2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

An Effective Vector-driven Biologically-motivated Neural Network Algorithm to Real-time Autonomous Robot Navigation

Chaomin Luo[1], Simon X. Yang[2], Mohan Krishnan[1], and Mark Paulik[1]

***Abstract*— A novel biologically-motivated neural networks approach associated with developed vector-driven autonomous robot navigation is proposed in this paper. The biologically-motivated neural networks (BNN) algorithm is employed to guide an autonomous robot to reach goal with obstacle avoidance motivated by Grossberg's model for a biological neural system. As the robot plans its trajectory toward the goal, unreasonable path will be inevitably planned. A vector-based guidance paradigm is developed for guidance of the robot locally so as to plan more reasonable trajectories. In addition, square cell map representations are proposed for real-time autonomous robot navigation. The BNN based scheme demonstrates that the algorithms avoid the issue of local minima in path planning. In this paper, both simulation and comparison studies of an autonomous robot navigation demonstrate that the proposed model is capable of planning more reasonable and shorter collision-free paths in non-stationary and unstructured environments compared with other approaches.**

I. INTRODUCTION

Real-time collision-free navigation of an autonomous robot in a non-stationary and unstructured environment is a crucial issue in robotics. A number of methods for real-time collision-free robot navigation are based on neural networks, genetic algorithms, Ant Colony Optimization, and fuzzy logic models have been developed to perform navigation of an autonomous robot. For energy efficiency of real-time collision-free autonomous robot navigation, trajectory length planed and robot turning numbers should be reasonable and minimized.

There are plenty of studies on the real-time collision-free navigation for robotic systems using various approaches (e.g., [2], [3], [4], [5], [6], [7], [8]). Bin and Xiong [2] proposed a real-time collision-free robot trajectory formation in a non-stationary environment based on a modified neural network model without suffering from undesired local minima. However, trajectories generated by their model are unreasonable and thus robot traverses longer distance to reach the goal. Hu and Yang [3] proposed a knowledge based problem-specific genetic algorithm (GA) method for path planning of a mobile robot that outperforms standard GA in terms of real-time robot navigation. Yang and Meng [4] proposed a neural-dynamics-based path planning model, which is a successful approach to plan reasonable, safe and short paths for robot navigation. Glasius et al. [5] developed a neural network model for real-time collision-free robot path planning with reasonable trajectory generated. However, the model does not perform appropriately in a dynamically changing environments. Afsar et al. [6] proposed genetic algorithms (GA) for optimal path planning integrated with morphological image processing of the terrain. A specialized morphological preprocessing technique is employed to improve the computational efficiency of GAs. Peng and Xu [7] suggested a fuzzy logic and Bayesian network based method associated with a decision-making method for unmanned vehicle path planning. Recently, Xing et al. [8] described the applications of Ant Colony Optimization Systems (ACO) to intelligent vehicle navigation. It shows that ACO has been devised in vehicle path planning as it is more robust and faster in achieving the globally optimal solution.

In this paper, an effective vector-driven biologically-motivated neural network (BNN) model for real-time autonomous robot navigation is proposed. The state space of the discretely and topologically organized neural network is the Cartesian workspace of an autonomous robot. The dynamics of each neuron is characterized by a dynamical equation derived from Grossberg's model from a biological neural system [1], [9]. There are only local lateral connections among neurons. The neural activity can autonomously propagate by identifying the inter-neurons structure and detecting the strength of neural activities among neighboring inter-neurons by using vector-driven navigation approach. The effectiveness and efficiency of the proposed BNN integrated with the vector-driven approach is validated by *both* comparison and simulation studies, in which comparison studies demonstrate that the proposed model is capable of planning more reasonable and shorter collision-free paths in non-stationary and unstructured environments in comparison with other models. The major contribution addressed in this paper is that a novel biologically-motivated neural network model is derived from a biological neural system based on its electrical circuit elements and it is first applied to real-time autonomous robot navigation.

