University of Alberta

Neural Network Approaches to Real-time Motion Planning and Control of Robotic Systems

by

Xianyi Yang ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Electrical and Computer Engineering

Edmonton, Alberta

Fall 1999

0-612-46951-4

University of Alberta

Library Release Form

Name of Author: Xianyi Yang

Title of Thesis: Neural Network Approaches to Real-time Motion Planning and Control of Robotic Systems

Degree: Doctor of Philosophy

Year this Degree Granted: 1999

[illegible]

Xianyi Yang
10305 - 116 Street, Apt. 202
Edmonton, Alberta
Canada, T5K 1W5

Date: June 25, 1999

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Neural Network Approaches to Real-Time Motion Planning and Control of Robotic Systems** submitted by Xianyi Yang in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

[illegible]

Dr. Max Meng, *Supervisor*

[illegible]

Dr. Tongwen Chen

[illegible]

Dr. Horacio J. Marquez

[illegible]

Dr. Hong Zhang

[illegible]

Dr. Ning Xi

Date: May 27, 1999

Dedication

To my great grandmother,

for feeding my young body and soul;

To my parents,

for raising me and the ever-growing standards;

To my wife,

for burning her immense love to energize me;

To my daughters,

for bringing me lucks and uppermost happiness.

Abstract

Real-time motion planning and control are fundamentally important in robotics. In this thesis, a framework, based on biologically inspired neural networks, is developed for real-time robot motion planning with obstacle avoidance in a nonstationary environment. Each neuron in the topologically organized neural network is characterized by a shunting equation or an additive equation. The developed algorithms can be applied to point mobile robots, manipulation robots, holonomic and nonholonomic robots, and multi-robot systems. The planned real-time robot motion with safety consideration does not suffer from either the "too far" or the "too close" problems. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network without explicitly searching over the free workspace or the collision paths, without explicitly optimizing any cost functions, without any prior knowledge of the dynamic environment, without any learning procedures, and without any local collision checking procedures at each step of robot movement. Therefore the proposed algorithms are computationally efficient. The computational complexity linearly depends on the neural network size. The global stability and convergence of the neural network system is guaranteed by both qualitative analysis and the Lyapunov stability theory. The model algorithms are not sensitive to model parameter variations nor sensor noise.

The last part of the thesis presents an efficient neural network based approach to real-time fine motion control of robot manipulators with completely unknown robot dynamics and subject to significant uncertainties. The real-time fine robot motion control is achieved through only the on-line learning of the neural network, without any off-line training procedures. The proposed controller is capable of quickly compensating sudden changes in the robot dynamics. The neural network assumes a single-layer structure, and the learning algorithm is computationally efficient. The global asymptotic stability of the system and the convergence of the tracking error is proved by the Lyapunov stability theory.

Acknowledgments

First and foremost I wish to express my sincere gratitude to Dr. Max Meng, my supervisor, friend, and mentor, for his masterly guidance, constant support and encouragement, intelligent inspiration, and invaluable advice throughout my graduate education at the University of Alberta. His dedication and discussions with me are primarily responsible for the successful completion of this work. I thank him deeply for everything that he taught me.

I would like to thank Dr. Tongwen Chen, Dr. Horacio Maquez, and Dr. Hong Zhang for proof-reading this thesis and providing invaluable suggestions. I also thank Dr. Ning Xi for serving as the external examiner and offering expert suggestions.

I also offer my thanks to my fellow graduate students, faculty and staff members for creating the friendly environment at U of A Engineering. Inspiring talks with many graduate students and faculty members during the past years are acknowledged.

I would also like to acknowledge the financial supports by NSERC through grants to Professor Max Meng, and by University of Alberta through Ph.D. Fellowship and Dissertation Fellowship and in the format of awards and teaching assistantships .

Last bust not least, I owe a great deal of thanks to my family for their support during the past years. Especial thanks are due to my wife, Shirley. Without her encouragement, patience, and taking care of almost all family matters, this work would be far from complete. I would like to thank my parents for their constant encouragement. I also thank my daughters for bringing me lucks and happiness.

Contents

List of Figures

List of Symbols

Symbols in Robot Motion Planning Algorithms

V_m	Voltage across the membrane of a neuron
C_m	Membrane capacitance
E_K, E_{Na}, E_p	Nernst potentials for potassium ions, sodium ions and the passive leak current in the membrane
g_K, g_{Na}, g_p	Conductances of potassium, sodium and passive channels
ξ_i	Neural activity (membrane potential) of the i-th neuron
i, j	Indexes of a neuron in the neural network
A	Passive decay rate of the neural activity
B, D	Upper and lower bounds of the neural activity
S_i^+, S_i^-	Excitatory and inhibitory inputs to the i-th neuron
I_i	External input to the i-th neuron
w_{ij}	Connection weight between the i-th and j-th neurons
N	Number of neurons in the neural network
k	Number of neighboring neurons of a neuron
p_i	Position vector of the i-th neuron in the state space
d_{ij}	Euclidean distance between positions p_i and q_j
μ	Connection weight constant
r_0	Receptive field constant
$\mathcal{S}$	State space of the neural network
$\mathcal{W}$	Workspace of the robot
$\mathcal{C}$	Configuration space of the robot

$\mathcal{R}_i$	Receptive field of the i-th neuron
$v(t)$	Lyapunov function candidate
x, y	X and Y coordinates in Cartesian workspace
θ	Orientation angle of the robot
R	Turning radius of a nonholonomic car-like robot
K	Curvature of the curve followed by the nonholonomic robot

Symbols in Robot Motion Control Algorithms

q	Vector of joint position of the robot manipulator
$\dot{q}$	Vector of joint velocity of the robot manipulator
$\ddot{q}$	Vector of joint acceleration of the robot manipulator
$q_d, \dot{q}_d, \ddot{q}_d$	Vector of desired joint position, velocity and acceleration
$e, \dot{e}, \ddot{e}$	Error vector of joint position, velocity and acceleration
τ	Vector of control torque applied to the robot manipulator
τ_{NN}	Vector of control torque from the neural network
τ_{PD}	Vector of control torque from the PD loop
M	Mass matrix of the robot manipulator
$C\dot{q}$	Vector of centripetal and Coriolis torques
G	Vector of gravitational torques
Y	Regressor of the robot manipulator
W	Vector of connection weight of the neural network
K_p, K_d	Coefficient matrices of PD control torque
P, Γ	Positive definite constant matrices in Lyapunov function candidate

Chapter 1

Introduction

In the past three decades, there has been a great deal of research in the field of robotics. In this thesis, we will primarily concern with the motion planning and motion control of robotic systems.

Before a robot is to move, motion planning is required to tell the robot *where* to move. In a static, priorly known environment, the robot motion path from the starting position to the target position can be planned before the robot starts to move. However, when the environment is nonstationary, real-time motion planning with obstacle avoidance is needed. After knowing where to go, the immediate problem that the robot faces is to know *how* to move. Therefore real-time motion control of the robot is required.

Even if confining our scope only to the motion planning and motion control of robotic systems, the huge volume of published work makes a concise review very difficult if not impossible. Therefore, this chapter will start with an overview of some related previous works on robot motion planning, especially on neural network based approaches to motion planning, safety consideration in motion planning, and motion planning of car-like robots. Then the objectives of the proposal of a novel framework, based on biologically inspired neural networks, for real-time collision-free motion planning of robotic systems will be presented. After that, some related previous works on motion control of robot manipulators will be reviewed. The objectives of the proposal of a novel efficient neural network based approach to real-time fine motion control of robot manipulators will be presented. Finally, the contributions of this thesis will be summarized, and the organization of this thesis will be outlined.

1.1 Overview of Robot Motion Planning

Motion planning is a very important issue in robotics. It becomes much more difficult when the robot is in a nonstationary environment. In this section, we will start with a brief review of robot motion planning using various approaches. Then, some of the neural network based approaches to robot motion planning will be introduced. After that, the safety consideration in robot motion planning, and motion planning of holonomic and nonholonomic car-like robots will be addressed.

1.1.1 Various Approaches to Robot Motion Planning

There are a lot of studies on motion planning of robotic systems using various approaches (e.g. Lozano-Pérez, 1983; Thorpe, 1984; Brooks and Lozano-Pérez, 1985; Crowley, 1985; Kant and Zucker, 1986; Khatib, 1986; Lumelsky and Stepanov, 1986; Donald, 1987; Noborio *et al.*, 1989; Ritter *et al.*, 1989; Ilari and Torras, 1990; de Lamadrid and Gini, 1990; Payton, 1990; Barraquand and Latombe, 1991; Latombe, 1991; Zhu and Latombe, 1991; Seshadri and Ghosh, 1993; Chang *et al.*, 1994; Glasius *et al.*, 1994; Li and Öğmen, 1994; Gambardella and Versino, 1994; Zelinsky, 1994; Glasius *et al.*, 1995; Hyun and Suh, 1995; Kyriakopoulos and Saridis, 1995; Muñiz *et al.*, 1995; Stentz, 1995; Zalama *et al.*, 1995; Zelek, 1995; Zhang and Zhang, 1995; Al-Sultan and Aliyu, 1996; Ferrari *et al.*, 1996; Glasius *et al.*, 1996; Gaudiano *et al.*, 1996; Muraca *et al.*, 1996; Schmitt and Book, 1996; Azarm and Schmidt, 1997; LaValle and Sharma, 1997; Lumelsky and Harinarayan, 1997; Namgung and Duffy, 1997; Pruski and Rohmer, 1997; Bander and White, 1998; Chakravarthy and Ghose, 1998; Fang and Dissanayake, 1998; Haddad *et al.*, 1998; Katevas *et al.*, 1998; Kimmel *et al.*, n.d.; Kreczmer, 1998; LaValle and Hutchinson, 1998; Li and Bui, 1998; Marti and Qu, 1998; Meng and Yang, 1998; Nearchou, 1998; Podsedkowski, 1998; Tsoularis and Kambhampati, 1998; Yang and Meng, 1998*c*; Yih and Ro, 1998; Yung and Ye, 1998*b*; Kassim and Kumar, 1999; Kruse *et al.*, 1999; Saab and VanPutte, 1999; Tsoularis and Kambhampati, 1999; Yang and Meng, 1999*b*; Yang and Meng, 1999*e*). Some of the previous models (e.g. Lozano-Pérez, 1983; Brooks and Lozano-Pérez, 1985; Crowley, 1985; Kant and Zucker, 1986; Donald, 1987; Barraquand and Latombe, 1991; Zelinsky, 1994; Ratering and Gini, 1995; Wyard-Scott and Meng, 1995; Al-Sultan and Aliyu, 1996; Vandorpe *et al.*, 1996; Meeran and Shafie, 1997; Namgung and Duffy, 1997; Chen and Hwang, 1998; de Leon and Sossa, 1998; Jerez *et al.*, 1998; Kreczmer, 1998; Li and Bui, 1998; Podsedkowski, 1998; Saab and VanPutte, 1999) use global methods to search the possible paths in the workspace, which are computationally expensive when the environment is complex, and most of them deal with static environment only. Ong and Gilbert (1998) proposed a new searching based model for path planning with penetration growth distance, which searches over the collision paths instead of searching over free space as most other models. Ong and Gilbert's (1998) model is capable of planning optimal, continuous robot paths in a static environment. Some of the searching based models (e.g. Khatib, 1986; Glasius *et al.*, 1994; Wyard-Scott and Meng, 1995) suffer from undesired local minima, which may be trapped in the deadlock situations, such as with concave U-shaped obstacles, or in maze-like environment. For example, Wyard-Scott and Meng (1995) proposed a potential maze-solving algorithm. It can effectively obtain a locally optimized path that is a solution to the maze-solving type problems. Figure 1.1 shows the generated solution to the well-known Beam Robot Competition Micromouse Maze. Obviously the obtained solution is not globally optimal.

Many other models were proposed for robot motion planning. Seshadri and Ghosh (1993) proposed a path planning model using an iterative approach, but it is computationally complicated, particularly in a complex environment. Li and Bui (1998) proposed a fluid model for robot path planning in a static environment, which is based on the theory of fluid mechanics. Oriolo *et al.* (1997) proposed a model for real-time map building and navigation for a mobile robot using a global path planning plus a local graph searching algorithms, where several cost functions are used (Oriolo *et al.*, 1997; Oriolo *et al.*, 1998). Lumelsky and Harinarayan (1997) proposed a cocktail

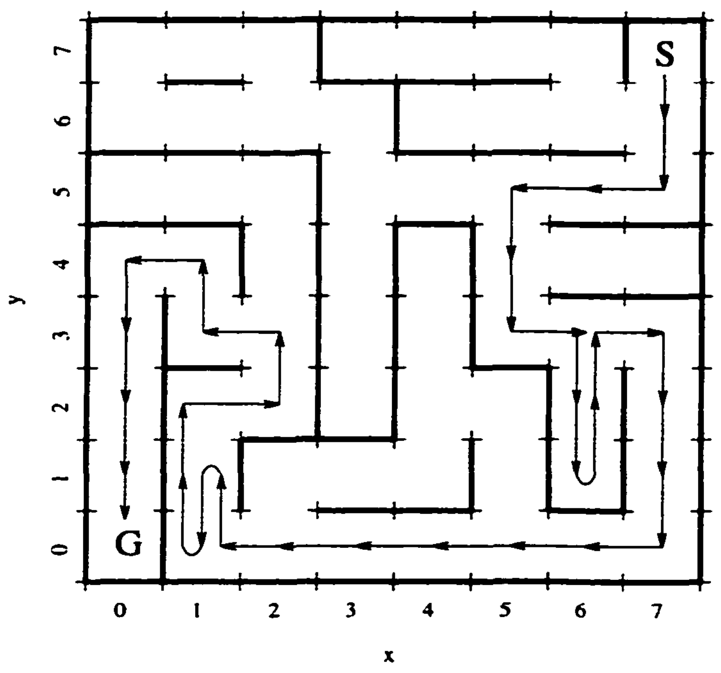


Figure 1.1: *Solution to a maze-solving type problem using a searching based potential model. (from Wyard-Scott and Meng, 1995)*

party model for motion planning of multiple mobile robots in a static environment, which is based on the maze-searching techniques. Fiorini and Shiller (1998) proposed a model for motion planning of robot in a dynamic environment by selecting avoidance maneuvers to avoid static and moving obstacles in the velocity space. Gambardella and Versino (1994) proposed a learning method, based on an artificial potential field, for robot path planning in a cluttered workspace by combining the sensor perception, field information and planner experience. It can detect the dynamic local minima, and incrementally learn to escape or prevent deadlock situations. Figure 1.2 shows the motion path of a 3-degree of freedom (d.o.f.) mobile robot in a room-like environment, where the robot can eventually reach the target without collisions through the learning procedures. The planned robot motion path using learning based approaches may be much longer than the shortest path from the starting robot location to the target (Gambardella and Maex, 1993; Gambardella and Versino, 1994).

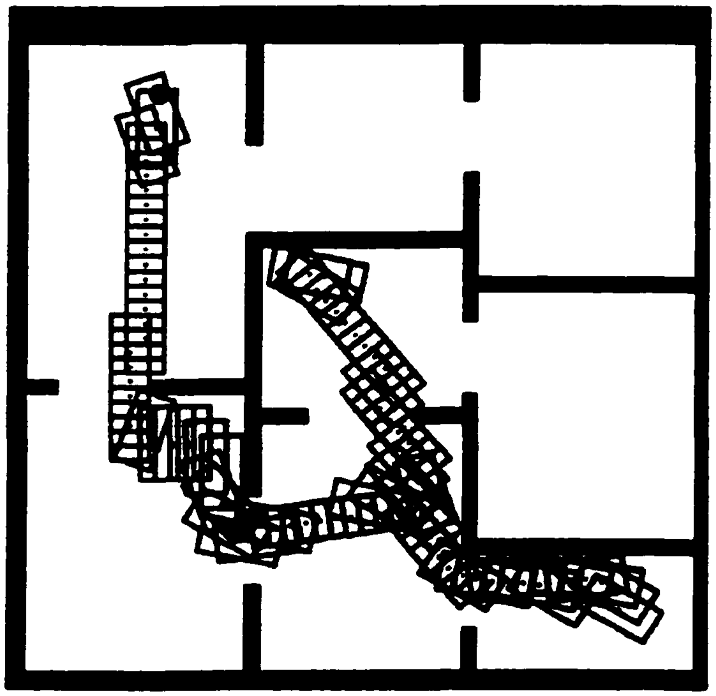


Figure 1.2: *Planned robot motion using an artificial potential field based planner through learning. (from Gambardella and Versino, 1994)*

1.1.2 Neural Network based Approaches to Robot Motion Planning

Many neural network based approaches were proposed for motion planning of robotic systems (e.g. Ritter *et al.*, 1989; Chan *et al.*, 1993; Li and Öğmen, 1994; Glasius *et al.*, 1994; Beom and Cho, 1995; Castellano *et al.*, 1995; Glasius *et al.*, 1995; Muñiz *et al.*, 1995; Zalama *et al.*, 1995; Chang and Song, 1996; Gaudiano *et al.*, 1996; Glasius *et al.*, 1996; Kawakami and Kakazu, 1996; Naruse and Leu, 1996; Matric, 1997; Nagrath *et al.*, 1997; Zhang *et al.*, 1997; Chohra and Benmehrez, 1998; Fang and Dissanayake, 1998; Meng and Yang, 1998; Nearchou, 1998; Pulakka and Kujanpaa, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1998*c*; Kassim and Kumar, 1999; Yang and Meng, 1999*a*; Yang and Meng, 1999*b*; Yang and Meng, 1999*c*; Yang and Meng, 1999*d*; Yang and Meng, 1999*e*; Yang and Meng, 1999*f*). Most of the previous neural network models for robot motion planning are learning based (e.g. Ritter *et al.*, 1989; Simon, 1993; Baginski and Eldracher, 1994; Li and Öğmen, 1994; Millan and Torras, 1994; Beom and Cho, 1995; Li, 1995; Muñiz *et al.*, 1995; Zalama *et al.*, 1995; Gaudiano *et al.*, 1996; Bourbakis *et al.*, 1997; Grolinger *et al.*, 1997; Zhang *et al.*, 1997; Araujo and Vieira, 1998; Fang and Dissanayake, 1998; Kanaya and Tanaka, 1998; Touzet and Santos, 1998; Yamaura and Onozuka, 1998; Kassim and Kumar, 1999). For example, Ritter *et al.* (1989) proposed a neural network model based on Kohonen's self-organizing mapping algorithm, which can learn the transformation from Cartesian workspace to the robot manipulator joint space. Li and Öğmen (1994) proposed a neural network model for real-time trajectory generation by combining an adaptive sensory-motor mapping model and an on-line visual error correction model. However, these models deal with static environment only and assume no obstacles in the workspace.

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. However, this model is computationally complicated since it incorporated the vector associative map (VAM) model and the direction-to-rotation effector control transform (DIRECT) model (Zalama *et al.*, 1995; Gaudiano *et al.*, 1996). The left panel of Figure 1.3 shows four planned paths of a mobile robot in a static environment (from Zalama *et al.*, 1995), where there is no obstacle in the workspace. This neural network model is capable of avoiding the undesired local minima in the deadlock situations such as with concave U-shaped obstacles through learning procedures (Muñiz *et al.*, 1995; Zalama *et al.*, 1995; Gaudiano *et al.*, 1996). The right panel of Figure 1.3 shows the planned real-time robot motion path with concave U-shaped obstacles (from Muñiz *et al.*, 1995), where the robot initially runs into the concave obstacles, then moves around, and finally reaches the static target with obstacle avoidance. These examples demonstrate that the planned robot motion using learning based neural network approaches is not optimal, particularly during the initial learning period of the neural network.

There are various learning based neural network approaches for robot motion planning with obstacle avoidance. Beom and Cho (1995) proposed a robot motion

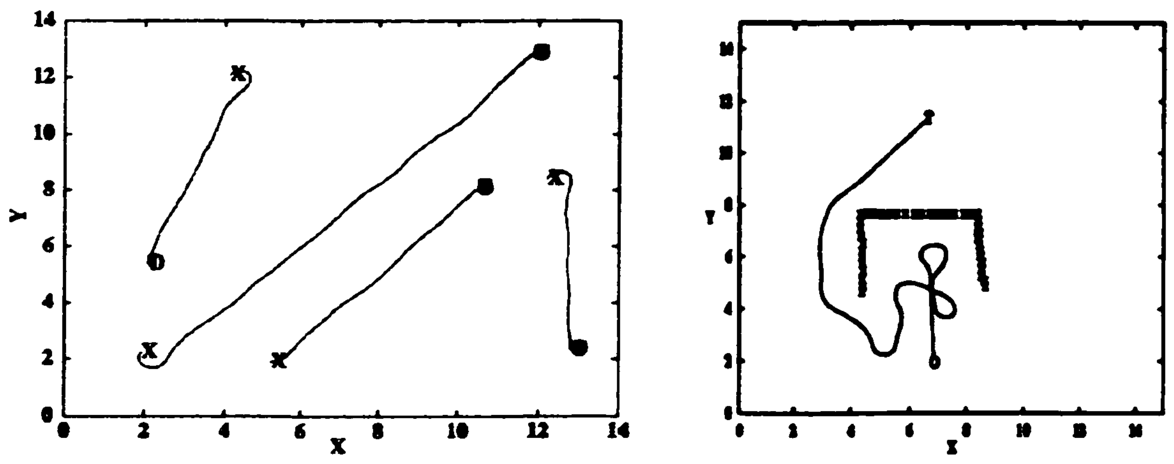


Figure 1.3: *Planned robot motion of a mobile robot using learning based neural network approaches. (Left panel: from Zalama et al., 1995; Right panel: from Muñiz et al., 1995)*

planning model using reinforcement learning. Kassim and Kumar (1997) proposed learning based robot path planner using a wave expansion neural network. Chang and Song (1996) proposed a virtual force guidance model for dynamic motion planning of a mobile robot in a predictable environment, where an artificial neural network is used to predict the future environment through a relative-error-back-propagation learning algorithm (Chang and Song, 1997). Kawakami and Kakazu (1996) proposed a reactive motion planning model for robot manipulators using classifier-systems-based reinforcement learning. Nagrath *et al.* (1997) proposed a Kohonen's topology conserving neural network model for real-time navigation of a mobile robot, where the reinforcement learning based on stochastic real-valued technique is also used. Matric (1997) proposed a reinforcement learning model for motion planning of multiple robots in a dynamic environment, which involves minimizing the learning space through the use of behaviors and conditions, and dealing with the credit assignment problem through shaped reinforcement in the form of heterogeneous reinforcement functions and progress estimators. Thrun (1998) proposed a model for indoor mobile robot navigation, which integrates two paradigms: grid-based map and topological map, where the grid-based maps are learned using artificial neural networks and naive Bayesian integration, while the topological maps are generated on top of the grid-based maps by partitioning the latter into coherent regions. Kanaya and Tanaka (1998) proposed a path planning model for multiple robots using a cellular neural network. Fujii *et al.* (1998) proposed a multi-layer reinforcement learning model for navigation of multiple mobile robots. A schematic diagram of its learning procedure is shown in the upper panel of Figure 1.4. After the neural network is well learned, the planned collision-free motion of two mobile robots in the same workspace is shown in the lower panel of Figure 1.4. However, in addition to extra learning procedures required by the neural network, the model algorithm is computationally complicated.

Some neural network and fuzzy logic based models were proposed for robot motion planning (e.g., Song and Sheen, 1995; Wu, 1995; Zhang *et al.*, 1997; Yung and Ye, 1998*a*). However, the learning procedures are still needed in these models. For example, Yung and Ye (1998*b*) proposed an adaptive fuzzy approach to motion planning with obstacle avoidance, which uses a supervised learning method based on back propagation to determine the parameters for the membership functions.

Glasius *et al.* (1995) proposed a non-learning based Hopfield type neural network model for real-time trajectory generation with obstacle avoidance in a nonstationary environment. This model does not suffer from undesired local minima (Glasius *et al.*, 1995). Glasius *et al.* (1996) later proposed another model by cascading two neural network layers where each layer has a similar architecture to the model in Glasius *et al.* (1995), which becomes an unsupervised learning model but doubles the computational burden as well. However, all these models suffer from slow dynamics, i.e., they cannot perform properly in a fast changing environment. For example, they require that the robot must move faster than the target and obstacles (Glasius *et al.*, 1995; Glasius *et al.*, 1996).

1.1.3 Safety Consideration in Robot Motion Planning

Safety consideration is a very important issue in motion planning of robotic systems. The clearance from obstacles should not be ignored. Many models for robot

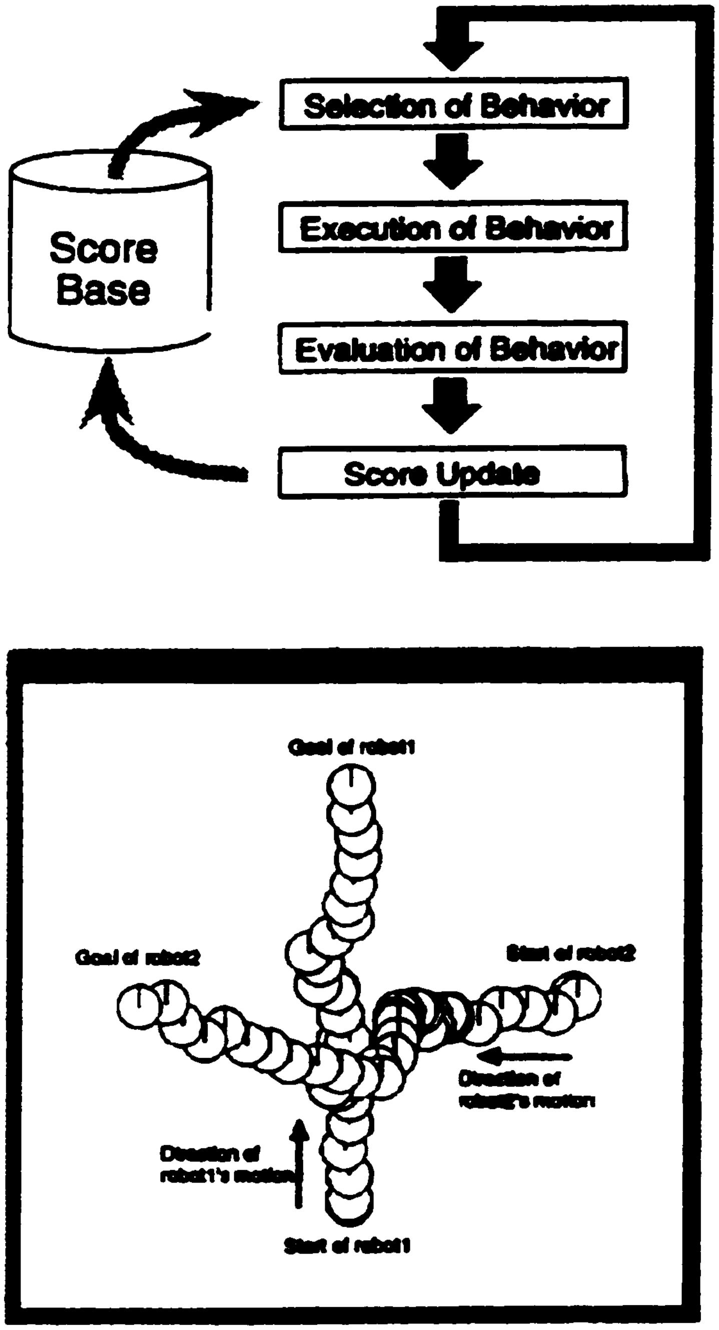


Figure 1.4: *Robot motion planning using multi-layer reinforcement learning model. Upper panel: procedure of reinforcement learning; Lower panel: planned motion of two mobile robots after the learning procedures of the neural network are completed. (from Fujii et al., 1998)*

motion planning concentrate on minimizing the distance between the starting position and target (e.g. Lozano-Pérez, 1983; Brooks and Lozano-Pérez, 1985; Donald, 1987; Payton, 1990; Laumond *et al.*, 1994; Desaulniers and Soumis, 1995; Glasius *et al.*, 1995; Alexander *et al.*, 1998; Bander and White, 1998; Meng and Yang, 1998; Wang, 1998; Szczerba *et al.*, 1998; Yang and Meng, 1998*a*; Saab and VanPutte, 1999; Yang and Meng, 1999*a*; Yang and Meng, 1999*e*). In a static environment, the robot motion planned by the neural network models in Glasius *et al.* (1995), Glasius *et al.* (1996), Meng and Yang (1998), Yang and Meng (1998*a*) and Yang and Meng (1999*e*) has the shortest distance as well, although they do not explicitly minimize any cost functions. They assume that the shortest path is the "best" path. The clearance from obstacles is not considered during the robot motion planning. Therefore, the planned robot motion path clips the corners of obstacles and runs down the edges of obstacles. This is the so called "too close" problem (Thorpe, 1984; Zelinsky, 1994; Yang and Meng, 1999*b*; Yang and Meng, 1999*h*). Such a "too close" problem can be avoided by expanding the obstacles by an extra size, but some possible solution paths are blocked out. This strategy is not acceptable, particularly when all the possible solution paths are blocked out after the expansion.

On the other hand, some models (e.g. Ilari and Torras, 1990) maximize the clearance from obstacles while minimizing the distance from the starting position to the target. Ilari and Torras (1990) proposed a path finding approach by constructing a Voronoi diagram of the free space and then removing branches in the diagram that are not relevant to planning. The left panel of Figure 1.5 shows a complete Voronoi diagram. The planned path after removal of the irrelevant path branches is shown in the right panel of Figure 1.5. It shows that the planned path passes through the middle of free space (Ilari and Torras, 1990; Zelinsky, 1994). Therefore it may deviate significantly from the shortest path. This is the so called "too far" problem (Thorpe, 1984; Zelinsky, 1994; Yang and Meng, 1999*b*; Yang and Meng, 1999*h*). In addition, Ilari and Torras's (1990) model is sensitive to noise. For example, the small triangular obstacle on the boundary causes an unnecessary deviation in the global path. In this model there is no mechanism to limit the effect of the irrelevant obstacles (Zelinsky, 1994). An example of motion planning of a mobile robot using Ilari and Torras's (1990) model is shown in Figure 1.6. It shows that the planned robot motion path deviates significantly from the shortest path.

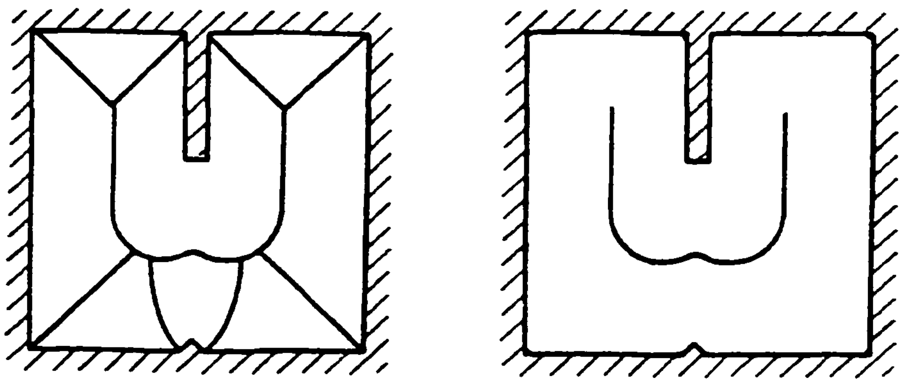


Figure 1.5: *A complete Voronoi diagram (left panel) and the planned path after removal of the irrelevant path branches (right panel). (from Ilari and Torras, 1990)*

Several models were proposed to reduce or solve the "too far" problem and/or the "too close" problems (e.g. Noborio *et al.*, 1989; Zelinsky, 1994; Zhu and Latombe, 1991; Lambert and Le-Fort, 1998). Zelinsky (1994) proposed a path transform model for robot motion planning by combining the distance transform and the obstacle transform approaches, which is considered as an extension to Jarvis and Byrne's (1986) distance transform approach. A schematic diagram of the distance transform (left panel) and obstacle transform (right panel) is shown in Figure 1.7. The cost function in Zelinsky's (1994) model is referred as the path transform (PT) defined as

$$PT(c) = \min_{p \in P} \left(\text{length}(p) + \sum_{c_i \in P} \alpha \, \text{obstacle}(c_i) \right), \tag{1.1}$$

where P is the set of all possible paths from the cell c to the goal, and $p \in P$ is a single path to the goal. Function length(p) is the length of path p to the goal, while function obstacle(c) is a cost function generated by using the values of the obstacle transform. The weight $\alpha \leq 0$ represents the degree of discomfort the nearest obstacle exerts on a cell c. By minimizing the cost function PT, Zelinsky's (1994) model can find a neither "too far" nor "too close" path in a static environment.

