathbf{P}_{i,j}}\right\|$, $\Delta u_i = \left\|\overrightarrow{\mathbf{P}_{i,j}\mathbf{P}_{i+1,j}}\right\|$, $\Delta w_{j-1} = \left\|\overrightarrow{\mathbf{P}_{i,j-1}\mathbf{P}_{i,j}}\right\|$, $\Delta w_j = \left\|\overrightarrow{\mathbf{P}_{i,j}\mathbf{P}_{i,j+1}}\right\|$, $\Delta q_{i-1} = \left\|\overrightarrow{\mathbf{P}_{i-1,j-1}\mathbf{P}_{i,j}}\right\|$, $\Delta q_i = \left\|\overrightarrow{\mathbf{P}_{i,j}\mathbf{P}_{i+1,j+1}}\right\|$, $\Delta s_{j-1} = \left\|\overrightarrow{\mathbf{P}_{i+1,j-1}\mathbf{P}_{i,j}}\right\|$ and $\Delta s_j = \left\|\overrightarrow{\mathbf{P}_{i,j}\mathbf{P}_{i-1,j+1}}\right\|$. ∇ represents Laplace operator, $y_{i,j}$ is the activity at point $\mathbf{P}_{i,j}$, and $\left\|\overrightarrow{\mathbf{P}_{i-1,j}\mathbf{P}_{i,j}}\right\|$ and other similar terms represent the magnitudes of vector $\overrightarrow{\mathbf{P}_{i-1,j}\mathbf{P}_{i,j}}$ and other similar vectors.

From Eq. (8), it can be easily seen that the local interaction of Laplace operator has the property that the sum of the weights at each node and its neighbourhood is zero.

Thus, the local connectivity of the neuron at point $\mathbf{P}_{i,j}$ can be obtained as Eq. (9).

$$
A_{ij} = \begin{pmatrix}
\frac{2h}{\Delta s_j(\Delta s_{j-1}+\Delta s_j)} & \frac{2h}{\Delta w_j(\Delta w_{j-1}+\Delta w_j)} & \frac{2h}{\Delta q_i(\Delta q_{i-1}+\Delta q_i)} \\
\frac{2h}{\Delta u_{i-1}(\Delta u_{i-1}+\Delta u_i)} & & \frac{2h}{\Delta u_i(\Delta u_{i-1}+\Delta u_i)} \\
\frac{2h}{\Delta q_{i-1}(\Delta q_{i-1}+\Delta q_i)} & \frac{2h}{\Delta w_{j-1}(\Delta w_{j-1}+\Delta w_j)} & \frac{2h}{\Delta s_{j-1}(\Delta s_{j-1}+\Delta s_j)}
\end{pmatrix} \tag{9}
$$

where *h* is a positive constant.

It is not difficult to see that the property of the local interaction of Laplace operator is also inherited by the locally connected neurons, i.e., the sum of the connection weights for each neuron and its neighbouring neurons is zero.

B. Stability Analysis

The generalized Hopfield-type neural network is globally stable, if the connected weights are symmetric [16]. However, in the proposed neural network, it can be seen from Eq. (9) that the local connectivity of neurons is asymmetric. Therefore, it is necessary to analyze the stability of the proposed neural network.

One of the most effective techniques for analyzing the convergence properties of dynamic neural networks is Lyapunov's method. The stability and convergence of the proposed neural network can be strictly proved by using the Lyapunov stability theory.

The Lyapunov's function of the proposed neural network is:

$$
L = \sum_{i=1}^{n}\left[\frac{y_i^2}{2R_i} - \left(\sum_{j\in N_r(i)} A_{ij}y_j + I_i\right)y_i\right] \tag{10}
$$

By differentiating Eq. (10), the following equation may be obtained:

$$
\frac{dL}{dy_i} = \frac{y_i}{R_i} - \sum_{j\in N_r(i)} A_{ij}y_j - I_i \tag{11}
$$

Thus, the differential of Eq. (11) with respect to time may be written as:

$$
\frac{dL}{dt} = \sum_{i=1}^{n}\frac{dL(y)}{dy_i}\frac{dy_i}{dt} = -\sum_{i=1}^{n}\left(\frac{dy_i}{dt}\right)^2 \le 0 \tag{12}
$$

Therefore, the proposed neural network system is stable. The dynamics of the neural network are guaranteed to converge to an equilibrium state of the system. The rigorous proof of the Lyapunov stability theory can be found in [16, 20].