II. BIO-MOTIVATED NEURAL NETWORK ALGORITHM

A computational model for a biological neural system using electrical circuit elements was first proposed by Grossberg (1982) [1]. The model of a gradient descent based neural system is represented using electrical components shown in Fig. 1 motivated from a biological neural system [1], [9]. A neural system in this paper consists of one or more neurons. Each neuron is a simple processing device that has inputs (dendrites) and outputs (axons). Therefore, a neural system

[1]C. Luo, M. Krishnan and M. Paulik are with The Advanced Mobility Laboratory (AML), Department of Electrical and Computer Engineering, University of Detroit Mercy, Michigan, USA luoch@udmercy.edu;
[2]S. X. Yang is with The School of Engineering, University of Guelph, Canada syang@uoguelph.ca.

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

is modeled as a circuit shown in Figure 1, which is made up of n neurons, mapping its input voltage v_i into the output u_i through the activation function $f(v_i)$.

The circuit representation of a Grossberg network delivers mathematical description of the neural networks [1]. The conductances g_1, g_2, ..., and g_n are synaptic weights between the ith neuron and ground, which represents the nonzero input conductance of the ith neuron. In the Hopfield-type neural network, conductance w_{ij} connects the input of ith neuron to output of jth neuron.

Each amplifier has an input capacitance, C_i, and an input conductance (resistance), g_i. Each amplifier is supplied a constant current by the external signals.

Inhibitory signals are simulated by inverting amplifiers, which establish inhibitory connection. In the circuit,the time constant of the exponential and synaptic weights are approximated by a set of resistors. Description of electrical circuits results in the equation to express the neuron activation.

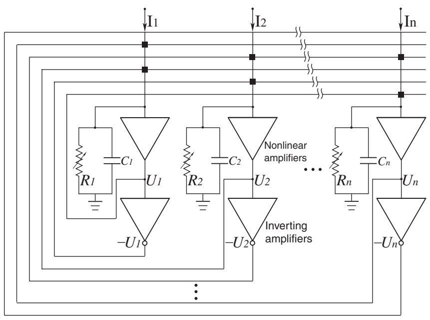

Fig. 1. The electrical component representation of a neural system.

The potential v_i is given by the following equation.

$$C_i \frac{dv_i}{dt} = -\,(\sum_{j=1}^{n} w_{ij} + g_i)v_i + \sum_{j=1}^{n} w_{ij}u_j + I_i \qquad (1)$$

The total current entering through the capacitance C_i is sum of $n-1$ currents of value $(u_j - v_i)w_{ij}$ (for j = 1,2,...,n), the current I_i, and the current $-g_i v_i$. The total conductance connected to the input of neuron i, G_i, and and its resistance R_i is defined as Equation (2) in Figure 1.

$$\frac{1}{R_i} = -\sum_{j=1}^{n} w_{ij} + g_i \qquad (2)$$

The weights w_{ij} transform the voltage $u_j - v_i$ into currents. Therefore, the potential v_i is given by Equation (3).

$$C_i \frac{dv_i(t)}{dt} = -\,\frac{1}{R_i} v_i(t) + \sum_{j=1}^{n} w_{ij}u_j(t) + I_i \qquad (3)$$

The dynamics with memory of a neuron are modeled by the capacitive term $C_i(dv_i(t)/dt)$ on the Equation (4).

$$C_i \frac{dv_i(t)}{dt} + \frac{1}{R_i} v_i(t) \qquad (4)$$

Therefore, Equation (3) can be reformed as the following equations:

$$\frac{dv_i(t)}{dt} = -\,\frac{1}{R_i C_i} v_i(t) + \frac{1}{C_i} \sum_{j=1}^{n} w_{ij}u_j(t) + I_i \qquad (5)$$

where the time constant $\tau_i = R_i C_i$ of the ith neuron in Equation (5) is same as the jth neuron.