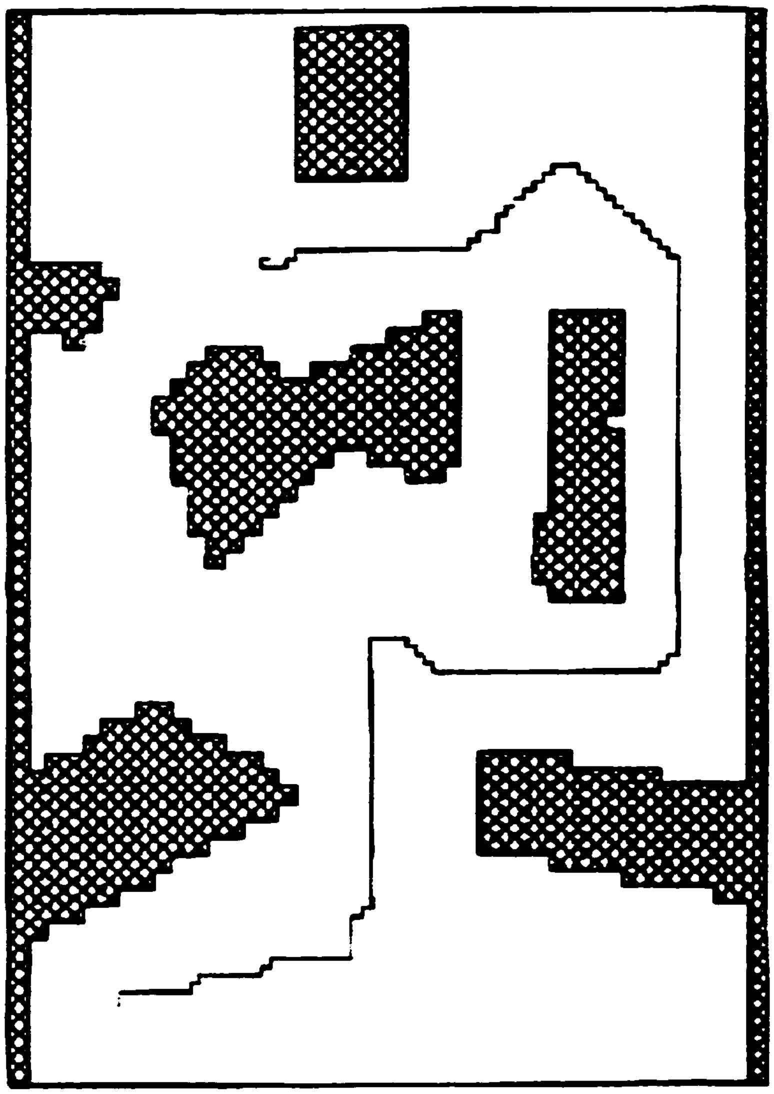


Figure 1.6: *Planned robot motion path of a mobile robot by maximizing the clearance from obstacles while minimizing the robot traveling path. (from Ilari and Torras, 1990)*

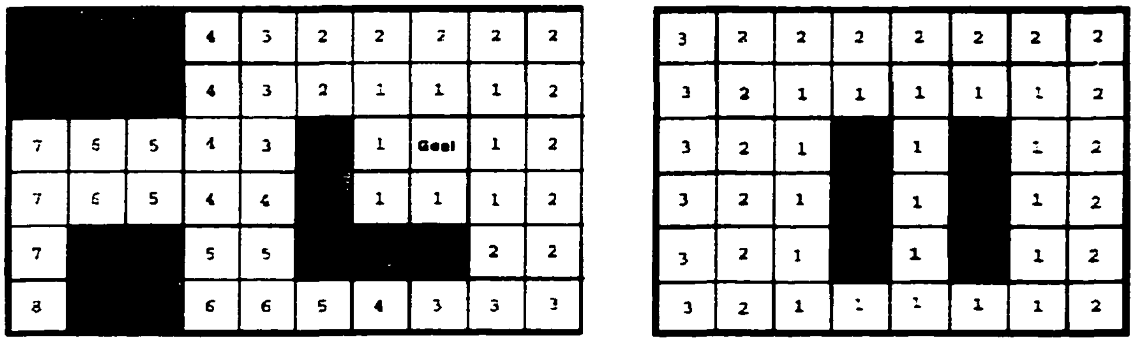


Figure 1.7: *Schematic diagrams of the distance transform (left panel) and the obstacle transform (right panel). (from Zelinsky, 1994)*

1.1.4 Motion Planning of Holonomic and Nonholonomic Car-like Robots

A small and maneuverable mobile robot can be treated as a point robot, with the comparison of the size of the robot and its maneuvering possibilities with the size of free workspace space. For example, in practice, a car in the traffic planning in large cities or tanks in field military operations can be treated as point robots. A point robot has 2-d.o.f., i.e., the translation along X and Y axes in the 2-dimensional (2D) Cartesian workspace. However, in many situations, e.g., when the size of the robot is comparable to the free space, or the length of the robot is obviously larger than its width (e.g., a rectangular robot), the robot should be considered with its shape and size. A holonomic car-like robot is a freely movable object with 3-d.o.f., i.e., two are the translation of the robot base point in the 2D workspace, and one is the rotation with respect to the base point. Therefore, the motion planning of holonomic car-like robots is referred as the motion planning of 3-d.o.f. robots in 2D workspace. The left panel of Figure 1.8 shows a simple rectangular robot with 3-d.o.f., where the configuration of the robot can be uniquely represented by the spatial Cartesian position of its base point and its orientation angle with respect to the base point.

A nonholonomic car-like robot is not a freely movable object. The turning radius of the robot is lower-bounded, i.e., the curvature of the curve followed by the

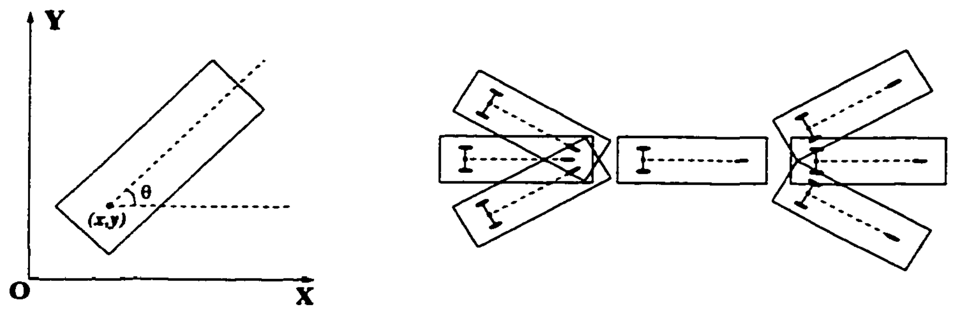


Figure 1.8: *Schematic diagram of car-like robot. Left panel: model of a holonomic 3-d.o.f. robot; Right panel: possible next configurations for a nonholonomic car-like robot.*

robot is upper-bounded. The kinematic constraint makes the degree of freedom for a nonholonomic car-like robot be two. In order to plan the motion path of a nonholonomic car-like robot, the control variables of the robot, the robot velocity and the curvature of the robot moving curve, must be discretized (Latombe, 1991; Podsedkowski, 1998; Kreczmer, 1998). In Podsedkowski's (1998) model, for a given robot configuration there are at most six successors (i.e., possible next configurations) by setting the velocity and curvature of the robot to six discrete values. The right panel of Figure 1.8 shows an example of the possible next configurations for a given configuration of a nonholonomic car-like robot. In Kreczmer's (1998) model, there are at most ten possible next configurations for a given robot configuration.

Motion planning in 2D workspace for mobile robots with size and shape is an important issue in robotics. There are a lot of studies on motion planning of 3-d.o.f. robots using various methods (e.g. Ilari and Torras, 1990; Barraquand and Latombe, 1991; Latombe, 1991; Gambardella and Versino, 1994; Zelinsky, 1994; Fraichard and Mermond, 1998; Kreczmer, 1998; Ong and Gilbert, 1998; Podsedkowski, 1998; Kassim and Kumar, 1999). Most of the previous models deal with static environment only (e.g. Ilari and Torras, 1990; Barraquand and Latombe, 1991; Latombe, 1991; Laumond *et al.*, 1994; Zelinsky, 1994; Kreczmer, 1998; Laumond *et al.*, 1998; Kreczmer, 1998; Ong and Gilbert, 1998). Most of them use a global motion planner plus a local collision checking procedure (e.g. Ilari and Torras, 1990; Zelinsky, 1994; Podsedkowski, 1998). A local collision checking procedure is required at each step of the robot movement. This is not a trivial task. Therefore those models are computationally expensive, particularly when the environment is complex. For example, to detect local collisions, Barraquand and Latombe's (1991) model uses a divide-and-conquer technique; Zelinsky's (1994) model uses a hierarchical collision testing procedure based on "distance space bubbles". Figure 1.9 shows an example of collision detection between a rectangular robot and an obstacle using the distance space bubble hierarchy (from Zelinsky, 1994).

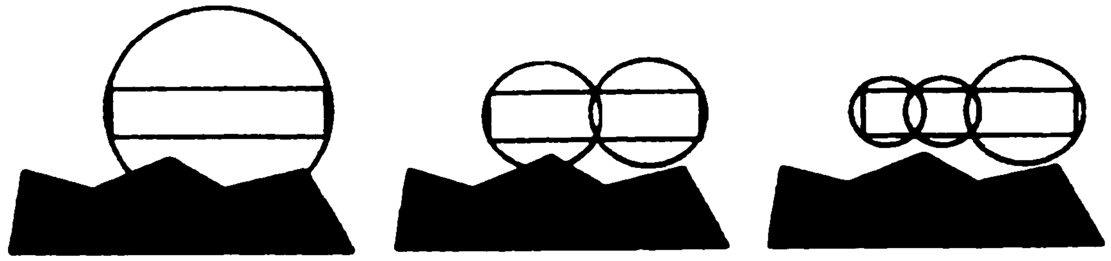


Figure 1.9: *An example of local collision checking between a rectangular robot and an obstacle using distance space bubble hierarchy. (from Zelinsky, 1994)*

There are many studies of motion planning of nonholonomic car-like robots (e.g. Vasseur *et al.*, 1992; Samuel and Keerth, 1993; Fraichard and Scheuer, 1994; Jagannathan *et al.*, 1994; Laumond *et al.*, 1994; Wang *et al.*, 1994; Bicchi *et al.*, 1995; Desaulniers and Soumis, 1995; Bemporad *et al.*, 1996; Desaulniers, 1996; Hong *et al.*, 1996; Jiang *et al.*, 1996; Moutarlier *et al.*, 1996; Paromtchik and Laugier, 1996; Wang, 1996; Jiang *et al.*, 1997; Khatib *et al.*, 1997; Pruski and Rohmer, 1997; Scheuer and Fraichard, 1997; Simeon *et al.*, 1997; Svestka and Overmars, 1997; Bicchi *et al.*, 1998; Fraichard and Mermond, 1998; Kreczmer, 1998; Laumond *et al.*, 1998; Pei and Horng, 1998; Podsedkowski, 1998; Schlegel, 1998; Sekhavat *et al.*, 1998). Most of these models deal with static environment only, or are computationally complicated. Some previous models use two-step approaches that consist of first computing a collision-free holonomic path, and then transforming 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. For example, Moutarlier *et al.* (1996) proposed a model for planning the shortest path in configuration space of a car-like robot, which is based on a Lagrange method for optimizing a function. Paromtchik and Laugier (1996) proposed a searching based iterative algorithm for motion generation for parking a car-like robot. Svestka and Overmars (1997) proposed a probabilistic learning approach to motion planning of car-like robots, which involves a learning phase and a query phase and uses a local method to compute the feasible paths for the robots. Khatib *et al.*'s (1997) motion planning model for car-like robots used a bubble method to find the locally reachable space and a parameterization method to satisfy the kinematic constraint. Podsedkowski (1998) proposed a path planner for nonholonomic car-like robot using a searching algorithm, which requires a local collision checking procedure and the minimization of cost functions. Jiang *et al.* (1997) proposed a time-optimal motion planning method for a robot with kinematic constraints, which consists of three stages: planning for a point mobile robot; planning for a car-like robot; and optimizing cost functions for a time-optimal solution. Sekhavat *et al.* (1998) proposed a multi-level approach to motion planning of nonholonomic robots, where at the first level, a path is found that disrespects the nonholonomic constraints; at each of the next levels, a new path is generated by transformation of the path generated at the previous level; at the final level, all nonholonomic constraints are respected. This model is computationally expensive.

1.2 Objectives of the Proposal of a Novel Framework for Robot Motion Planning

Inspired by Hodgkin and Huxley's (1952) membrane model for a biological neural system and Grossberg's (1973) shunting model, we will propose a novel neural network approach to real-time robot motion planning with obstacle avoidance in a nonstationary environment. The proposed neural network will be topologically organized.

Each neuron in the neural network will be characterized by a shunting equation or an additive equation, which is derived from Hodgkin and Huxley's (1952) membrane equation. The target globally attracts the robot in the whole robot workspace through neural activity propagation, while the obstacles only locally push the robot away in a small region to avoid the possible collisions. The robot motion will be generated through the dynamic activity landscape of the neural network. In contrast to most previous models for robot motion planning, the optimal real-time robot motion will be planned without explicitly optimizing any cost functions, without explicitly searching over the free workspace or the collision paths, without any prior knowledge of the dynamic environment, and without any learning procedures. Therefore, the proposed approach will be computationally efficient. The generated solution to maze-solving type problems or the planned robot motion in a static environment is globally optimal in the sense of the shortest path from the starting position to the target position, if it exists. The optimality of the real-time robot motion planning in a nonstationary environment is in the sense of a continuous, smooth path toward the target. The term "real-time" is in the sense that the robot motion planner responds immediately to the dynamic environment, including the robot, target, obstacles and sensor noise. In order to make the computational complexity of the proposed algorithms linearly depend on the neural network size, each neuron will have only local lateral connections to its neighboring neurons. The model algorithm will be not sensitive to model parameter variations, connection weight function, nor sensor noise. The global stability and convergence of the proposed neural network system will be proved by both qualitative analysis and a Lyapunov stability theory.

Secondly, safety consideration in robot motion planning will be studied. Based on the biologically inspired neural network model for a point mobile robot, by introducing inhibitory lateral connections among neurons and thresholds of the neural connections in the neural network, we will propose an extended model for real-time robot motion planning with safety consideration in a nonstationary environment. In addition, the state space of the neural network will be extended to either the Cartesian workspace or the joint space of multi-joint robot manipulators. This extended neural network model will be capable of planning a real-time "comfortable" robot motion path without suffering from either the "too close" problem nor the "too far" problem. The strength of the clearance from obstacles will be adjustable. By selecting suitable model parameter values, this model will be able to plan the shortest path, a comfortable path, or the safest path for a point mobile robot or a multi-joint robot manipulator.

Thirdly, real-time collision-free motion planning of holonomic 3-d.o.f. robots in 2D Cartesian workspace will be investigated based on the proposed neural network model for a point mobile robot. By extending the state space of the neural network to the configuration space of holonomic car-like robots, an extended neural network model for real-time motion planning with obstacle avoidance of holonomic 3-d.o.f. robots will be proposed. In contrast to the previous models that require local collision checking procedures at each step of the robot movement, no local collision checking will be needed in the proposed algorithms. In addition, some complicated robot motion planning problems, such as real-time motion planning with sudden environmental changes, motion planning of a robot with multiple moving targets, and motion planning of multi-robot system, will be studied.

Furthermore, real-time motion planning with obstacle avoidance of nonholonomic car-like robots will be studied based on the neural network model for motion planning of holonomic 3-d.o.f. robots. By introducing directionally selective lateral neural connections, we will propose an extended neural network model for real-time collision-free motion planning of nonholonomic car-like robots in a nonstationary environment.

In summary, a framework, using biologically inspired neural networks, for real-time collision-free motion planning of robotic systems will be developed. It will be able to deal with point mobile robots, multi-joint manipulation robots, holonomic and nonholonomic car-like robots, and multi-robot systems. In comparison to previous motion planning methods, the proposed framework will have several feature properties, e.g., efficient computational algorithm; global stability; global optimality; robustness to model parameter variations and sensor noise; safety consideration; no learning procedures; no explicit minimization of cost functions; no explicit searching procedures; and biological plausibility.

1.3 Overview of Motion Control of Robot Manipulators

Motion control of robot manipulators in real-time is very important in robotics but very difficult to achieve, particularly when without any prior knowledge of the robot dynamics and with sudden changes in the robot dynamics. There are a lot of studies on motion control of robot manipulator using various approaches (e.g. Slotine and Li, 1987; Craig, 1988; Ortega and Spong, 1988; Slotine and Li, 1988; Lewis *et al.*, 1990; Miller *et al.*, 1990; Sadegh and Horowitz, 1990; van der Smagt and Kröse, 1991; Zomaya and Morris, 1992; Lu and Meng, 1993; Wada and Kawato, 1993; Khemaissia and Morris, 1993; Gomi and Kawato, 1993; Behera *et al.*, 1994; Meng and Lu, 1994; Pham and Oh, 1994; Ahmed *et al.*, 1995; Lewis *et al.*, 1995; Meng, 1995; Watanabe *et al.*, 1995; Behera *et al.*, 1996; Jung and Hsia, 1996*a*; Lewis *et al.*, 1996; Meng, 1996; Morris and Khemaissia, 1996; Chiaverini *et al.*, 1997; Ciliz and Isik, 1997; Meddah and Benallegue, 1997; Yan and Li, 1997; ElDeeb and ElMaraghy, 1998; Li *et al.*, 1998; Song and Sun, 1998; Tomiyama *et al.*, 1998; Yang and Meng, 1998*b*; Dixon *et al.*, 1999; Meng and Yang, 1999; Yang and Meng, 1999*g*). In this section, we will start will a brief introduction to the conventional approaches to motion control of robot manipulators. Then, some neural network based approaches to robot motion control will be reviewed.

1.3.1 Conventional Approaches to Robot Motion Control

The traditional proportional and derivative (PD) controller is very simple and does not require any knowledge of the robot dynamics. But it requires very large actuation to achieve fine motion control robot manipulators, which is not practical but highly demanded in many cases (Craig, 1988; Meng, 1992; Meng, 1995; Meng, 1996). The computed torque control approach and other model-based approaches (e.g. Meng, 1995; Dixon *et al.*, 1999) are capable of achieving fine robot motion control. However, they requires the exact model of the robot dynamics, which is almost impossible in practice (Meng, 1992; Meng, 1996). The adaptive controllers (e.g. Slotine and Li, 1987; Slotine and Li, 1988; Meng, 1992; Lu and Meng, 1993; Meng and Lu, 1994; Chiaverini *et al.*, 1997; ElDeeb and ElMaraghy, 1998) can achieve fine robot motion control and compensate the partially known manipulator dynamics. But, they often require complicated on-line estimation of the robot dynamics (Slotine and Li, 1987; Slotine and Li, 1988; Lu and Meng, 1993; Meng, 1996).

1.3.2 Neural Network based Approaches to Robot Motion Control

A lot of neural network based controllers were proposed for motion control of robot manipulators (e.g. van der Smagt and Kröse, 1991; Goh *et al.*, 1993; Gomi and Kawato, 1993; Khemaissia and Morris, 1993; Kim *et al.*, 1993; Wada and Kawato, 1993; Sanger, 1994; Hamavand and Schwartz, 1995; Kguchi and Fukuda, 1995; Lewis *et al.*, 1995; Tso and Lin, 1995; Behera *et al.*, 1996; Chen and Gill, 1996; Jung and Hsia, 1996*b*; Kim and Lee, 1996; Lewis *et al.*, 1996; Meng, 1996; Morris and Khemaissia, 1996; Xu *et al.*, 1996; Ciliz and Isik, 1997; Ertugrul and Kaynak, 1997; Jeong *et al.*, 1997; Jung and Hsia, 1997; Meddah and Benallegue, 1997; Nam *et al.*, 1997; Sun and Sun, 1997; Yan and Li, 1997; Jung and Hsia, 1998; Song and Sun, 1998), which succeed in some areas where the model-based approaches failed. However, these models need an off-line learning procedures, or are computationally complicated. Khemaissia and Morris (1993) proposed a neuro-adaptive controller by using a neural network plus a servo feedback controller, which does not require an off-line training. However, it suffers from computational complexity and very slow convergence, since for an n-d.o.f. robot manipulator it requires n three-layer neural networks. Later Morris and Khemaissia (1996) proposed a new neural network based controller by using recursive prediction error technique to improve the convergence speed. But it is still computationally complicated. Kguchi and Fukuda (1995) proposed a fuzzy neural controller for control of robot manipulators, where the neural networks control has a learning ability from experiments and the fuzzy control has an ability of dealing with human knowledge. Behera *et al.* (1996) proposed a neuro-adaptive hybrid controller for tracking control of robot manipulators, where three multi-layer neural networks are used to learn the mass matrix, centrifugal and Coriolis force matrix, and the gravitational torque vector, respectively, but it is computationally expensive.

Meng (1996) proposed a signal-layer neural network based model for fine motion control of robot manipulators. This model takes advantage of the robot regressor dynamics and is computationally simple. However, it requires an off-line training procedure for the neural network to perform satisfactorily (Meng, 1996).

1.4 Objectives of the Proposal of a Novel Neural Network based Controller

A novel neural network based method will be proposed for real-time fine motion control of multi-joint robot manipulators. Unlike most previous neural network based approaches, no off-line training procedures will be required in the proposed controller. Unlike the model-based approaches, no knowledge of the robot dynamics will be needed. The proposed neural network based controller will be capable of performing real-time fine motion control of robot manipulators with the ability of quickly compensating sudden changes in the robot dynamics. To achieve these objectives, the proposed controller will consist of two parts: a control torque from a feedforward neural network, and a control torque from a conventional PD feedback loop. The PD feedback loop will be used to guarantee the global stability during the learning period of the neural network at the initial phase *or* when sudden changes in the robot dynamics happen. The real-time fine motion control of robot manipulators will be achieved through only the on-line learning of the neural network. Therefore the proposed neural network based controller will inherit advantages from both the neural network based controllers and the PD type controllers.

In addition, to obtain a simple neural network architecture and an efficient learning algorithm of the neural network, the proposed controller will take advantage of the robot regressor dynamics formulation that expresses the highly nonlinear robot dynamics in a linear form in terms of the robot dynamic parameters. The neural network will assume a single-layer structure. The learning algorithm of the neural network will be derived from the global stability analysis of a Lyapunov function candidate. Therefore, the control algorithm will be computationally efficient. In addition, the global asymptotic stability of the control system and the convergence of the tracking error will be guaranteed.

1.5 Contributions of this Thesis

The proposed neural network approaches to real-time motion planning and motion control of robotic systems have potential industrial applications, and offer insight into the biological mechanism. The contributions of this thesis can be summarized as follows.

1. A novel framework, based on biologically inspired neural networks, for real-time collision-free motion planning of robotic systems in a nonstationary environment is developed. The proposed motion planning approach can deal with point mobile robots, multi-joint robot manipulators, holonomic and nonholonomic car-like robots, and multi-robot systems. The state space of the neural network can be either the Cartesian workspace, the joint space of multi-joint robot manipulators, or the configuration space of the car-like robots. Each neuron in the topologically organized neural network is characterized by a shunting equation or an additive equation. There are only local connections among neurons. The computational complexity of the proposed algorithms linearly depend on the neural network size. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network without explicitly optimizing any cost functions, without explicitly searching over the free workspace or the collision paths, without any prior knowledge of the dynamic environment,

without any learning procedures. Therefore, the model algorithms are computationally efficient. The stability and convergence of the neural network system is guaranteed by both qualitative analysis and the Lyapunov stability theory. The proposed algorithms are not sensitive to model parameters, connection weight function, nor sensor noise.

2. A novel neural network model for real-time robot motion planning with safety consideration is proposed. By employing both excitatory and inhibitory lateral connections among neurons and thresholds of the neural connections, the proposed neural network model is capable of planning a real-time comfortable robot motion path without suffering from the "too far" nor the "too close" problems. The strength of the clearance from obstacle is adjustable. By choosing suitable model parameters, the proposed model is capable of generating either the shortest path, a comfortable path, or the safest path for a point mobile robot and a multi-joint robot manipulator.

3. A novel neural network model for real-time motion planning with obstacle avoidance of holonomic 3-d.o.f. robots in a nonstationary environment is proposed. Based on the biologically inspired neural network model for motion planning of point mobile robot, a neural network is proposed by extending the state space of the neural network to the configuration space of 3-d.o.f. robot. Unlike most previous models that require local collision checking procedures at each step of the robot movement, no local collision checking procedure is needed in the proposed neural network model.

4. A novel neural network model for real-time collision-free motion planning of nonholonomic car-like robot in a nonstationary environment is proposed. In contrast to the neural network model for motion planning of holonomic 3-d.o.f. robot, the lateral connections among neurons in the proposed neural network is directionally selective, and the neural activity propagation is subject to the nonholonomic kinematic constraint of the car-like robot. To the best of our knowledge, it is the first time that the real-time collision-free motion planning of

car-like robot are studied using a non-learning based neural network approach.

5. Some complicated robot motion planning problems are studied using the proposed framework for real-time robot motion planning with obstacle avoidance, including motion planning with sudden environmental changes, motion planning of a robot with multiple moving targets, and motion planning of multiple robots in the same workspace.

6. An efficient neural network based model is developed for real-time fine motion control of robot manipulators with completely unknown robot dynamics and subject to significant uncertainties. No off-line training procedure is needed in the proposed controller. The real-time fine robot motion control is achieved through only the on-line learning of the neural network. In addition, the proposed neural network based controller is capable of quickly compensating sudden changes in the robot dynamics. The proposed controller inherits advantages from both the PD type controllers and the neural networks based controllers. By taking advantage of the robot regressor dynamics, the neural network assumes a single-layer structure, and the learning algorithm is computationally efficient. The global asymptotic stability of the control system and the convergence of the tracking error to zero is guaranteed by a Lyapunov stability theory.

1.6 Organization of this Thesis

In Chapter 2 we develop a novel biologically inspired neural network approach to real-time robot motion planning with obstacle avoidance in a nonstationary environment. The originality, the model algorithm, and the system stability analysis of the proposed neural network approach are presented. Several simulation studies of real-time motion planning of a point mobile robot under various situations are included to demonstrate the effectiveness and efficiency of the proposed algorithm. The parameter sensitivity of the proposed model is discussed by both descriptive analysis and computer simulations. Three model variations are introduced and a comparison study among these models is presented. Finally, several feature properties of the proposed approach are outlined.

In Chapter 3 we develop a novel neural network approach, based on the neural network model presented in Chapter 2, to real-time robot motion planning with safety consideration in a nonstationary environment. Both excitatory and inhibitory connections among neurons and threshold of the neural connections are used in the proposed neural network model. In addition, the state space of the neural network is extended to either the Cartesian workspace or the joint space of a multi-joint robot manipulator. Several simulations studies of real-time motion planning of a point mobile robot and a robot manipulator, in static and nonstationary environment, are carried out with various strength of the clearance from obstacles. A model variation is presented, and the feature properties of the proposed approach is summarized.

In Chapter 4 we develop a novel neural network approach to real-time collision-free motion planning of holonomic 3-d.o.f. robots in a nonstationary environment. The configuration space of a 3-d.o.f. robot is used as the state space of the neural network. Some complicated robot motion planning problems, such as real-time robot motion planning with sudden environment changes, motion planning of a robot with multiple moving targets, and motion planning of multiple robots in the same workspace, are investigated. Finally we introduce a model variation and outline some feature properties of the proposed neural network model.

In Chapter 5 we develop an novel neural network approach, based on the model presented in Chapter 4, to real-time collision-free motion planning of nonholonomic car-like robots in a nonstationary environment. To satisfy the kinematic constraint of the nonholonomic car-like robot, the lateral connections among neurons in the neural network are directionally selective. Simulation studies of real-time motion planning of a nonholonomic car-like robot in both static and nonstationary environment are carried out. A model variation is introduced and the feature properties of the proposed neural network model is highlighted at the end.

In Chapter 6 we develop an efficient neural network based approach to real-time fine motion control of multi-joint robot manipulators. The philosophy of the proposed approach, the control algorithm, the on-line learning algorithm of the neural network are presented. The global asymptotic stability of the system and the convergence of tracking error are proved using a Lyapunov stability theory. A case study of a two-link robot manipulator is carried out to track an elliptic trajectory in real-time with demonstration of the capability of quickly compensating sudden changes in the robot dynamics. In addition, an alternative neural network based control algorithm, based on the same concept, is presented. Finally, the feature properties of the proposed neural network based controller is outlined.

Chapter 7 summarizes the work presented in this thesis and discusses possible future research projects as extension to the presented work.

Chapter 2

A Biologically Inspired Neural Network Approach to Real-time Robot Motion Planning with Obstacle Avoidance

In this chapter, a novel biologically inspired neural network approach is developed for real-time robot motion planning with obstacle avoidance in a nonstationary environment. The proposed neural network is topologically organized, where the dynamics of each neuron is characterized by a shunting equation or an additive equation. There are only local connections among neurons in the neural network. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network, which represents the dynamic environment. The stability and convergence of the neural network system is proved by both qualitative analysis and the Lyapunov stability theory. In addition, this model is not very sensitive to the model parameters nor connection weight function. Three model variations are introduced, and a comparison among these models is presented. The effectiveness and efficiency of the proposed approaches are demonstrated through simulation and comparison studies.

2.1 Introduction

Motion planning with obstacle avoidance is a very important issue in robotics. There are many studies on motion planning for robots using various approaches. Most of the previous models use global methods to search the possible paths in the workspace. Ong and Gilbert (1998) proposed a new searching based model for path planning with penetration growth distance, which searches over collision paths instead of the free workspace. Most searching based models can deal with static environment only and are computationally complicated when the environment is complex. Some of the previous models, especially the potential based approaches, suffer from undesired local minima, which may be trapped in the deadlock situations such as with concave U-shaped obstacles. To obtain an optimal solution to the motion planning problems, most of the previous motion planning models require the minimization of some cost functions. For example, Oriolo *et al.* (1997) proposed a motion planning model for a mobile robot by a global path planning plus a local graph search algorithm. Several cost functions are used to achieve a satisfactory performance (Oriolo *et al.*, 1997; Oriolo *et al.*, 1997). Some of the previous robot motion planning models require the prior information of the nonstationary environment, including the varying target and obstacles. For example, Chang and Song (1996) proposed a virtual force guidance model for dynamic motion planning of a mobile robot in a predictable environment, where an artificial neural network is used to predict the future environment through a relative-error-back-propagation learning algorithm (Chang and Song, 1996; Chang and Song, 1997).

Many neural network based models were proposed for motion planning of robotic systems. Some of the previous neural network models for robot motion planning deal with static environment only (e.g. Ritter *et al.*, 1989; Li and Öğmen, 1994). Most of the neural network models are learning based (e.g. Ritter *et al.*, 1989; Li and Öğmen, 1994; Muñiz *et al.*, 1995; Zalama *et al.*, 1995; Fujii *et al.*, 1998; Kassim and Kumar, 1999). The learning based approaches suffer from extra computational cost because of the learning procedures. In addition, the planned robot motion using learning based approaches is not optimal, especially during the initial learning phase of the neural network. A detailed review of various robot motion planning approaches can be found in Chapter 1.

Glasius *et al.* (1995) proposed a non-learning based Hopfield type neural network model for real-time trajectory generation with obstacle avoidance in a nonstationary environment. This model does not suffer from undesired local minima (Glasius *et al.*, 1995). Glasius *et al.* (1996) later proposed another model by cascading two neural network layers where each layer has a similar architecture to the model in Glasius *et al.* (1995), which becomes an unsupervised learning model but doubles the computational burden as well. However, all those models suffer from slow dynamics and cannot perform properly in a fast changing environment, e.g., they require that the robot dynamics must be faster than the target and obstacle dynamics (Glasius *et al.*, 1995; Glasius *et al.*, 1996).

In this chapter, inspired by Hodgkin and Huxley's (1952) membrane model for a biological neural system and Grossberg's (1973) shunting model, a novel neural network approach is proposed for real-time robot motion planning with obstacle avoidance in a nonstationary environment. The proposed neural network is topologically organized. The state space of neural network is the Cartesian workspace. The dynamics of each neuron is characterized by a shunting equation or a simple additive equation, which is derived from Hodgkin and Huxley's (1952) membrane equation. In the proposed neural network, the target *globally* attracts the robot in the whole state space through neural activity propagation, while the obstacles *locally* push the robot away to avoid possible collisions. Such a property is guaranteed by the fact that each neuron has only local, excitatory lateral connections to its neighboring neurons in the neural network. Therefore, unlike some previous models such as Ilari and Torras (1990), the proposed model is not sensitive to any irrelevant obstacles nor sensor noise. In addition, the computational complexity *linearly* depends on the neural network size.

In the proposed neural network model for robot motion planning, the dynamically varying environment is represented by the dynamic neural activity landscape of the neural network. The *real-time optimal* robot motion is directly planned through the dynamic activity landscape of the neural network without any prior knowledge of the dynamic environment. Unlike previous searching based models whose robot motion is planned by searching over the free space or the collision paths, there are no explicit searching procedures in the proposed model. In contrast to most previous models where the optimal robot motion is generated by optimizing some cost functions, there are no explicit optimization procedures in the proposed model. The optimal robot motion results from the nature of the neural network design. Distinct from most neural network based motion planning models, no learning procedures are needed in the proposed model. Therefore, the model algorithm is computationally *efficient*. The term "real-time" is in the sense that the robot motion planner responds immediately to the dynamic environment, including the robot, target, obstacles and sensor noise. The generated solution to a maze-solving type problem or the planned robot motion in a static environment is globally optimal in the sense of the shortest path from the starting position to the target if it exists. If there exist more than one shortest paths, the proposed model provides only one of them. The optimality of the real-time collision-free motion planning in a nonstationary environment is in the sense that the robot travels a continuous, smooth path toward the target.

2.2 The Model

In this section, the originality of the proposed neural network approach to real-time collision-free robot motion planning is briefly introduced. Then, the philosophy of the proposed neural network approach and the model algorithm are presented. Finally, the stability of the proposed model is proved by using both qualitative analysis and the Lyapunov stability theory.

2.2.1 Originality

Hodgkin and Huxley (1952) proposed a computational model for a patch of membrane in a biological neural system using electrical circuit elements. This modeling work together with other experimental work led Hodgkin and Huxley to a Nobel Prize in 1963, for their discoveries concerning the ionic mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane. In Hodgkin and Huxley's (1952) membrane model, the dynamics of voltage across the membrane, V_m, is described using state equation technique as

$$C_m \frac{dV_m}{dt} = -(E_p + V_m)g_p + (E_{Na} - V_m)g_{Na} - (E_K + V_m)g_K, \tag{2.1}$$

where C_m is the membrane capacitance. Parameters 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 (Hodgkin, 1964; Plonsey and Fleming, 1969).

By substituting $C_m = 1, \xi_i = E_p + V_m, A = g_p, B = E_{Na} + E_p, D = E_k - E_p, S_i^+ = g_{Na}$ and $S_i^- = g_K$ in Equation (2.1), a shunting equation is obtained (Öğmen and Gagné, 1990*a*; Öğmen and Gagné, 1990*b*),

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)S_i^+(t) - (D + \xi_i)S_i^-(t), \tag{2.2}$$

where i is the index of the neuron. Variable ξ_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^+ and S_i^- are the excitatory and inhibitory inputs to the neuron (Öğmen and Gagné, 1990*a*; Öğmen and Gagné, 1990*b*; Yang, 1996). This shunting model was first proposed by Grossberg to understand the real-time adaptive behavior of individuals to complex and dynamic environmental contingencies (Grossberg, 1973; Grossberg, 1982; Grossberg, 1983; Grossberg, 1988), and has a lot of applications in biological and machine vision, sensory motor control, and many other areas (e.g. Grossberg, 1982; Bullock and Grossberg, 1988*a*; Bullock and Grossberg, 1988*b*; Grossberg, 1988; Bullock and Grossberg, 1989; Öğmen and Gagné, 1990*a*; Öğmen and Gagné, 1990*b*; Gaudiano and Grossberg, 1991; Öğmen, 1991; Öğmen, 1993; Li and Öğmen, 1994; Muñiz *et al.*, 1995; Yang and Öğmen, 1995; Zalama *et al.*, 1995; Gaudiano *et al.*, 1996; Maguire and Yang, 1996; Yang, 1996).

2.2.2 Model Algorithm

The fundamental concept of the proposed model to develop a neural network architecture, whose dynamic neural activity landscape represents the dynamically varying 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. The target can globally attract the robot in the whole workspace through the neural activity propagation, while the obstacles have only local effect to avoid collisions. The real-time collision-free robot motion is planned through the dynamic activity landscape of the neural network.

The neural network architecture of the proposed model is a discrete topologically organized map that has been used in many neural network models (e.g. Kohonen, 1982; Linsker, 1986; Zelinsky, 1994; Gambardella and Versino, 1994; Glasius *et al.*, 1995; Podsedkowski, 1998; Kassim and Kumar, 1999). The proposed model is expressed in a finite ($F-$) dimensional ($F-$D) state space $\mathcal{S}$, which is the Cartesian workspace. The location of the i-th neuron at the grid in the F-D state space $\mathcal{S}$, denoted by a vector $p_i \in R^F$, uniquely represents a position in the workspace. 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.