C. Robot Movement

The proposed neural network model guarantees that the activity of the target can propagate to the whole state space through local neural connections, while the activities of the obstacles remain locally only. Therefore, the target globally influences the whole state space to attract the robot, while the obstacles have only local effect to avoid collisions. In addition, the activity propagation from the target is blocked when it hits the obstacles. Such a property is very important for a maze-solving type of problem.

The positions of the target and obstacles may vary with time. As shown in Eq. (7), the activity of each neuron dynamically changes according to the input, the local connectivity of the neuron and the activities of the neuron's neighbouring neurons. As shown in Eqs. (3) and (4), the target corresponds to the input, and the obstacles have a fixed activity and the position change of an obstacle corresponds to the activity change of a neuron. Therefore, each neuron only responds to the real-time changes of the target and obstacles, and there is no prior knowledge of the varying environment in the proposal model. The real-time path is generated from the dynamic activity landscape by a steepest gradient ascent rule. For a given present position $\mathbf{P}_i$ in the state space of the neural network (i.e. a position in the Cartesian workspace or a configuration in the robot manipulator joint space), assuming the corresponding neuron is e(i), the next position $\mathbf{P}_n$ is obtained by:

$$\mathbf{P}_n \Leftarrow y_n = \max\{y_j, e(j) \in N_r(i)\} \tag{13}$$

After the present position gets to its next position, the next position becomes a new present position. If the found next position is the same as the present position, the robot remains there without any movement. The present position adaptively changes according to the varying environment.

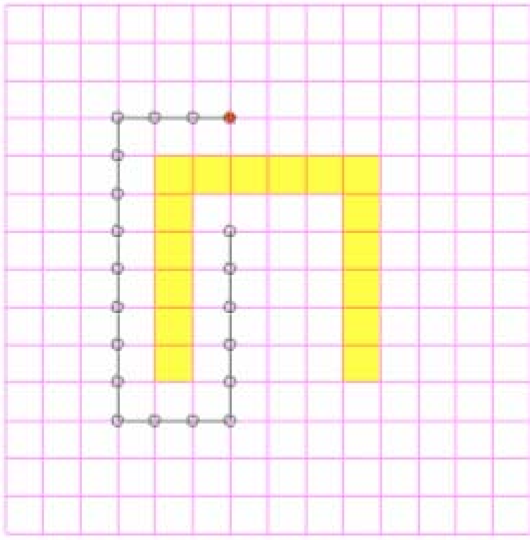
The dynamic activity of the topologically organized neural network is used to determine where the next robot position will be. However, when the next robot position is generated, it is determined by the robot moving speed. In a static environment, the neural network activity will reach a steady state. In most cases, the robot reaches the target earlier before the neural network activity reaches the steady state. When a robot is in a changing environment, the neural network activity will never reach the steady state. With the evolution of the neural network activity, the target always keeps remaining at the top and the obstacles always keep remaining at the bottom. The robot keeps moving toward the target with obstacle avoidance according to the steepest gradient ascent rule till the designated objective is obtained.

Since the local connectivity of neurons is harmonic and the target activity is propagated among neurons in the manner of physical heat conduction, the proposed methodology can generate optimal robot paths with obstacle avoidance in both static and dynamic environments. The generated path in a static environment is globally optimal in the sense of a shortest path from the starting position to the target, and the robot always reaches the target along a shortest path. In a dynamic environment, the optimality is in the sense that the robot travels a continuous, smooth route to the target.

