15th Triennial World Congress, Barcelona, Spain

REAL-TIME PATH PLANNING AND TRACKING CONTROL USING A NEURAL DYNAMICS BASED APPROACH ⋆

Simon X. Yang and Eric Hu

Advanced Robotics and Intelligent Systems (ARIS) Lab
School of Engineering, University of Guelph
Guelph, ON N1G 2W1, Canada

Abstract: Real-time collision-free path planning and tracking control of a nonholonomic mobile robot in a dynamic environment is investigated using a neural dynamics based approach. The real-time robot path is generated through a dynamic neural activity landscape of a topologically organized neural network that represents the changing environment. The dynamics of each neuron is characterized by an additive neural dynamics model. The real-time tracking velocities are generated by a novel non-time based controller, which is based on the conventional event based control technique and an additive model. The effectiveness and efficiency of this approach are demonstrated through simulation and comparison studies. *Copyright ©2002 IFAC*

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

1. INTRODUCTION

Real-time collision-free path planning and tracking control of mobile robots 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. Zelinsky, 1994; Jiang *et al.*, 1997; Podsedkowski, 1998; Svestka and Overmars, 1997; Xi, 1993; Kang *et al.*, 1999). There are many studies on robot path planning using various approaches. Most of the previous models for path planning deal with static environment only and are computationally complicated. A local collision checking procedure is required at each step of the robot movement, e.g., to detect local collisions, Zelinsky (1994) model uses a hierarchical collision testing procedure based on "distance space bubbles".

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

Most previous models for nonholonomic mobile robots use two-step approaches that first compute a collision-free holonomic path, and then transform this path by a sequence of feasible ones. The quality of the solution and the computational cost of the second step depend on the shape of the holonomic path, e.g., Jiang *et al.* (1997) proposed a time-optimal path planning method for a robot with kinematic constraints, which consists of three stages: planning for a point mobile robot; planning for a mobile robot; and optimizing cost functions for a time-optimal solution. Podsedkowski (1998) proposed a path planner for nonholonomic mobile robot using a searching algorithm, which requires a local collision checking procedure and the minimization of cost functions.

There are some learning based path planning models, e.g., Svestka and Overmars (1997) proposed a probabilistic learning approach to path planning of mobile robots, which involves a learning phase and a query phase and uses a local method to compute the feasible paths for the robots. However, the learning procedures require extra computational cost, and the planned path is not optimal at its initial learning phase.

Effective robot control design is a fundamentally important issue in robotics. There are many studies on control of robotic systems. Most of the previous control approaches are time based, where the time plays an important role of action reference in both desired trajectory information and measurable system feedback. A typical conventional time-based control system is shown in Fig. 1A, where $y_d(t)$ is the desired robot trajectory, $y(t)$ is the actual robot position, and $e(t)$ is the tracking error.

Xi (1993) proposed a non-time based control method, which produces a better solution to some control problems. The basic idea of non-time based control design is to introduce the concept of an action reference parameter other than time, which is directly relevant to the sensory measurement and the event (thus it is also called event-based control). A typical non-time based control system is shown in Fig. 1B, where s is the action reference parameter, $y_d(s)$ and $e(s)$ are the desired robot path and the tracking error as a functions of s. There are many successful theoretical and practical studies of non-time based controllers, such as robot motion control, multi-robot coordination, force and impact control, robotic teleoperation, manufacturing automation, and Internet based teleoperation. Recently, the concept of non-time based control design is applied to tracking control of nonholonomic mobile robots (Kang *et al.*, 1999), which can track an arbitrary twice differentiable robot path. In addition, the construction of the control system has integrated planning capability, thus the planning and control become a closed-loop system. However, it can deal with small tracking errors only (Kang *et al.*, 1999).