In the proposed neural network model, the dynamics of the i-th neuron in the neural network is characterized by a shunting equation derived from Equation (2.2). The excitatory input S_i^+ results from the target and the lateral connections from its neighboring neurons, while the inhibitory input S_i^- results from the obstacles only. Thus the differential equation for the i-th neuron is given by

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)\left([I_i]^+ + \sum_{j=1}^{N} w_{ij}[\xi_j]^+\right) - (D + \xi_i)[I_i]^-, \qquad (2.3)$$

where N is the total number of neurons in the neural network. The terms $[I_i]^+ + \sum_{j=1}^{N} w_{ij}[\xi_j]^+$ and $[I_i]^-$ are the excitatory and inhibitory inputs, S_i^+ and S_i^- in Equation (2.2), respectively. The external input I_i to the i-th neuron is defined as

$$I_i = \begin{cases} E, & \text{if there is a target} \\ -E, & \text{if there is an obstacle} \\ 0, & \text{otherwise} \end{cases}, \tag{2.4}$$

where $E \gg B$ is a very large positive constant. 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 j-th neuron to the i-th neuron is a function of distance defined as

$$w_{ij} = f(d_{ij}), \tag{2.5}$$

where $d_{ij} = |p_i - p_j|$ represents the Euclidean distance between positions p_i and p_j in the state space $\mathcal{S}$ of the neural network. The connection weight function $f(d_{ij})$ is a monotonically decreasing function, e.g., a function defined as

$$f(d_{ij}) = \begin{cases} \mu/d_{ij}, & \text{if } 0 < d_{ij} < r_0 \\ 0, & \text{if } d_{ij} \geq r_0 \end{cases}, \tag{2.6}$$

where μ and r_0 are positive constants. Therefore, it is obvious that the connection weight w_{ij} is symmetric, $w_{ij} = w_{ji}$, and is a function of distance only, not depending on directions. In addition, the neuron has only local connections in a small region $(0, r_0)$, i.e., its receptive field $\mathcal{R}_i$ is the space whose distance to the i-th neuron is less than r_0. The neurons located within the receptive field of the i-th neuron is referred as its *neighboring neurons*. Therefore, the dynamics of the i-th neuron can be further described as,

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[\xi_j]^+\right) - (D + \xi_i)[I_i]^-, \tag{2.7}$$

where k is the total number of neighboring neurons of the i-th neuron.

The proposed neural network characterized by Equation (2.7) guarantees that the positive neural activity can propagate from one neuron to its neighboring neurons through lateral neural connections, but the negative activity stays locally only. Thus, the target *globally* influences the whole state space to attract the robot through neural activity propagation, while the obstacles have only *local* effect to avoid possible collisions due to no inhibitory lateral neural connections in the neural network. Therefore, unlike some previous models such as Ilari and Torras (1990) where the planned global path is sensitive to irrelevant obstacles, the proposed neural network model is not sensitive to any irrelevant obstacles because the obstacles have only local effect. In addition, the activity propagation from the target is blocked when it hits the obstacles. Such a property is very important for maze-solving type problems.

The positions of the target and obstacles may vary with time. The activity landscape of the neural network dynamically changes according the varying external inputs and the lateral excitatory connections. As shown in Equation (2.7), the neuron responds to only the immediate inputs from the target and obstacles. Therefore no prior knowledge of the varying environment is needed in the proposed model. The real-time robot motion is planned from the dynamic activity landscape by a steepest gradient ascent rule. For a given present position in the state space of the neural network (i.e., a position in the Cartesian workspace, or a neuron in the neural network), denoted by p_p, the next position p_n (also called "command position") is obtained by

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \max\{\xi_j, j = 1, 2, \cdots, k\}, \tag{2.8}$$

where k is the total number of all the neighboring neurons of the p_p-th neuron in the neural network, i.e., all the possible next positions of the present position p_p. After the present position reaches its next position, the next position becomes a new present position. The present position *adaptively* changes according to the varying environment.

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.

2.2.3 Stability Analysis

In the shunting model in Equations (2.2), (2.3) or (2.7), the neural activity ξ_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 ξ_i and B are, the slower ξ_i increases. When the activity ξ_i is below B, the excitatory term is positive causing an increase in the neural activity. If ξ_i is equal to B, the excitatory term becomes zero and ξ_i will no longer increase no matter how strong the excitatory input is. In case the activity ξ_i exceeds B, $B - \xi_i$ becomes negative and the shunting term pulls ξ_i back to B. Therefore, ξ_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 (Yang, 1996).

The stability and convergence of the proposed model can also be rigorously proved using a Lyapunov stability theory. Introducing the new variables, $\eta_i = \xi_i - B$, the proposed model in Equation (2.3) or (2.7) can be written into the general form proposed by Grossberg (1983),

$$\frac{d\eta_i}{dt} = a_i(\eta_i)\left(b_i(\eta_i) - \sum_{j=1}^{N} c_{ij}d_j(\eta_j)\right), \tag{2.9}$$

by the following substitutions:

$$a_i(\eta_i) = -\eta_i, \tag{2.10}$$

$$b_i(\eta_i) = \frac{1}{\eta_i}\left(AB + \eta_i(A + [I_i]^+ + [I_i]^-) + (B + D)[I_i]^-\right), \tag{2.11}$$

$$c_{ij} = -w_{ij}, \tag{2.12}$$

and

$$d_j(\eta_j) = [\eta_j + B]^+. \tag{2.13}$$

Since the neural connection weight is symmetric, $w_{ij} = w_{ji}$, then $c_{ij} = c_{ji}$ (symmetry). Since η_i varies within the finite interval $[-B - D, 0]$, where B and D are nonnegative constants, then η_i is a nonpositive number. Hence the amplification function $a_i(\eta_i)$ is nonnegative, i.e., $a_i(\eta_i) \geq 0$ (positivity). From the definition of function $[a]^+$, have $d_j'(\eta_j) = 0$ at $\eta_j < -B$ and $d_j'(\eta_j) = 1$ at $\eta_j > -B$. Hence the signal function $d_j(\eta_j)$ has a nonnegative derivation, i.e., $d_j'(\eta_j) \geq 0$ (monotonicity). Therefore, Equation (2.7) satisfies all the three stability conditions required by Grossberg's general form (Grossberg, 1983; Grossberg, 1988). The Lyapunov function candidate for Equation (2.9) can be chosen as

$$v = -\sum_{i=1}^{N} \int^{\xi_i} b_i(\eta_i) d_i'(\eta_i) d\eta_i + \frac{1}{2} \sum_{j,k=1}^{N} c_{jk} d_j(\eta_j) d_k(\eta_k). \tag{2.14}$$

The derivative of v along all the trajectories is given by

$$\frac{dv}{dt} = -\sum_{i=1}^{N} a_i d_i' \left(b_i - \sum_{j=1}^{N} c_{ij} d_j \right)^2 . \tag{2.15}$$

Since $a_i \geq 0$ and $d_i' \geq 0$, then $dv/dt \leq 0$ along all the trajectories. The rigorous proof of the stability and convergence of Equation (2.9) can be found in Grossberg (1983). Therefore, the proposed neural network system is stable. The dynamics of the network is guaranteed to converge to an equilibrium state of the system.

2.3 Simulation Studies

The proposed neural network model is capable of planning real-time optimal robot motion with obstacle avoidance in a nonstationary environment. The generated solution to a maze-solving type problem or the planned robot motion in a static environment is globally optimal in the sense of the shortest path from the starting position to the target, if it exists. Such a property results from the fact that the connection weight of the neural network is a function of the distance only. The neural activity propagation is omnidirectional, i.e., the propagation from the target to all directions is exactly in the same manners. In a static environment, the neural activity propagation from the target always reach the robot location along the shortest path. For real-time robot motion planning in a nonstationary environment, the optimality is in the sense that the robot travels a continuous, smooth route toward the target. The term "real-time" is in the sense that the robot motion planner responds immediately to the dynamic environment, including the robot, target, obstacles and sensor noise.

In this section the proposed model is applied to motion planning of a point mobile robot in a 2D Cartesian workspace. A small and maneuverable mobile robot can be treated as a point mobile robot, when comparing the size of the robot and its maneuvering possibilities to the size of the free workspace. For example, in practice, a car in the traffic planning in large cities or tanks in field military operations can be treated as point mobile robots. Several cases of robot motion planning are studied, including a maze-solving type problem, motion planning to avoid concave U-shaped obstacles, a moving target tracking problem, and varying obstacles avoiding problem. In the simulation, the responses of the neural network are numerically calculated using the adaptive step-size Runge-Kutta-Fehlberg 4-5 formula.

2.3.1 Motion Planning in a Static Environment

The proposed model is first applied to the obstacle avoidance problem for a set of U-shaped obstacles. Potential field methods and other strictly local obstacle avoidance schemes cannot deal with this type problems (Muñiz *et al.*, 1995; Meng and Yang, 1998; Yang and Meng, 1999*e*). The concave U-shaped obstacles are shown in Figure 2.1A by solid squares. The neural network has 30×30 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 planned robot motion is shown in Figure 2.1A by solid circles. It shows that the planned robot motion path is a continuous, smooth route from the starting position to the target with obstacle avoidance. The stable (the time is long enough) activity landscape of the neural network is shown in Figure 2.1B, where the peak is at the target location, while the valley is at the obstacle location.

The solution to a maze-solving type problem can be treated as a special case

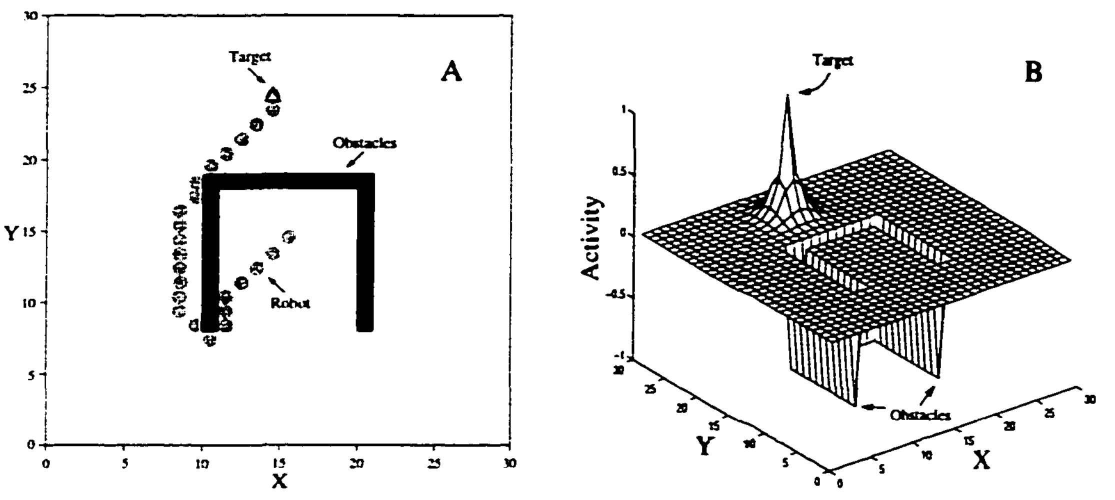


Figure 2.1: *Motion planning of a point mobile robot to avoid a set of concave U-shaped obstacles.* A: *the planned robot motion path;* B: *the stable activity landscape of the neural network.*

of robot motion planning problem in a 2D workspace, along which a point mobile robot can reach the target from a given starting position with obstacle avoidance. The example of the well-known Beam Robot Competition Micromouse Maze (solid squares in Figure 2.2 shows a typical quarter of the maze) is used. Wyard-Scott and Meng (1995) proposed a searching based potential algorithm that can effectively obtain a locally optimized solution to this maze-solving type problem (see Figure 1.1). The proposed neural network model is used to solve the maze type problem shown in Figure 2.2. The neural network has 17 $\times$ 17 neurons. The model parameters are chosen as the same as in the previous case. The generated globally optimal solution is shown in Figure 2.2, where the robot motion path is represented by solid circles. Therefore, the proposed model does not suffer from undesired local minima, i.e., will not be trapped in the deadlock situations, even with the concave U-shaped obstacles and the complex maze-solving type problems.

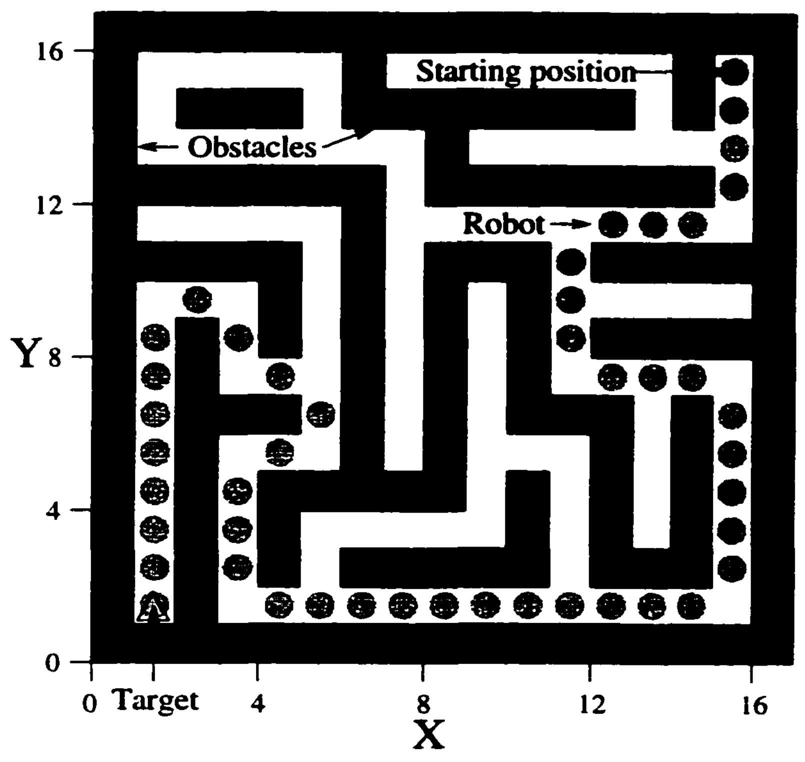


Figure 2.2: *Solution to a maze-solving type problem. The travel route of the robot is represented by solid circles, while the obstacles (wall) are represented by solid squares.*

2.3.2 Motion Planning for Tracking a Moving Target

The proposed model is then applied to a real-time motion planning problem for a point mobile robot tracking a moving target. The neural network has 30×30 neurons with zero initial neural activities. The model parameters are chosen as the same as in previous cases, i.e., $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. In a 2D workspace free of obstacles, the travel route of the target is shown in Figure 2.3A as indicated by hollow triangles, with an initial position at $(X, Y) = (5, 5)$. The target moves at a speed of 25 *block/minute* (it is convenient to assume that the space and time units are block and minute, respectively), and stops at position (25,25) after it arrives there. Note that the proposed neural network dynamically responds to the immediate location of the targets and obstacles. No *prior* knowledge of the dynamically varying environment is needed. The robot starts to move from position

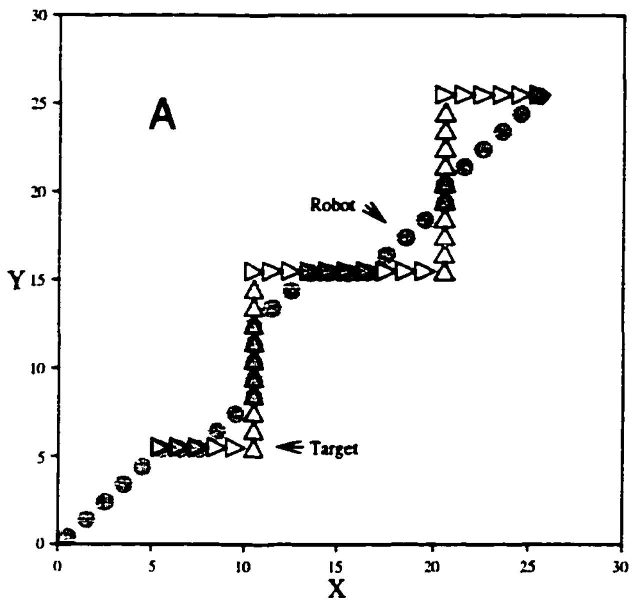
(0,0) at a speed of 10 *block/minute*. The planned robot motion path is shown in Figure 2.3A by solid circles. The activity landscapes of the neural network at two time instants during the motion are shown in Figures 2.3B and 2.3C, where the target arrives at positions (16,16) and (20,21), respectively.

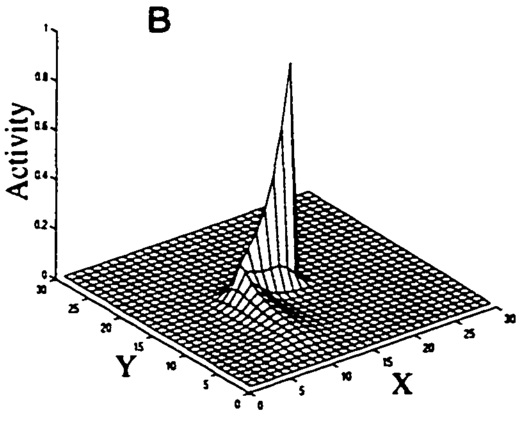
When the robot tracks a moving target, the relative moving speed between the target and the robot is an important factor to influence the robot tacking trajectory. With the same model parameters as in Figure 2.3, Figure 2.4A shows the planned real-time motion path of a robot moving at a speed of 20 *block/minute*, which is twice of that in Figure 2.3. When the robot moves faster than the target at a speed of 30 *block/minute*, the planned real-time robot motion is shown in Figure 2.4B. It shows that the target is caught before it reaches its final position. Comparing Figures 2.3A and 2.4, it is shown that the robot with a slower moving speed takes less steps (not time) to reach the target, since the robot has more time to "wait and see" which position is to go next. However, the faster moving robot spends less time to reach the target. It takes 0.67, 2.04 and 3.06 minutes for the robot at speeds of 30 (Figures 2.4B), 20 (Figures 2.4A) and 10 (Figure 2.3) *block/minute*, respectively, to reach the target.

When there are obstacles in the workspace, the robot is able to follow the target with collision avoidance. Choosing the same model parameters as in Figure 2.3, the planned real-time robot motion with presence of obstacles is shown in Figure 2.5. The robot takes more steps (and time) to reach the target due to the influence of the obstacles. Note that all the robot motion paths in Figures 2.3, 2.4 and 2.5 are continuous, smooth routes. The traveling path of the robot is generally shorter than that of the target.

2.3.3 Motion Planning for Reaching a Moving Target with Moving Obstacles

Then, the proposed model is applied to a more complex case, where both the target and the obstacles are moving. The neural network architecture and the model





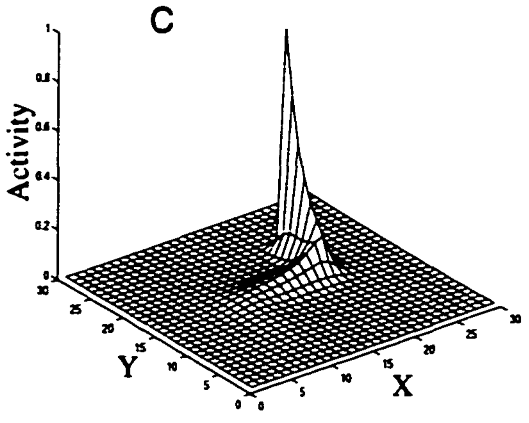


Figure 2.3: *Motion planning of a point mobile robot to track a moving target.* A: *the real-time motion paths of the target (hollow triangles) and the robot (solid circles);* B: *the neural activity landscapes when the target arrives at position (16,16);* C: *when the target arrives at (20,21).*

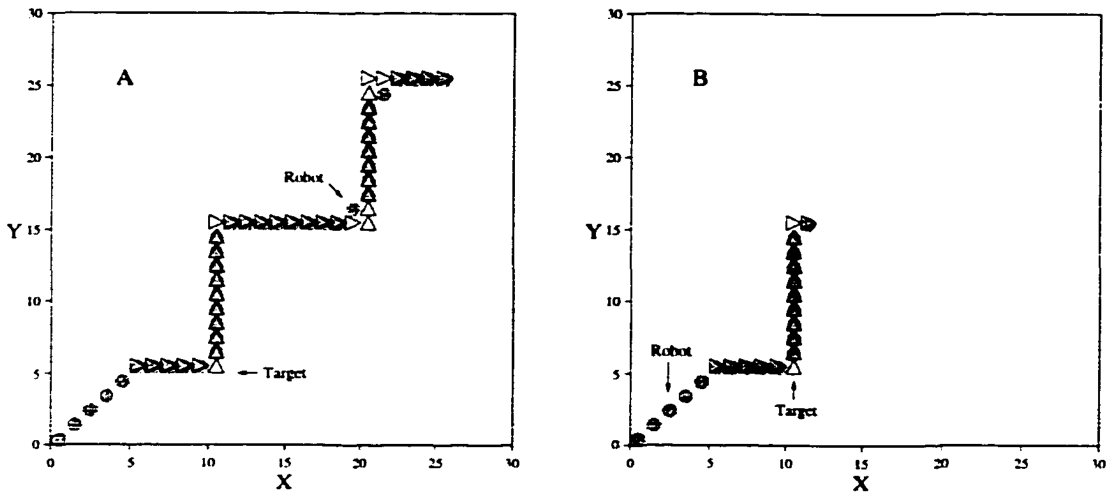


Figure 2.4: *Motion planning of a robot to track a moving target at a faster robot moving speed than in Figure 2.3. The target moves at 25 block/minute. The robot speed in Figure 2.3 is 10 block/minute.* A*: the dynamic robot motion path when the robot moves at 20 block/minute;* B*: the path when robot moves at 30 block/minute.*

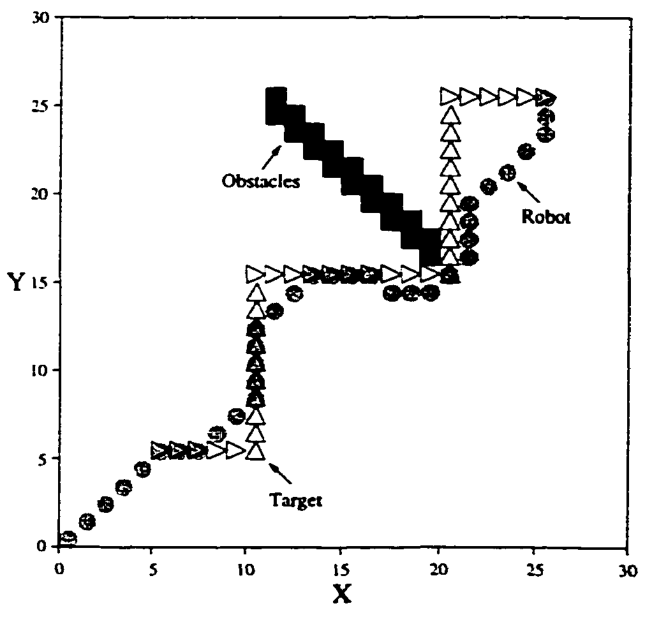


Figure 2.5: *Motion planning of a robot to reach a moving target with presence of obstacles. The obstacles are represented by sold squares.*

parameters are chosen as the same as in the previous cases, i.e., 30×30 neurons, zero initial neural activity, $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. The target starts at position (4,25) and continuously moves back and forth along the line between positions (4,25) and (24,25) at a speed of 10 *block/minute* (shown in Figure 2.6 by hollow triangles). The static obstacles shown in Figure 2.6 by solid squares form two possible channels for the robot to reach the target. In addition, there are 10 movable obstacles. They initially stop for 0.5 *minute* at positions from (5,19) to (14,19) inside the left channel, where they completely block the left channel. Then the obstacles start to move toward the right at a speed of 20 *block/minute*, and finally stop at positions from (14,19) to (23,19), where they completely block the right channel. Note that no *prior* knowledge of the dynamically varying environment is needed in the proposed model, because it dynamically responds to the instantaneous locations of the targets and obstacles. The robot starts to move from position (14,1) at a speed of 20 *block/minute*. The planned robot motion is shown in Figure 2.6 by solid circles. Initially the robot moves toward the right channel, since the left channel is completely blocked while the right channel is open. However, during the time the robot is moving toward the target through the right channel, the obstacles are gradually moving to close the right channel and open the left channel. Before the robot is able to pass through the right channel, the moving obstacles completely block the right channel and leave the left channel completely open. The robot has to move away from the target, passes around the middle static obstacles, and finally catches the moving target through the left channel. Here again the robot travels a continuous, smooth route to catch the target free of collisions.

2.4 Parameter Sensitivity

The sensitivity of a system to parameter variations is a factor of prime importance to be considered when proposing or evaluating a model. An acceptable model should be not very sensitive to the variations in its parameters. There are few parameters in the proposed model: parameters A, B and D for the shunting equation, parameters

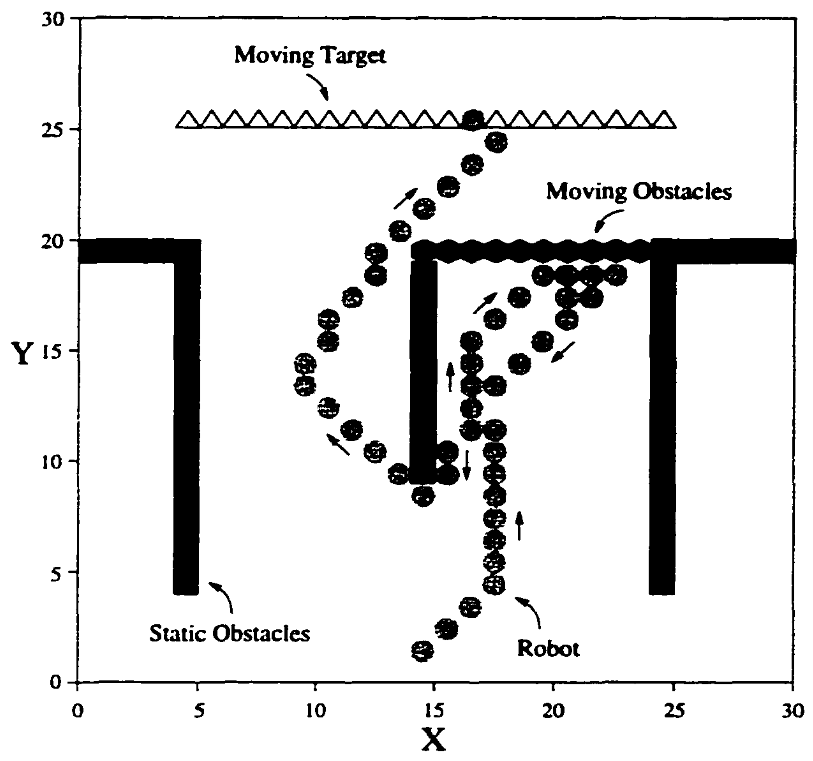


Figure 2.6: *Motion planning of a point mobile robot to reach a moving target with moving obstacles. The moving obstacles are represented by solid hexagons. The target moves back and forth along the hollow triangles.*

μ and r_0 for the lateral neural connections, and parameter E for the external inputs from the target and obstacles. In this section, the sensitivity of parameters for the shunting model, the lateral neural connections and the external inputs will be studied by qualitative analysis and quantitative simulations.

2.4.1 Parameters for the Shunting Equation

In the shunting equation (2.2), for simplification without losing generality, assume the excitatory and inhibitory inputs are step signals with different onset times, i.e., $S^+(t) = S_e u(t - t_e)$ and $S^-(t) = S_i u(t - t_i)$, where S_e, S_i, t_e and t_i are positive constants, and $u(t)$ is the unit step function. The steady-state neural activity ξ_i is given by

$$\xi_i = \frac{BS_e - DS_i}{A + S_e + S_i}. \tag{2.16}$$

It shows that at steady-state the neural activity ξ_i linearly depends on the parameters B and D, while nonlinearly depends on the parameter A.

To illustrate the parameter sensitivity of the shunting model, a set of simulations is carried out. The parameters for the step inputs are chosen as: $I_e = I_i = 10, t_e = 0.2$ and $t_i = 0.8$. Initially the parameters of the shunting equation are chosen as: $A = 10, B = D = 1$ (a set of parameters used in the simulations in Section 2.3). The neural activity $\xi_i(t)$ with various model parameters is shown in Figure 2.7. The solid curves in all panels correspond to the neural activity obtained without parameter change, i.e., $A = 10, B = D = 1$. In Figure 2.7A, the five curves from the uppermost to the lowermost correspond to the activities obtained with A=1, 5, 10, 50 and 100, respectively. It shows that a larger A value results in a shorter duration to reach its maximum activity and a smaller maximum activity value caused by the excitatory input. In Figure 2.7B, from the uppermost to the lowermost, the five curves correspond to the activities obtained with B=10, 5, 1, 0.5 and 0.1, respectively. It shows that a larger B value causes a larger maximum activity value, but nearly no change in the duration to reach the maximum activity. In Figure 2.7C, the five curves from the uppermost to the lowermost correspond to the activities obtained with D=0.1, 0.5, 1, 5, and 10, respectively. It shows that when D increases, the component caused by the inhibitory input shifts down further, but there is almost no change in the time-constant as well. In addition, Figures 2.7B and 2.7C show that the maximum values caused by the excitatory and inhibitory input is nearly proportional to the B and D values, respectively. Therefore, B and D determine the upper and lower bounds of the activity, respectively. Parameter A determines the time-constant of the system (Yang, 1996).

Therefore, in the shunting equations (2.2) or (2.7), parameters B and D are not important factors to the proposed model, because we concern only the relative value of the neural activity with an arbitrary unit. For example, B and D can be chosen as $B = D = 1$ for all cases. Parameter A determines the transient response to an input

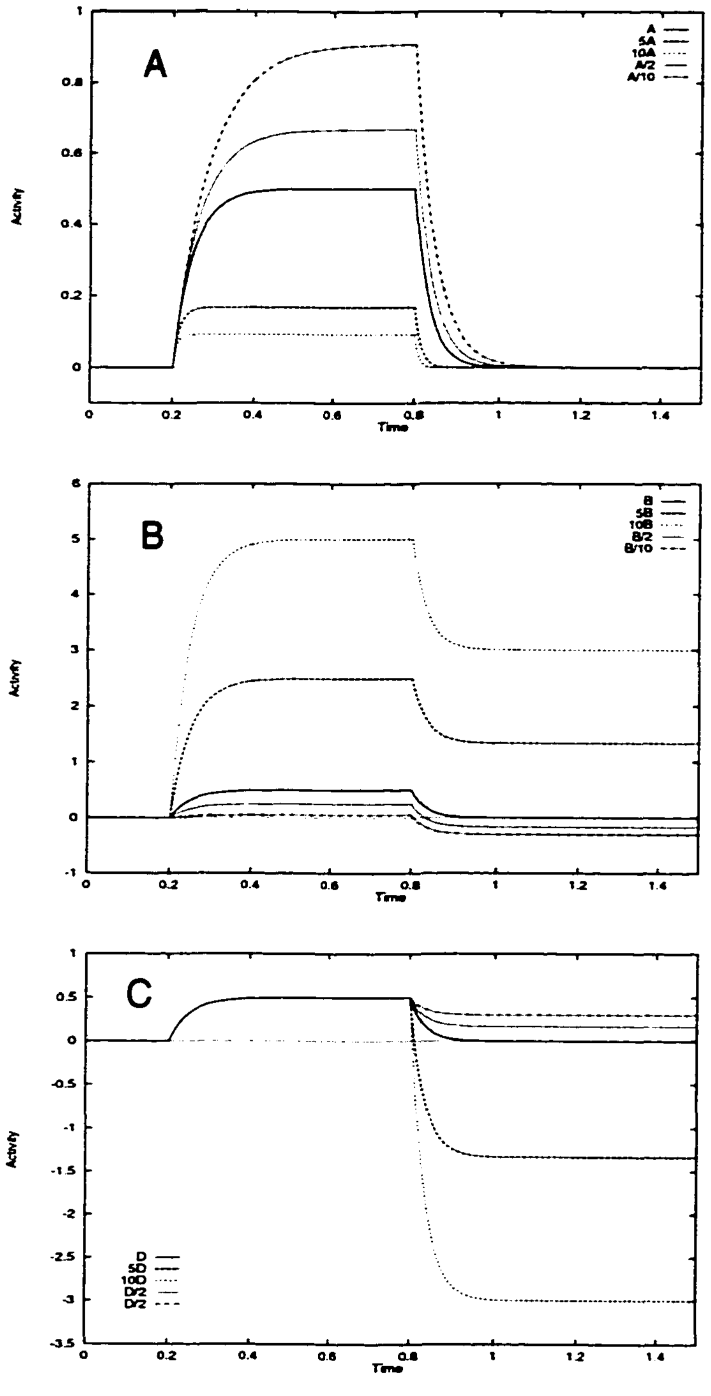


Figure 2.7: *Parameters sensitivity of a shunting equation. The solid curves in all panels are the neural activity with parameters* $A = 10$ *and* $B = D = 1$. A: *activities with* A *= 1, 5, 10, 50 and 100 (from the uppermost to the lowermost), respectively;* B: *with* B*= 10, 5, 1, 0.5 and 0.1 (from the uppermost to the lowermost), respectively;* C: *with* D*= 0.1, 0.5, 1, 5 and 10 (from the uppermost to the lowermost), respectively.*

signal. It plays an essential role in the system dynamics, which is very important when the target and obstacles are varying. To demonstrate the importance of parameter A, we do two simulations under the same condition as the case in Figure 2.3 except choosing two different values of parameter A.

First, a much smaller A value is chosen, $A = 2$ instead of $A = 10$ as in Figure 2.3. Figure 2.8B shows the neural activity landscape of the neural network when the target arrives at position (20,21), which is at the same time instant as in Figure 2.3C. Comparing Figures 2.8B and 2.3C, it shows that the activity landscape in Figure 2.8B has a much longer and wider "tail". This is because that a smaller A value results in a slower passive decaying of the neural activity. The planned robot motion path is shown in Figure 2.8A by solid circles, where the robot follows the target for a few steps and then completely stops at (10,10), failing to finally reach the target. The slow decaying of the activity causes a fast increase of the neural activity due to the lateral excitatory connections among neurons, yielding a quick saturation of the neural activity. When the robot moves to (10,10), the neuron activity over there is saturated. The robot cannot find its next position through the activity difference and has to stop there forever. The activity landscape of the neural network at time = 6.0 *minute* is shown in Figure 2.8C, where the neural activity is saturated. Therefore, when the value of A is too small, this model cannot function properly due to the quick activity saturation.

