15th Triennial World Congress, Barcelona, Spain

# A NEURAL COMPUTATION MODEL FOR REAL-TIME COLLISION-FREE ROBOT NAVIGATION ⋆

**Simon X. Yang[1], Max Meng[2] and Hao Li[1]**

[1] *Advanced Robotics and Intelligent Systems (ARIS) Lab*
*School of Engineering, University of Guelph*
*Guelph, ON N1G 2W1, Canada*
[2] *Advanced Robotics and Teleoperation (ART) Lab*
*Department of Electrical and Computer Engineering*
*University of Alberta, Edmonton, AB T6G 2G7, Canada*

Abstract: A biologically inspired neural computation model is proposed for dynamic planning and tracking control of robots. The dynamic environment is represented by a neural activity landscape of a topologically organized neural network, where each neuron is characterized by a shunting equation. The collision-free path is generated in real-time through the activity landscape without any prior knowledge of the dynamic environment. The real-time tracking control of robots to follow the planned path is also designed using shunting equations. The effectiveness is demonstrated through case studies. Simulation in several computer-synthesized virtual environments further demonstrates the advantages of the proposed approach. *Copyright ©2002 IFAC*

Keywords: path planning, obstacle avoidance, velocity control, neural dynamics, neural network

## 1. INTRODUCTION

Real-time path planning and tracking control of robotic systems are fundamentally important but very difficult issues in robotics, particularly in a nonstationary environment. Many previous works deal with the path planning and the tracking control separately (e.g., Lozano-Pérez, 1983; Khatib, 1986; Glasius *et al.*, 1995; Muñiz *et al.*, 1995; Zelinsky, 1994; Ong and Gilbert, 1998; Fierro and Lewis, 1997; Zhang *et al.*, 1999). There are many studies on path planning of robots using various approaches (e.g., Lozano-Pérez, 1983; Khatib, 1986; Glasius *et al.*, 1995; Muñiz *et al.*, 1995; Zelinsky, 1994; Ong and Gilbert, 1998). However, most of the previous models deal with static environment only and are computationally complicated. Most of the previous models use global methods to search the possible paths in the free space. Ong and Gilbert (1998) proposed a new model for path planning with penetration growth distance, which searches over collision paths instead of the free space. However, these models deal with static environment only and are computationally complicated when in a complex environment (e.g., Lozano-Pérez, 1983; Zelinsky, 1994; Ong and Gilbert, 1998).

Several neural network models were proposed to generate real-time trajectories through learning, e.g., Muñiz *et al.* (1995) proposed a neural network model for the navigation of a mobile robot, which can generate dynamic trajectory with obstacle avoidance through unsupervised learning. But the generated trajectory using learning based approaches is not optimal, particularly at the ini-

⋆ This work was partially supported by Natural Sciences and Engineering Research Council (NSERC) and Materials and Manufacturing Ontario (MMO) of Canada.

tial learning phase. Glasius *et al.* (1995) proposed a neural network model for dynamic trajectory generation without any learning process. However, it suffers from slow dynamics and cannot perform properly in a fast changing environment.

Many tracking controllers are proposed for robots using various methods, such as sliding mode, linearization, backstepping, neural networks, fuzzy systems, and neuro-fuzzy systems (Fierro and Lewis, 1997; Zhang *et al.*, 1999). However, most of the tracking controllers are very complicated, some can be used for mobile robots only.

In this paper, a biologically inspired neural network approach is proposed for real-time collision-free path planning *and* tracking control of robots in a nonstationary environment. The state space of the topologically organized neural network is the configuration space of the robot, which can be the Cartesian workspace for a mobile robot or the joint space for a multi-joint robot manipulator. The dynamics of each neuron is characterized by a shunting equation derived from Hodgkin and Huxley's (1952) membrane model for a biological neural system. There are only local lateral connections among neurons. Thus the computational complexity linearly depends on the neural network size. The varying environment is represented by the dynamic activity landscape of the neural network. The real-time robot path is directly planned through the dynamic neural activity landscape without any *prior* knowledge of the changing environment (note the *current* knowledge of the environment is assume to be completely known), without any explicit searching over the free space or obstacle paths, without any explicit optimizations of global cost functions, and without any learning procedures. Therefore the model algorithm is computationally efficient. Note that the robot responds instantaneously to the dynamic environment. It requires the real-time current knowledge of the changing environment, although it does not need any prior or past environmental information.