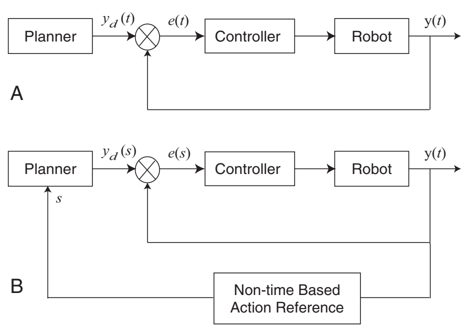

Fig. 1. Schematic diagrams of robot control system. A: conventional control; B: non-time based control.

In this paper, a novel biologically inspired neural network approach is proposed for real-time collision-free path planning *and* tracking control of a nonholonomic mobile robot in a nonstationary environment. The real-time collision-free path planner for nonholonomic mobile robots is based on the neural network model for path planning of a point mobile robot (Yang and Meng, 2001). 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 activity landscape of the neural network without any prior knowledge of the changing environment, without any explicit searching procedures, without any explicit optimizations of global cost functions, without any learning procedures, and without any local collision checking procedures at each step of the robot movement. The computational complexity linearly depends on the neural network size. Therefore the model algorithm is computationally *efficient*. Inspired by the features of neural dynamics in a shunting model, a novel non-time based tracking controller is proposed for tracking control of a mobile robot. The proposed control algorithm can generate a smooth and continuous velocity control commands, which removes the small tracking error limitation in the conventional non-time based controllers. 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. PATH PLANNING AND TRACKING CONTROL PROBLEM

The location of a mobile robot in the 2D Cartesian workspace $\mathcal{W}$ can be *uniquely* determined by the spatial position (x, y) of the base point and the orientation angle θ with respect to the base (see Fig. 2A, where ϕ is the steering angle). A robot location in $\mathcal{W}$, also called a robot configuration, uniquely corresponds to a point (x, y, θ) in the configuration space $\mathcal{C}$. Under the conditions of pure rolling and non-slipping, the kinematic constraint of a nonholonomic mobile robot is described as

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0. \qquad (1)$$

The kinematic constraint can be parameterized by time t. Given the robot linear velocity v and the angular velocity ω of the robot, the robot velocity is given by

$$\dot{x} = v\sin\theta, \quad \dot{y} = v\cos\theta, \quad \dot{\theta} = \omega = vK, \quad (2)$$

where $K = \omega/v$ is the curvature of the curve followed by the robot. The velocity v and the curvature K (the control variables) are limited to $v_{\max}$

and $K_{\max}$, respectively, i.e., $|K| \leq K_{\max}$ and $|v| \leq v_{\max}$. Thus the control space may be denoted as Cartesian product of two intervals, $[-v_{\max}, v_{\max}] \times [-K_{\max}, K_{\max}]$. The minimum turning radius R is given by $R = 1/K_{\max}$. When planning the robot path, the control variables v and K should be discretized (Podsedkowski, 1998). For a given robot configuration, for simplicity there are at most six possible next configurations by setting the v and K to the six discretized values: $\{-v_{\max}, v_{\max}\} \times \{-K_{\max}, 0, K_{\max}\}$. Such a discretization is used to generate the robot movement (one step). After the integration over the time interval of one step, the next robot position is obtained. Fig. 2B shows an example of the possible next robot configurations of a given robot configuration. In realistic robot path planning, the length of one step of robot movement is significantly smaller, so the next robot configuration partially overlaps themselves. Note that an obstacle in $\mathcal{W}$ results in that several robot configurations in $\mathcal{C}$ are not allowed ("forbidden", called obstacle configurations in $\mathcal{C}$).

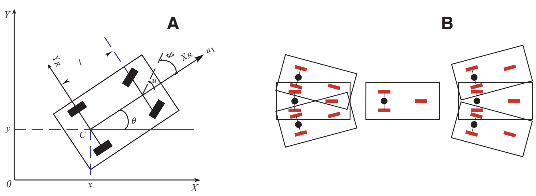

Fig. 2. Schematic diagram of a nonholonomic mobile robot. A: the kinematics of a mobile robot; B: the six possible next robot location of a given location.