Then, a much larger A value is chosen, $A = 50$ instead of $A = 10$ as in Figure 2.3. Figure 2.9B shows the neural activity landscape of the neural network when the target arrives at position (20,21), which is at the same time instant as in Figures 2.3C and 2.8B. It shows that the activity landscape has a much shorter "tail" than those in Figures 2.3C and 2.8B. This is because that a larger A results in a faster passive decaying of the neural activity. The planned robot motion path is shown in Figure 2.9A, where the robot takes much less steps to reach the target (the travel route of the robot becomes a straight line). Since the robot moves at the same speed as in Figure 2.3, it certainly takes less time (2.55 minutes, in comparison to 3.06 minutes in Figure 2.3) to catch the target. The faster decaying of the remaining activity after

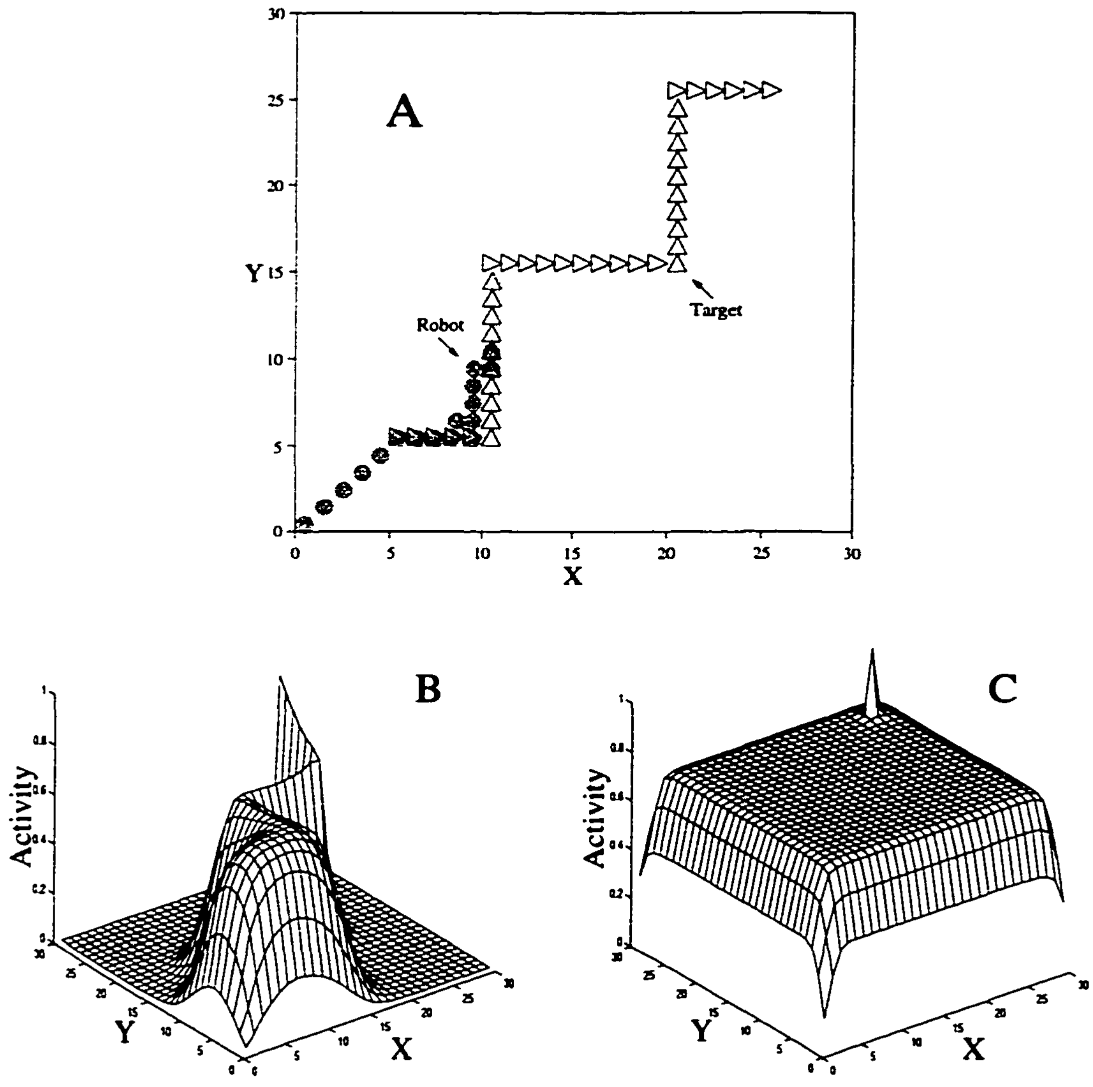


Figure 2.8: *Motion planning with a much smaller A value than in Figure 2.3. $A = 2$ instead of $A = 10$ as in Figure 2.3.* A*: the planned robot motion path;* B*: the neural activity landscapes when the target arrives at position (20,21);* C*: the activity landscape at time= 6.0 minute.*

the target passes away makes the travel "history" of the target disappear faster. The activity propagation from target becomes the domain contribution in forming the neuron activities. Hence, the motion of the robot highly aims at the current position of the moving target. Therefore, choosing a large enough A value is necessary for the robot to aim at the target. However, as we will see later in Figures 2.11A and 2.16, if the robot is required to tightly follow the travel route of the target, a smaller value of parameter A is necessary.

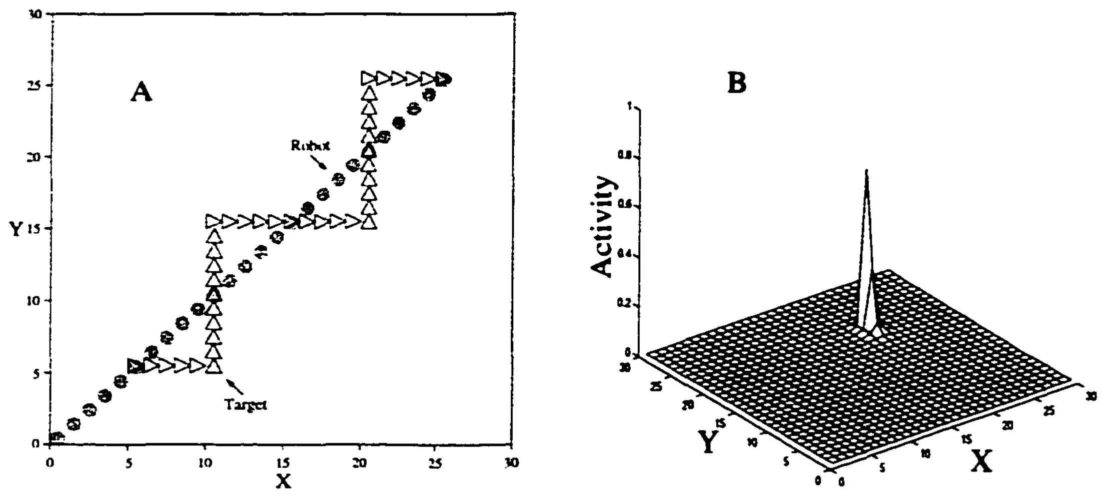


Figure 2.9: *Motion planning with a much larger A value than in Figure 2.3. $A = 50$ instead of $A = 10$ as in Figure 2.3.* A: *the planned robot motion path;* B: *the neural activity landscapes when the target arrives at position (20,21).*

2.4.2 Parameters for the Lateral Neural Connections

Although each neuron has only local connections in a small region and the target is the only positive stimulus, the positive neural activity can propagate to the whole state space of the neural network through neural activity propagation. Therefore the lateral connections among neurons are essential in forming the dynamic neural activity landscape. It shall be pointed out that the proposed model is not sensitive to the connection weight function $f(d_{ij})$ in Equation (2.6), which can be chosen as any monotonically decreasing function. The connection weight is *solely* determined by parameter μ. Therefore, μ is an important factor in the proposed model. To illustrate the role played by μ, a simulation is carried out under the same condition as in Figure 2.3, except choosing a much smaller μ value, $\mu = 0.2$ instead of $\mu = 1$ over there. The neural activity landscape of the neural network when the target arrives at position (20,21) is shown in Figure 2.10B. It shows that the activity landscape has a much narrower "tail" than in Figure 2.3C, since a small μ value results in weak lateral connections. The robot motion is shown in Figure 2.10A, which is very similar to that in Figure 2.3A, since they have the same A value, and thus have the same system dynamics. However, there is a difference: the travel route of the robot in Figure 2.10A takes more steps to reach the target. This is because that a smaller μ value weakens the propagation from the target, and results in a relatively stronger contribution from the remaining activity after the target leaves there. Therefore, to aim at the target, a large enough μ value is necessary.

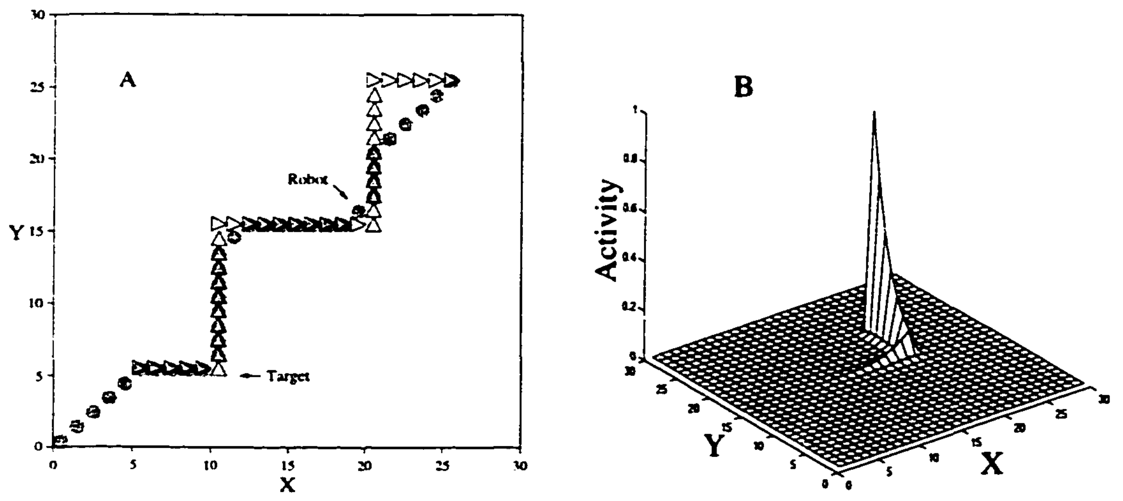
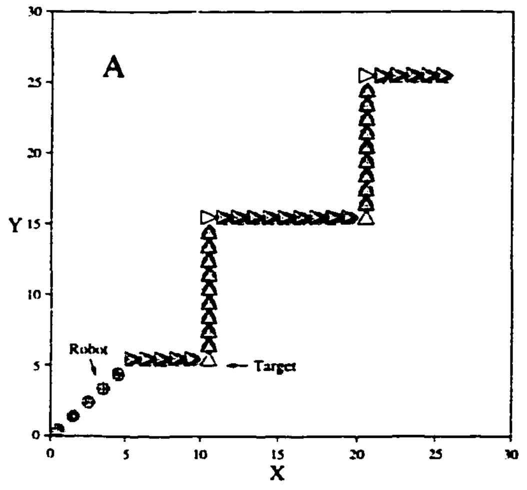


Figure 2.10: *Motion planning with a much smaller μ value than in Figure 2.3. $\mu = 0.2$ instead of $\mu = 1$ as in Figure 2.3.* A*: the planned robot motion path;* B*: the neural activity landscapes when the target arrives at position (20,21).*

To further illustrate the role played by μ, one more simulation is carried out under the same condition as in Figure 2.8, except choosing a much smaller μ value, $\mu = 0.2$

instead of $\mu = 1$ over there. Figure 2.11B shows the neural activity landscape of the neural network when the target arrives at position (20,21). As expected, the "tail" is very narrow in comparison to the very wide "tail" in Figure 2.8B, where they are at the same time instant. It also shows that both of them have a very long "tail", because they have the same very small A value, $A = 2$, which results in a very slow passive decaying of the neural activity. The planned robot motion is shown in Figure 2.11A. It shows that the robot is able to catch the target, whereas the robot in Figure 2.8 fails to do so due to the activity saturation. A small μ value results in weak lateral connections and can prevent the possible saturation in neural activity. In addition, Figure 2.11A shows that the travel route of the robot tightly follows the travel "history" of the target. This result is caused by both the small A value *and* the small μ value: the small A slows down the passive decaying of the neural activity and increases the influence from remaining activity; the small μ weakens the propagation from target activity and decreases the direct influence from the target.



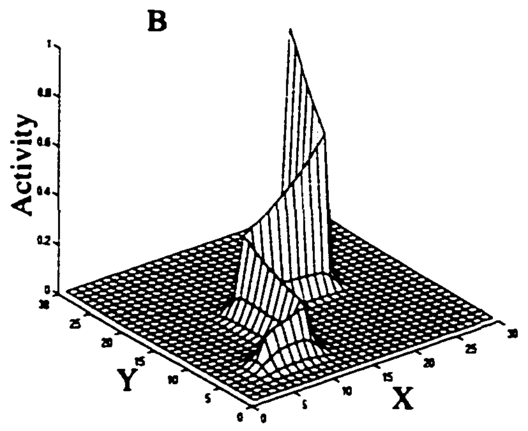


Figure 2.11: *Motion planning with a much smaller μ value than in Figure 2.9. $\mu = 0.2$ instead of $\mu = 1$ as in Figure 2.9.* A: *the planned robot motion path;* B: *the neural activity landscapes when the target arrives at position (20,21).*

When parameter $\mu > 1$, the propagated neural activity is amplified and the neural activity is very easy to saturate. Therefore, to prevent the possible saturation a smaller μ is necessary; to strengthen the influence from the target, a larger μ is needed. Thus, parameter μ is usually chosen in the region $\mu \in (0, 1]$.

Parameter r_0 determines the size of the receptive field of the neuron. It is not an important factor in the proposed model. A larger value r_0 increases the propagation of the neural activity. However, when applying the proposed neural network model to solve maze type problems, a small value is necessary, e.g., $r_0 = 2$, since it is required that the neural activity cannot pass through any obstacles ("wall"). Therefore parameter r_0 can be chosen as $r_0 = 2$ for all cases.

2.4.3 Parameter for the External Inputs

Parameter E determines the amplitude of the external inputs from the target and the obstacles. To keep the target and obstacles staying at the peak and valley, respectively, the value E should be chosen as a very large value over the total input from the lateral neural connections. Since the neural activity is bounded at the interval $[-D, B]$, by choosing $B = D = 1$ and $r_0 = 2$, the maximum total input from lateral connections is 8, then choosing any value $E > 40$ is good enough. Therefore, parameter E is not an important factor in the proposed model.

In summary, only two parameters, A and μ, are fundamentally important in the proposed neural network model for real-time collision-free robot motion planning. The system dynamics is determined by the value of A. Parameter μ determines the activity propagation among neurons. Note that the parameter values in the above simulations is chosen in a very wide range, and all the simulations for various cases in Section 2.3 choose exactly the same model parameters. Therefore, it is apparent that the proposed model is not very sensitive to the model parameter variations.

2.5 Model Variations

The neural network model for robot motion planning characterized by Equation (2.7) has only excitatory lateral neural connections. Based on the same philosophy presented in Section 2.2, a shunting model with only inhibitory lateral connections is developed. In addition, by lumping together the excitatory and inhibitory terms and removing the auto gain control terms in the shunting models, two simple neural network models for real-time collision-free robot motion planning, characterized by the additive equations, are obtained. Finally, a comparison among these models and Glasius *et al.*'s (1995) model is studied by qualitative analysis and quantitative simulation.

2.5.1 A Shunting Model with Only Inhibitory Lateral Connections

The proposed neural network model for real-time robot motion planning with obstacle avoidance is characterized by the shunting equation in (2.7), where there are only excitatory lateral connections amon neurons. As presented in Section 2.2, the fundamental concept is that the dynamically varying environment is represented by the dynamic neural activity landscape that is used for real-time collision-free robot motion planning. In the dynamic activity landscape of the neural network, the target is always at the peak and the obstacles are always at the valley (e.g., see Figure 2.1B). The procedure to plan the real-time robot motion can be viewed as that the robot is climbing up the dynamic neural activity landscape to reach the activity peak. Such a network property is guaranteed by the fact that there are only excitatory connections among neurons.

Alternatively, based on the same philosophy, a different neural network model is proposed, which is characterized by a shunting equation derived from Equation (2.2), where there are only inhibitory connections among neurons. In this inhibitory-lateral-connection shunting model, the total excitatory input S_i^+ in Equation (2.2) results from the obstacles only, while the total inhibitory input S_i^- results from the target and the lateral connections to its neighboring neurons, Thus the dynamics of the i-th neuron activity ξ_i is characterized by

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)[J_i]^+ - (D + \xi_i)\left([J_i]^- + \sum_{j=1}^{k} w_{ij}[\xi_j]^-\right). \qquad (2.17)$$

The definitions of $[a]^+, [a]^-$, w_{ij} and k are the same as those in Equation (2.7). The external input J_i is defined as: $J_i = E$, if there is an obstacle; $J_i = -E$, if there is a target; and $J_i = 0$, otherwise.

The shunting model characterized by Equation (2.17) guarantees that only the negative neural activity can propagate to the other neurons. In addition, the target and obstacles are guaranteed to stay at the valley and peak of the dynamic activity landscape of the neural network, respectively. The real-time robot motion is planned through the dynamic neural activity landscape using a steepest descent rule. For a given present position p_p of the robot in the state space $\mathcal{S}$) of the neural network, the next robot position p_n is obtained by

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \min\{\xi_j, j = 1, 2, \cdots, k\}. \tag{2.18}$$

The present robot position adaptively changes according to the varying environment. The procedure to plan the real-time collision-free robot motion can be viewed as that a ball (the robot) is naturally falling down to reach the valley of the dynamic activity landscape.

The inhibitory-lateral-connection shunting model in Equation (2.17) is a stable system, because the neural activity is bounded in the finite region $[-D, B]$. In addition, the stability and convergence can also be rigorously proved using a Lyapunov stability theory. Introducing the new variables, $\eta_i = \xi_i + D$, i.e., where η_i is a nonnegative number varying in the finite interval $[0, B+D]$, Equation (2.17) can be rewritten into Grossberg's general form in Equation (2.9) via the following substitutions:

$$a_i(\eta_i) = \eta_i, \tag{2.19}$$

$$b_i(\eta_i) = \frac{1}{\eta_i}\left(AD - \eta_i(A + [J_i]^+ + [J_i]^-) + (B + D)[J_i]^+\right), \tag{2.20}$$

$$c_{ij} = -w_{ij}, \tag{2.21}$$

and

$$d_j(\eta_j) = -[\eta_j - D]^-. \tag{2.22}$$

Obviously, have $c_{ij} = c_{ji}$ (symmetry), and $a_i(\xi_i) \geq 0$ (positivity). From the definition of function $[a]^-$, have $d_j'(\eta_j) = 0$ at $\eta_j > D$ and $d_j'(\eta_j) = 1$ at $\eta_j < D$. Hence the signal function $d_j(\eta_j)$ has $d'_j(\eta_j) \geq 0$ (monotonicity). Therefore, Equation (2.17) satisfies all the three stability conditions required by the Grossberg's general form in Equation (2.9) (Grossberg, 1983; Grossberg, 1988). Therefore, this inhibitory-lateral-connection neural network system in Equation (2.17) is stable. The dynamics of the neural network is guaranteed to converge to an equilibrium state of the system.

A case study using the inhibitory-lateral-connection shunting model in Equation (2.17) under the same condition as in Figure 2.1 is simulated. The neural network architecture and all the model parameters are chosen as the same as in Figure 2.1. The planned robot motion is exactly the same as in Figure 2.1A. The stable (time is long enough) neural activity landscape is shown in Figure 2.12. In contrast to Figure 2.1B, the activity peak is at the obstacle location, while the valley is at the target location.

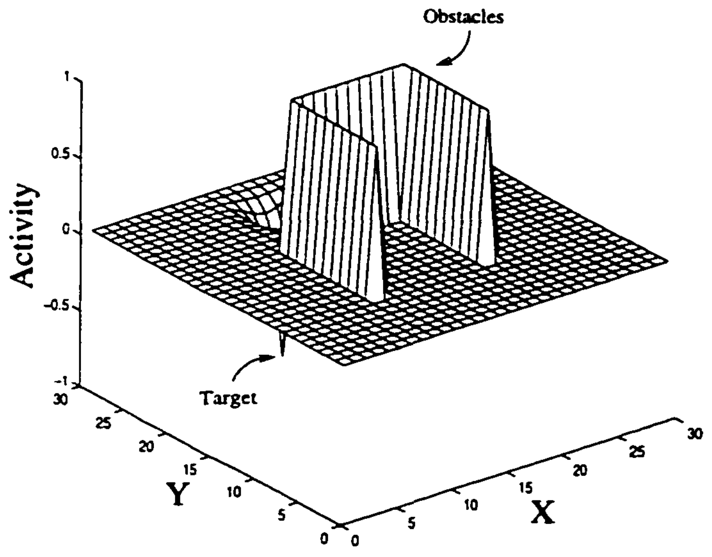


Figure 2.12: *The stable neural activity landscape of the neural network using the inhibitory-lateral-connection shunting model in Equation (2.17).*

2.5.2 Simple Additive Models

If the excitatory and inhibitory terms in the excitatory-lateral-connection shunting equation in Equation (2.7) are lumped together and the auto gain control terms are removed, then Equation (2.7) can be written into a simpler form

$$\frac{d\xi_i}{dt} = -A\xi_i + I_i + \sum_{j=1}^{k} w_{ij}[\xi_j]^+. \tag{2.23}$$

This is an additive equation first proposed by Grossberg (1982), which is widely applied to a lot of areas such as vision, associative pattern learning and pattern recognition (Grossberg, 1982; Grossberg, 1988). The term $I_i+\sum_{j=1}^{k} w_{ij}[\xi_j]^+$ represents the total input to the i-th neuron from the external and lateral connections. The definitions of I_i, $[a]^+$ and w_{ij} are the same as those in Equation (2.7). The nonlinear function $[a]^+$ guarantees that only the positive neural activity can propagate to the other neurons. In addition, because of the very large external input constant $E >> B$, the target and obstacles are guaranteed to stay at the peak and the valley of the dynamic neural activity landscape, respectively. Therefore, this simple additive model also satisfy the fundamental concept of the proposed approach to real-time collision-free robot motion presented in Section 2.2. The procedure to plan the real-time robot motion is the same as the procedure presented in Section 2.2, which is described in Equation (2.8). This additive model is capable of planning real-time robot motion with obstacle avoidance in most situations.

Unlike the shunting model whose neural activity is bounded, the neural activity characterized by the additive model in Equation (2.23) does not have any bounds. However, it is easy to prove that this additive neural network model is a stable system using a Lyapunov stability theory. Equation (2.23) can be rewritten into the Grossberg's general form in Equation (2.9) by substituting

$$a_i(\xi_i) = 1, \tag{2.24}$$

$$b_i(\xi_i) = -A\xi_i + I_i, \tag{2.25}$$

$$c_{ij} = -w_{ij}, \tag{2.26}$$

and

$$d_j(\xi_j) = [\xi_j]^+. \tag{2.27}$$

Obviously, have $c_{ij} = c_{ji}$ (symmetry), $a_i(\xi_i) \geq 0$ (positivity), and $d'_j(\xi_j) \geq 0$ (monotonicity). Thus Equation (2.23) satisfies all the three stability conditions required by the Grossberg's general form in Equation (2.9) (Grossberg, 1983; Grossberg, 1988). Therefore this additive neural network system is stable.

A case study using the simple additive model in Equation (2.23) under the same condition as in Figure 2.1 is simulated. The neural network architecture and all the model parameters are chosen as the same as in Figure 2.1. The planned robot motion is exactly the same as in Figure 2.1A. The stable (time is long enough) neural activity landscape is shown in Figure 2.13, which is qualitatively the same as the neural activity landscape in Figure 2.1B, except some quantitative differences. Such a difference is because that the neural activity of the additive model is not bounded.

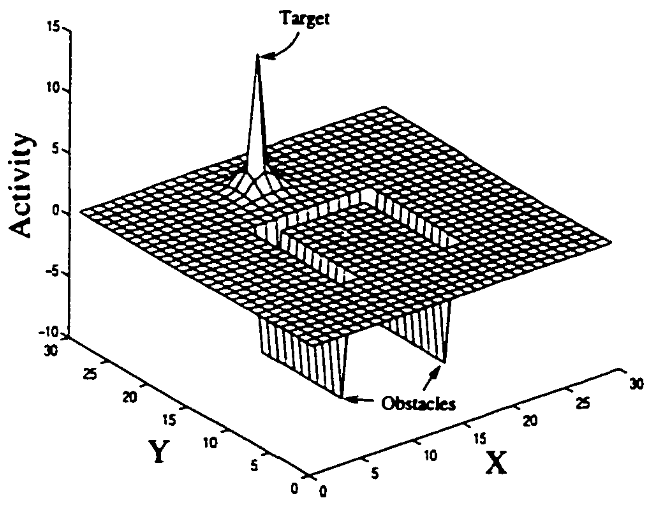


Figure 2.13: *The stable neural activity landscape of the neural network using the additive model in Equation (2.23).*

Similarly, a different additive model for real-time collision-free robot motion planning can be obtained from the inhibitory-lateral-connection shunting model in Equation (2.17). By lumping together the excitatory and inhibitory terms and removing the auto gain control terms in Equation (2.17), the dynamics of the i-th neuron is given by the additive equation,

$$\frac{d\xi_i}{dt} = -A\xi_i + J_i + \sum_{j=1}^{k} w_{ij}[\xi_j]^-, \tag{2.28}$$

where $J_i + \sum_{j=1}^{k} w_{ij}[\xi_j]^-$ is the total input to the i-th neuron from the external and lateral connections. The definitions of J_i, $[a]^-$ and w_{ij} are the same as those in Equation (2.17). The nonlinear function $[a]^-$ guarantees that only the negative neural activity can propagate to the other neurons. In addition, those definitions guarantees that the target and obstacles stay at the valley and peak of the dynamic activity landscape of the neural network, respectively. The procedure to plan the real-time robot motion is the same as the inhibitory-lateral-connection shunting model described in Equation (2.18). This additive model is also capable of planning real-time robot motion with obstacle avoidance in most situations.

The additive neural network in Equation (2.28) can also be rigorously proved to be stable using a Lyapunov stability theory, although its neural activity is not bounded. Equation (2.28) can be rewritten into Grossberg's (Grossberg, 1988) general form in Equation (2.9) by the following substitutions:

$$a_i(\xi_i) = 1, \tag{2.29}$$

$$b_i(\xi_i) = -A\xi_i + J_i, \tag{2.30}$$

$$c_{ij} = w_{ij}, \tag{2.31}$$

and

$$d_j(\xi_j) = -[\xi_j]^-. \tag{2.32}$$

Again, we have $c_{ij} = c_{ji}$ (symmetry), $a_i(\xi_i) \geq 0$ (positivity), and $d'_j(\xi_j) \geq 0$ (monotonicity). Therefore, all the three stability conditions required by the general form in Equation (2.9) are satisfied by Equation (2.28) and this additive neural network system is stable.

There are a lot of important differences between the shunting models (e.g., Equations (2.7) and (2.17)) and the additive models (e.g., Equations (2.23) and (2.28)), although the additive model is computationally simpler and can also plan real-time robot motion with obstacle avoidance in most situations. First, by rewriting them into the Grossberg's general form in Equation (2.9), unlike the additive model in Equation (2.23) with a constant $a_i(\xi_i) = 1$ and a linear function $b_i(\xi_i)$, for the shunting model in Equation (2.7) the amplification function $a_i(\eta_i)$ in Equation (2.10) is not a constant, and the self-signal function $b_i(\eta_i)$ in Equation (2.11) is nonlinear. Second, the shunting model in Equation (2.7) has two *auto gain control* terms, $(B - \xi_i)$ and $(D + \xi_i)$, which result in that the dynamics of Equation (2.7) remains sensitive to input fluctuations (Grossberg, 1988). Such a property is important for the real-time robot motion planning when the target and obstacles are varying. In contrast, the dynamics of the additive equation may saturate in many situations (Grossberg, 1988). Third, the activity of the shunting model is bounded in the finite interval $[-D, B]$, while the activity in the additive model does not has any bounds. A detailed analysis of the shunting model and the additive model can be found in Grossberg (1988), Öğmen and Gagné (1990*a*), Öğmen and Gagné (1990*b*) and Yang (1996).

Glasius *et al.* (1995) proposed a Hopfield type neural network model for real-time trajectory generation, where the *output* z_i of the i-th neuron is modeled by

$$\frac{dz_i}{dt} = -z_i + g\left(I_i + \sum_{j=1}^{N} w_{ij} z_j\right), \tag{2.33}$$

which is derived from a discrete input-output model,

$$z_i(t+1) = g\left(I_i + \sum_{j=1}^{N} w_{ij} z_j(t)\right), \tag{2.34}$$

where $g(a)$ is an input-output transfer function that can be any sigmoid function, e.g., a function defined as

$$g(a) = \begin{cases} 0, & \text{if } a < 0 \\ \gamma a, & \text{if } 0 \le a \le 1 \\ 1, & \text{if } a > 1 \end{cases}, \tag{2.35}$$

where γ is a constant, $\gamma \in [0, 1]$. The connection weight w_{ij} is defined as a function of the distance d_{ij} between i-th and j-th neurons, e.g., $w_{ij} = 1$, if $d_{ij} < 2$; $w_{ij} = 1$, otherwise (from Glasius *et al.*, 1996).

Comparing between Glasius *et al.*'s (1995) model in Equation (2.33) and the proposed shunting model in Equation (2.7) or simple additive model in Equation (2.23), the most important difference is that Equation (2.33) has a constant passive decay rate at $A = 1$. As discussed previously, parameter A plays an essential role in the real-time robot motion planning. Although parameters μ in Equation (2.23) or γ in Equation (2.33) can prevent the possible saturation in the neural activity, they do not have any effects to the transient response characteristics of the model, i.e., the system dynamics. Therefore, Glasius *et al.*'s (1995) model has limitations with fast dynamic systems. It cannot perform properly in a fast changing environment. For example, it requires that the dynamics of the robot is faster than that of the target and the obstacle (Glasius *et al.*, 1995; Glasius *et al.*, 1996).

Another major difference between Glasius *et al.*'s (1995) model in Equation (2.33) and the proposed shunting model in Equation (2.7) or simple additive model in Equation (2.23) is that Equation (2.33) describes the dynamics of the neuron *output* z_i, which is derived from a simple discrete input $(I_i + \sum w_{ij}z_j)$ -output (z_i) function; while Equations (2.7) or (2.23) characterize the dynamics of the neural *activity* ξ_i, which is derived from Hodgkin and Huxley's (1952) biological model, and describes the relationship among the input (S_i^+ and S_i^-), output ($[\xi_i]^+$) and activity (ξ_i) of a neuron. In addition, unlike Glasius *et al.*'s (1995) model in Equation (2.33) which does not model the neural activity, the proposed shunting model has a continuous neural activity with both upper and lower bounds. Therefore the proposed model is more biologically plausible.

By doing a linear, invertible mathematical transformation, $u_i = I_i + \sum w_{ij}z_j$, where u_i is the input to the i-th neuron, Equation (2.33) can be rewritten into the following form,

$$\frac{du_i}{dt} = -u_i + I_i + \sum_{j=1}^{N} w_{ij}g(u_j). \tag{2.36}$$

This equation describes the dynamics of the ***input*** to the i-th neuron (biologically this is not meaningful). Therefore, Equation (2.36) is similar to the proposed Equation

(2.23) in the special case of a constant passive decay rate $A = 1$. This transformation from a nonlinear signal function of sum ($g(\sum z_j)$) to a sum of nonlinear signals ($\sum g(\xi_j)$) is usually called "S Σ exchange" (signal-sum exchange) in neural network analysis.

2.5.3 A Comparison Study among Models

A set of computer simulations is carried out to illustrate the differences among these models. A target catching case is designed. In a 2D Cartesian workspace, the target starts to move from position (10,5) at a speed of 25 *block/minute*. The travel route of the target is shown in Figure 2.14 by hollow triangles. After target reaches (20,29) at time=3.0 *minute*, it disappears, i.e., the target leaves the workspace. The robot moves at a speed of 20 *block/minute* starting from (0,0). The task of the robot is to *catch the target before the target disappears*. Since the robot moves slower than the target, it can catch the target only if it can find a shorter travel route than the target.

First, the proposed additive model in Equation (2.23) is used. The neural network has 30×30 neurons, and the parameters A and μ are chosen as $A = 1$ and $\mu = 0.01$. All the other model parameters are chosen the same as in previous cases, i.e., $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$, which are also used in the following simulations of the proposed models. The activity landscape of the neural network when the target leaves the workspace is shown in Figure 2.14B. As expected, it has a very long and very narrow "tail", since a very small A value and a very small μ value are used. The planned robot motion is shown in Figure 2.14A by solid circles. It shows that the robot ends at (12,5) and fails to catch the target. This is caused by the limitations of the additive equation.

Second, Glasius *et al.*'s (1995) model in Equation (2.33) is used. Parameter γ is chosen as $\gamma = 0.1$ (from Glasius *et al.*, 1996). The planned robot motion is shown Figure 2.15A, where the robot ends at (20,6) and fails to catch the target. The activity landscape when the target leaves the workspace is shown in Figure 2.15B. This result is qualitatively the same as that in Figure 2.14 except some quantitative differences.