$$\frac{dv_i(t)}{dt} = -\,\frac{1}{R_i C_i} v_i(t) + S_i\beta \sum_{j=1}^{n} w_{ij}u_j(t) + I_i \qquad (6)$$

where $S_i\beta = C_i$. S_i and β are positive constants, and j = 1,2,...,n. The biologically-motivated neural network is a particular kind of function motivated by neurons in the brain. The proposed discretely and topologically organized model is expressed in 2D Cartesian workspace $\mathcal{W}$ of the cleaning robots. The location of the ith neuron the state space $\mathcal{S}$ of the neural network, denoted by a vector $q_i \in R^2$, *uniquely* represents a location in $\mathcal{W}$. In the proposed model, the excitatory input results from the goal location and the lateral neural connections, while the inhibitory input results from the obstacles only. Each neuron has local lateral connections to its adjacent neurons that constitute a subset $\mathcal{R}_i$ in $\mathcal{S}$. The subset $\mathcal{R}_i$ is called the receptive field of the ith neuron in neurophysiology. The neuron responds only to the stimulus within its receptive field. Thus, the dynamics of the ith neuron in the biologically-motivated neuron network is characterized by the following equation as

$$\frac{dv_i(t)}{dt} = -\,A + S_i\beta \sum_{j=1}^{k} w_{ij}u_j(t) + I_i \qquad (7)$$

where $S_i\beta = C_i$, $A = 1/(R_i C_i) > 0$ is a positive constant representing feedback gain, and k is the total number of neural connections of the ith neuron to its adjacent neurons within the receptive field $\mathcal{R}_i$. v_i is the neural activity. S_i and β are positive constants. S_i is given as

$$S_i = \begin{cases} 0, & \text{obstalce} \\ 1, & \text{otherwise} \end{cases}, \qquad (8)$$

Parameter I_i is the external input to the ith neuron is defined as

$$I_i = \begin{cases} E, & \text{if it is the goal} \\ 0, & \text{otherwise} \end{cases}, \qquad (9)$$

where $E \gg \beta \gg 0$ is a large positive constant. The connection weight w_{ij} from the ith neuron to the jth neuron is given by

$$w_{ij} = w_{ji} = (k\beta/A)^{\sqrt{2}-1} \qquad (10)$$

where, again, k is the number of neural connections of the ith neuron to its neighboring neurons within the receptive field $\mathcal{R}_i$. Each neuron has only local lateral connections in a small region $[0, r_0]$. It is obvious that the weight w_{ij} is symmetric, i.e., $w_{ij} = w_{ji}$. The proposed network characterized by Equation (7) guarantees that the positive neural activity can propagate to all the state space, but the negative activity only stays locally. Therefore, the goal globally attracts the robot, while the obstacles only locally avoid the collision. Fig. 2) illustrates the architecture of a 2D neural network with adjacent neurons with regard to the central neuron $C(p, q)$. The activity landscape of the neural network dynamically changes due to the varying external inputs from the unclean areas and obstacles and the internal activity propagation among neurons. The optimal robot path is planned from the dynamic activity landscape, and the previous robot location to avoid least navigation direction changes. The robot will move to the neuron with maximal neural activity or the neuron with the second largest neural activity, which is addressed in the following sections of Navigation and Vector-driven Algorithms (Algorithm 2 and Algorithm 3 in the next sections).

After the current location reaches its next location, the next location becomes a new current location. The current robot location *adaptively* changes according to the varying environment.

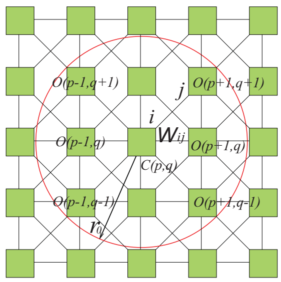

Fig. 2. The architecture of a 2D neural network with adjacent neurons with regard to the central neuron C.