Inspired by the features of neural dynamics in a shunting equation, a novel tracking controller is proposed for real-time tracking control of robots. Distinct from the previous neural networks based approaches, no learning procedures are needed. The proposed controller is capable of generating smooth, continuous control commands. The proposed path planner generates the next robot location based on the *current* environment that includes the target, the robot and obstacles. The proposed tracking controller generates real-time velocity commands driving the robot to follow the generated path. The *current* robot position that results from the tracking commands is used for the real-time collision-free path planning.

## 2. THE PROPOSED MODEL

In this section, the biological inspiration will be briefly presented. Then the fundamental concept of using a recurrent shunting neural network for real-time collision-free path planning will be pointed out, and the model algorithm will be presented. Finally, the shunting neural model for tracking control will be presented.

### 2.1 *Biological Inspiration*

Hodgkin and Huxley (1952) proposed a computational model for a patch of membrane in a biological neural system using electrical circuit elements. In this model, the dynamics of voltage across the membrane, $V_m$, is described using state equation technique as

$$C_m \frac{dV_m}{dt} = -(E_p + V_m)g_p + (E_{Na} - V_m)g_{Na} - (E_K + V_m)g_K, \quad (1)$$

where $C_m$ is the membrane capacitance, $E_K$, $E_{Na}$ and $E_p$ are the Nernst potentials (saturation potentials) for potassium ions, sodium ions and the passive leak current in the membrane, respectively. Parameters $g_K, g_{Na}$ and $g_p$ represent the conductances of potassium, sodium and passive channels, respectively. This model provided the foundation of the shunting model and led to a lot of model variations and applications. By substituting $C_m = 1, \xi_i = E_p + V_m, A = g_p, B = E_{Na} + E_p, D = E_k - E_p, S_i^e = g_{Na}$ and $S_i^i = g_K$ in Eq. (1), a shunting equation is obtained (Öğmen and Gagné, 1990)

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)S_i^e(t) - (D + \xi_i)S_i^i(t), \quad (2)$$

where $\xi_i$ is the neural activity (membrane potential) of the $i$th neuron. Parameters $A$, $B$ and $D$ are nonnegative constants representing the passive decay rate, the upper and lower bounds of the neural activity, respectively. Variables $S_i^e$ and $S_i^i$ are the excitatory and inhibitory inputs to the neuron. This shunting model was first proposed by Grossberg to understand the real-time adaptive behavior of individuals to complex and dynamic environmental contingencies, and has a lot of applications in visual perception, sensory motor control, and many other areas (Grossberg, 1988).

### 2.2 *Real-time Path Planning*

The fundamental concept of the proposed model is to develop a neural network architecture, whose dynamic neural activity landscape represents the dynamically varying environment, not only marking the currently target and obstacle locations, but also memorizing some history of the changing environment. By properly defining the external inputs from the varying environment and internal neural connections, the target and obstacles are guaranteed to stay at the peak and the valley of the activity landscape of the neural network, respectively. The target globally attracts the robot in the whole state space through neural activity propagation, while the obstacles have only local effect to avoid collisions. The real-time collision-free robot path is planned through the dynamic activity landscape of the neural network.

The proposed topologically organized model is expressed in a finite ($F-$) dimensional ($F-$D) state space $\mathcal{S}$, which can be either the Cartesian workspace $\mathcal{W}$ or the configuration joint space $\mathcal{C}$ of a multi-joint manipulator. The location of the $i$th unit ("neuron") at the grid in the $F$-D state space, denoted by a vector $q_i \in R^F$, *uniquely* represents a position in $\mathcal{W}$ or a configuration in $\mathcal{C}$. In the proposed model, the excitatory input results from the target and the lateral connections among neurons, while the inhibitory input results from the obstacles only. Each neuron has a local lateral connections to its neighboring neurons that constitute a subset $\mathcal{R}_i$ in $\mathcal{S}$. The subset $\mathcal{R}_i$ is called the receptive field of the $i$th neuron in neurophysiology. The neuron responds only to the stimulus within its receptive field. Thus, the dynamics of the $i$th neuron in the neuron network is characterized by the shunting equation given by