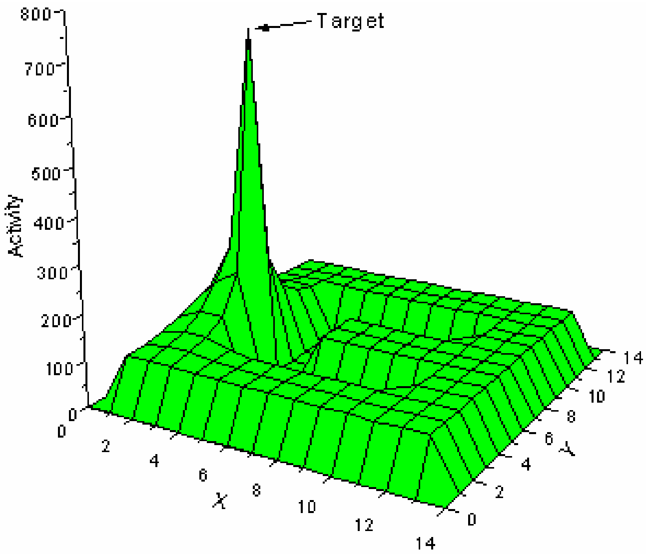
IV. SIMULATION STUDIES AND DISSCUSSIONS

A. Path Planning in Static Environments

The obstacle avoidance for U-shaped obstacles has been achieved by the proposed methodology. Potential field based methods and other strictly local avoidance schemes cannot handle this type of problems. Fig. 2 shows an example for robot path planning with avoidance of concave U-shaped obstacles. The neural network has 15×15 topologically organized neurons. A set of concave U-shaped obstacles are represented by yellow solid squares. The model parameters $E = 1$ and $h = 1$. The generated trajectory is shown in Fig. 2 by black hollow circles connected with black lines. The target is represented as a solid red circle, and the starting position is indicated by the first black hollow circle, which is the farthest from the target along the black line. It can be seen from Fig. 2(a) that that the generated trajectory is a continuous, smooth route from the start point to the target with obstacle avoidance. The stable activity landscape of the neural network is shown in Fig. 2(b), where the peak is at the target location and the bottom is at the obstacle location.



(a) Generated robot path



(b) Stable CNN activity landscape

Figure 2: Robot path planning with avoidance of a set of concave U-shaped obstacles

The solution to the maze-solving type problem can be treated as a special case of the path planning problem in a 2D workspace, in which a mobile robot reaches the target from a given starting position with obstacle avoidance. Fig. 3 shows an example of the well-known beam robot competition micro mouse maze, where yellow solid squares represents a typical maze. The neural network has 30×30 neurons, and has the same model parameters as the neural network model shown in Fig. 2. The generated globally optimal solution is illustrated in Fig. 3, where the trajectory of the robot is indicated by black hollow circles connected with black lines. It can be seen from these two examples that the proposed methodology does not suffer from undesired local minima, i.e., the robot will not be trapped in the situation with concave U-shaped obstacles or with complex maze-like obstacles.