A. The Navigation Algorithm

In order to guide the robot autonomously to traverse in the dynamical environment with obstacle avoidance, the proposed model consists of an initialization phase, a navigation phase and a vector-driven phase that will be presented in the next section. In these algorithms, it is necessary to define a flag, denoted by $f(p,q)$, for a neuron at Point (p,q) to indicate its status as visited, unknown (unvisited), obstacle, or goal.

$$f(k,l) = \begin{cases} 0 & \text{if it is unknown/unvisited} \\ 1 & \text{if it is visited} \\ 2 & \text{if it is obstacle} \end{cases} . \tag{11}$$

Initially, the autonomous robot has no *priori* knowledge of any obstacles in the environment, except the whole workspace dimension. The model algorithm is composed of following three phases: initialization phase, navigation phase and vector-driven guidance phase. The Initialization Phase: The initialization algorithm aims to initialize the starting point of the robot, to set all the neural activities as zeros, etc., which is given in Algorithm 1. The Navigation Phase: The goal globally attracts the robot in the whole state space through neural activity propagation, while the obstacles only exercise a local influence on a small region to prevent from collisions. The navigation algorithm for the autonomous robot is given in Algorithm 2. In this algorithm, for a robot, once it traverses from the current point to its next point, the next point becomes a new current point, and the previous point is marked as visited (see Algorithm 2). The definition of "next point" is in the sense that the next point is selected based on the neural activity (see Equation (7) and Algorithm 2). The Vector-driven Guidance Phase: The vector-driven guidance algorithm aims to modify the trajectory generated by the neural-dynamic-based navigation model so as to obtain more reasable path, which is indeed a developed vector-based method. It is presented in the next section.

The following notations will be utilized to describe the proposed three algorithms. N_x and N_y are the discretized size of the Cartesian workspace.

$\mathcal{O}(p,q)$	The set of the discretized workspace, $\{(p,q), 1 \le p \le N_x, 1 \le q \le N_y\}$.
$O(p,q)$	The unknown point (p,q), $1 \le p \le N_x, 1 \le q \le N_y$.
$\dot{O}(p,q)$	The visited point (p,q), $1 \le p \le N_x, 1 \le q \le N_y$.
$\ddot{O}(p,q)$	The point (p,q) with the second largest neural activity $1 \le p \le N_x, 1 \le q \le N_y$.
$x(p,q)$	Neural activity at unknown point (p,q).
(p_c,q_c)	The current point (p_c,q_c).
(p_n,q_n)	The next point (p_n,q_n) with the maximal activity.
$x_m(p_n,q_n)$	The maximal neural activity at Point (p_n,q_n).
$I(p,q)$	External input to neuron $O(p,q)$.

The neighborhood of a central neuron $C(p,q)$ is composed of some neighboring neurons that enclose the central neuron $C(p,q)$. The neighborhood of a central neuron $C(p,q)$ in the Grossberg neural network is defined by $\mathcal{O}_r(p,q) = \{N(m,n) | \max\{|m-p|, |n-q|\} \le r, 1 \le m \le N_x, 1 \le n \le N_y\}$, where r is the number of circles enclosing the central neuron (see Fig. 2). The position of an adjacent neuron $O(m,n)$ near to the central neuron $C(p,q)$ has the following property: $m \in \{p-1, p, p+1\}$ and $n \in \{q-1, q, q+1\}$ illustrated in Fig. 2, in which the architecture of a 2D neural network with adjacent neurons with regard to the central neuron $C(p,q)$ is shown. The central neuron $C(p,q)$ illustrated by dark shaded square has eight neighboring neurons if r=1. In this model, the central neuron locally connects with closest neurons, i.e., r=1.

Algorithm 1 Initialization Algorithm

1) set $p_c := p_0$; set $q_c := q_0$, where (p_0, q_0) is a starting point
   // Set starting point to a current neuron
2) set $f(p,q) := 0$; set $I(p,q) := 0$, $\forall\ 1 \le p \le N_x, 1 \le q \le N_y$
   // Set all areas as unvisited except obstacles
3) set $x(p,q) := 0$, $\forall\ 1 \le p \le N_x, 1 \le q \le N_y$
   // Set all neural activities as zero

Algorithm 2 Navigation Algorithm

1) Compute neural activity by Equation (7)
2) $\mathcal{O}_r(p,q) = \{N(m,n)|\ m \in (p-1, p, p+1)$ and $n \in (q-1, l, l+1)\}$
   // scan unknown adjacent neurons $(p_n, q_n) =$