A nonholonomic mobile robot can be controlled by two control commands: the linear velocity u_1 and the angular velocity u_2 (Kang & *et al.*, 1999). The tracking control design in this paper is to implement a mapping between the known information (e.g., the desired path information and the measurable system output) and the velocity commands designed to achieve the robot's task. The controller design problem can be described as: given the desired robot path $x_d(s)$, $y_d(s)$, $\theta_d(s)$ and $\phi_d(s)$, design a control law for the linear velocity $u_1(t)$ and angular velocity $u_2(t)$, which drive the robot to move, such that the actual robot position $x(s)$, $y(s)$, $\theta(s)$ and $\phi(s)$ will precisely track the desired robot path $x_d(s)$, $y_d(s)$, $\theta_d(s)$ and $\phi_d(s)$. Note that the planned robot path by most models is discretized.

3. THE PROPOSED MODEL

In this section, the biological inspiration from a biological membrane model to a additive dynamics model is first outlined. Then the proposed neural dynamics based approach to real-time navigation of mobile robot in a dynamic environment is presented, including the neural network model for path planning and the neural dynamics based tracking controller are presented.

3.1 *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

$$C_m \frac{dV_m}{dt} = -(E_p + V_m)g_p + (E_{Na} - V_m)g_{Na} - (E_K + V_m)g_K, \qquad (3)$$

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 setting $C_m = 1$ and substituting $\xi_i = E_p + V_m, A = g_p, B = E_{Na} + E_p, D = E_k - E_p, S_i^e = g_{Na}$ and $S_i^i = g_K$ in eqn (3), a shunting equation is obtained (Yang and Meng, 2001)

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)S_i^e(t) - (D + \xi_i)S_i^i(t), \qquad (4)$$

where ξ_i is the neural activity (membrane potential) of the ith neuron, 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 (1988) to understand the real-time adaptive behavior of individuals to complex and dynamic environmental contingencies, and has applications in various areas (Yang and Meng, 2001).

In the shunting model in eqn (4), if the excitatory and inhibitory inputs are lumped together and the auto gain control terms are removed, then eqn (4) can be written into a simpler form,

$$\frac{dx_i}{dt} = -Ax_i + S_i(t) \qquad (5)$$

This is an additive equation (Grossberg, 1988), where $S_i(t)$ represents the total input to the i-th neuron from the external and lateral connections. This additive model is widely applied to a lot of areas such as vision, associative pattern learning and pattern recognition (Grossberg, 1988).

3.2 *Neural Network for Path Planning*

The neural network architecture of the proposed model is a discrete topographically organized map, which is expressed in a 3D state space $\mathcal{S}$, where two represent the spatial position of the robot base point in the 2D Cartesian workspace and one represents the orientation of the robot with respect to the base point, i.e., the state space $\mathcal{S}$ is the configuration space $\mathcal{C}$ of the mobile robot. The location of the ith neuron at the grid in $\mathcal{S}$, denoted by a vector $p_i \in R^3$, *uniquely* represents a configuration in $\mathcal{C}$ or a location in $\mathcal{W}$ of the robot. In the proposed model, the excitatory input results from the target and its neighboring neurons, while the inhibitory input results from the obstacles only. The dynamics of the ith neuron in the neural network is characterized by a shunting equation,

$$\frac{d\xi_i}{dt} = -A\xi_i + \sum_{j=1}^{k} w_{ij}[\xi_j]^+ + I_i, \qquad (6)$$