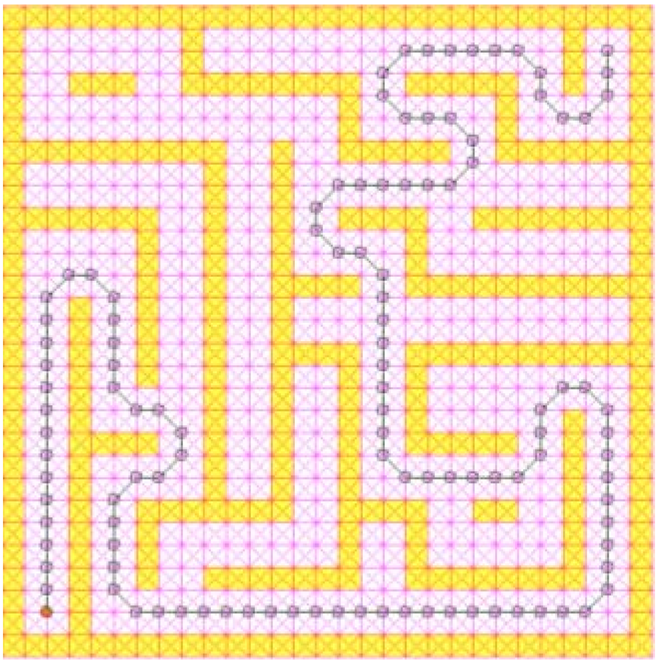


Figure 3. Solution to the maze-solving problem

B. Path Planning in Changing Environments

Real-time path planning in an environment with moving obstacles has been achieved by the proposed methodology. Fig. 4 shows an example for real-time path generation with avoidance of moving obstacles. The neural network has 40×40 neurons, and has the same model parameters as the neural network model in Fig. 2. The origin is at the up left corner. The robot starts from position (19, 33) and moves at a speed of 25 blocks/min. The static obstacles displayed by solid yellow boxes formulate two possible channels for the robot to reach the target. There also are eleven moving obstacles displayed by solid blue boxes. The moving obstacles are initially located at positions from (9, 14) to (19, 15) to completely obstruct the left channel. They move towards the right channel at a speed of 20 blocks/min, and finally stop at positions from (19, 14) to (30, 15) to completely block the right channel and free the other. The robot initially moves to the target through the right channel since the left channel is blocked. With the proceeding of the robot moving, the moving obstacles are moving to block the right channel and leave the left channel open. The right channel is completely blocked before the robot passes through the right channel, and thus the robot has to turn back to catch the target through the left channel. The generated robot path is shown in Fig. 4. The generated robot path is displayed by black hollow circles connected with black lines, where the robot reverses and reaches the target via the left channel due to a rapid change of the neural network activities and the direction of the neural network activity gradient.

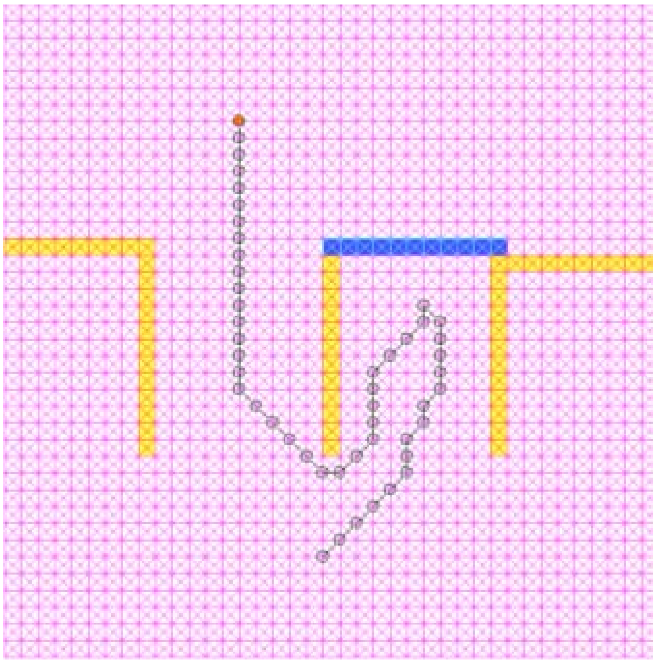


Figure 4. Path generation for tracking a static target with avoidance of moving-and-stopping obstacles