$\underset{m,n}{\operatorname{argmax}} x(m,n) \in \{\mathcal{O}_r | m \in (p-1, p, p+1)$ and $n \in (q-1, q, q+1)\}$
// find the next adjacent neuron with the maximal neural activity
3) set $p_c := p_n$; set $q_c := q_n$
// Set current neuron to neighboring neuron
4) if $\exists (p,q) \in \mathcal{O}_r(m,n)$, s.t. $x(p,q) \leq x(p_c, q_c)$
// if adjacent neural activity $\leq$ current neural activity
then
   - set $I(p,q) := 0$
   - set $\dot{O}(p,q) := O(p,q)$
   - flag $f(p,q) := 1$
   // Mark it as visited and external input as zero

end if
5) if $\exists (p,q) \in \mathcal{O}_r(m,n)$, s.t. $x(p,q) \leq x(p_n, q_n)$
// if adjacent neural activity $\leq$ maximal neural activity
then
   - set $\ddot{O}(p,q) := O(p,q)$
   - flag $f(p,q) := 3$
   // Mark it as the second largest neural activity

end if
6) if $\exists (g,h) \in \mathcal{O}_r(m,n)$, $\forall\ g \in \{p-1, p, p+1\}$ and $h \in \{q-1, q, q+1\}$, s.t. $f(g,h) = 1$
// if adjacent neurons are all visited
then
   - set $I(g,h) := 0$
   - set $\dot{O}(g,h) := O(p,q)$
   // Mark them as visited and external input as zero

end if
7) go to 1).

The computational complexity depends linearly on the state space size of the neural network, which is proportional to the workspace size. The number of neurons required is equal to $M = N_x \times N_y$. The workspace is discretized with dimension of width N_x and height N_y. If the workspace is an $N \times N$ square in shape, there are N^2 neurons and each neuron has at most eight local neural connections. The computational complexity of the proposed algorithm is $O(N^2)$.

B. The Vector-driven Algorithm

In this section, a vector-based approach is developed to assist in navigation to plan more reasonable path. A vector is a specific mathematical structure, which result primarily from its ability to represent magnitude and direction simultaneously. In Fig. 3, magnitude and direction of a vector associated with the proposed BNN model are utilized to guide the robot to plan a more reasonable and shorter trajectory. There are two set of adjacent neural architectures populated in the workspace with starting point S and goal G in Fig. 3. A vector $mathcalV$ containing magnitude and direction is defined as

$$\mathcal{V} \triangleq \{|\overrightarrow{V}|, \varphi\} \tag{12}$$

, where $|\overrightarrow{V}|$ and φ represent magnitude and direction of the vector, respectively.

According to the Grossberg's BNN model [1], the robot is guided to move to the neuron with the maximal neural activity based on Algorithm 2 described previously. Once a robot reaches a point/neuron, all the neural activities of adjacent neurons are investigated. The vector-driven navigation strategy is introduced to investigate the two neural activities: one is the neuron with the maximal neural activity; the other is the neuron with the second largest neural activity. The vectors of two neurons are compared to adjust which point the robot should move to. If the angle and distance to the goal represented by the defined vector indicate the position with the second largest neural activity is better than the the position with the maximal neural activity, then the robot moves the position with the second largest neural activity.

In the first neighboring neural architecture in Fig. 3, the positions (neurons) with the maximal neural activity and the second largest neural activity are denotes by ${N_1}^s$ and ${N_1}^m$, respectively. d_1 represents the distance from the current neuron to the goal G employed to guide the robot move to the goal effectively. ${\theta_1}^L$ and ${\theta_1}^R$ represent the angles to judge which position of neurons with the maximal neural activity or the second largest neural activity should be selected to move.