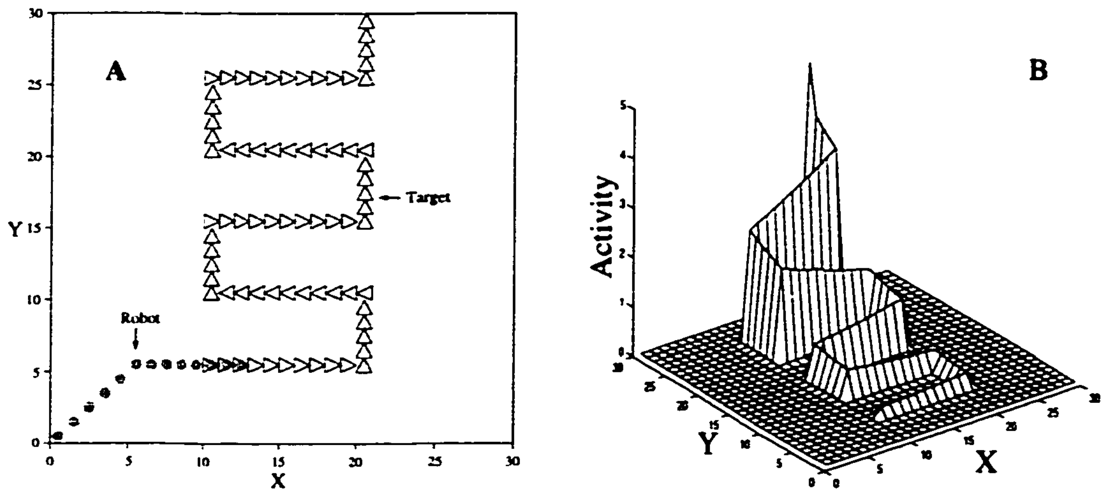


Figure 2.14: *Motion planning of a point mobile robot to catch a moving target using the additive model in Equation (2.23). $A = 1$ and $\mu = 0.01$.* A*: the planned robot motion path;* B*: the neural activity landscape when the target leaves the workspace.*

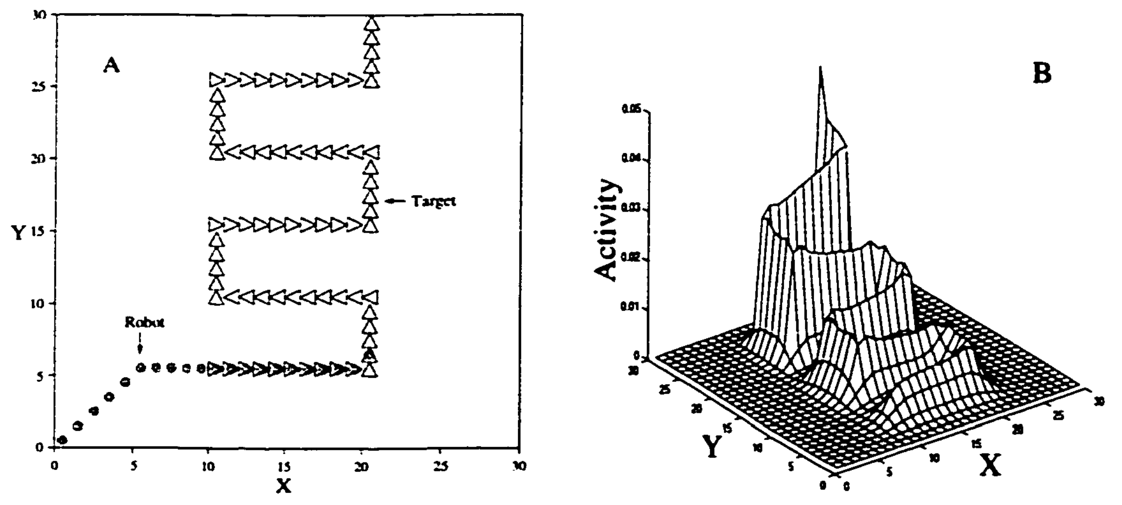


Figure 2.15: *Motion planning of a robot to catch a moving target using the additive model in Equation (2.33).* A*: the planned robot motion path;* B*: the neural activity landscape when the target leaves the workspace.*

Third, the proposed shunting model in Equation (2.7) is used with $A = 1$ and $\mu = 0.1$. The neural activity landscape when the target leaves the workspace is shown in Figure 2.16B. Unlike those in Figures 2.14B and 2.15B, the neural activity in Figure 2.16B is bounded in $[0, 1]$, although their activity landscapes are qualitatively the same. The planned robot motion is shown in Figure 2.16A. It shows that the robot tightly follows the travel route of the target. However, the robot also fails to catch the target before the target leaves the workspace, because it does not travel a shorter route than the target. If the task is to detect and follow the travel route of the target, the robot does a very good job. Unlike the cases in Figures 2.14A and 2.15A where the robot ends at a location forever, the robot in Figure 2.16A can continuously follow the travel route of the target. This result demonstrates the advantage of the shunting model.

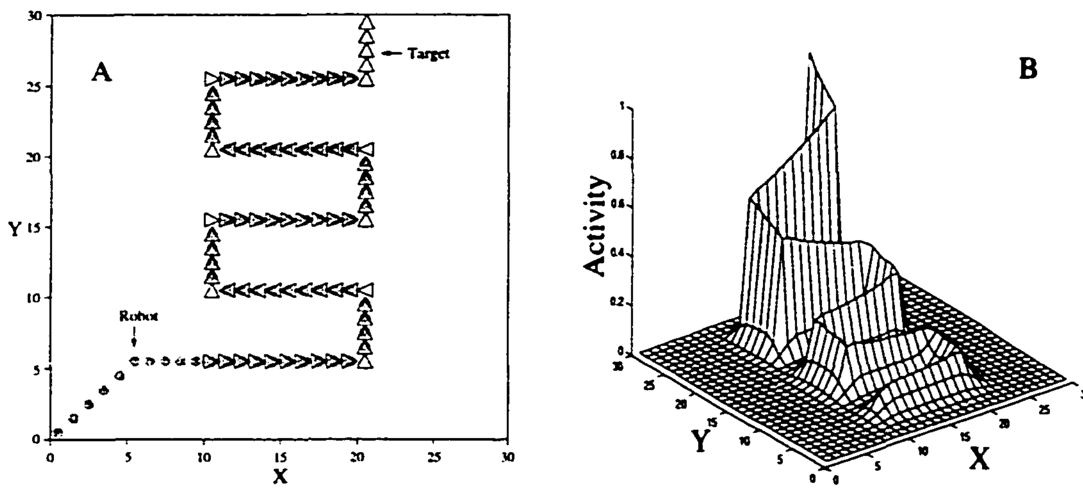


Figure 2.16: *Motion planning of a robot to catch a moving target using the shunting model in Equation (2.7). $A = 1$ and $\mu = 0.1$.* A: *the planned robot motion path;* B: *the neural activity landscape when the target leaves the workspace.*

Forth, the proposed additive model in Equation (2.23) is used with $A = 50$ and $\mu = 1$. The planned robot motion is shown in Figure 2.17A. It shows that the robot travels a *shorter* route than the target and catches the target at position (11,15). The neural activity landscape when the robot catches the target is shown in Figure 2.17.

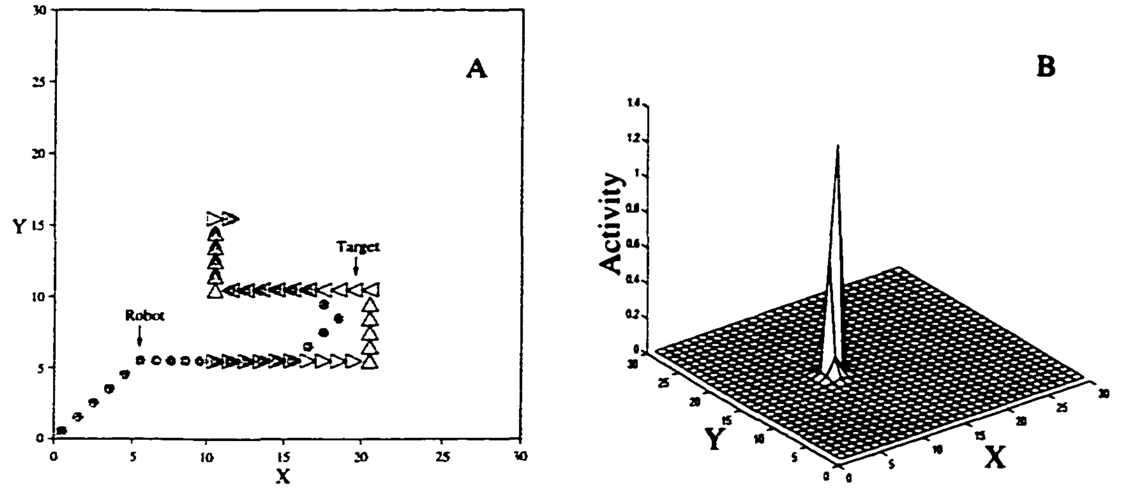


Figure 2.17: *Motion planning of a robot to catch a moving target using the additive model in Equation (2.23)* $A = 50$ *and* $\mu = 1$. A: *the planned robot motion path;* B: *the activity landscape when the robot catches target.*

As expected, it has a very short "tail".

Fifth, the proposed shunting model in Equation (2.7) is used with the same parameters as in Figure 2.17, i.e., $A = 50$ and $\mu = 1$. The planned robot motion path is the same as that shown in Figure 2.17A. The neural activity landscape when the robot catches the target is qualitatively the same as that in 2.17B.

In summary, the above simulations demonstrate that parameter A is fundamentally important in real-time robot motion planning, particularly when the environment is changing in a fast manner. The neural dynamics of the additive models may saturate in some situations.

2.6 Conclusion

In this chapter, a novel biologically inspired neural network approach is developed for real-time robot motion planning with obstacle avoidance in an arbitrarily dynamic environment. Based on the same philosophy, three model variations are presented and the differences are compared by descriptive analysis and quantitative simulation. The *optimal real-time* robot motion is planned through the dynamic activity landscape of the neural network. The global stability and convergence of the proposed models are guaranteed by the qualitative analysis and the Lyapunov stability analysis. Some points are worth mentioning about the proposed model:

- The model algorithm is computationally *efficient*. The real-time optimal robot motion is planned *without* explicitly searching over the free workspace or the collision paths, *without* explicitly optimizing any cost functions, *without* any prior knowledge of the dynamic environment, and *without* any learning procedures.

- The computational complexity *linearly* depends on the state space size of the neural network. Each neuron in the neural network has only *local* connections, which does not depend on the size of the overall neural network.

- This model is biologically plausible. It is derived from Hodgkin and Huxley's (1952) biological membrane model. 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; Marcus *et al.*, 1990).

- This model is not very sensitive to the model parameters nor the connection weight function. Only two model parameters, A and μ, are important factors. The model parameters can be chosen in a very wide range. The weight function can be any monotonically decreasing function.

- This model is not sensitive to any irrelevant obstacles. There are no inhibitory lateral connections in the neural network. The negative neural activity from the obstacle location stays locally only without propagating to any other neurons. Thus the obstacles have only local effect to push the robot away to avoid collisions. Therefore, unlike some previous models (e.g. Ilari and Torras, 1990) where the planned global path is sensitive to irrelevant obstacles, the irrelevant obstacles do not influence the robot motion planning.

- The proposed models do not suffer from undesired local minima, i.e., it will not be trapped in the deadlock situations, e.g., even with the concave U-shaped obstacles and the complex maze-solving type problems. The target is the only source of positive neural activities. The target globally influences the whole workspace through neural activity propagation. The neural activity propagates to all directions exactly in the same manners. There is no activity source from inside the deadlock.

Chapter 3

A Novel Neural Network Approach to Real-time Motion Planning with Safety Consideration

An novel neural network approach, based on the biologically inspired neural network model presented in Chapter 2 for motion planning of a point mobile robot, is proposed for real-time motion planning with safety consideration in a nonstationary environment. In the proposed shunting neural network model, the state space is extended to either the Cartesian workspace or the joint space of a multi-joint robot manipulator. Both excitatory and inhibitory lateral connections in the neural network and the threshold of neural connections are used in the proposed algorithms. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network that incorporates the clearance from obstacles. The proposed neural network model is capable of planning real-time "comfortable" robot motion that does not suffer from either the "too close" problem nor the "too far" problem. The stability and convergence of the proposed neural network system are proved using qualitative analysis and the Lyapunov stability theory. The effectiveness and the efficiency of the proposed model are demonstrated through simulation and comparison studies.

3.1 Introduction

Safety consideration is very important in robot motion planning. The clearance from obstacles should not be ignored (Thorpe, 1984; Zelinsky, 1994; Yang and Meng, 1998*c*; Yang and Meng, 1999*b*). Many models for path planning concentrate on minimizing the distance between the starting position and target (e.g. Lozano-Pérez, 1983; Brooks and Lozano-Pérez, 1985; Donald, 1987; Payton, 1990; Bander and White, 1998; Wang, 1998; Saab and VanPutte, 1999). In a static environment, the planned robot motion by some neural network models (e.g. Glasius *et al.*, 1994; Glasius *et al.*, 1995; Glasius *et al.*, 1996; Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1999*e*) has the shortest path as well, although they do not explicitly minimize any cost functions. They assume that the shortest path is the "best" path. The clearance from obstacles is not considered during the motion planning. Therefore, the planned path clips the corners of obstacles and runs down the edges of obstacles. This is the so called "too close" problem (narrow safety margin) (Thorpe, 1984; Zelinsky, 1994; Yang and Meng, 1998*c*; Yang and Meng, 1999*b*). Such a "too close" problem can be avoided by expanding the obstacles by an extra size, but some possible solution paths are blocked. Therefore, this strategy is not acceptable, particularly when all the possible solution paths are blocked after the expansion (Zelinsky, 1994; Yang and Meng, 1999*b*).

On the other hand, some models (e.g. Ilari and Torras, 1990) maximize the clearance from obstacles while minimizing the distance from the starting position to the target. The found path passes through the middle of free space. Therefore it may deviate significantly from the shortest path. This is the so called "too far" problem (waste) (Thorpe, 1984; Zelinsky, 1994; Yang and Meng, 1999*b*; Yang and Meng, 1999*h*). Several models (e.g. Noborio *et al.*, 1989; Zhu and Latombe, 1991; Zelinsky, 1994) were proposed to solve or reduce the "too far" and/or the "too close" problems. For example, Zelinsky (1994) proposed a path transform model for finding a neither "too far" nor "too close" path in a static environment by combining the distance transform and the obstacle transform.

In this chapter, a novel neural network approach for real-time motion planning with safety consideration in a nonstationary environment is proposed, based on the biologically inspired neural network model for motion planning of a point mobile robot presented in Chapter 2 (also in Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1999*e*). The dynamics of each neuron in the neural network is characterized by a shunting equation or an additive equation derived from the previous model for a point mobile robot. Unlike those models presented in Chapter 2 that has either excitatory or inhibitory lateral connections among neurons, there are *both* excitatory *and* inhibitory connections in the proposed model (Yang and Meng, 1999*b*). In addition, the threshold of neural connections is introduced, which guarantees that the negative neural activity is confined in a small region only. Furthermore, the state space of the topographically ordered neural network is extended to either the Cartesian workspace or the joint space of a multi-joint robot manipulator. The proposed model is capable of planning real-time "comfortable" motion path of a point mobile robot or a robot manipulator without suffering from the "too close" problem and/or the "too far" problem (Yang and Meng, 1999*b*). The stability of the proposed neural network system is proved by the Lyapunov stability analysis.

The proposed neural network model for robot motion planning with safety consideration Inherits advantages from the previous model for motion planning of a point mobile robot. There are only local connections among neurons, thus the computational complexity linearly depends on the neural network size. The real-time optimal robot motion is planned through the dynamic neural activity landscape, which integrates the clearance from obstacles, *without* any prior knowledge of the dynamic environment, *without* explicitly searching over the free workspace or the collision paths, *without* explicitly optimizing any cost functions, and *without* any learning procedures. Therefore the model algorithm is computationally *efficient*. When there is no clearance from obstacles, the planned robot motion path is one of the shortest paths from the starting position to the target (Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1999*a*; Yang and Meng, 1999*e*), which is referred as the shortest path that may suffer from the "too close" problem. When the strength of the clearance from obstacles is strong, the planned robot motion keeps away from the obstacles as far as possible while reaching the target, which is referred as the safest path that may suffer from the "too far" problem. When a suitable strength of the obstacle clearance is selected, the planned robot motion does not suffer from either the "too close" problem nor the "too far" problem, which is referred as a comfortable path of the robot.

3.2 The Model

The basic concept of the proposed neural network model for real-time robot motion planning with safety consideration is to enlarge the obstacle influence by employing inhibitory later neural connections, and to confine the obstacle influence to a desired small region by employing threshold of the neural connections in the neural network. The strength of the clearance from obstacles is adjustable by changing the relative inhibitory neural connection weight and the threshold of the inhibitory connections in the neural network.

The neural network architecture of the proposed model for real-time robot motion planning with safety consideration is a discrete topographically organized map. The proposed model is expressed in a finite ($F-$) dimensional ($F-$D) state space. Unlike those neural network models presented in Chapter 2 where the state space is the Cartesian workspace of the point mobile robot, the state space of the proposed neural network is either the Cartesian workspace or the joint space of a multi-joint manipulator. For example, a 2D workspace has $F = 2$; a 6 d.o.f. robot manipulator has $F = 6$. The location of the i-th neuron at the grid in the F-D state space, denoted by a vector $p_i \in R^F$, represents a position in the Cartesian workspace or a configuration in the robot joint space. Like the previous neural network models, each neuron has a local lateral connections to its neighboring neurons that constitute a subset in the state space, which is the receptive field of the neuron.

In contrast to those neural network models presented in Chapter 2 where there are only excitatory *or* inhibitory lateral connections in the neural network, there are *both* excitatory *and* inhibitory lateral neural connections in the proposed neural network model for robot motion planning with safety consideration. The target globally attracts the robot through neural activity propagation, while the obstacles push the robot away only locally in a small region to take into account of the clearance from obstacles. The dynamics of the i-th neuron in the neural network is given by a shunting equation,

$$\begin{aligned}\frac{d\xi_i}{dt} &= -A\xi_i + (B-\xi_i)\Big([I_i]^+ + \sum_{j=1}^{N} w_{ij}[\xi_j]^+\Big) \\ &\quad -(D+\xi_i)\Big([I_i]^- + \sum_{j=1}^{N} v_{ij}[\xi_j-\sigma]^-\Big), \end{aligned} \tag{3.1}$$

where N is the total number of the neurons in the neural network. Parameters A, B and D are nonnegative constants representing the passive decay rate, the upper and lower bounds of the neural activity, respectively. Variable ξ_i is the neural activity of the i-th neuron, which has a continuous value in the finite interval $\xi_i \in [-D, B]$ The external input I_i to the i-th neuron is defined as the same as in Equation (2.7), i.e., $I_i = E$, 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 excitatory input S_i^- in Equation (2.2) is given by $[I_i]^+ + \sum_{j=1}^{N} w_{ij}[\xi_j]^+$, which is the total input from the target and the lateral neural connections in the neural network. In contrast to the neural network model presented in Chapter 2 where the inhibitory input is from *only* the obstacles, in the proposed model the inhibitory input S_i^- in Equation (2.2) is given by $[I_i]^- + \sum_{j=1}^{N} v_{ij}[\xi_j - \sigma]^-$, which is from the obstacles *and* the later neural connections in the neural network. Function $[a]^+$ is defined as $[a]^+ = \max\{a, 0\}$, and $[a]^-$ is defined as $[a]^- = \max\{-a, 0\}$. The weights of the excitatory and inhibitory connections, w_{ij} and v_{ij}, from the j-th neuron to the i-th neuron are defined as

$$w_{ij} = f(d_{ij}) \tag{3.2}$$

and

$$v_{ij} = \beta w_{ij}, \tag{3.3}$$

respectively, where β is a positive constant, $\beta \in [0,1]$, and $d_{ij} = |p_j - p_i|$ represents the Euclidean distance between positions p_j and p_i in the state space. Function $f(d_{ij})$

is a monotonically decreasing function, e.g., a function defined as $f(d_{ij}) = \mu/d_{ij}$, if $0 < d_{ij} < r_0$; $f(d_{ij}) = 0$, otherwise, where μ and r_0 are positive constants. Therefore, it is obvious that the weights w_{ij} and v_{ij} are symmetric. The neuron has only local connections in a small region $(0, r_0)$, i.e., its receptive field is the space whose distance to the i-th neuron is less than r_0. Therefore, the dynamics of the i-th neuron can be further written as,

$$\begin{aligned}\frac{d\xi_i}{dt} &= -A\xi_i + (B - \xi_i)\Big([I_i]^+ + \sum_{j=1}^{k} w_{ij}[\xi_j]^+\Big) \\ &\quad -(D + \xi_i)\Big([I_i]^- + \sum_{j=1}^{k} v_{ij}[\xi_j - \sigma]^-\Big), \end{aligned} \tag{3.4}$$

where k is the total number of neighboring neurons of the i-th neuron. Parameter σ is the threshold of the inhibitory lateral neural connections. The threshold of the excitatory connections is chosen as a constant zero.

The proposed neural network characterized by Equation (3.4) guarantees that the positive neural activity can propagate to the whole state space, while the negative activity stays locally only in a small region, because of the existence of the threshold σ of the inhibitory lateral neural connections in the neural network. Therefore, the target *globally* influences the whole state space to attract the robot, while the obstacles have only *local* effect in a small region to avoid the possible collisions and to take into account of the clearance from obstacles. In addition, the local influence from the obstacles is adjustable by changing the relative lateral connection strength β and/or the threshold σ of the inhibitory lateral neural connections. Therefore, by selecting a suitable strength of clearance from obstacles, the proposed model is capable of generating either the shortest path from the starting position to the target, a comfortable path, or the safest path of a point mobile robot and a multi-joint robot manipulator, depending on the different requirement.

The positions of the target and obstacles may vary with time. The activity landscape of the neural network dynamically changes due to the varying external inputs and the internal lateral neural connections. Similar to those neural network models presented in Chapter 2, the robot motion is planned through the dynamic activity landscape by a steepest gradient ascent rule. For a given *present position* (a position in the Cartesian workspace or a configuration in the robot joint space), denoted by p_p, the *next position* p_n (also called "command position") is obtained by

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \max_k \{\xi_k, k = 1, 2, \cdots, k\}, \tag{3.5}$$

where k is the number of all the neighboring positions of the present position, i.e., all the possible next positions. After the *present position* reaches its *next position*, the *next position* becomes a *new present position*. The present robot position *adaptively* changes according to the varying environment.

The proposed neural network is a stable system, where the neural activity is bounded in a finite interval $[-D, B]$. In addition, similar to the stability analysis in Chapter 2, the stability and convergence of the proposed model can be rigorously proved using a Lyapunov stability theory. From the definition of $[a]^+$, $[a]^-$ and v_{ij}, Equation (3.4) can be rewritten into Grossberg's (1983) general form,

$$\frac{dy_i}{dt} = a_i(\xi_i)\left(b_i(\xi_i) - \sum_{j=1}^{N} c_{ij} d_j(\xi_j)\right), \tag{3.6}$$

by the following substitutions:

$$a_i(\xi_i) = \begin{cases} B - \xi_i, & \text{if } \xi_j \geq 0 \\ D + \xi_i, & \text{if } \xi_j < 0 \end{cases}, \tag{3.7}$$

$$b_i(\xi_i) = \frac{1}{a_i(\xi_i)}\Big(B[I_i]^+ - D[I_i]^- - (A + [I_i]^+ + [I_i]^-)\xi_i\Big), \tag{3.8}$$

$$c_{ij} = -w_{ij}, \tag{3.9}$$

and

$$d_j(\xi_j) = \begin{cases} \xi_j, & \text{if } \xi_j \geq 0 \\ \beta(\xi_j - \sigma), & \text{if } \xi_j < \sigma \\ 0, & \text{otherwise} \end{cases}. \tag{3.10}$$

Since B and D are positive constants and $\xi_i \in [-D, B]$, then $a_i(\xi_i) \geq 0$ (positivity); Since $w_{ij} = w_{ji}$, then $c_{ij} = c_{ji}$ (symmetric); Since $d_j'(\xi_j) = 1$ at $\xi_j > 0$; $d_j'(\xi_j) = \beta \geq 0$

at $\xi_j < \sigma$; and $d_j'(\xi_j) = 0$, otherwise, then $d_j'(\xi_j) \geq 0$ (monotonicity). Thus, Equation (3.4) satisfies all the three stability conditions required by Grossberg's general form (Grossberg, 1983; Grossberg, 1988). 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.

3.3 Simulation Studies

In this section, to illustrate the concept of safety consideration in robot motion planning, the proposed neural network model is first applied to very simple case, where a point mobile robot is to reach a target in a static environment. Then, a complicated real-time motion planning case of a point mobile robot to catch a moving target with moving obstacles is studied. After that, the proposed neural network model is applied to a two-link planner robot to reach a target in both static and dynamic environment, where there are two targets in the joint space. The joint space of robot manipulators is used as the state space of the neural network.

3.3.1 Motion Planning of a Point Mobile Robot with Static Obstacles

To illustrate the concept of safety consideration, the proposed model is first applied to a very simple case of motion planning, where a point mobile robot is to reach a target in a static environment. The workspace is shown in Figure 3.1, where the target located at position (14,14) is shown by an empty triangle, while the obstacles are shown by solid squares. The robot starts to move at position (0,0). The neural network has 15×15 topographically ordered neurons. The model parameters are chosen as: $A = 10$ and $B = D = 1$ for the shunting equation; $\mu = 1$, $\beta = 1$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs.

First, the threshold of the inhibitory connection is chosen as $\sigma = -1.5$, the planned robot motion is shown in Figure 3.1A by solid circles. Since the neural activity is

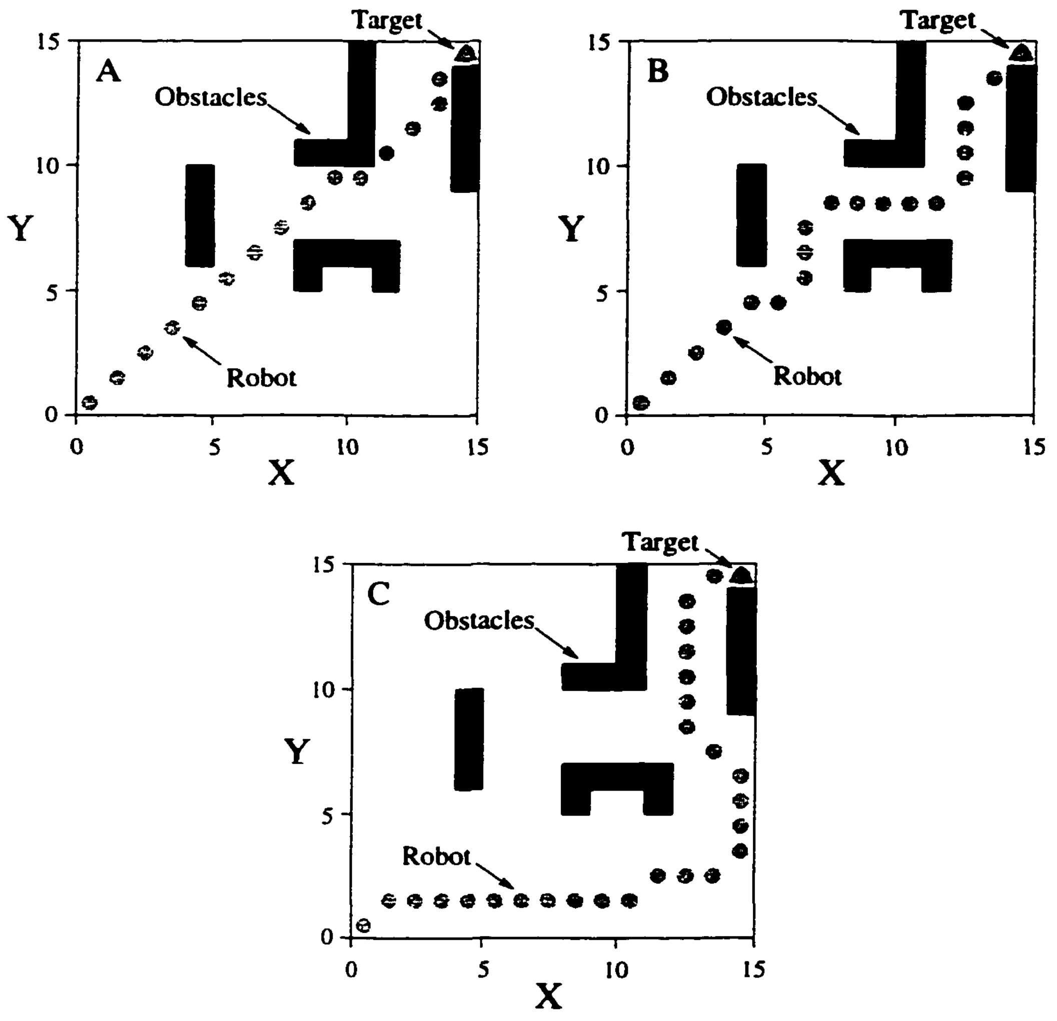


Figure 3.1: *Motion planning of a point mobile robot in a static environment.* A: *no clearance from obstacles yields the shortest robot motion path;* B: *a moderate clearance from obstacles yields a comfortable robot motion path;* C: *a strong clearance from obstacles yields the safest robot motion path.*

bounded in [-1, 1], when $\sigma < -1$, the lateral inhibitory term, $\sum_{j=1}^{k} v_{ij}[\xi_j - \sigma]^-$, in Equation (3.4) is equal to zero. Note that when $\beta = 0$, i.e., $v_{ij} \equiv 0$, this term becomes zero as well. Thus no negative neural activity is able to propagate to the other neurons. There is no clearance from the obstacles. Therefore, the planned robot motion in Figure 3.1A has the shortest path from the starting position to the target (Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1999*e*). Second, choosing $\sigma = -0.7$, the robot motion with a moderate obstacle clearance shown in Figure 3.1B is generated. Third, σ is chosen as $\sigma = -0.5$, a strong obstacle clearance is selected. We have the safest path from the starting position to the target shown in 3.1C. It takes 15, 19 and 26 steps to reach the target when parameter σ is chosen as -1.5 (Figure 3.1A), -0.7 (Figure 3.1B) and -0.5 (Figure 3.1C), respectively. It shows that along the shortest path in Figure 3.1A the robot clips the corners of obstacles and runs down the edges of obstacles. Thus it suffers from the "too close" problems. The safest path in Figure 3.1C has a very strong clearance from obstacles. It stays far away from the obstacles whenever possible, which usually suffers from the "too far" problems. However, it is capable of being very close to the obstacles when it *has to* be so in order to reach the target (the last phase of the path in Figure 3.1C). The safer path in Figure 3.1B is considered as a "comfortable" path, which takes a moderate clearance from obstacles. It not also does not clip the corners of obstacles nor runs down the edges of obstacles, but also takes a much shorter path than the safest path. Thus it avoids to be either "too close" to or "too far" from the obstacles. Note that in this example, those models suffering from "too close" problems (e.g. Lozano-Pérez, 1983; Brooks and Lozano-Pérez, 1985; Donald, 1987; Payton, 1990; Glasius *et al.*, 1994; Glasius *et al.*, 1995; Glasius *et al.*, 1996; Meng and Yang, 1998; Yang and Meng, 1998*a*) will fail to reach the target by expanding the obstacles with an extra size, because *all* possible solution paths are blocked after the expansion.

3.3.2 Motion Planning of a Point Mobile Robot with Moving Target and Obstacles

The proposed model in Equation (3.4) characterizes the dynamics of the neural activity. It is capable of planning real-time motion with obstacle clearance in a nonstationary environment. This model is applied to a more complex motion planning case first introduced in Section 2.3.3, where both the target and the obstacles are moving. There are 30×30 neurons in the neural network. The model parameters are chosen as: $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. The target starts at position (4,25) and continuously moves back and forth along the path between positions (4,25) and (24,25) at a speed of 10 *block/minute* (shown in Figure 3.2 by empty triangles). The static obstacles are shown in Figure 3.2 by solid squares, which form two possible channels for the robot to reach the target. In addition, there are 10 movable obstacles. They initially stay for 0.5 *minute* along positions (5,19) to (14,19) inside the left channel, where they completely block the left channel. Then these obstacles start to move toward the right at a speed of 20 *block/minute*, and finally stay along positions (14,19) to (23,19), where they completely block the right channel.

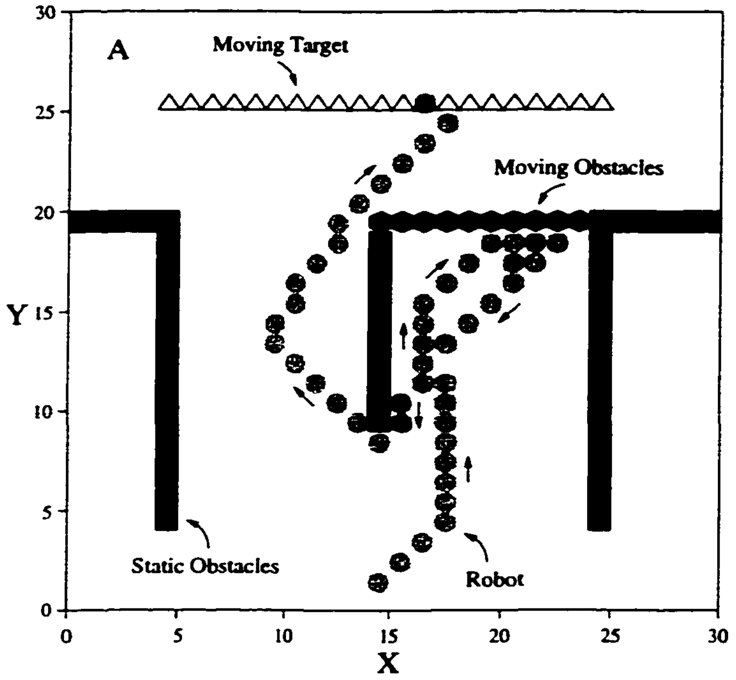
First, choosing $\beta = 0$, no clearance from obstacles is considered. The planned robot motion is shown in Figure 3.2A by solid circles. Initially the robot moves toward the right since the left channel is completely blocked while the right channel is open. However, during the time the robot is moving toward the target in the right channel, the obstacles are gradually moving to close the right channel, while gradually leaving the left channel open. Before the robot is able to pass through the right channel, the moving obstacles completely block the right channel and leave the left channel completely open. The robot has to move *away* from the target, passes around the middle static obstacles, and finally catch the moving target through the left channel. Because no obstacle clearance is taken into account, as shown in Figure 3.2A the planned robot path clips the corners of obstacles and runs down the edges of obstacles, thus it suffers the "too close" problem.

Then, we take the safety consideration by choosing $\beta = 0.01$ and $\sigma = -0.6$. The planned robot motion is shown in Figure 3.2B. Unlike the real-time path in Figure 3.2A, the robot in Figure 3.2 travels a continuous, smooth, "comfortable" route to catch the moving target without clipping the corners of obstacles nor running down the edges of obstacles.

3.3.3 Motion Planning of a Robot Manipulator in a Static Environment

The proposed neural network model is capable of planning real-time robot motion 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 must disappear from the state space once it is caught. The proposed model is applied to motion planning of a two-link planner robot, where there are two targets in the joint space. The state space of the neural network is the joint space of the robot manipulator. The robot link lengths are $l_1 = 1m$ and $l_2 = 1.1m$, respectively. The robot is located in a $5m \times 5m$ Cartesian workspace (Figure 3.3A), where the base of the first link is at in the center $(0, 0)$. The initial robot manipulator configuration in joint space is at $(\theta_1, \theta_2) = (30^o, 30^o)$ (Figure 3.3B), where θ_1 and θ_2 are the joint angles of the first and second links, respectively. Thus Initially the tip of the second link is at position $(1.366, 1.366)$ in the workspace (Figure 3.3A). The task is to move the tip of the second link to position $(1.366, -1.366)$ in the workspace (Figure 3.3A), i.e., to plan a continuous path to change the initial robot configuration to one of the two target configurations in joint space, $(\theta_1, \theta_2) = (330^o, 330^o)$ and $(300^o, 30^o)$ (shown in Figure 3.3B by empty triangles).