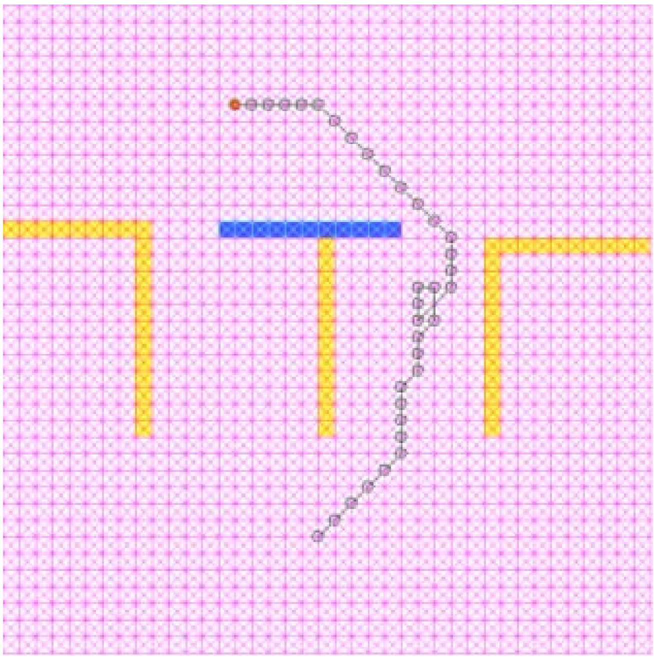


Figure 5. Path generation for tracking a static target with avoidance of continuously moving obstacles

A more complex case is that the presence of continuously moving obstacles. Fig. 5 shows an example for real-time path generation to catch a target with avoidance of continuously moving obstacles. The neural network model is the same as that in Fig. 4 except that the obstacles are continuously moving back and forth between positions from (9, 14) to (19, 15) and positions from (19, 14) to (30, 15). Compared with Fig. 4, the robot is turning back and moving away from the target since the right channel is being closed, and then is moving forward since the right channel is being opened again. Therefore, there is a loop generated in the robot path.

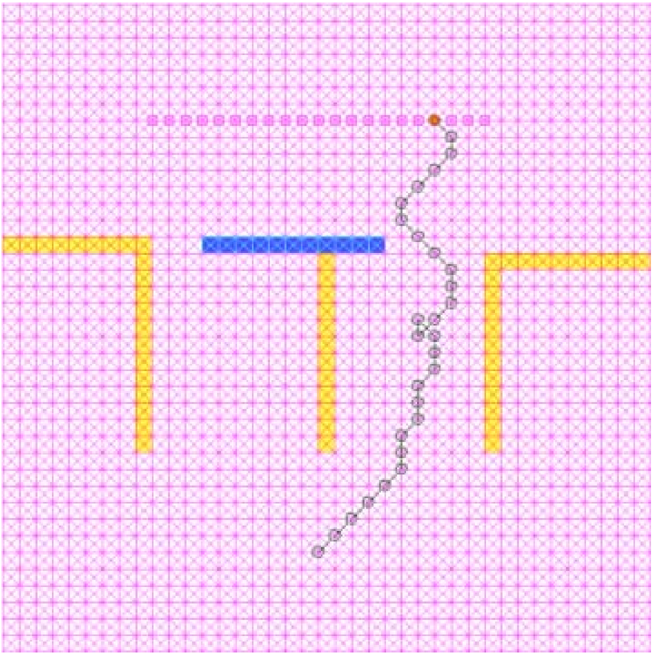


Figure 6. Path generation for tracking a moving target with avoidance of moving obstacles

Real-time path planning with both a moving target and moving obstacles has also been achieved by the proposed methodology. Fig. 6 shows an example for tacking a moving target with avoidance of moving obstacles. The neural network model is the same as that in Fig. 5 except that the moving target. The target starts from position (9, 7) and moves back and forth between position (9, 7) and position (29, 7) at the same speed (20 blocks/min) of the moving obstacles. The target moving path is displayed by solid pink circles. The real-time generated robot path is displayed in Fig. 6. There still is a turning loop in the generated path since the moving obstacles close and then open the right channel. However, the loop becomes smaller since the target is moving closely to the robot. In addition, after passing through the right channel, the robot moves in a zigzag way to catch the moving target.

V. CONCLUSIONS

This paper presents a new neural dynamics based methodology for real-time robot path generation with obstacle avoidance. The contribution of the paper is that the elegant properties of harmonic functions are incorporated in a neural system for real-time path generation in both static and dynamic environments. An improved Hopfield-type neural model is established to propagate the target activity among neurons in the manner of physical heat conduction, which guarantees that the target and obstacles remain at the peak and the bottom in the activity landscape, respectively. The real-time collision-free robot motion is planned through the dynamic neural network activity without any prior knowledge of the dynamic environment, without explicitly searching over the global free workspace or searching collision paths, and without any learning procedures. Examples are presented to demonstrate the effectiveness and efficiency of the proposed methodology.