Algorithm 3 Vector-driven Algorithm

1) Compute neural activity by Equation (7)
2) $\mathcal{O}_r(p,q) = \{N(m,n)|\ m \in (p-1, p, p+1)$ and $n \in (q-1, l, l+1)\}$
// scan unknown adjacent neurons $(p_n, q_n) =$
$\underset{m,n}{\operatorname{argmax}} x(m,n) \in \{\mathcal{O}_r | m \in (p-1, p, p+1)$ and $n \in (q-1, q, q+1)\}$
// find the next adjacent neuron with the maximal neural activity
3) if $\exists (p,q) \in \mathcal{O}_r(m,n)$, s.t. $x(p,q) \leq x(p_n, q_n)$
// if adjacent neural activity $\leq$ maximal neural activity
then
   - set $\ddot{O}(p,q) := O(p,q)$
   - flag $f(p,q) := 3$
   // Mark it as the second largest neural activity

end if
4) if $\exists (p,q) \in \mathcal{O}_r(m,n)$, s.t. ${\theta_1}^R \leq {\theta_1}^L$
// The angles make better judgement
then
   - set $\ddot{O}(p,q) := O(p,q)$ // move to point with the second largest neural activity

end if
5) go to 1).

III. COMPARISON AND SIMULATION STUDIES

The proposed vector-guided bio-motivated neural network approach is applied for an autonomous robot navigation in room-like, and unstructured workspace. In this section, this model with *square* map representation is applied to various simulation environments. The proposed approach

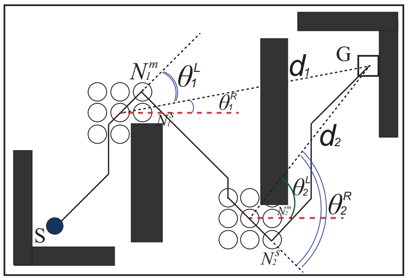

Fig. 3. The vector-driven navigation for a robot with magnitude and direction).

enables an autonomous mobile robot to reach the target along reasonably long trajectory and body turning numbers. Simulation and comparison studies are performed in this section to validate the effectiveness and efficiency of proposed autonomous robot navigation algorithms.

A. Vector-driven Bio-motivated Navigation in a Room-like Environment

The proposed model is applied to a known indoor room environment case with some obstacles in the known workspace. The workspace is shown in Fig. 4, where S (2,2) indicates the starting point and the squares represent the obstacles. As describe previously, Bin and Xiong [2] modified a neural network model for path planning application. Fig. 4 illustrates the generated unreasonable path of the robot by their model. The neural network contains 30×30 discretely and topologically organized neurons, where all the neural activities are initialized to zero. The model parameters are set as: $r_0 = 2$ for the lateral connections; and $E = 200$ for the external inputs. There are five set of wall-like obstacles populated in the workspace. The mobile robot starts from S(2,2). The generated robot path is shown in Fig. 5A, where the robot is autonomously capable of traversing the entire workspace with obstacle avoidance along a more reasonable path. Finally, the robot reaches the target at $G(27, 27)$. The varying environment is represented by the dynamic activity landscape of the neural network. The real-time robot trajectory is planned from the dynamic neural activity landscape and the previous robot location driven by vector-based methodology. Fig. 5B illustrates the neural activity landscape of the BNN in this room-like environment. The peak point indicates the goal at $G(27, 27)$ while the valley areas illustrate the obstacles. The trajectory length and robot's body turning number by the proposed model and the model of Bin and Xiong were calculated (Table I). It shows that the trajectory length by our proposed model is shorter than their model. The length of the trajectory by the proposed model is 34.79% shorter than theirs. The turns of the proposed model is only 1/3 of theirs.

B. Vector-driven Bio-motivated Navigation in a Structured Environment

The proposed model is then compared with the Hu and Yang's GA model [3]. Fig. 6 reproduces the navigation result presented by Hu and Yang whereas Fig. 7A implements

TABLE I

COMPARISON OF TRAJECTORY LENGTH AND NUMBERS OF TURNING OF BIN AND XIONG'S MODEL AND THE PROPOSED MODEL

Model	Length	Turns
Bin and Xiong's model	72.62	27
Proposed model	47.35	9

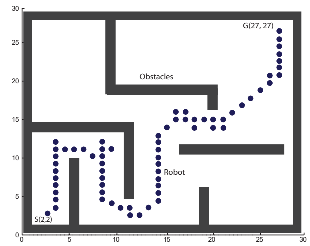

Fig. 4. The generated path by Bin and Xiong's model (redrawn from Bin and Xiong 2004).