where ξ_i is the neural activity of the ith neuron, which has a continuous value (Yang and Meng, 2001). The term $\sum_{j=1}^{k} w_{ij}[\xi_j]^+$ is the excitatory input. The external input I_i to the ith 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. Function $[a]^+$ is a linear-above-threshold function defined as $[a]^+ = \max\{a, 0\}$. The lateral connection weight, w_{ij}, are defined a function of the robot orientation θ *and* the Euclidean distance, $d_{ij} = |p_i - p_j|$, between positions p_j and p_i in $\mathcal{S}$, $w_{ij} = f(d_{ij})$, if the ith and jth neurons are *neighboring*; $w_{ij} = 0$, otherwise, where function $f(d_{ij})$ is a monotonically decreasing function, e.g., $f(d_{ij}) = \mu/d_{ij}$, if $0 < d_{ij} < r_0$; $f(d_{ij}) = 0$, otherwise, where μ 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 lateral neural connection weight is symmetric, $w_{ij} = w_{ji}$. In the proposed model, the *neighboring neurons* is defined as all neurons satisfying the kinematic constraint in eqn (2) *and* whose distance to the ith neuron is less than r_0. Parameter k is the number of all the neighboring neurons of the ith neuron. Therefore, due to the kinematic constraint, the ith neuron has at most six neighboring neurons, $k \leq 6$ (see Fig. 2B).

The proposed neural network characterized by eqn (6) guarantees that only the positive neural activity can propagate to the whole state space. The negative activity stays 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. The activity propagation is directionally selective, which is subject to the kinematic constraint in eqn (2). 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 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

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \max\{\xi_j, j = 1, 2, \cdots, k\}, \quad (7)$$

where k is the number of *neighboring neurons* of the p_pth 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. The current robot location *adaptively* changes according to the varying environment.

3.3 *Neural Dynamics based Tracking Controller*

By using the state-to-reference projection, the desired robot path can be represented as a function of the action reference parameter s, $x_d(s)$ and $y_d(s)$. The measurable robot position is also represented by s as $x(s)$ and $y(s)$. Thus the tracking error can be obtained as $e_x = x - x_d$, $e_y = y - y_d$ and $e_\theta = \theta - \theta_d$, which is also a function of s. The velocity commands of a conventional controller is defined as (Kang & *et al.*, 1999)

$$u_2 = \left(u_1(a_1e_1 + a_2e_2 + a_3e_3) - \beta_1\right)/\beta_2, \quad (8)$$

where

$$e_1 = -e_x \sin\theta_d + e_y \cos\theta_d, \qquad (9)$$

$$e_2 = \sin e_\theta, \qquad (10)$$

$$e_3 = \frac{1}{l}\cos e_\theta tan\phi, \qquad (11)$$

$$\beta_1 = -\frac{1}{l}\sin e_\theta \tan\phi \tan\phi, \qquad (12)$$

$$\beta_2 = \frac{1}{l}\cos e_\theta \sec\phi \sec\phi, \qquad (13)$$

where a_1, a_2 and a_3 are control parameters.

In the proposed non-time based tracking control design, since the oscillation results from the sudden change of tracking error e_θ and $\tan\phi$, control components from two additive models are used to replace the e_θ and $\tan\phi$ that directly cause the oscillations. Thus, the proposed novel neural dynamics based controller is given as the same as eqn (8) with a different control input e_2, e_3, β_1 and β_2, which are defined as,

$$e_2 = \sin(k_1 v_1), \tag{14}$$

$$e_3 = \frac{1}{l} k_2 v_2 \cos(k_1 v_1), \tag{15}$$

$$\beta_1 = -\frac{1}{l} k_2 v_2^2 \sin(k_1 v_1), \tag{16}$$

$$\beta_2 = \frac{\cos(k_1 v_1)}{l \cos^2 \phi}, \tag{17}$$

$$\frac{dv_1}{dt} = -A_1 v_1 + e_\theta \tag{18}$$

$$\frac{dv_2}{dt} = -A_2 v_2 + \tan phi \tag{19}$$

where k_1 and k_2 are positive parameters. The new control variables v_1 and v_2 are characterized by two additive equations. It can be easily proved that the proposed non-time based control system is stable. The tracking error is guaranteed to converge to zero.