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[\xi_j]^+\right) - (D + \xi_i)[I_i]^-, \tag{3}$$

where $k$ is number of neural connections of the $i$th neuron to its neighboring neurons within the receptive field $\mathcal{R}_i$. The external input $I_i$ to the $i$th neuron is defined as $I_i = E$, if there is a target; $I_i = -E$, if there is an obstacle; $I_i = 0$, otherwise, where $E \gg B$ is a very large positive constant. The terms $[I_i]^+ + \sum_{j=1}^{n} w_{ij}[x_j]^+$ and $[I_i]^-$ are the excitatory and inhibitory inputs, $S_i^e$ and $S_i^i$ in Eq. (2), respectively. Function $[a]^+$ is a linear-above-threshold function defined as, $[a]^+ = \max\{a, 0\}$, and the nonlinear function $[a]^-$ is defined as $[a]^- = \max\{-a, 0\}$, The connection weight $w_{ij}$ from the $i$th neuron to the $j$th neuron is given by $w_{ij} = f(|q_i - q_j|)$, where $|q_i - q_j|$ represents the Euclidean distance between vectors $q_i$ and $q_j$ in the state space, and $f(a)$ is a monotonically decreasing function, such as a function defined as $f(a) = \mu/a$, if $0 \le a < r_0$; $f(a) = 0$, if $a \ge r_0$, where $\mu$ and $r_0$ are positive constants. Therefore 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., $w_{ij} = w_{ji}$. A schematic diagram of the neural network in 2D is shown in Fig. 1, where $r_0$ is chosen as $r_0 = 2$. The receptive field of the $i$th neuron is represented by a circle with a radius of $r_0$.

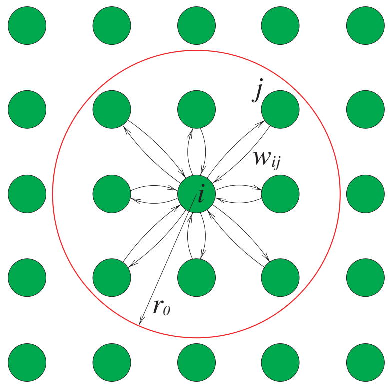


Fig. 1. Schematic diagram of the neural network for robot path planning when the state space is 2D. The $i$th neuron has only 8 lateral connections to its neighboring neurons that are within its receptive field.

The proposed network characterized by Equation (3) guarantees that the positive neural activity can propagate to all the state space, but the negative activity only stays locally. Therefore, the target globally attract the robot, while the obstacles only locally avoid the collision. In addition, the activity propagation from the target is blocked when it hits the obstacle. The locations of the target and obstacles may vary with time. The activity landscape of the neural network dynamically changes due to the varying external inputs from the targets and obstacles and the internal activity propagation among neurons. The optimal robot path is planned from the dynamic activity landscape by a steepest gradient ascent rule. For a given present robot location in $\mathcal{S}$ (i.e., a location in $\mathcal{W}$ or a configuration in $\mathcal{C}$), denoted by $p_p$, the next robot location $p_n$ (also called "command location") is obtained by

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \max\{\xi_j, j = 1, 2, \cdots, k\}, \tag{4}$$

where $k$ is the number of *neighboring neurons* of the $p_p$th neuron, i.e., all the possible next locations of the present location $p_p$ that are subject to the nonholonomic constraint. After the present location reaches its next location, the next location becomes a new present location (if the found next location is the same as the present location, the robot stays there without any movement). The current robot location *adaptively* changes according to the varying environment.

Note that the dynamic activity landscape is used to determine *where* the next robot position is. However, *when* to generate the next robot position is determined by the robot moving speed. In a static environment, the activity landscape of the neural network will reach a steady state. Mostly the robot reaches the target much earlier than the activity landscape reaches the steady state of the neural network. When a robot is in a dynamically changing environment, the neural activity landscape will never reach a steady state. Due to the very large external input constant $E$, the target and the obstacles keep staying at the peak and the valley of the activity landscape of the neural network, respectively. The robot keeps moving toward the target with obstacle avoidance till the designated objectives are achieved.

### 2.3 *Real-time Tracking Control*

The function of a tracking controller in this paper is to implement a mapping between the known information (e.g., the desired path information for the path planner and the measurable sensory information of the current robot position) and the velocity commands designed to achieve the robot's task. The controller design problem can be described as: given the desired robot configuration $q^d(t) = [q_1^d(t), q_2^d(t), \cdots, q_F^d(t)]^T$ in $F$-D, design a control law for the velocity $v(t) = [v_1(t), v_2(t), \cdots, v_F(t)]^T$, which drive the robot to move, such that the actual robot position $q(t)$ will precisely track the desired robot position $q^d(t)$.