the proposed BNN model. The same layout is used to run the BNN navigation model. The simulation results depict that the path produced by the proposed vector-driven BNN model is shorter than the one by Hu and Yangs GA model. The results are summarized in the following table. Fig. 7B illustrates the neural activity landscape of the bio-motivated neural networks in this structured environment. The peak point indicates the goal at $G(16, 16)$ while the valley areas illustrate the obstacles. The trajectory length by the proposed model and the model of Hu and Yang were calculated (Table II). It shows that the trajectory length by our proposed model is shorter than their model. The trajectory length of proposed model obviously calculated in Fig. 7A is $10 \times \sqrt{2} + 16$ = 30.14. In this case, the length of the trajectory by the proposed model is 23.01% shorter than theirs.

TABLE II

COMPARISON OF TRAJECTORY LENGTH OF THE PROPOSED MODEL WITH HU AND YANG'S MODEL

Model	Length
Hu and Yang's model	39.15
Proposed model	30.14

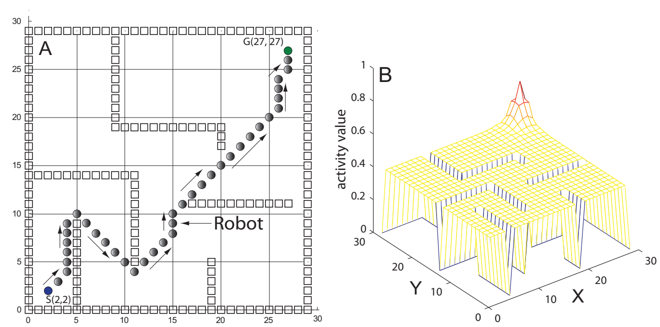

Fig. 5. The simulation result by the proposed model. A: The generated trajectory; B: The neural activity landscape of the bio-motivated neural networks.

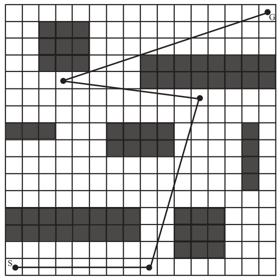

Fig. 6. The robot path generated by Hu and Yang's GA model (redrawn from Hu and Yang 2004).

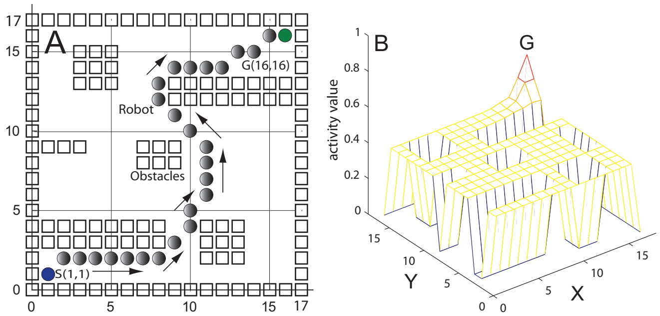

Fig. 7. The simulation result by the proposed model. A: The generated trajectory; B: The neural activity landscape of the bio-motivated neural networks.

C. Vector-driven Bio-motivated Navigation in a Double U-shaped Workspace

In order to validate the efficiency and effectiveness of proposed model, a simulation has been performed to navigate the robot in a two U-shaped environment in Fig. 8A. It demonstrates the the trajectory generated in a square cell map environment representation is reasonable without local minima. The robot position at starting point $S(18, 8)$ is indicated by a blue dot whereas the path in square cell map representation is denoted by dark solid circles with the final goal at $G(19, 21)$. Fig. 8B illustrates the neural activity landscape of the bio-motivated neural networks in the double U-shaped environment. The peak point denotes the goal at $G(19, 21)$, while the valley areas indicate two U-shaped obstacles.