ACKNOWLEDGMENTS

This research is supported by Australian Research Council (ARC).

REFERENCES

[1] A. Zelinsky, Using path transforms to guide the search for findpath in 2D, International Journal of Robotics Research, vol. 13, no. 4, 1994, pp315-325

[2] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, International Journal of Robotics Research, vol. 5, no. 1, 1986, pp90-98.

[3] C. Seshadri and A. Ghosh, Optimum path planning for robot manipulators amid static and dynamic obstacles, IEEE Trans. Systems, Man, and Cybernetics, vol. 23, 1993, pp576-584

[4] Z. X. Li and T. D. Bui, Robot path planning using fluid model, Journal of Intelligent and Robotic Systems, vol. 21, no. 1, 1998, pp29–50

[5] C. J. Ong and E. G. Gilbert, Robot path planning with penetration growth distance, Journal of Robotic Systems, vol. 15, no. 2, 1998, pp57-74

[6] P. Svestka and M. H. Overmars, Motion planning for carlike robots using a probabilistic approach, International Journal of Robotics Research, vol. 16, no. 2, 1997, pp119-145

[7] E. Zalama, P. Gaudiano, and J. L. Coronado, A real-time, unsupervised neural network for the low-level control of a mobile robot in a nonstationary environment, Neural Networks, vol. 8, 1995, pp103-123

[8] E. Zalama, J. Gomez, M. Paul and J. R. Peran, Adaptive behavior navigation of a mobile robot, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 32, no. 1, 2002, pp160-169

[9] M. Quoy, S. Moga and P. Gaussier, Dynamical Neural Networks for Planning and Low-Level Robot Control, IEEE Transactions on Systems, Man, and Cybernectics-Part A: Systems and Humans, vol. 33, no. 4, 2003, pp523-532

[10] A. Willms and S. X. Yang, An Efficient Dynamic System for Real-Time Robot-Path Planning, IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 36, no. 4, 2006

[11] R. Glasius, A. Komoda, and S. C. A. M. Gielen, Neural network dynamics for path planning and obstacle avoidance, Neural Networks, vol. 8, no. 1, 1995, pp125-133

[12] S. X. Yang and M. Meng, An efficient neural network approach to dynamic robot motion planning, Neural Networks, vol. 13, no. 2, 2000, pp143-148

[13] S. X. Yang and M. Meng, Neural network approaches to dynamic collision-free robot trajectory generation, IEEE Transactions on Systems, Man, Cybernetics-Part B: Cybernetics, vol. 31, no. 3, 2001, pp302-318

[14] S. X. Yang and M. Q.-H. Meng, Real-time collision-free motion planning of mobile robots using neural dynamics based approaches, IEEE Transactions on Neural Networks, vol. 14, no. 6, 2003, pp1541-1552

[15] S. X. Yang and C. Luo, A neural network approach to complete coverage path planning, IEEE Transactions Systems, Man, Cybernetics-Part B: Cybernetics, vol. 34, no. 1, 2004, pp718-725

[16] J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, Proceedings of the National Academy of Sciences of the United States of America, 1984, vol. 81, pp3088-3092

[17] R. Glasius, A. Komoda and S. Gielen, A biologically inspired neural net for trajectory formation and obstacle avoidance, Biological Cybernetics, vol. 74, no. 6, 1996, pp511-520

[18] G. E. Karniadakis and R. M. Kirby II, Parallel scientific computing in C++ and MPI: a seamless approach to parallel algorithms and their implementation, New York: Cambridge University Press, 2003

[19] H.K. Versteeg and W. Malalasekera, An introduction to computational fluid dynamics: the finite volume method, Harlow, Essex, England: Longman Scientific & Technical, 1995

[20] M. A. Cohen and S. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, IEEE Transaction on Systems, Man, and Cybernetics, vol. SMC-13, 1983, pp815-821