4. SIMULATION STUDIES

In this section, the neural network for real-time path planning is applied to a dynamic environment with sudden changes. Then, the neural dynamics based controller is applied to track a simple discretized path. After that, the proposed path planning and tracking control approach is applied to a complicated house-like environment.

4.1 *Path Planning with Sudden Changes*

The proposed neural network for robot path planning can perform properly in an arbitrarily dynamic environment, even with sudden environmental changes, such as suddenly adding or removing obstacles or targets. A case with sudden placement of obstacles in front of a nonholonomic mobile robot is studied. The neural network has $50 \times 30 \times 24$ neurons, and the model parameters are chosen as: $A = 10$, $\mu = 1$, $r_0 = 2$ and $E = 100$. The initial and target robot locations are at locations (5,5,6) and (40,25,0) in $\mathcal{W}$, respectively. First, the planned robot path without any obstacles in $\mathcal{W}$ is shown in Fig. 3A. In the second case under the same initial condition, when the robot reaches (15,22,2) on the way toward the target, a set of V-shaped obstacles shown in Fig. 3B with dark sold squares are *suddenly* placed in front of the robot. The real-time robot path is shown in Fig. 3B, where the robot first has to move *away* from the target, then passes around these obstacles, and finally reaches the target without any collisions. satisfying the kinematic constraint.

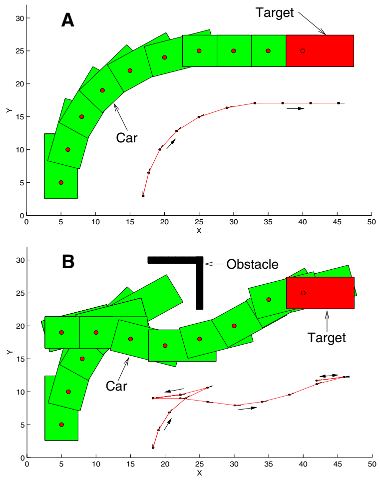

Fig. 3. Real-time path planning of a mobile robot with sudden environmental changes. A: the planned path with no obstacles; B: the real-time path with V-shaped obstacles suddenly placed in front of the robot.

4.2 *Tracking Control with a Simple Discretized Path*

To demonstrate the effectiveness of the proposed neural dynamics based tracking controller, application to a simple discretized path is conducted in Fig. 4A. The desired robot path is shown in Fig. 4, where the starting and target locations are (2,2) and (25,25), respectively. The desired linear and angular velocities are chosen as $v_d = 1$ and $u_d = 0$, respectively. The model parameters are chosen as: $A_1 = 10$, $k_1 = 0.5$, $A_2 = 10$, $k_2 = 0.5$, $a_1 = -5$, $a_2 = -9$, $a_3 = -5$ and $l = 1$. The generated angular velocity command u_2 using the proposed controller is shown in Fig. 4B, while the generated angular velocity u_2 using a conventional controller is shown in Fig. 4C. It is obvious that the control command by the proposed controller smooth, while the command by the conventional controller suffers from sharp changes at the turning positions.

4.3 *Planning and Tracking in a House-like Environment*

The proposed model is then applied to a complex house-like environment, where there are several dead-lock situations that the robot may be trapped in. The neural network has $90 \times 90 \times 24$ neurons, and the parameters are chosen as: $A = 10$, $\mu = 1$, $r_0 = 2$ and $E = 100$. The parameters for tracking control are chosen as the same as in previous case. In case that Door L is opened, the planned robot path is shown in Fig. 5A, where the robot moves to the target along the shortest path. When Door L is closed, the planning path is shown in Fig. 5B, where the