The neural network architecture has 60×60 topographically ordered neurons, which represent the joint angles from 0^o to 354^o with a step of 6^o. Because geometrically $360^o = 0^o$, the $(0, 0)$-th neuron in the neural network is an neighboring neuron of the $(59, 59)$-th and $(0, 59)$-th neurons, and likewise. The model parameters are chosen as: $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 50$. First, simulations are carried out without clearance from the obstacles by choosing $\beta = 0$ or $\sigma = -1.5$. The



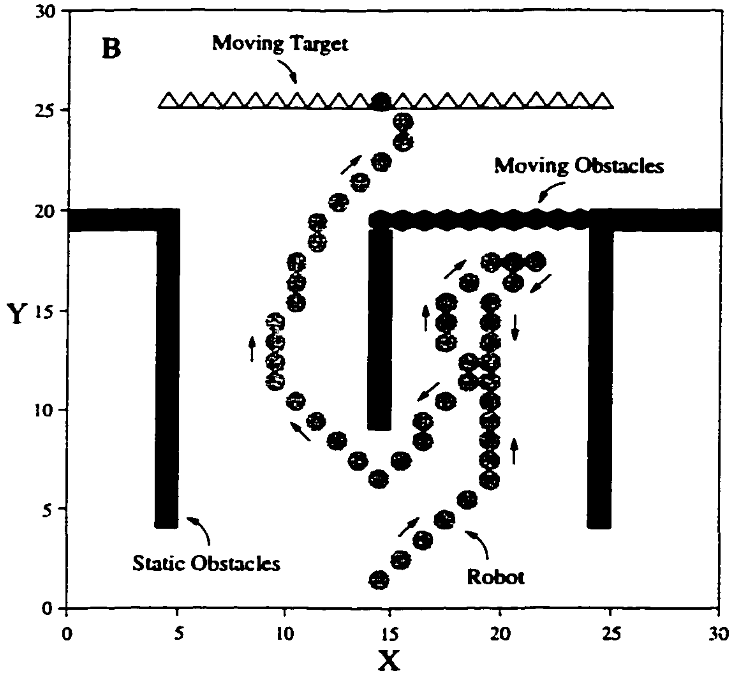


Figure 3.2: *Motion planning of a point mobile robot with moving target and moving obstacles.* A: *the real-time robot motion path without clearance from obstacles;* B: *the path with clearance from obstacles.*

solid circles in Figure 3.3B gives the planned robot motion from the starting position to the first target $(\theta_1, \theta_2) = (330^o, 330^o)$, the closest target, in robot joint space. Note that neuron $(59, 59)$ is an neighboring neuron of neuron $(0, 0)$ due to the geometrical relationship, although it looks that they are very far in the plot. It shows that the robot is able to plan the shortest path to reach the closest target, Target 2. The corresponding dynamic robot performance in the workspace is shown in Figure 3.3A.

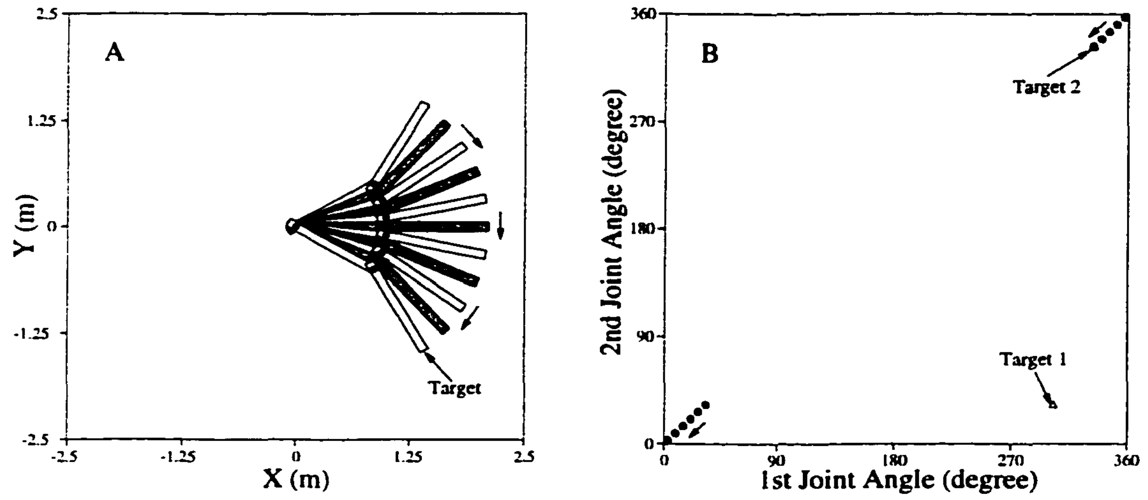


Figure 3.3: *Motion planning of a two-link robot manipulator without any obstacles in the workspace.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space of the robot manipulator.*

Second, an obstacle is placed at position $(0.8, 0)$ in the workspace (Figure 3.4A with solid circle). The corresponding obstacles in joint space is shown in Figure 3.4B by sold squares. Choosing the same model parameters as in Figure 3.3, the planned path in joint space is shown in Figure 3.4B, and the corresponding dynamic robot performance in workspace is shown in Figure 3.4A. It shows the robot is able to avoid the obstacle and reach Target 1 at $(\theta_1, \theta_2) = (300^o, 30^o)$, the closest target. Then, one more obstacle is placed at position $(0, -1.5)$. With the same model parameters as in Figure 3.3, the planned path in joint space and the corresponding dynamic robot performance in workspace is shown in Figures 3.4B and 3.4A, respectively. In both cases, the robot can reach the closest target, Target 1, with obstacle avoidance.

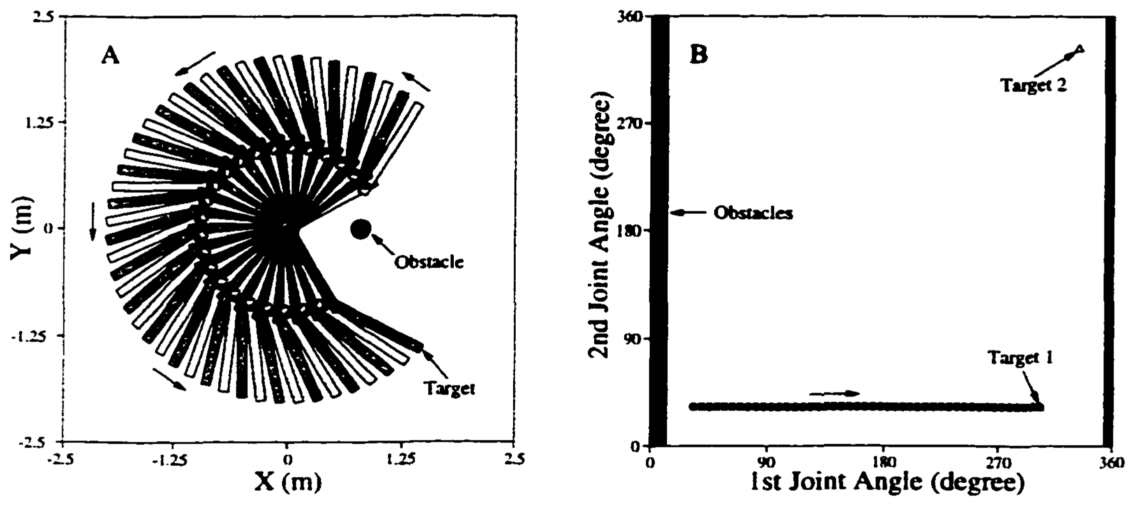


Figure 3.4: *Motion planning of a two-link robot manipulator with one obstacle in the workspace.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space.*

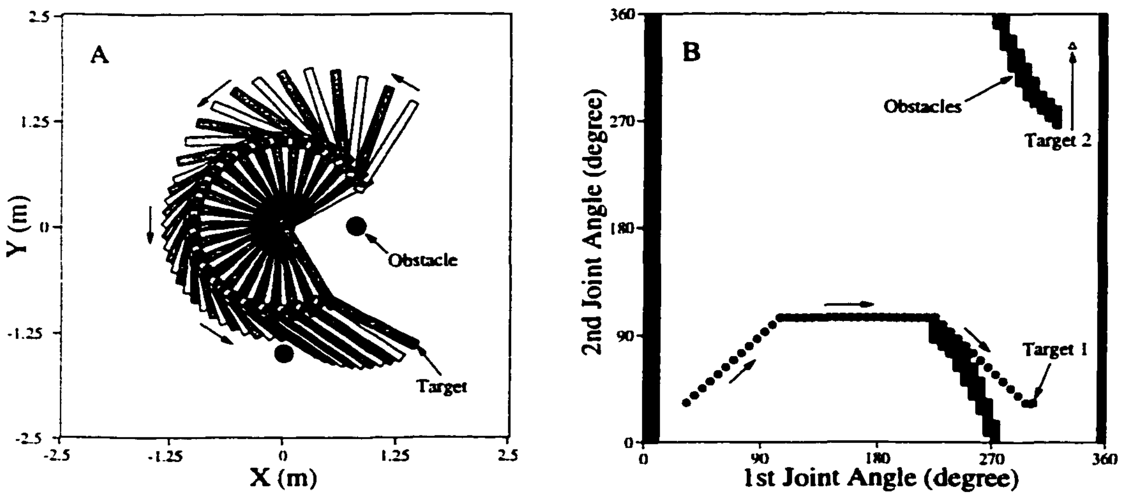
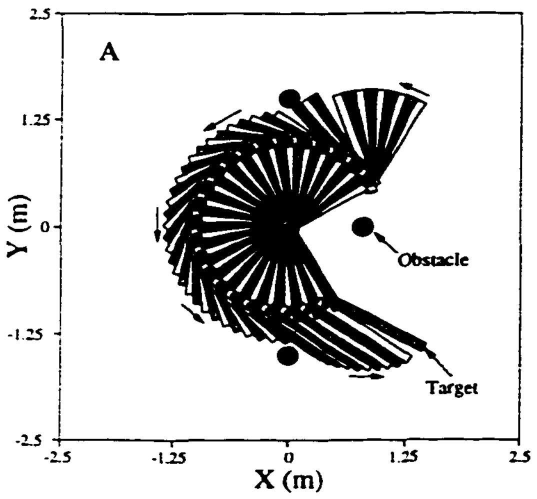


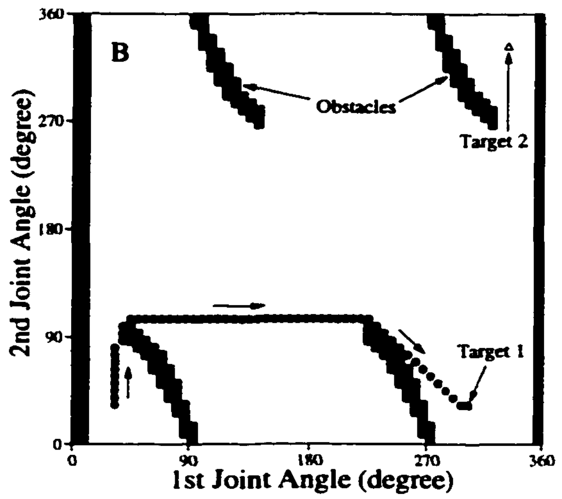
Figure 3.5: *Motion planning of a two-link robot manipulator with two obstacles in the workspace.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space.*

Third, one more obstacle is placed at position (0, 1.5) in the workspace (Figure 3.6A). With the same model parameters as in Figures 3.3 and 3.4, the planned robot path in joint space and the dynamic robot performance in workspace are shown in Figures 3.6A and 3.6B, respectively. Note that the planned paths in Figures 3.3B, 3.4A and 3.6B have the shortest distance from the starting position to the closest targets, due to no clearance from obstacles is considered. The stable (time is long enough) activity landscape of the neural net is shown in Figure 3.6C, where the peak of the activity landscape is at the target location, while the valley is at the obstacle location.

Because no clearance from obstacles is considered, the planned path without safety consideration shown in Figure 3.6 clips the corners of obstacles and runs down the edges of obstacles. Two more simulations with obstacle clearance are carried out under the same condition as in Figure 3.6, i.e., there are three obstacles in the workspace. By choosing $\beta = 1$ and $\sigma = -0.8$, a "comfortable" path from the starting position to the closest target is planned (Figure 3.7). Figures 3.7A and 3.7B shows the dynamic robot performance in workspace and the path in joint space, respectively. It shows that the robot does not clip the corners of obstacles nor run down the edges of obstacles. The stable (time is long enough) activity landscape of the neural net is shown in Figure 3.7C. In contrast to the neural activity landscape where there is no propagation of negative neural activity, the negative neural activity in Figure 3.7C slightly propagates from the obstacles to its neighboring positions.

Then, by choosing $\beta = 1$ and $\sigma = -0.5$, a strong clearance from obstacles is chosen. We have the safest path from the initial robot location to the closest target, Target 1 (Figure 3.8). The dynamic robot performance in workspace and the path in joint space are shown in Figures 3.8A and 3.8B, respectively. It shows that the robot does not pass through the small gate at the lower-left corner in the joint space, instead it goes across the figure to reach the closest target. The stable (time is long enough) activity landscape of the neural net is shown in Figure 3.8C. Comparing with the stable neural activity landscapes in Figures 3.7C and 3.6C, there is a more neural activity propagation from the obstacles to their neighboring positions.





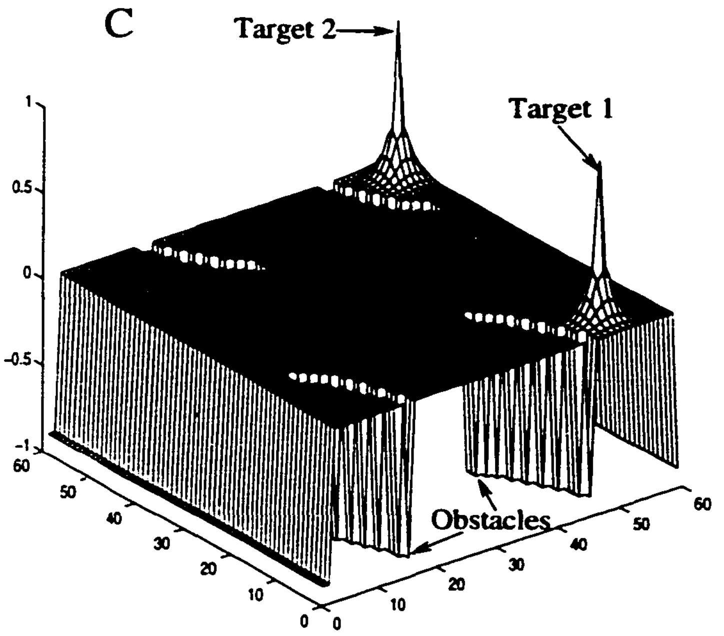


Figure 3.6: *Motion planning of a two-link robot manipulator with three obstacles in the workspace. No obstacle clearance yields the shortest path.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space;* C: *the stable activity landscape of the neural network.*

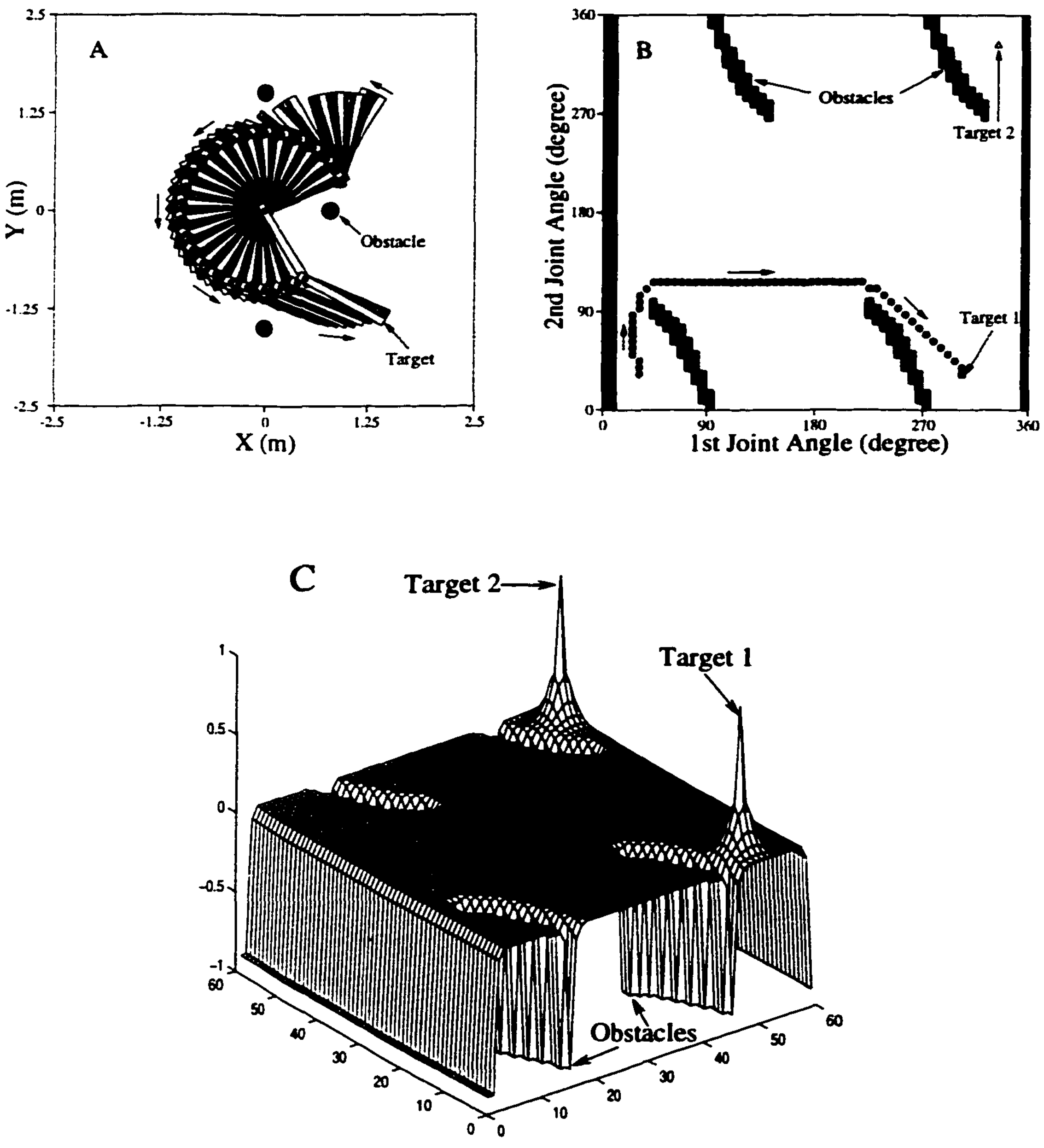
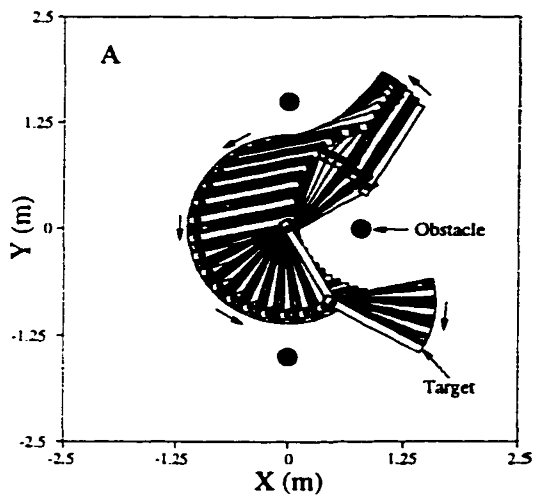
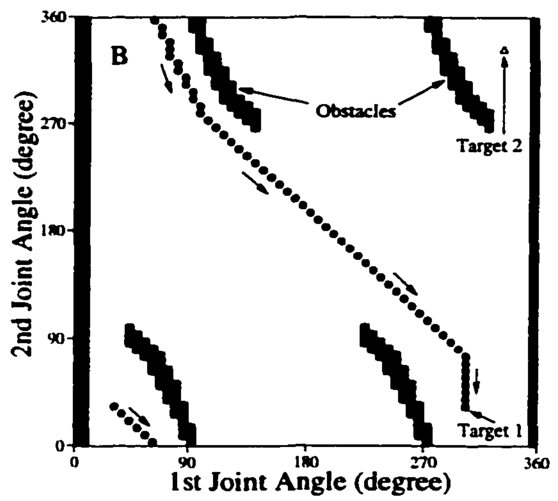


Figure 3.7: *Motion planning of a two-link robot manipulator with three obstacles in the workspace. A moderate obstacle clearance yields a comfortable path.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space;* C: *the stable activity landscape of the neural network.*





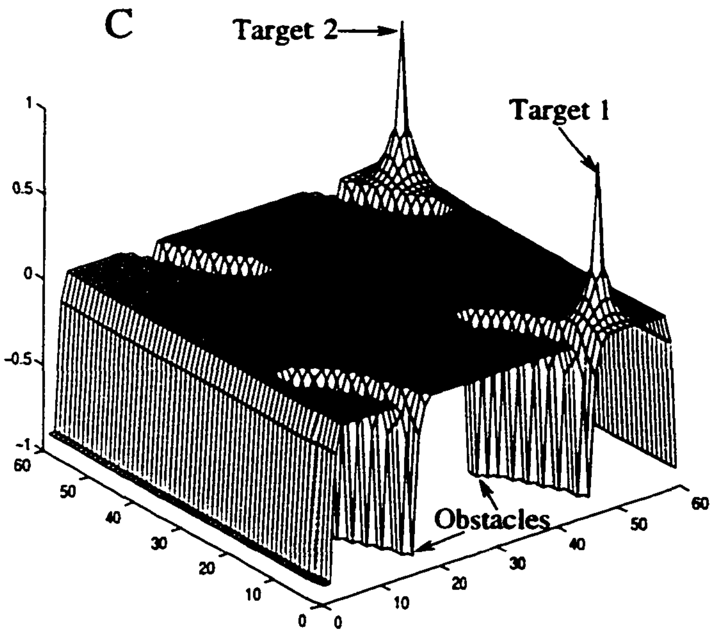


Figure 3.8: *Motion planning of a two-link robot manipulator with three obstacles in the workspace. A strong obstacle clearance yields the safest path.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space;* C: *the stable activity landscape of the neural network.*

3.3.4 Motion Planning of a Robot Manipulator with Sudden Environment Changes

The proposed neural network model can perform properly in an *arbitrary* environment, even with sudden environmental changes, such as suddenly adding or removing obstacles or targets in the dynamic environment. A case with sudden placement of obstacles in front of the robot manipulator is studied. The neural network architecture and the robot manipulator model are the same as in previous section. The task of the robot is the same as well, i.e., reach one of the targets in the joint space of the robot manipulator. The model parameters are chosen as: $A = 20$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 80$. There is one static obstacle located at $(0, -1.5)$. The initial configuration of the robot manipulator in joint space is at $(\theta_1, \theta_2) = (30^\circ, 30^\circ)$, i.e., the initial tip position of the second link is at $(1.366, 1.366)$ in the workspace (Figure 3.9A). First, no clearance from obstacles is considered by choosing $\beta = 0$. When the robot moves toward to the closest target, Target 2, there is an obstacle suddenly placed in its front at position $(0.8, 0)$ in the workspace (Figure 3.9A). The neural activity landscape of the neural network right after the sudden obstacle placement is shown in Figure 3.9B. It shows that there appears a large negative neural activity at the sudden obstacle locations in the joint space. The proposed neural network is sensitive to any sudden stimulus from the dynamic environment. The planned real-time path in joint space is shown in Figure 3.10B, and the corresponding dynamic robot performance in workspace is shown in Figure 3.10A. It shows that the robot reaches the closest target, Target 1, without any collisions with both the static and sudden obstacles.

Then, clearance from obstacles is taken into account by choosing $\beta = 0.1$ and $\sigma = -0.7$. The planned robot path in joint space is shown in Figure 3.11B, and the corresponding dynamic robot performance in workspace is shown in Figures 3.11A. Unlike the real-time path in Figure 3.10 where the robot motion suffers from the "too close" problem, the planned robot motion in Figure 3.11 travels a continuous, smooth, "comfortable" route to catch the moving target without clipping the corners

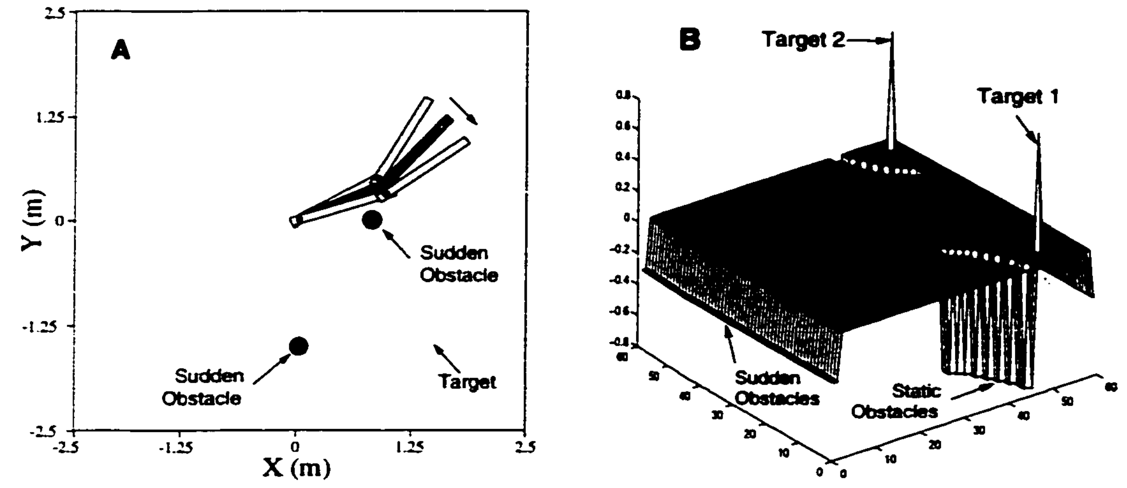


Figure 3.9: *Motion planning of a two-link robot manipulator when an obstacle is suddenly placed in front of the robot.* A: *the robot performance in workspace.* B: *the activity landscape of the neural network right after the obstacle is placed.*

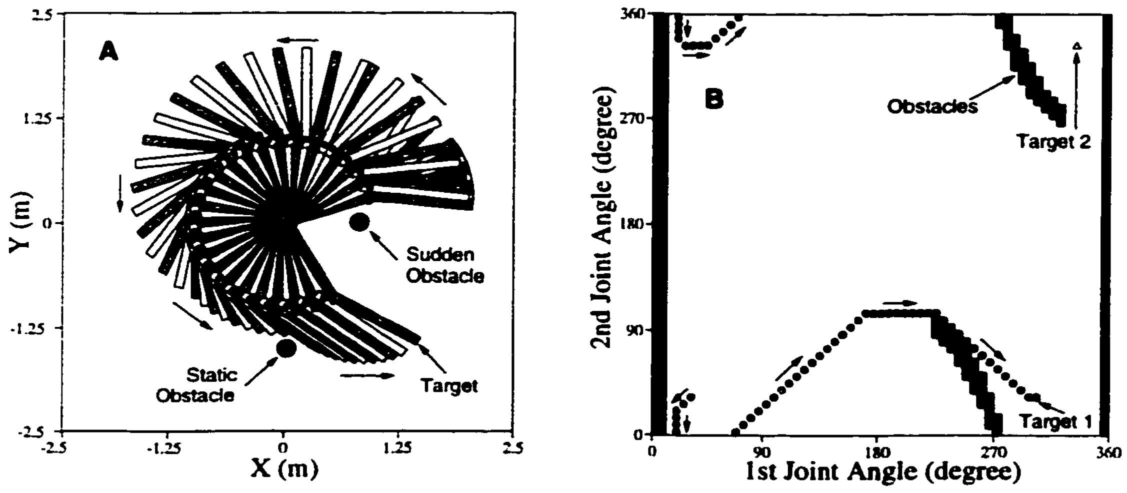


Figure 3.10: *Motion planning without obstacle clearance of a two-link robot manipulator with sudden environmental changes.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space.*

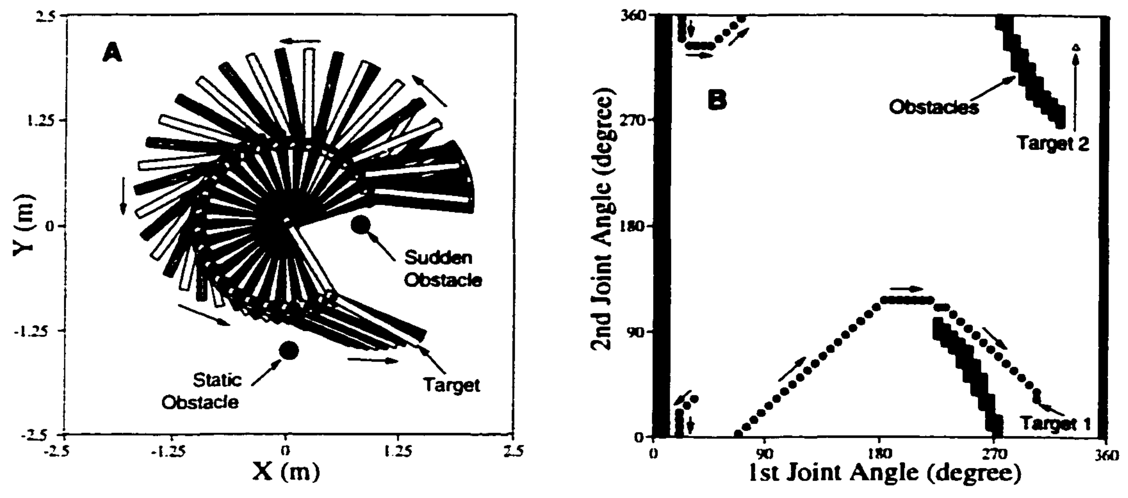


Figure 3.11: *Motion planning with obstacle clearance of a two-link robot manipulator with sudden environmental changes.* A: *the real-time robot motion in workspace;* B: *the planned path in joint space.*

of obstacles nor running down the edges of obstacles.

3.4 Model Variation

If the excitatory and inhibitory connections in the shunting equation in Equation (3.4) are lumped together and the auto gain control terms are removed, then a simpler form can be obtained from Equation (3.4),

$$\frac{d\xi_i}{dt} = -A\xi_i + I_i + \sum_{j=1}^{k} w_{ij}[\xi_j]^+ - \sum_{j=1}^{k} v_{ij}[\xi_j - \sigma]^-. \tag{3.11}$$

This is an additive equation. The term $I_i + \sum_{j=1}^{k} w_{ij}[\xi_j]^+ - \sum_{j=1}^{k} v_{ij}[\xi_j - \sigma]^-$ represents the total inputs to the i-th neuron from the external and internal connections. The nonlinear functions $[a]^+$, $[a]^-$, the connection weight w_{ij} and v_{ij}, the threshold σ, and the external input I_i are defined as the same as in Equation (3.4). These definitions together guarantee that the positive neural activity can propagate to the whole workspace, while the negative activity can propagate locally in a small region only. From the definitions of $[a]^+$, $[a]^-$ and v_{ij}, Equation (3.11) can be rewritten into a compact form as

$$\frac{d\xi_i}{dt} = -A\xi_i + I_i + \sum_{j=1}^{k} w_{ij} d(\xi_j), \tag{3.12}$$

where $d(\xi_j)$ is defined in Equation (3.10). In this additive neural network system, the targets and obstacles are guaranteed to stay at the peak and valley of the dynamic neural activity landscape. By properly choosing the strength and threshold of the inhibitory lateral neural connections, the influence of the obstacles to the activity landscape is confined in a desired small region. Therefore, the additive neural network model in Equation (3.12) is capable of planning real-time robot motion with clearance from obstacles in most situations. Although the additive model is computationally simpler, there are several important differences between the shunting and additive models (Grossberg, 1988; Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1999*e*), which are discussed in Chapter 2.

The stability of the additive neuron network in Equations (3.11) or (3.12) can be proved using a Lyapunov stability theory. Equation (3.12) can be rewritten into Grossberg's general form of Equation (3.6) by the following substitutions:

$$a_i(\xi_i) = 1, \tag{3.13}$$

$$b_i(\xi_i) = -A\xi_i + I_i, \tag{3.14}$$

and

$$c_{ij} \quad = -w_{ij}. \tag{3.15}$$

Obviously, we have $c_{ij} = c_{ji}$ (symmetry), $a_i(\xi_i) \geq 0$ (positivity), and $d'_j(\xi_j) \geq 0$ (monotonicity). Therefore, Equation (3.12) satisfies all the three stability conditions required by the Grossberg's general form. The global stability and convergence of the additive neural network system is proved (Grossberg, 1983; Grossberg, 1988).

A case using the proposed additive neural network model in Equation (3.12) under the same condition as in Figure 3.6 is simulated, where there are three obstacles in the workspace. The neural network architecture is the same as in Figure 3.6. The model parameters are chosen as: $A = 5$, $B = D = 1$, $\mu = 1$, $r_0 = 2$, and $E = 50$. By choosing $\beta = 0$ and $\sigma = 0$, $\beta = 0.1$ and $\sigma = -1$, and $\beta = 0.1$ and $\sigma = -0.8$, the additive model in Equation (3.12) is capable of generating the shortest path in Figure (3.6), the comfortable path in Figure (3.7), and the safest path in Figure (3.8), respectively. The stable landscape activity (time is long enough) under the same condition as in Figure (3.8) is shown in Figure 3.12. This result is qualitatively the same as that in Figure 3.8C except some quantitative differences because there is no bounds of neural activity in the additive model.

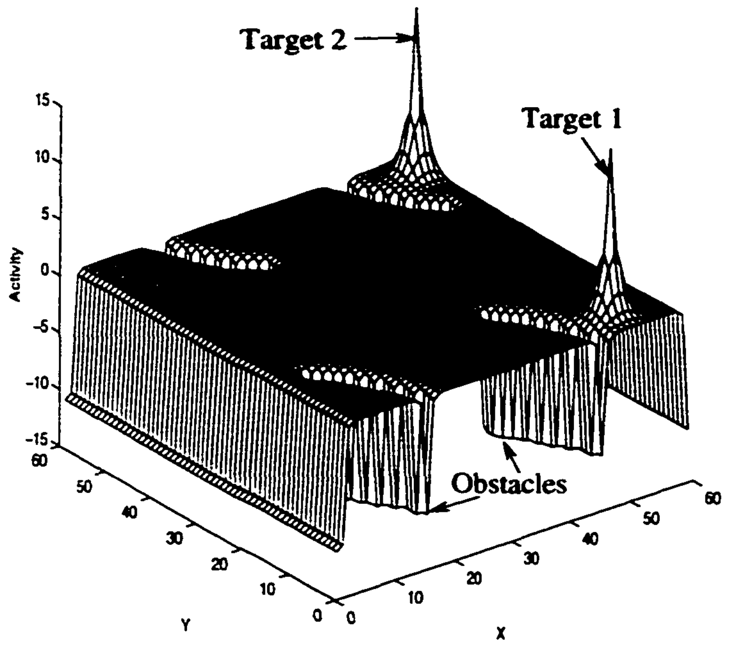


Figure 3.12: *The stable neural activity landscape of neural network using the additive model. There are three obstacle in the workspaces. A strong clearance from obstacles is considered.*

3.5 Conclusion

In this chapter, a novel neural network approach, based on the biologically inspired neural network work model for motion planning of a point mobile robot, is proposed for real-time robot motion planning with safety consideration in a nonstationary environment. The dynamics of each neuron in the neural network is characterized by a shunting equation or an additive equation. In contrast to the neural network models presented in Chapter 2, both excitatory and inhibitory neural connections are employed, and the threshold of neural connections are introduced. In addition, the state space of the topographically organized neural network is extended to either the Cartesian workspace or the joint space of a multi-joint robot manipulator. The proposed model is capable of planning real-time motion with safety consideration for a point mobile robot or a robot manipulator. The stability and convergence of the proposed neural network system is proved using a Lyapunov stability theory. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network that integrates the clearance from obstacles. The planned robot motion in a static environment without obstacle clearance is globally optimal in the sense of the shortest path from the starting position to the target if it exists. When the clearance from obstacles is considered, the optimality is in the sense of a "comfortable" path not being "too close" to nor "too far" from the obstacles. The term "real-time" is in the sense that the robot motion planner responds immediately to the dynamic environment, including the robot, target, obstacles and sensor noise. The proposed neural network approach is based on the robot motion planning model presented in Chapter 2. Therefore it inherits the feature properties from the neural network model presented in Chapter 2 plus the capacity of obstacle clearance. Several points are worth to highlight about the proposed neural network model for real-time robot motion planning with safety consideration:

- The strength of the obstacle clearance is adjustable. By selecting suitable model parameters, this model can plan the shortest path, a comfortable path, or the safest path of a point mobile robot or a robot manipulator.
- This model is biologically plausible.
- This model does not suffer from undesired local minima.
- This model is not very sensitive to the model parameters nor the connection weight function.