Inspired by the dynamic characteristics of the biological shunting model, a novel tracking controller for robots is proposed using a set of shunting equations. In the proposed tracking control, the real-time velocity commands are obtained by

$$\begin{aligned}\frac{dv_i}{dt} = & -A_v v_i + (B_v - v_i)[e_i]^+ \\ & -(D_v + v_i)[e_i]^-,\end{aligned} \tag{5}$$

where $v_i, i = 1, 2, \cdots, F$ are the velocity commands for the robot. For a point mobile robot in a 3D Cartesian workspace, $F = 3$, and $v(t) = [v_x, v_y, v_z]^T$ are the velocity commands in the $X$, $Y$ and $Z$ directions. The control input is from the tracking error $e_i$, which is defined as $e_i = q_i^d - q_i$. The nonlinear function $[a]^+$ and $[a]^+$ are defined as the same as in Eq. (2).

### 2.4 *Stability Analysis of Shunting Neural Network*

In the shunting model in (2), the neural activity $\xi_i$ increases at a rate of $(B - \xi_i)S_i^+$, which is not only proportional to the excitatory input $S_i^+$, but also proportional to an auto gain control term $B - \xi_i$. Thus, with an equal amount of input $S_i^+$, the closer the values of $\xi_i$ and $B$ are, the slower $\xi_i$ increases. When the activity $x_i$ is below $B$, the excitatory term is positive causing an increase in the neural activity. If $\xi_i$ is equal to $B$, the excitatory term becomes zero and $\xi_i$ will no longer increase no matter how strong the excitatory input is. In case the activity $\xi_i$ exceeds $B$, $B - \xi_i$ becomes negative and the shunting term pulls $\xi_i$ back to $B$. Therefore, $\xi_i$ is forced to stay below $B$, the upper bound of the neural activity. Similarly, the inhibitory term forces the neural activity stay above the lower bound $-D$. Therefore, once the activity goes into the finite region $[-D, B]$, it is guaranteed that the neural activity will stay in this region for any value of the total excitatory and inhibitory inputs.

The stability and convergence of the proposed shunting neural network model can also be rigorously proved using a Lyapunov stability theory. Introducing the new variables, $y_i = \xi_i - B$, by variable substitutions the proposed shunting model can be written into Grossberg's general form (Grossberg, 1988)

$$\frac{dy_i}{dt} = a_i(y_i)\left(b_i(y_i) - \sum_{j=1}^{N} c_{ij} d_j(y_j)\right). \tag{6}$$

It can be proved that the proposed shunting model satisfies all the three stability conditions of Grossberg's general form in (6. Therefore, the proposed neural network system is stable. The dynamics of the neural network is guaranteed to converge to an equilibrium state of the system.

## 3. SIMULATION STUDIES

Virtual reality environments have been used for various applications. In this paper, the virtual reality modeling language (VRML) is used to implement the proposed real-time collision-free path planning and tracking control algorithms for mobile robots and manipulation robots. Simulation in several computer-synthesized virtual environments further demonstrates the advantages of the proposed approach with encouraging experimental results.

### 3.1 *A Point Robot to Reach a Target in 3D*

The proposed model is first applied to the obstacle avoidance problem for a 3D U-shaped obstacles. Potential field based methods and other strictly local avoidance schemes cannot deal with this type of deadlock problems (Muñiz *et al.*, 1995). The 3D U-shaped obstacles are shown in Fig. 2 by semi-transparent walls (a box with one side open). The neural network has $30 \times 30 \times 30$ topologically organized neurons, where all the neural activities are initialized to zero. The model parameters are chosen as: $A = 10$ and $B = D = 1$ for the shunting equation; $\mu = 1$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs. The initial robot position is at (15,15,15) that is inside box, while the target is at (12,12,25) that is on opposite side of the open side. The generated robot path is shown in Fig. 2 (two panels are viewed from two different points) by dark balls, where the robot has to first move *away* from the target, then pass through the 3D U-shaped obstacles from the open side, and eventually reach the target position. It shows that the generated robot path is a continuous, smooth route from the starting position to the target with obstacle avoidance.