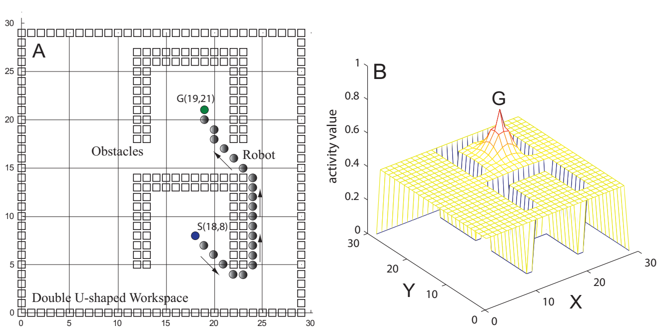

Fig. 8. The simulation result by the proposed model. A: The generated trajectory; B: The neural activity landscape of the bio-motivated neural networks.

In the proposed model, the next location of the neuron propagation is determined by the maximal neural activity or the second largest neural activity among the neighboring neurons by the proposed vector-driven algorithm. No any local minima are found in this simulation with two U-shaped obstacles by the proposed algorithm. The negative activity stays locally only. Since the goal point in the workspace is set as $I_i = E$, the positive neural activity propagates to the whole workspace until it reaches the goal at $G(19, 21)$.

With regard to the implementation of the proposed model onto an actual robot. An actual autonomous vehicle will be developed as a test-bed for real-time navigation of an autonomous vehicle. The vehicle incorporates six sensors into its compact design as follows: a LIDAR, a DGPS (Novatel's ProPak-LB Plus DGPS system), a digital compass (PNI TCM6 digital compass), a camera (The AVT Stingray CCD camera), and an IMU, each of which is enclosed in a waterproof case and firmly mounted to the robot.

IV. CONCLUSIONS

A novel biologically-motivated neural networks approach integrated with developed vector-driven autonomous robot navigation is proposed in this paper. Both simulation and comparison studies validated that the proposed model was capable of planning more reasonable and shorter collision-free paths in non-stationary and unstructured environments. In the future, the proposed model will be implemented on an actual robot [10].

REFERENCES

[1] S. Grossberg (1982), Studies of mind and brain. *Reidel*, Boston.

[2] N. Bin and C. Xiong (2004), "New approach of neural network for robot oath planning". In: *Proc. of IEEE Intl. Symp. on Systems, Man amd Cybernetics*, pp. 735-739.

[3] Y. Hu and S. X. Yang (2004), "A knowledge based genetic algorithm for Path Planning of a mobile robot". *Proc. of IEEE Intl. Conf. on Robotics and Automation*, pp. 4350-4355.

[4] S. X. Yang and M. Meng (2000), "An efficient neural network method for real-time motion planning with safety consideration". *Robotics and Autonomous Systems*, vol. 32, no. 2-3, pp. 115-128.

[5] R. Glasius, A. Komoda and S. Gielen (1996), "A biologically inspired neural net for trajectory formation and obstacle avoidance". *Biological Cybernetics*, vol. 74, no. 6, pp. 511-520.

[6] F. A. Afsar, M. Arif and M. Hussain (2006), "Genetic algorithm based path planning and optimization for autonomous mobile robots with morphological". In: *Proc. of IEEE Intl. Multitopic Conf.*, Philadelphia, USA, pp. 182-187.

[7] X. Peng and D. Xu (2012), "Intelligent online path planning for UAVs in adversarial environments". In: *Intl. Journal of Advanced Robotic Systems*, no. 9, pp. 14-25.

[8] F. B. Xing, W.-J. Gao, K. Battle, T. Marwala and F. V. Nelwamondo (2010), "Can ant algorithms make automated guided vehicle system more intelligent?". In: *Proc. of IEEE Intl. Conf. on Systems Man and Cybernetics*, pp. 3226-3234.

[9] S. O. Haykin (2008) Neural Networks and Learning Machines (3rd Edition). *Prentice Hall.*

[10] J. Chen, C. Luo, M. Krishnan, and M. Paulik (2010), "An enhanced dynamic Delaunay triangulation-based path planning algorithm for autonomous mobile robot navigation". In: *Proc. of SPIE Conf. on Intelligent Robots and Computer Vision: Algorithms and Techniques*, San Jose, California, USA, vol. 7539, pp. 1-12.