- The computational complexity linearly depends on the state space size of the neural network.
- The model algorithm is computationally efficient. The real-time robot motion with safety consideration is planned through the dynamic activity landscape of the neural network that integrates the obstacle clearance, *without* any prior knowledge of the dynamic environment, *without* explicitly searching over the free space or the collision paths, *without* explicitly optimizing any cost functions, and *without* any learning procedures.

Chapter 4

A Novel Neural Network Approach to Real-time Collision-free Motion Planning of 3-D.O.F. Robots in 2D

In this chapter, a novel neural network approach, based on the biologically inspired neural network model presented in Chapter 2 for motion planning of a point mobile robot, is proposed for real-time collision-free motion planning of 3-d.o.f. robots in a nonstationary environment. The state space of the proposed neural network is 3D, where two represent the spatial position in the 2D Cartesian workspace and one represents the orientation of the robot. The proposed neural network model is capable of planning real-time optimal motion path for 3-d.o.f. robots through the dynamic neural activity landscape. In addition to the feature properties of the neural network model for a point mobile robot, unlike most previous models that require a local collision checking procedure at each step of the robot movement, no local collision checking is needed in the proposed neural network model. Therefore the model algorithm is computationally efficient. Some complicated robot motion planning problems, such as real-time motion planning with sudden environmental changes, motion planning of a robot with multiple targets, and motion planning of multiple robots in the same workspace, are studied in this chapter. The effectiveness and efficiency of the proposed algorithm are demonstrated through simulation studies.

4.1 Introduction

Real-time collision-free motion planning of a mobile robot in 2D is a very important issue in robotics. A small and maneuverable mobile robot can be treated as a point mobile robot that has 2-d.o.f., i.e., the translation along X and Y axes. However, in many situations, e.g., when the size of the robot is comparable to the free workspace, the robot should be considered with its shape and size, where the robot has 3-d.o.f., i.e., two are the translation of the robot base point in the 2D Cartesian workspace, and one is the rotation with respect to the base point. A freely movable 3-d.o.f. robot is referred as a holonomic car-like robot.

There are a lot of studies on motion planning of 3-d.o.f. robots using various approaches (e.g. Ilari and Torras, 1990; Barraquand and Latombe, 1991; Latombe, 1991; Zelinsky, 1994; Gambardella and Versino, 1994; Podsedkowski, 1998; Fraichard and Mermond, 1998; Kreczmer, 1998; Ong and Gilbert, 1998; Kassim and Kumar, 1999). Most of the previous models are searching based and deal with static environment only. These models are computationally complicated, particularly when the environment is complex, because a local collision checking procedure is needed for obstacle avoidance at each step of the robot movement. For example, to detect local collisions, Barraquand and Latombe's (1991) model uses a divide-and-conquer technique; Zelinsky's (1994) model uses a distance space bubble hierarchy.

Some learning based models were proposed for the motion planning of 3-d.o.f. robots in 2D workspace. For example, Gambardella and Versino (1994) proposed a learning method for motion planning of a robot in a cluttered workspace where dynamic local minima can be detected. Through learning it can avoid the local minima, such as in the deadlock situations. Fujii *et al.* (1998) proposed a multi-layer reinforcement learning model for motion planning of multiple mobile robots, However, the robot motion planned by learning based approaches is not optimal, especially in its initial learning phase (e.g., see Figure 1.2).

In this chapter, a novel neural network approach to real-time collision-free motion planning of 3-d.o.f. robots in a nonstationary environment is proposed, which is based on the biologically inspired neural network model presented in Chapter 2 for motion planning of a point mobile robot. Unlike those models presented in Chapters 2 and 3 (also in Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1998*c*; Yang and Meng, 1999*a*; Yang and Meng, 1999*b*; Yang and Meng, 1999*e*) where the state space of the neural network is either the Cartesian workspace or the joint space of a multi-joint robot manipulator, the proposed neural network model is expressed in a 3D state space, where two represent the spatial position in the 2D Cartesian workspace and one represents the orientation of the robot. Unlike most previous models that require a local collision checking procedure at each step of the robot movement, no local collision checking procedure is needed in the proposed model. Similar to the neural network models presented in Chapters 2 and 3, the real-time optimal robot motion is planned through the dynamic activity landscape of the neural network. The model algorithm is computationally efficient. In addition, some complicated robot motion planning problems, such as real-time motion planning with sudden environmental changes, motion planning of a robot with multiple moving targets, and motion planning of multiple robots in the same workspace, are studied in this chapter.

4.2 The Model

A freely moving robot with shape and size in the 2D Cartesian workspace $\mathcal{W}$ can be modeled as a 3-d.o.f robot, whose location can be ***uniquely*** determined by the spatial position (x, y) of the base point and the orientation angle θ with respect to the base point (see Figure 4.1A). A robot location in $\mathcal{W}$, called a robot configuration, uniquely corresponds to a point in the robot configuration space $\mathcal{C}$. The robot configurations not allowed are called obstacle configurations in $\mathcal{C}$.

The proposed model is based on the biologically inspired neural network approach to real-time collision-free motion planning of a point mobile robot in a nonstationary environment. The neural network architecture is a discrete topographically organized map, where the neural dynamics of each neuron is characterized by a shunting equation or an additive equation. Unlike those neural network models presented in Chapters 2 and 3 where the state space $\mathcal{S}$ of the neural network is either the Cartesian workspace or the joint space of a multi-joint robot manipulator, the proposed neural network model 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 robot configuration space $\mathcal{C}$. The location of the i-th 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 3-d.o.f robot. The dynamics of the i-th neuron in the neural network is characterized by a shunting equation,

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[\xi_j]^+\right) - (D + \xi_i)[I_i]^-, \tag{4.1}$$

where all the parameters and functions are defined as the same as those in Equation (2.7). Parameters A, B and D are nonnegative constants that represent the passive decay rate, the upper and lower bounds of the neural activity, respectively. Variable ξ_i is the neural activity of the i-th neuron, which has a continuous value $\xi_i \in [-D, B]$. The excitatory input, $[I_i]^+ + \sum_{j=1}^{n} w_{ij}[\xi_j]^+$, results from the targets and the lateral connections among neurons, while the inhibitory input $[I_i]^-$ results from the obstacles. Function $[a]^+$ is defined as, $[a]^+ = \max\{a, 0\}$, while $[a]^-$ is defined as $[a]^- = \max\{-a, 0\}$. The lateral connection weight w_{ij} from the j-th neuron to the i-th neuron are defined as $w_{ij} = f(d_{ij})$, where $d_{ij} = |p_i - p_j|$ represents the Euclidean distance between vectors p_i and p_j in the state space. Function $f(d_{ij})$ is a monotonically decreasing function, such as a function defined as: $f(d_{ij}) = \mu/d_{ij}$, if $0 < d_{ij} < r_0$; $f(d_{ij}) = 0$, if $d_{ij} \geq r_0$, where μ and r_0 are positive constants. Therefore the neuron has only local connections in a small region $(0, r_0)$. It is obvious that the weights w_{ij} is symmetric, $w_{ij} = w_{ji}$. 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.

In the proposed neural network model for 3-d.o.f. robots, if there is one obstacle in the 2D Cartesian workspace $\mathcal{W}$, several configurations in $\mathcal{C}$ of the 3-d.o.f. robot are not allowed ("forbidden"), i.e., there are several obstacle configurations in $\mathcal{C}$. For example, when the obstacle is at position (0,0) and the base point of the robot is at position (1,0) in $\mathcal{W}$, the forbidden robot locations are shown in Figure 4.1B, where the obstacles are shown by dark solid squares, while the robot is shown by light solid rectangular with its base point shown by a dark solid circle.

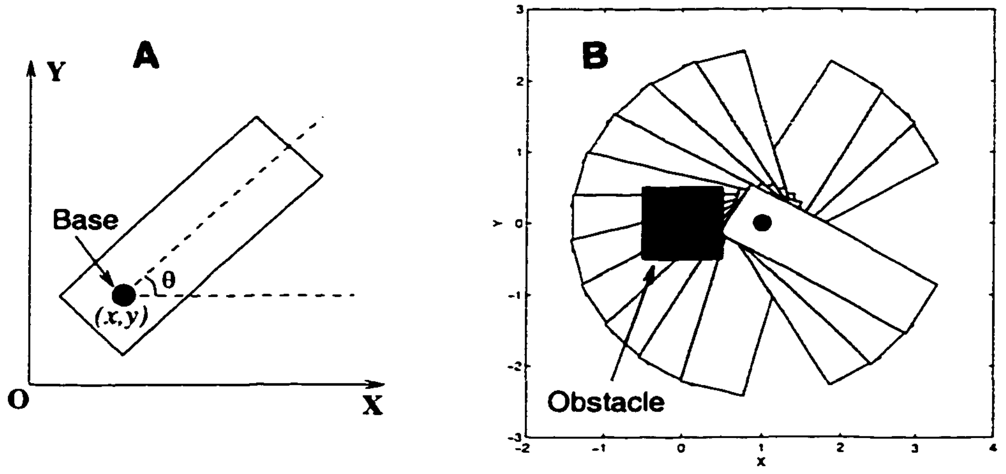


Figure 4.1: *A schematic diagram of a 3-d.o.f. robot* (A) *and the forbidden robot locations* (B) *when the obstacle and the robot base point are at positions (0,0) and (1,0), respectively.*

The proposed neural network characterized by Equation (4.1) 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 locations of the target and obstacles may vary with time. The neural activity landscape dynamically change due to the varying external inputs from the targets and obstacles and the internal activity propagation among neurons. The optimal robot motion is planned from the dynamic activity landscape by a gradient ascent rule. For a given present location in $\mathcal{S}$ (i.e., a location $\mathcal{W}$ or a configuration in $\mathcal{C}$), denoted by

p_p, the next location p_n is obtained by

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \max_k \{\xi_k, k = 1, 2, \cdots, k\}, \tag{4.2}$$

where k is the number of all the neighboring neurons of the present robot location p_p, i.e., all the possible next locations. The current robot location adaptively changes according to the varying environment.

The proposed neural network is a stable system, where the neural activity varies in the finite interval $[-D, B]$. In addition, by rewriting Equation (4.1) into Grossberg's general form in Equation (2.9), it can be proved that Equation (4.1) satisfies all the three stability conditions required by Grossberg's general form. Therefore, the stability and convergence of proposed neural network system characterized by Equation (4.1) can be rigorously proved using the Lyapunov stability analysis (Grossberg, 1983; Grossberg, 1988; Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1999*d*; Yang and Meng, 1999*e*).

4.3 Simulation Studies

In this section, several simulations are carried out to demonstrate the effectiveness of the proposed neural network approach to real-time collision-free motion planning of 3-d.o.f. robots. First this model is applied to motion planning of a robot in a static environment. Then, some complicated robot motion planning problems, such as real-time motion planning with sudden environmental changes, motion planning of a robot with multiple moving targets, and motion planning of multiple robots in the same workspace, are studied.

4.3.1 Motion Planning in a Static Environment

The proposed neural network model is first applied to motion planning of a 3-d.o.f. robot in a deadlock situation. The initial robot location is shown in Figure 4.2A, where it is in a deadlock situation. The task is to replace the robot to face in the opposite direction as shown in Figure 4.2B. The neural network has $20 \times 20 \times 24$

topographically ordered neurons, where 20×20 represents the size of the discretized 2D Cartesian workspace, and 24 represents number of the orientation angles from $0°$ to $345°$ with a step of $15°$. Since geometrically $360° = 0°$, the neuron at location $(X_i, Y_i, 0)$ in the neural network is an neighboring neuron of the neuron at $(X_i, Y_i, 23)$, and likewise. 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 input. The planned robot motion path to move the robot out is shown in Figure 4.2C, while the path to move the robot in is shown in Figure 4.2D. It shows that the robot changes to the opposite direction without any collisions.

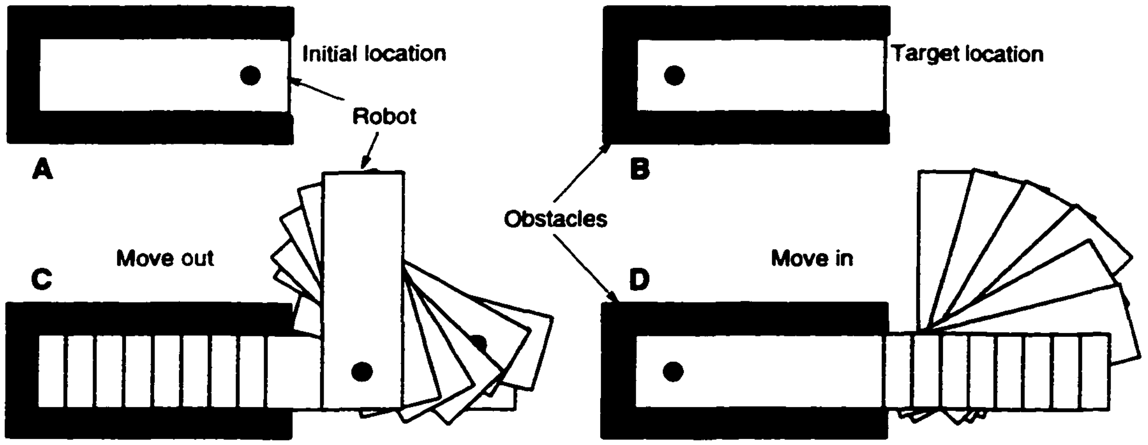


Figure 4.2: *Motion planning of a robot to replace to the opposite direction in a deadlock situation.* A *and* B*: the initial and target location of the robot, respectively;* C *and* D*: the planned robot motion for moving out and moving in, respectively.*

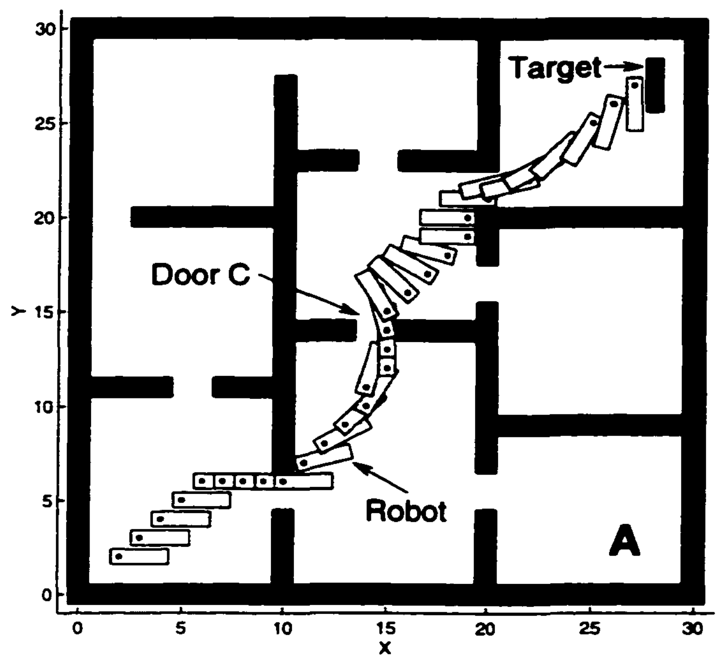
Then this model is applied to a more complex house-like environment, where there are several deadlock situations that the robot may be trapped in. Figure 4.3 shows a house-like environment, where the doors can be opened or closed. The neural network has $30 \times 30 \times 24$ neurons with the same model parameters as in the previous case. Figure 4.3A shows the planned real-time robot motion in the case that the central door C is opened. When the central door C is closed, the planned robot motion is shown in Figure 4.3B, where the robot has to travel a much longer route to reach the target location. It shows that without any learning procedures, the robot is capable of reaching the target without any collisions and without being trapped in any deadlock situations. In a static environment, the robot motion is globally optimal in the sense of the shortest route from starting location to the target.

4.3.2 Motion Planning with Sudden Environmental Changes

The proposed neural network model can perform properly in an arbitrary environment, even with sudden environmental changes. A difficult motion planning case with sudden placement of obstacles in front of the robot is studied. The neural network architecture is a $30 \times 20 \times 24$ neural network map. The model parameters are chosen as: $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. The initial and target robot locations are at locations (2,5,0) and (27,17,12) in $\mathcal{W}$, respectively. First, the planned robot motion without any obstacles is shown in Figure 4.4A. In the second case, when the robot arrives at location (11,14,15) on the way toward the target, the obstacles (V-shaped, shown in Figure 4.4C by dark sold squares) are *suddenly* placed in front of the robot. The planned real-time robot motion path is shown in Figure 4.4B, where the robot first has to first move *away* from the target, then passes around these obstacles, and finally reaches the target. It shows that the robot is capable of avoiding the sudden obstacles to reach the designated target.

4.3.3 Motion Planning with Multiple Moving Targets

The proposed neural network model is capable of planning real-time motion for a robot 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 must disappear from the state space once it is caught. A complicated case is carried out where there are two moving targets with different speed in the 2D Cartesian workspace. The task of the robot is to catch one of moving targets. The neural network has $30 \times 30 \times 24$ neurons. The model parameters are chosen as: $A = 50$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. Figure 4.5A shows the travel routes of the targets. The moving



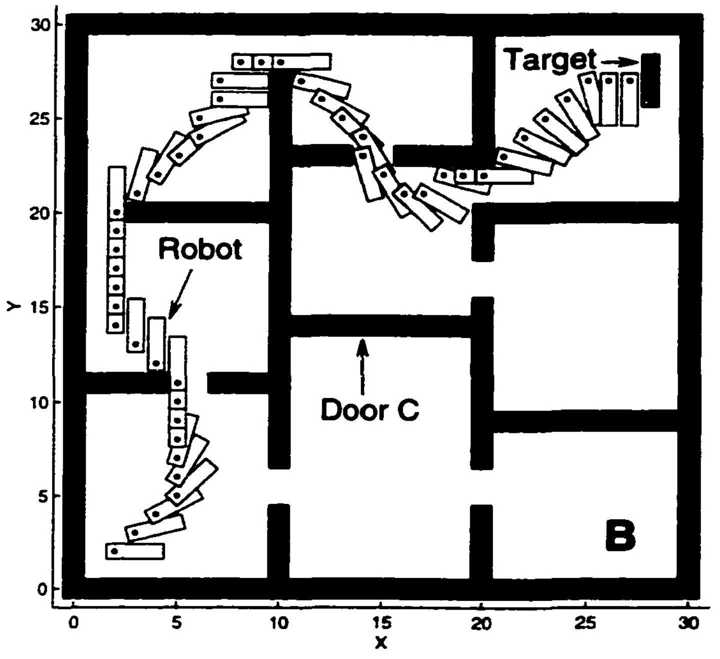
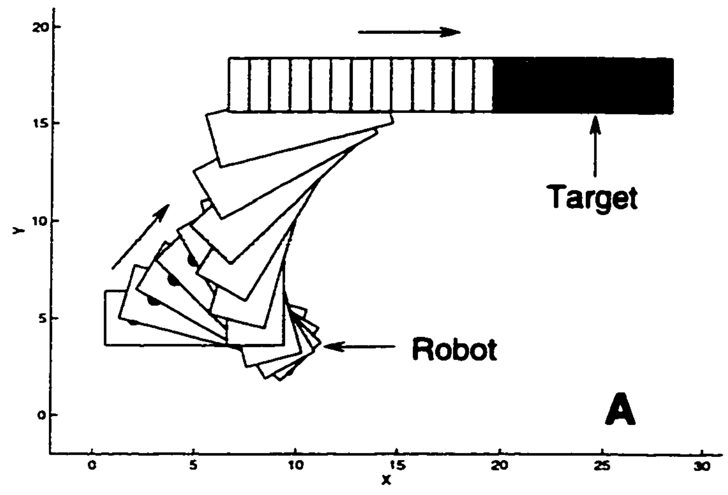


Figure 4.3: *Motion planning of a robot in a house-like environment with several deadlock situations.* A*: the planned robot motion when the central door C is opened;* B*: the planned robot motion when door C is closed.*



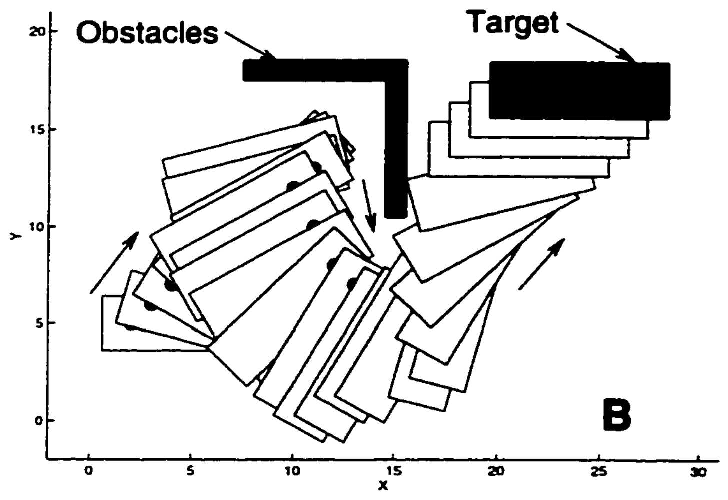


Figure 4.4: *Real-time motion planning of a robot with sudden placement of obstacles.* A*: the robot motion in case of no obstacles;* B*: the real-time robot motion in case that a set of V-shaped obstacles are suddenly placed in front of the robot.*

speeds of Targets 1 and 2 are 30 and 50 *block/minute*, respectively (it is convenient to assume that the space in $\mathcal{S}$ and time units are *block* and *minute*, respectively). The orientation varying speeds of both targets are 30 *block/minute*. Note that the proposed neural network dynamically responds to the real-time location of the targets and obstacles. No *prior* knowledge of the dynamic varying environment is needed. First, the robot starts at location (1,1,0), with a speed of 100 *block/minute* for both space and orientation. Figure 4.5B shows the dynamic routes of the robot and targets (the route of Target 2 is hidden by the route of the robot when plotted). The robot catches Target 2 at location (12,16,14), since the robot moves much faster than the targets. Then, the robot moves at a much slower speed, 50 *block/minute*. The real-time motion paths of the robot and targets are shown in Figure 4.5C. It shows that initially the robot moves toward Target 2, since it is closer to Target 2. However, finally the robot catches Target 1 at location (10,7,4), because Target 1 is moving toward the robot, while Target 2 is moving away from the robot.

4.3.4 Motion Planning of a Multi-robot System

The proposed neural network model is capable of planning real-time collision-free motion paths for multiple robots. The dynamic motion of four robots in the same 2D Cartesian workspace is studied. The overall neural network architecture consists of four $20 \times 20 \times 24$ neuron nets that have no lateral connections among neurons in different neuron net, which produce four dynamic neural activity landscapes for these four robots, respectively. The neuron net for Robot 1 treats the other three robots as moving obstacles, and likewise for the other robots. The model parameters are chosen as: $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. All the robots move at the same speed of 100 *block/minute*. The planned real-time motion paths for Robots 1, 2, 3 and 4 are shown in Figure 4.6A. To illustrate more clearly, the overall path in Figure 4.6A is plotted in two parts that are shown in Figures 4.6B and 4.6C, respectively. It shows that all these four robots travel smooth, continues routes to reach their targets without any collisions.

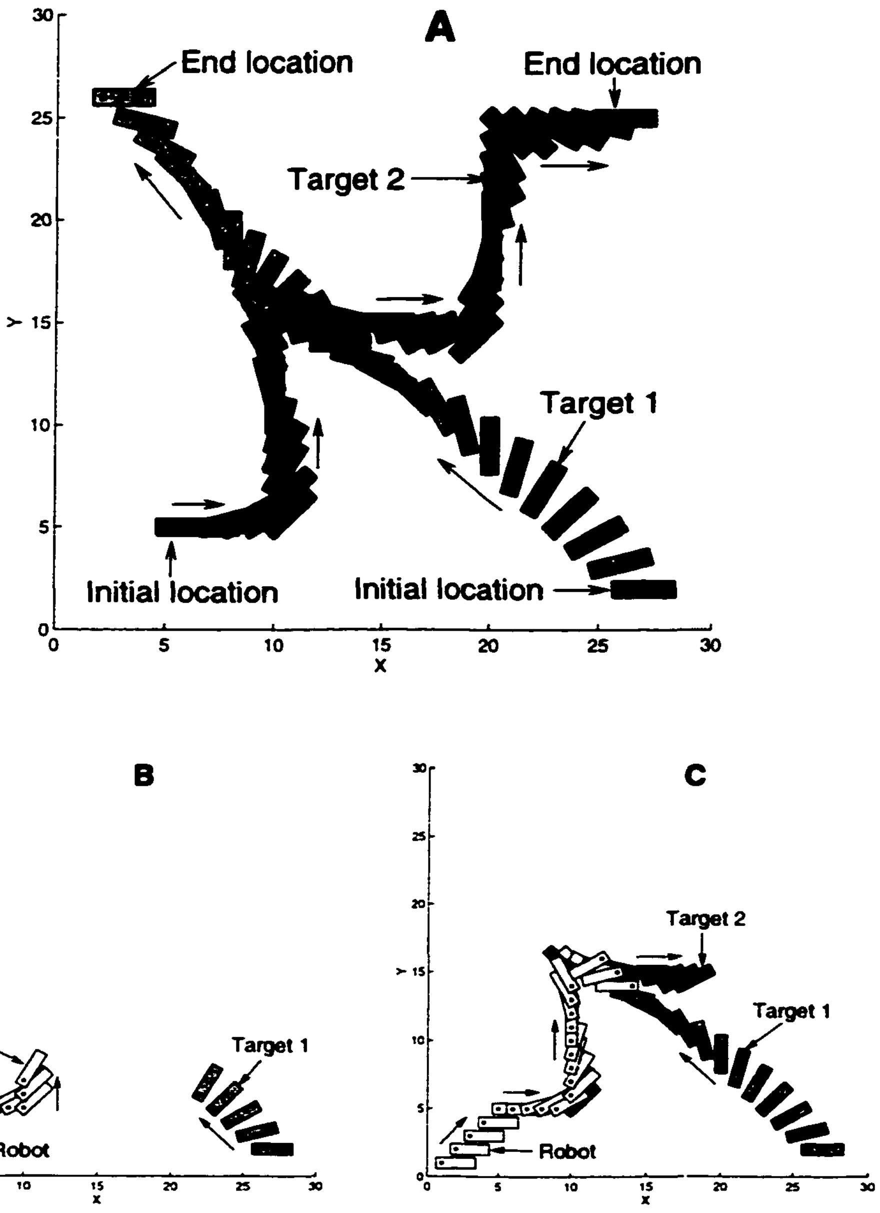
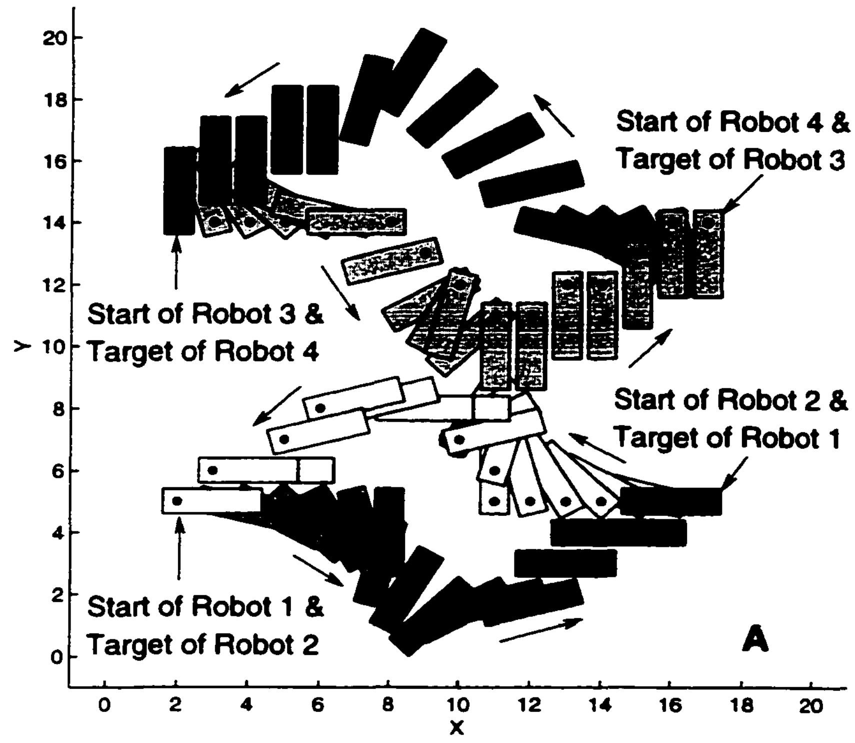
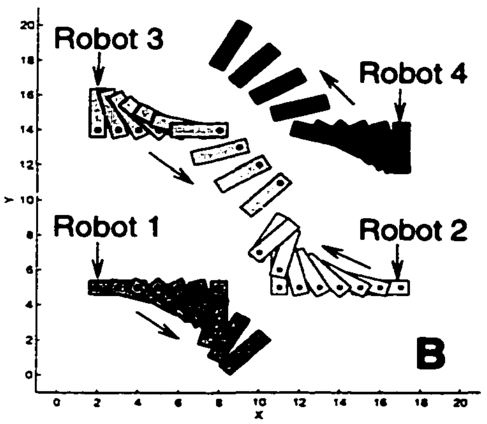


Figure 4.5: *Real-time motion planning of a robot with two moving targets.* A*: the moving routes of Targets 1 and 2;* B*: the real-time motion paths of the robot and the targets when the robot speed is twice of the speed of Target 2;* C*: the real-time motion paths when the robot moves at the same speed as Target 2.*





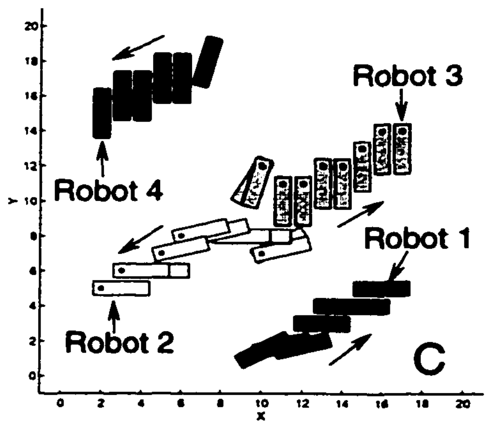


Figure 4.6: *Real-time motion planning of four robots in the same 2D workspace.* A*: the real-time paths of Robots 1, 2, 3 and 4;* B *and* C*: the first and second parts of the whole path, respectively.*

4.4 Model Variation

If the excitatory and inhibitory connections in the shunting equation in Equation (4.1) are lumped together and the auto gain control terms are removed, then Equation (4.1) may be written into a simpler additive equation,

$$\frac{d\xi_i}{dt} = -A\xi_i + I_i + \sum_{j=1}^{n} w_{ij}[\xi_j]^+, \tag{4.3}$$

where $I_i + \sum_{j=1}^{n} w_{ij}[\xi_j]^+$ represents the total input to the i-th neuron from the external and lateral connections. The definitions of I_i, $[a]^+$ and w_{ij} are the same as those in Equation (4.1). The nonlinear function $[a]^+$ guarantees that only the positive neural activity can propagate to the other neurons. This additive neural network system can be easily proved to be a stable system by rewriting the model equation into the general form in Equation (2.9). There are a lot of important differences between the shunting and the additive models (see Chapter 2), although the additive model is computationally simpler and can also plan real-time collision-free motion planning of 3-d.o.f. robots with in most situations.

4.5 Conclusion

In this chapter, a novel neural network approach is proposed for real-time collision-free motion planning of 3-d.o.f. robots in a nonstationary environment, which is based on the biologically inspired neural network model for motion planning of a point mobile robot presented in Chapter 2. In contrast to the neural networks presented in Chapters 2 and 3, the state space of the neural network is the 3D configuration space of the robot, where two represent the spatial position in the 2D Cartesian workspace and one represents the orientation of the robot. The real-time optimal robot motion is planned through the dynamic neural activity landscape of the neural network that represents the dynamic environment. Several points are worth to notice about the proposed neural network model for motion planning of 3-d.o.f. robots:

- This model can perform properly in an *arbitrary* varying environment, even with sudden environmental changes, such as suddenly adding or removing obstacles and targets. The neural network system is characterized by a continuous shunting equation, it is stable and keeps sensitive to changes of the target and obstacles.

- This model is not sensitive to sensor noise nor disturbance. The obstacles have only local effect due to no inhibitory lateral connections among neurons. The irrelevant obstacles and/or sensor noise (false-detected obstacles by the sensors) far from the robot will not influence the global robot motion planning. On the other side, because the proposed model is capable of dealing with the sudden environmental changes, the obstacles un-detected by the sensors at a far distance still can be avoided by robot motion planner at a short distance, even when they are very close to the robot.

- This model is capable of planning real-time motion for a robot with multiple moving targets, and real-time motion for multiple robots in the same workspace.

- This model is biologically plausible.

- This model will not be trapped in the deadlock situations.

- This model is not very sensitive to the model parameters nor the connection weight function.

- The computational complexity linearly depends on the neural network size.

- The model algorithm is computationally *efficient*. The real-time optimal robot motion is planned without explicitly searching over the free workspace or the collision paths, without explicitly optimizing any cost functions, without any prior knowledge of the dynamic environment, without any learning procedures, and without any local collision checking procedures at each step of the robot movement.