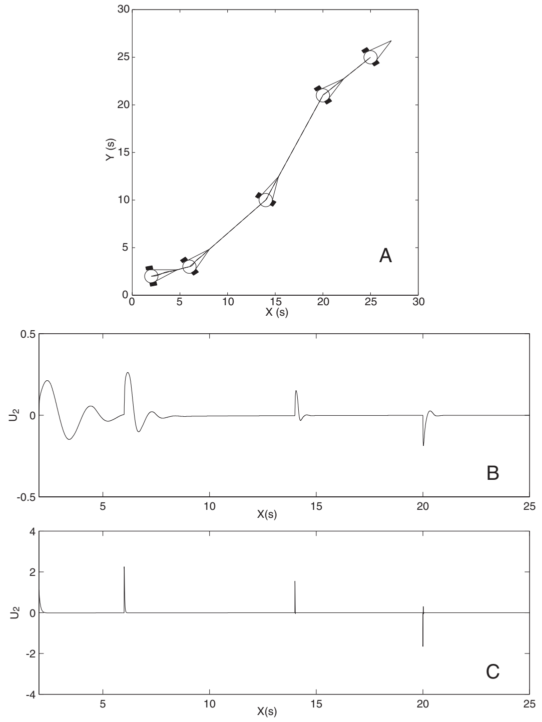

Fig. 4. Tracking control with a simple discretized path. A: the planned and the actual robot paths (almost overlapped); B: the generated angular velocity command; C: the generated angular velocity command using a conventional non-time based controller;

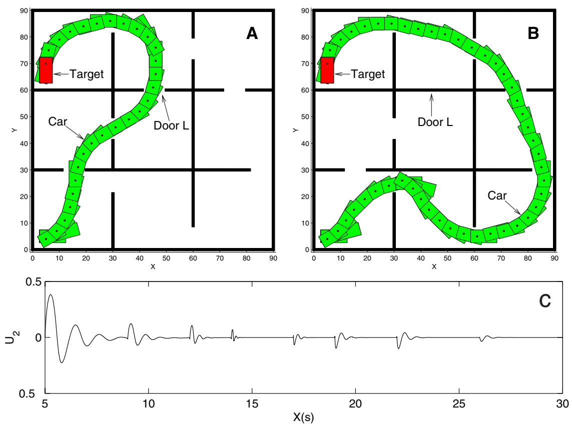

Fig. 5. Path planning and tracking control of a mobile robot in a house-like environment. A: the planned robot path when door L is opened; B: the path when door L is closed; C: the generated angular velocity in case A at the initial phase.

robot has to travel a much longer path to reach the target. Note there are no learning procedures. The robot is capable of reaching the target along the shortest path without any collisions, without violating the kinematic constraint, and without being trapped in any deadlock situations. Fig. 5C shows the generated angular velocity command when Door L is open during the initial period.

5. CONCLUSION

In this paper, a novel neural dynamics inspired approach to real-time collision-free path planning and tracking control of a nonholonomic mobile robot is proposed. The developed approach is capable of planning real-time collision-free path, and generating real-time smooth velocity tracking commands for a nonholonomic mobile robot in a nonstationary environment. The mobile robot can precisely travel along the planned robot path.

6. REFERENCES

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.

Jiang, K., L. D. Seneviratne and S. W. E. Earles (1997). Time-optimal smooth-path motion planning for a mobile robot with kinematic constraints. *Robotica* **15**(5), 547–553.

Kang, W., N. Xi and J. Tan (1999). Analysis and design of non-time based motion controller for mobile robots. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Detroit, USA. pp. 2964–2969.

Podsedkowski, L. (1998). Path planner for nonholonomic mobile robot with fast replanning procedure. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Leuven, Belgium. pp. 3588–3593.

Svestka, P. and M. H. Overmars (1997). Motion planning for carlike robots using a probabilistic approach. *Intl. J. Robotics Research* **16**(2), 119–145.

Xi, N. (1993). Event-based planning and control for robotic systems. PhD thesis. Washington University.

Yang, S. X. and M. Meng (2001). Neural network approaches to dynamic collision-free robot trajectory generation. *IEEE Trans. on Systems, Man, and Cybernetics* **31**(3), 302–318.

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