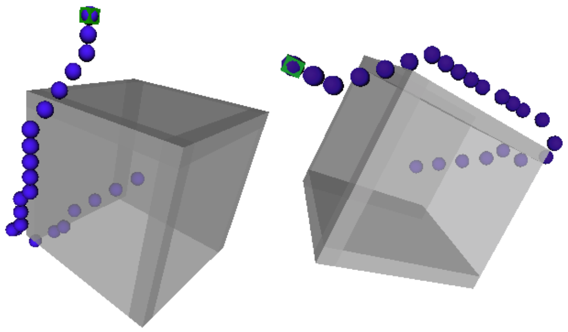

Fig. 2. *Path planning of a point robot to reach a target in 3D.*

### 3.2 *A Robot Manipulator with Multiple Targets*

The proposed model is capable of generating real-time path with multiple targets as well, where the task can be designed as either catching the closest target or catching all the targets. In the latter case, a target should disappear from the state space once it is caught. The proposed model is applied a 2-link planar robot with two targets in the state space of the neural network. The base of the first link is fixed at the center (0,0) of the Cartesian workspace. Initially the tip of the second link is at position (1.366,1.366) in the workspace (Fig. 3A). The task is to move the tip of the second link to position (1.366,-1.366) in the workspace (Fig. 3A). Thus there are two target configurations in the joint space, $(330^o, 330^o)$, and $(300^o, 30^o)$ (shown in Fig. 3B by hollow triangles). There are three obstacles in the workspace (shown in Fig. 3A by solid circles), and the corresponding obstacles in the configuration joint space are shown in Fig. 3B by solid squares. The task in the robot joint space is to generate a trajectory from the initial robot configuration to the closest target configuration.

The neural network has $60 \times 60$ topologically organized neurons, which represents the joint angles from $0^o$ to $354^o$ with a step of $6^o$. Since geometrically $360^o = 0^o$, the neuron at (0,0) is an immediate neighboring neuron of the neuron at (59,59) or (0,59) in the state space, and likewise.

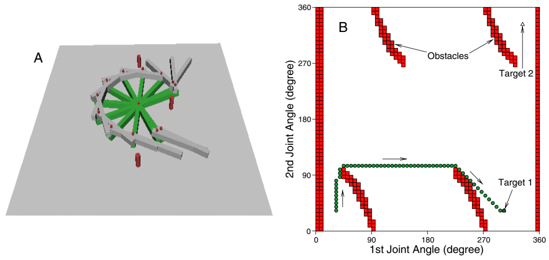


Fig. 3. Path planning of a 2-link planner robot with two target configurations. A: the robot performance in the workspace using VRML; B: the trajectory in the joint space.

The model parameters are chosen as the same as in previous case, i.e., $A = 10$, $B = D = 1$, $\mu = 1$ and $r_0 = 2$. The dynamic robot performance in Cartesian workspace are shown in Fig. 3A implemented by VRML. The path in joint space are shown in Fig. 3B by solid circles. It shows that the robot travels a smooth, continuous, collision-free route in both the workspace and the joint space, and reaches the closest target, Target 1. When there is no obstacle in the workspace, the robot reaches the closest target, Target 2 instead of Target 1.

### 3.3 *A Mobile Robot to Track a Moving Target*

The proposed model is then applied to a real-time trajectory generation problem for a robot to track a moving target. The neural network assumes $30 \times 30$ neuron structure with the same model parameters as in previous cases, i.e., $A = 10$, $B = D = 1$, $\mu = 1$ and $r_0 = 2$. In a 2-D workspace without any obstacles, the traveling route of the target is shown in Fig. 4A as indicated by hollow triangles, with an initial position at $(X, Y)$=(5,5). The target moves at a speed of 25 block/min (it is convenient to assume that the space and time units are block and minute, respectively), and stops at (25,25) after it arrives there. Note that the proposed neural network responds to the real-time location of the targets and obstacles. No prior knowledge of the varying environment is needed. The robot starts to move from position (0,0) at a speed of 10 block/min. The generated trajectory of the robot is shown in Fig. 4A by solid circles. In the virtual reality environment, the activity landscapes of the neural network at a time instant during the motion are shown in Fig. 4B, where the peak location in the activity landscape is the current target location due to the very large external input parameter $E$ from the target. It shows that after the target moving ahead, there is a long tail following the target pick, which results from the natural decay of neural activity. Therefore, the activity landscape not only indicate the current target and obstacle locations, but also has some memory of the changing environment, which is determined by the passive decay parameter $A$ in Eq. (3).

The parameters for the tracking controller is chosen as: $A_v = 3$ and $B_v = D_v = 1$. The generated velocity in $X-$ and $Y-$ directions are shown in Fig. 4C and 4D, respectively. It shows that the proposed simple tracking controller is capable of generating smooth, continuous velocity commands. The actual robot navigation path is almost overlapped with the desired robot path generated by the neural network path planner shown in Fig. 4A.

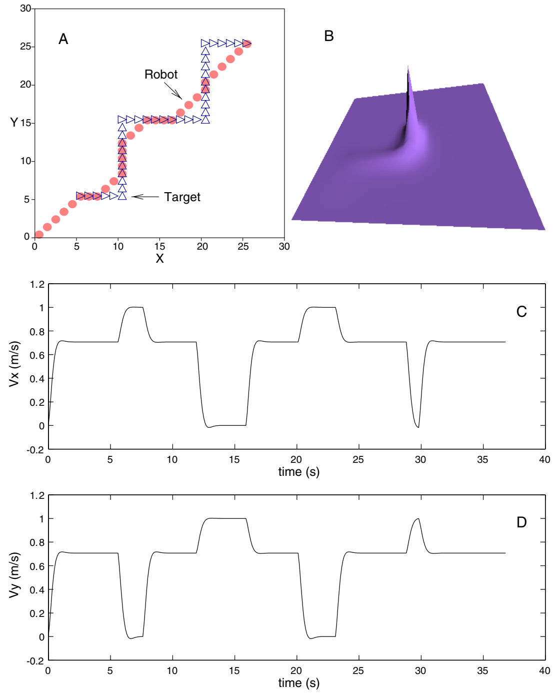


Fig. 4. Path planning and tracking control of a mobile robot to track a moving target. A: the dynamic trajectories of the target (hollow triangles) and the robot (solid circles); B: the activity landscapes at a time point in virtual reality environment; C and D: the generated velocity control commands in $X-$ and $Y-$ directions, respectively.

## 4. CONCLUSION

In this paper, a novel biologically inspired approach to dynamic path planning and tracking control of robots is proposed. The developed approach is capable of planning real-time collision-free path, and generating real-time smooth velocity tracking commands for a mobile or manipulation robot in a nonstationary environment. The robot can precisely travel along the planned robot path. Some points are worth to mention about the proposed neural network approach: (1) the proposed model does not very sensitive to model parameters. There are only very few model parameters, which can be chosen in a very wide arrange. Only two model parameters $A$ and $\mu$ are important factors; (2) the model algorithm is computationally efficient. The robot path is is generated *without* explicitly searching over the free workspace or the collision paths, *without* explicitly optimizing any global cost functions, *without* any prior knowledge of the dynamic environment, and *without* any learning procedures; (3) This model is biologically plausible. The neural activity is a continuous analog signal and has both upper and lower bounds. In addition, the continuous activity prevents the possible oscillations related to parallel dynamics of discrete neurons (Glasius *et al.*, 1995).

## 5. REFERENCES

Fierro, R. and F. V. Lewis (1997). Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. *J. of Robotic Systems* **14**(3), 149–163.

Glasius, R., A. Komoda and S. C. A. M. Gielen (1995). Neural network dynamics for path planning and obstacle avoidance. *Neural Networks* **8**(1), 125–133.

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

Hodgkin, A. L. and A. F. Huxley (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. Lond.* **117**, 500–544.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Research* **5**, 90–98.

Lozano-Pérez, T. (1983). Spatial planning: a configuration space approach. *IEEE Trans. Computers* **32**, 108–320.

Muñiz, F., E. Zalama, P. Gaudiano and J. López-Coronado (1995). Neural controller for a mobile robot in a nonstationary environment. In: *Proc. of 2nd IFAC Conf. on Intelligent Autonomous Vehicles*. Helsinki, Finland. pp. 279–284.

Ong, C. J. and E. G. Gilbert (1998). Robot path planning with penetration growth distance. *J. of Robotic Systems* **15**(2), 57–74.

Öğmen, H. and S. Gagné (1990). Neural models for sustained and on-off units of insect lamina. *Biological Cybernetics* **63**, 51–60.

Zelinsky, A. (1994). Using path transforms to guide the search for findpath in 2d. *Intl. J. of Robotics Research* **13**(4), 315–325.

Zhang, Q., J. Shippen and B. Jones (1999). Robust backstepping and neural network control of a low-quality nonholonomic mobile robot. *Intl. J. of Machine Tools and Manufacture* **39**(7), 1117–1134.