Chapter 5

A Novel Neural Network Approach to Real-time Motion Planning with Obstacle Avoidance of Nonholonomic Car-like Robot

In this chapter, a novel neural network approach, based on the neural network model for motion planning of 3-d.o.f robots presented in Chapter 4, is proposed for real-time motion planning with obstacle avoidance of nonholonomic car-like robots in a non-stationary environment. The dynamics of each neuron in the biologically inspired, topologically organized, locally connected neural network is characterized by a shunting equation or an additive equation. In contrast to those neural network models presented in Chapters 2, 3, and 4 where the neural activity propagation is omni-directional, the lateral neural connections in the proposed model are directionally selective, and the neural activity propagation is subject to the nonholonomic kinematic constraint of the car-like robot. The real-time collision-free robot motion is planned through the dynamic neural activity landscape of the neural network, under the kinematic constraint, without any local collision checking procedures at each step of the robot movement. Therefore it is computationally efficient.

5.1 Introduction

Real-time collision-free motion planning of nonholonomic mobile robots is one of the most important issue in robotics. A small and maneuverable mobile robot can be treated as a point robot that has 2-d.o.f., i.e., the translation along X and Y axes. However, in many situations, the robot should be considered with its shape and size. A robot with size and shape has 3-d.o.f., i.e., two are the translation of the robot base point in the 2D Cartesian workspace, and one is the rotation with respect to the base point. In realistic robotic applications, most mobile robots are nonholonomically constrained and the robots are not freely movable object. For a nonholonomic car-like robot, because of the kinematic constraint that the turning radius is lower-bounded, the degree of freedom of the robot becomes two.

There are more and more studies on motion planning of car-like robots using various approaches (e.g. Laumond *et al.*, 1994; Paromtchik and Laugier, 1996; Khatib *et al.*, 1997; Svestka and Overmars, 1997; Scheuer and Fraichard, 1997; Pruski and Rohmer, 1997; Fraichard and Mermond, 1998; Kreczmer, 1998; Laumond *et al.*, 1998; Podsedkowski, 1998; Sekhavat *et al.*, 1998). However, most of them deal with static environment only, or are computationally expensive. In addition, the local collision checking procedures are required at the step of the robot movement. Most previous models use two-step approaches that consist of first computing a collision-free holonomic path, and then transforming 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. For example, Paromtchik and Laugier (1996) proposed a searching-based iterative algorithm for motion generation for parking an car-like robot. Jiang *et al.* (1997) proposed a time-optimal motion planning method for a robot with kinematic constraints, which consists of three stages: planning for a point mobile robot; planning for a car-like robot; and optimizing cost functions for a time-optimal solution. Podsedkowski (1998) proposed a path planner for nonholonomic robot using a searching algorithm, it requires a local collision checking procedure and the minimization of cost functions. Sekhavat *et al.* (1998) proposed a multi-level approach to motion planning of nonholonomic robots, where at the first level, a path is found that disrespects the nonholonomic constraints; at each of the next levels, a new path is generated by transformation of the path generated at the previous level; at the final level, all nonholonomic constraints are respected. This model is computationally complicated.

In this chapter, a novel neural network approach, based on the neural network model presented in Chapter 4 for motion planning of 3-d.o.f. robots (also in Yang and Meng, 1999*c*; Yang and Meng, 1999*d*), is proposed for real-time collision-free motion planning of nonholonomic car-like robots in a nonstationary environment. Like the neural network model for motion planning of 3-d.o.f. robots, the proposed neural network is topologically organized, locally connected, where the dynamics of each neuron is characterized by a shunting equation or an additive equation. The state space of the neural network is the 3D configuration space of the car-like robot. Unlike those neural network models present in Chapters 2, 3, and 4 (also in Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1998*c*; Yang and Meng, 1999*a*; Yang and Meng, 1999*b*; Yang and Meng, 1999*c*; Yang and Meng, 1999*d*; Yang and Meng, 1999*e*) where the lateral connection strength of one neuron to its neighboring neurons is a function of the distance only and the neural activity propagation is omnidirectional, in the proposed neural network model the lateral connection strength is directionally selective, and the neural activity propagation is subject to the kinematic constraint of the car-like robot. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network, under the nonholonomic constraint, without any local collision checking procedures at each step of the robot movement. Therefore the model algorithm is computationally efficient. The planned robot motion in a static environment is globally optimal in the sense of the least steps from the starting location to the target if it exists. The optimality in the real-time motion planning in a nonstationary environment is in the sense of a continuous, smooth path toward the target. The term "real-time" is in the sense that the robot motion planner responds immediately to the dynamic environment, including the robot, target, obstacles and sensor noise. To the best of our knowledge, it is the first time that the real-time motion planning of car-like robots are studied using a non-learning based neural network approach.

5.2 The Model

In this section, we will briefly introduce the mathematical model of a nonholonomic car-like robot. Then the proposed neural network model for real-time collision-free motion planning of nonholonomic car-like robot will be presented. The stability of the proposed neural systems will be proved using a Lyapunov stability theory.

5.2.1 The Nonholonomic Car-like Robot Model

A freely movable robot with shape and size in the 2D Cartesian workspace $\mathcal{W}$ can be modeled as a 3-d.o.f. robot, which is referred as a holonomic car-like robot. If a 3-d.o.f. robot is subject to nonholonomic constraint, the kinematically constrained 3-d.o.f. is referred as nonholonomic car-like robot. The location of a car-like robot in the 2D Cartesian workspace can be *uniquely* determined by the spatial position (x, y) of the base point and the orientation angle θ with respect to the base. A robot location in $\mathcal{W}$, also called a robot configuration, uniquely corresponds to a point (x, y, θ) in the robot configuration space $\mathcal{C}$.

The kinematic constraint of a nonholonomic car-like robot is described as

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0. \tag{5.1}$$

Thus a nonholonomic car-like robot has 2-d.o.f. Equation (5.1) can be parameterized by time t. Given the robot velocity v and the curvature K of the curve followed by the robot (see Figure 5.1A), the so called *control variables*, the velocity parameters of the robot are given by

$$\dot{x} = v\sin\theta, \quad \dot{y} = v\cos\theta \quad \text{and} \quad \dot{\theta} = vK, \tag{5.2}$$

where the control variables v are K are limited to $v_{\max}$ and $K_{\max}$, respectively, i.e.,

$$|K| \leq K_{\max} \quad \text{and} \quad |v| \leq v_{\max}. \tag{5.3}$$

The minimum turning radius R is given by $R = 1/K_{\text{max}}$. When planning the robot motion, the control variables v and K must be discretized (Latombe, 1991; Podsedkowski, 1998). As shown in Latombe (1991) and Podsedkowski (1998), for a given robot configuration, there are at most six possible next robot configurations by setting the v and K to the six discretized values:

$$\{-v_0, v_0\} \times \{-K_{max}, 0, K_{max}\}, \tag{5.4}$$

where v_0 is the discretized robot moving velocity. 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 given by

$$\begin{aligned} \theta(t + \Delta t) &= \theta(t) + vK\Delta t, \\ x(t + \Delta t) &= x(t) + \frac{1}{K}(\sin\theta(t + \Delta t) - \sin\theta(t)), \\ y(t + \Delta t) &= y(t) + \frac{1}{K}(\cos\theta(t + \Delta t) - \cos\theta(t)), \end{aligned} \tag{5.5}$$

where Δt is time interval of one step. Figure 5.1B shows an example of the possible next robot configurations of a given robot configuration. In realistic robot motion planning, the length of one step of robot movement is significantly smaller, so the next robot configuration partially overlaps themselves.

If there is an obstacle in the 2D Cartesian workspace $\mathcal{W}$, several configurations of the car-like robot in $\mathcal{C}$ are not allowed ("forbidden", called the obstacle configurations in $\mathcal{C}$). For example, when the obstacle and the robot base point are at positions (0,0) and (1,0) in $\mathcal{W}$, respectively, the forbidden robot configurations are shown in Figure 4.1C, where the obstacles are shown by dark solid squares, while the robot is shown by light solid rectangular with its base point shown by a dark solid circle.

5.2.2 The Neural Network Model

The proposed neural network model for real-time collision-free motion planning of nonholonomic car-like robots in a nonstationary environment is based on the neural network approach to motion planning of holonomic 3-d.o.f. robots presented in Chap-

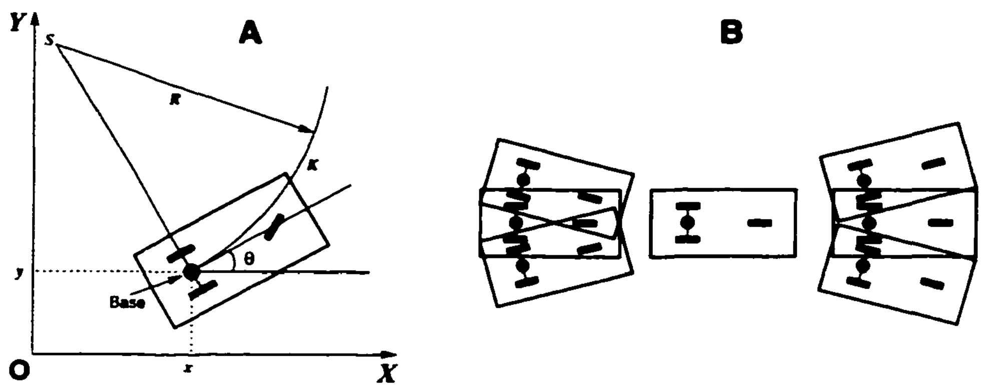


Figure 5.1: *Schematic diagram for a nonholonomic car-like robot.* A: *the kinematics of a car-like robot, where* (x, y) *is the base point,* θ *is the orientation angle,* R *is the turning radius, and* K *is the curvature of curve followed by the car;* B: *the six possible next robot configurations of a given robot configuration.*

ter 4 (also in Yang and Meng, 1999*c*; Yang and Meng, 1999*d*). Like the model for 3-d.o.f. robots, the proposed neural network architecture is a topologically organized, locally connected map, where the dynamics of each neuron is characterized by a shunting equation or an additive equation. Unlike those models presented in Chapters 2, 3, and 4 (Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1998*c*; Yang and Meng, 1999*a*; Yang and Meng, 1999*b*; Yang and Meng, 1999*d*; Yang and Meng, 1999*e*) where the lateral connection strength of each neuron to its neighboring neuron is a function of the distance only and the neural activity propagation is omnidirectional, in the proposed neural network model the lateral neural connection strength is directionally selective, and the neural activity propagation is subject to the kinematic constraint of the nonholonomic car-like robot.

The proposed neural network architecture is expressed in a 3D state space $\mathcal{S}$, where two represent the spatial position in the 2D Cartesian workspace and one represents the orientation of the robot, i.e., the state space $\mathcal{S}$ is the robot configuration space $\mathcal{C}$. The location of the i-th 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 car-like robot. The dynamics of the i-th neuron in the neural network is characterized by a shunting equation,

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)\left([I_i]^+ + \sum_{j=1}^{N} w_{ij}[\xi_j]^+\right) - (D + \xi_i)[I_i]^-. \tag{5.6}$$

where all the parameters and functions are defined as the same as in Equation (4.1), except the lateral neural connection weight w_{ij}. Parameters A, B and D represent the passive decay rate, the upper and lower bounds of the neural activity, respectively. Variable ξ_i is the neural activity of the i-th neuron. The excitatory input, $[I_i]^+ + \sum_{j=1}^{n} w_{ij}[\xi_j]^+$, results from the targets and the lateral connections among neurons, while the inhibitory input $[I_i]^-$ results from the obstacles. Functions $[a]^+$ and $[a]^-$ are defined as, $[a]^+ = \max\{a, 0\}$ and $[a]^- = \max\{-a, 0\}$. 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.

Unlike the previous neural network models presented in Chapters 2, 3 and 4 where the lateral connection weight is defined as a function of the distance only, in the proposed model the lateral connection weight, w_{ij}, from the j-th neuron to the i-th neuron 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 the state space of the neural network,

$$w_{ij} = \begin{cases} f(d_{ij}), & \text{if the } i\text{-th and } j\text{-th neurons are neighboring} \\ 0 & \text{otherwise} \end{cases}, \tag{5.7}$$

where function $f(d_{ij})$ is a monotonically decreasing function, e.g., a function defined as the same as in previous chapters: $f(d_{ij}) = \mu/d_{ij}$, if $0 < d_{ij} < r_0$; $f(d_{ij}) = 0$, otherwise, where μ and r_0 are positive constants. In contrast to those neural network models presented Chapters 2, 3 and 4 where the neighboring neurons of the i-th neuron are defined in Equations (2.5) and (2.6) as all neurons whose Cartesian distance d_{ij} to the i-th neuron is less than a constant r_0, in the proposed model the *neighboring neurons* is defined in Equation (5.5), i.e., the neighboring neurons of the i-th neuron are all neurons satisfying the kinematic constraint *and* whose distance to the i-th neuron is less than r_0. Therefore, due to the kinematic constraint of the nonholonomic car-like robot, the i-th neuron has at most six neighboring neurons.

An example illustrating all the neighboring configurations (next car locations) of a given robot configuration is shown in Figure 5.1B, where the relationship between the neighboring locations and the given robot location satisfies the kinematic constraint described in Equation (5.5). There are at most six neurons in the receptive field of the i-th neuron in the neural network. 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, $w_{ij} = w_{ji}$. The dynamics of the i-th neuron can be further written as,

$$\frac{d\xi_i}{dt} = -A\xi_i + (B - \xi_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[\xi_j]^+\right) - (D + \xi_i)[I_i]^-, \tag{5.8}$$

where $k \leq 6$ is the number of all the neighboring neurons of the i-th neuron.

The proposed neural network characterized by Equation (5.8) 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. In contrast to those neural network models presented in Chapters 2, 3, and 4 (Meng and Yang, 1998; Yang and Meng, 1998*a*; Yang and Meng, 1998*c*; Yang and Meng, 1999*a*; Yang and Meng, 1999*b*; Yang and Meng, 1999*d*; Yang and Meng, 1999*e*) where the neural activity propagation to all directions is exactly in the same manner, the activity propagation in the proposed model is directionally selective, which is subject to the kinematic constraint described in Equations (5.1) or (5.2). 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 optimal robot motion is planned from the dynamic activity landscape by a 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

$$p_n \quad \Leftarrow \quad \xi_{p_n} = \max\{\xi_j, j = 1, 2, \cdots, k\}, \tag{5.9}$$

where k is the number of all the neighboring neurons of the present location p_p, i.e., all the possible next locations described in Equation (5.5). 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.

The proposed neural network is a stable system. The neural activity in the shunting equation (5.8) is bounded in the finite interval $[-D, B]$. Although the lateral neural connections are directionally selective, the connection weight w_{ij} in proposed neural network model in Equation (5.8) is *symmetric*. Similar to the Lyapunov stability analysis presented in Chapter 2, the neural network model in Equation (5.8) can be rewritten into Grossberg's general form in Equation (2.9), and we can prove that Equation (5.8) satisfies all the three stability conditions required by Grossberg's general form (Yang and Meng, 1999*f*). Therefore, the stability and convergence of the proposed neural network system be rigorously proved using a Lyapunov stability theory (Grossberg, 1983; Grossberg, 1988; Yang and Meng, 1999*f*).

5.3 Simulation Studies

In this section, several simulations are carried out to demonstrate the effectiveness of the proposed neural network approach to real-time collision-free motion planning of nonholonomic car-like robots. The real-time motion planning for parallel parking, motion planning in a house-like environment with several deadlock situations, and motion planning with sudden environmental changes are studies.

5.3.1 Motion Planning for Parallel Parking

The proposed neural network model is first applied to the famous parallel parking problem. Motion planning for parallel parking of a nonholonomic car-like robot under various situations are studied. The neural network has $40 \times 30 \times 24$ topologically ordered neurons, where 40×30 represents the discretized 2D workspace at a size of 40 *block* $\times$ 30 *block*, and 24 represents the discretized orientation angles from $0°$ to $345°$ with a step of $15°$. 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 $E = 100$

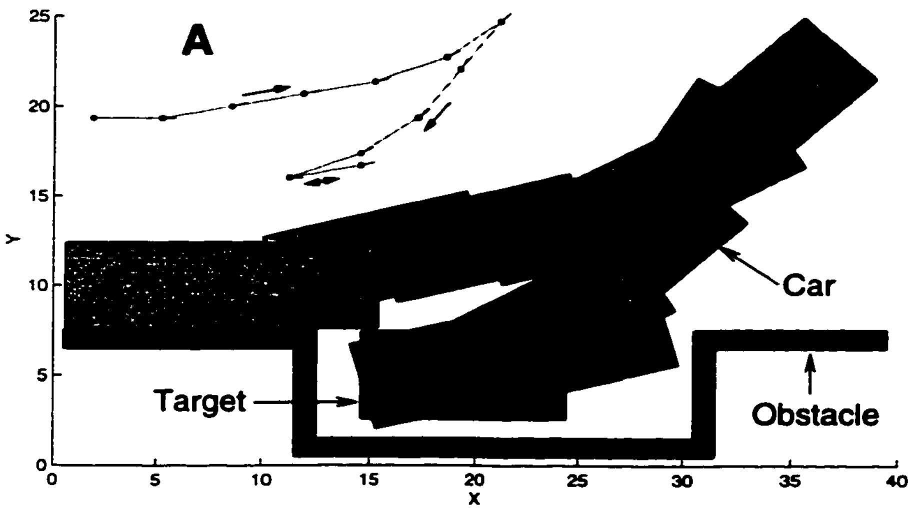
for the external input. The minimum turning radius of the car is R=20 *block*, i.e., the maximum curvature of the curve followed by the car-like robot is $K = 1/R = 0.05$. The planned motion to park the car is shown in Figure 5.2A, where the inset shows the dynamic trace of the base point and orientation of the car. It shows that the car first moves forward to the left, then turns backward to park the car at the target location without any collisions. In case of a narrow road (Figure 5.2B), the car has to turn back and forth for several times, and eventually the car is also able to properly park at the target location with obstacle avoidance. The real-time robot motion is shown in Figure 5.2B, where the inset shows the dynamic trace of base point and orientation of the car.

5.3.2 Motion Planning in a House-like Environment

Then, the proposed neural network model is applied to a complex house-like environment, where there are several deadlock situations that the robot may be trapped in. Figure 5.3 shows a house-like environment, where the doors can be opened or closed. The neural network has $90 \times 90 \times 24$ neurons. The model parameters are chosen as the same as the previous case, i.e., $A = 10$, $B = D = 1$, $\mu = 1$, $r_0 = 2$ and $E = 100$. In case that Door L is opened, the planned car motion is shown in Figure 5.3A, where the car moves to the target along the shortest path. When Door L is closed, the planned motion path is shown in Figure 5.3B. The car has to travel a much longer path to reach the target. Note there are no learning procedures. The car is capable of reach the target along the shortest path without any collisions, without violating the kinematic constraint, and without being trapped in any deadlock situations.

5.3.3 Motion Planning with Sudden Environmental Changes

The proposed neural network model can perform properly in an arbitrarily dynamic environment, even with sudden environmental changes. A case with sudden placement of obstacles in front of a nonholonomic car is studied. The neural network has $50 \times 30 \times 24$ neurons, and the model parameters are chosen as the same as in the previous



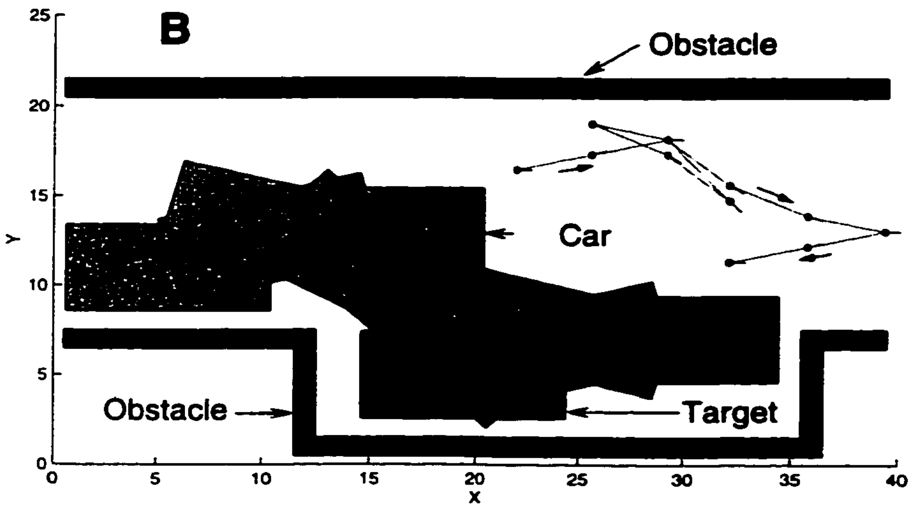
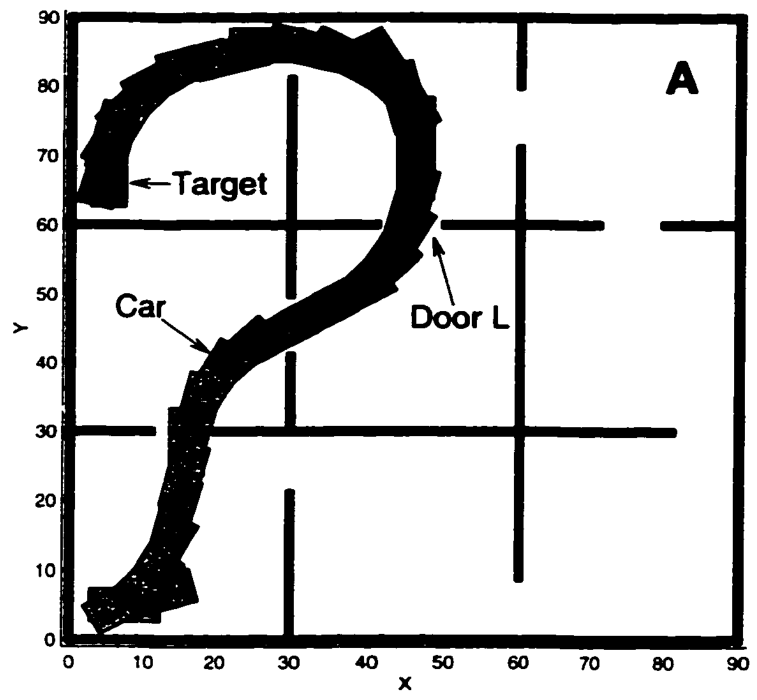


Figure 5.2: *Motion planning of a nonholonomic car-like robot in parallel parking problem.* A: *parallel parking in a wide area;* B: *parallel parking in a narrow road.*



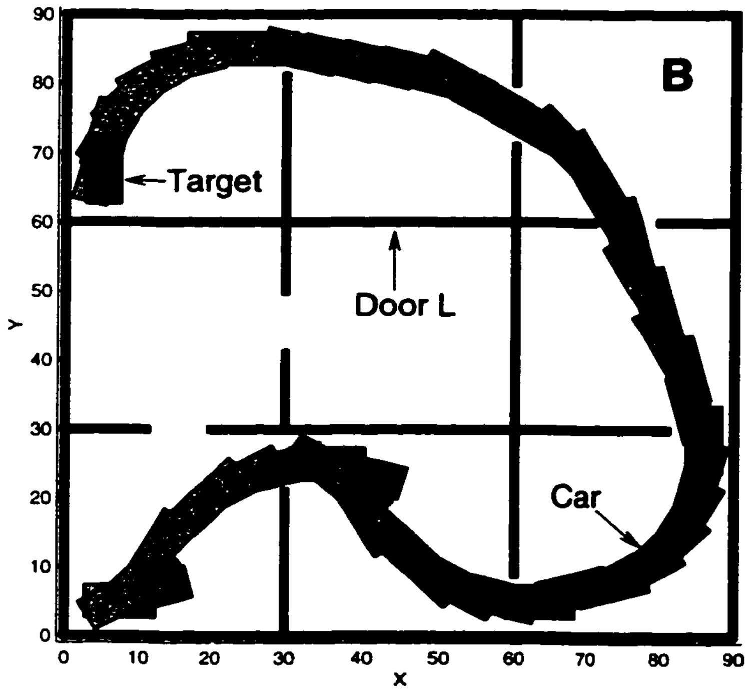


Figure 5.3: *Motion planning of a nonholonomic car-like robot in a house-like environment with several deadlock situations.* A: *the planned robot motion path when door L is opened;* B: *the path when door L is closed.*

cases. The initial and target locations of the car are at locations (5,5,6) and (40,25,0) in $\mathcal{W}$, respectively. First, the car motion path without any obstacles in the workspace is shown in Figure 5.4A. In the second case with the same initial condition, when the car reaches location (15,22,2) on the way toward the target, the obstacles (V-shaped, shown in Figure 5.4B with dark sold squares) are ***suddenly*** placed in front of the car. The real-time car motion is shown in Figure 5.4B, where the car 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.

5.4 Model Variation

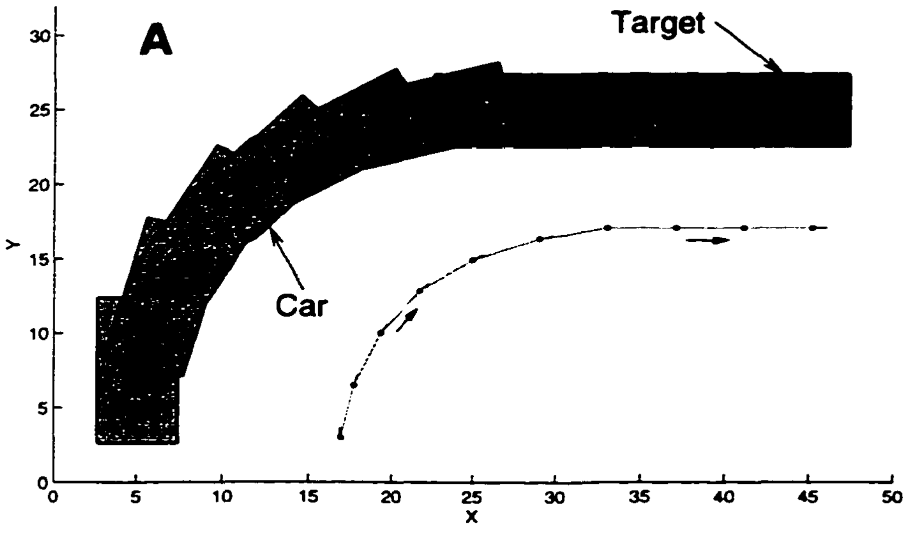
If the excitatory and inhibitory connections in the shunting model in Equation (5.8) are lumped together and the auto gain control terms are removed, then Equation (5.8) may be written into a simpler additive equation,

$$\frac{dx_i}{dt} = -Ax_i + I_i + \sum_{j=1}^{k} w_{ij}[x_j]^+, \tag{5.10}$$

where the term $I_i + \sum_{j=1}^{k} w_{ij}[x_j]^+$ represents the total input to the i-th neuron from the external and lateral connections. The definitions of I_i, $[a]^+$ and w_{ij} are the same as those in Equation (5.8). The stability and convergence of this simple additive model can be easily proved by rewriting Equation (5.10) into Grossberg's general form in Equation (2.9). This additive model is capable of planning real-time collision-free motion of nonholonomic car-like robots in most situations. There are a lot of important differences between the shunting and the additive models (see Chapter 2), although the additive model is computationally simpler.

5.5 Conclusion

In this chapter, a novel neural network approach is proposed for real-time collision-free motion planning of nonholonomic car-like robots in a nonstationary environment, which is based on the neural network model for motion planning of holonomic 3-



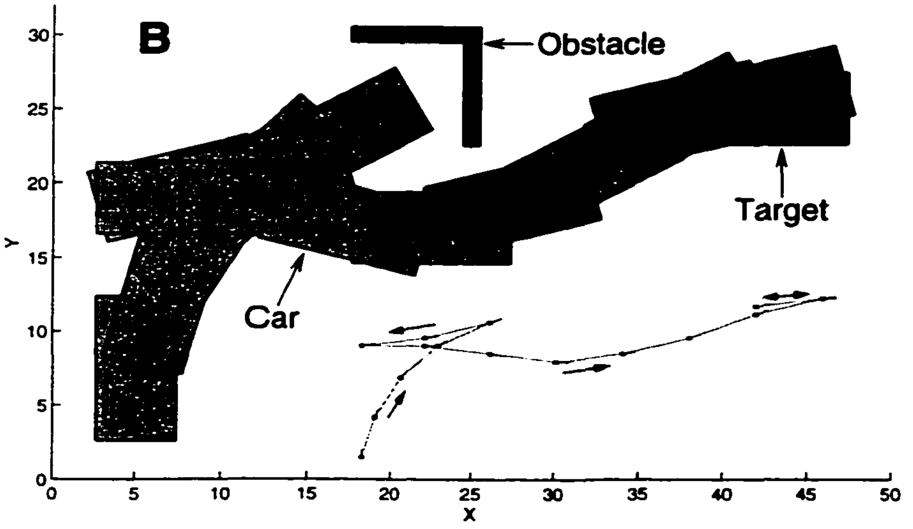


Figure 5.4: *Real-time planning of a nonholonomic car-like robot with sudden placement of obstacles.* A: *the planned car motion in case of no obstacles;* B: *the real-time motion in case that V-shaped obstacles suddenly are placed in front of the car.*

d.o.f. robots. The state space of the neural network is the configuration space of the robot. Unlike those neural network models present in Chapters 2, 3, and 4 where the lateral connection strength of one neuron to its neighboring neuron is a function of the distance only and the neural activity propagation is omnidirectional, in the proposed neural network model the lateral connection strength is directionally selective, and the neural activity propagation is subject to the kinematic constraint of the holonomic car-like robot. The optimal real-time robot motion is planned through the dynamic activity landscape of the neural network. The planned robot motion in a static environment is globally optimal in the sense of the least steps from the starting location to the target if it exists. The optimality of the real-time motion planning in a dynamic environment is in the sense of a continuous, smooth path toward the target. The stability and convergence of the proposed neural network system is also proved by a Lyapunov stability theory. To the best of our knowledge, it is the first time that the real-time motion planning of car-like robots are studied using a non-learning based neural network approach. Several feature properties are worth to notice about the proposed neural network model for motion planning of nonholonomic car-like robots:

- This model can perform properly in an arbitrary varying environment, even with sudden environmental changes.
- This model is biologically plausible.
- This model does not suffer from undesired local minima, i.e., it will not be trapped in the deadlock situations.
- This model is not very sensitive to the model parameters, connection weight function, nor sensor noise.
- The computational complexity linearly depends on the state space size of the neural network.
- The model algorithm is computationally efficient.

Chapter 6

An Efficient Neural Network Based Approach to Real-time Fine Motion Control of Robot Manipulators

In this chapter, a novel neural network based approach is proposed for real-time fine motion control of robot manipulators without any knowledge of the robot dynamics and subject to significant uncertainties. The controller structure consists of a simple feedforward neural network and a PD feedback loop, which inherits advantages from both the neural network based controllers and the traditional PD type controllers. By taking advantage of the robot regressor dynamics, the neural network assumes a single-layer structure, and the learning algorithm is computationally *efficient*. The real-time fine motion control of robot manipulators is achieved through *only* the on-line learning of the neural network, without any off-line training procedures. The PD control loop guarantees the global stability during the initial learning period of the neural network or sudden changes in the robot dynamics occur. In addition, the proposed neural network based controller does not require any knowledge of the robot dynamics, and is capable of quickly compensating sudden changes in the robot dynamics. The global asymptotic stability of the system and convergence of the tracking error are proved using a Lyapunov stability analysis. A model variation based on the same concept is presented. The proposed controller is applied to a two-link robot manipulator to track an elliptic trajectory and to compensate sudden changes in the robot dynamics in real-time. The effectiveness and the efficiency of the proposed controller are demonstrated through simulation and comparison studies.

6.1 Introduction

Motion control of robot manipulators in real-time is a very important but also difficult issue in robotics, especially when there is no knowledge of the robot dynamics and when sudden changes in the robot dynamics occur. There are a lot of studies on motion control of robot manipulators using various approaches. The PD controller is very simple and does not require any knowledge of the robot dynamics. However it requires very large actuation to achieve fine control, which is not practical but highly demanded in many cases (Meng, 1992; Meng, 1995; Meng, 1996). The computed torque control approach and other model-based approaches (e.g. Meng, 1995; Dixon *et al.*, 1999) are capable of achieving fine motion control. However, they require the exact model of the robot dynamics to achieve fine motion control, which is almost impossible in practice (Meng, 1992; Meng, 1996). The adaptive controllers (e.g. Slotine and Li, 1987; Slotine and Li, 1988; Meng, 1992; Lu and Meng, 1993; Meng and Lu, 1994; Chiaverini *et al.*, 1997; ElDeeb and ElMaraghy, 1998) can achieve fine control and compensate the partially known manipulator dynamics. But, they often require complicated on-line estimation of the robot dynamics (Slotine and Li, 1987; Slotine and Li, 1988; Meng, 1992; Lu and Meng, 1993; Meng, 1996).

Many neural network based controllers were proposed, which succeeded in some areas where the model-based approaches failed. Khemaissia and Morris (1993) proposed a neuro-adaptive controller by using a neural network plus a servo feedback controller, which does not require an off-line training. However, it suffers from computational complexity and very slow convergence, because for an n-d.o.f. robot manipulator this model requires n multi-layer neural networks. Later Morris and Khemaissia (1996) proposed a new neural network based controller by using recursive prediction error technique to improve the convergence speed. But it is still computationally complicated. Behera *et al.* (1996) proposed a neuro-adaptive hybrid controller for robot manipulator tracking control, where three multi-layer neural networks are used to learn the mass matrix, centrifugal and Coriolis force matrix and the gravitational torque vector, respectively, but it is computationally expensive.

Meng (1996) proposed a signal-layer neural network based model for fine control of robot manipulators. This model takes advantage of the robot regressor dynamics formulation that expresses the highly nonlinear robot dynamics in a linear form in terms of the robot dynamic parameters, thus the control algorithm is computationally simple. However, Meng's (1996) model requires an off-line training procedure for the neural network to perform satisfactorily.

In this chapter, a novel neural network based approach is proposed for real-time fine motion control of multi-joint robot manipulators without any knowledge of the robot dynamics and without any off-line training procedures (Yang and Meng, 1998*b*; Meng and Yang, 1999; Yang and Meng, 1999*h*). This model is also capable of compensating sudden changes in the robot dynamics. The proposed controller consists of two parts: a control torque from a feedforward neural network and a control torque from a PD feedback loop. The PD feedback loop guarantees the global stability during the learning period of the neural network at the initial phase *or* when a sudden change in the robot dynamics occurs. Unlike most neural network based controllers that require off-line training, no off-line training procedures are needed in the proposed controller. The real-time fine motion control of robot manipulator is achieved through only the *on-line* learning of the neural network. Therefore it inherits advantages from both the neural network based controllers and the PD type controllers. By taking advantage of the robot regressor dynamics formulation, the neural network assumes a single-layer structure. The learning algorithm of the neural network is derived from the global stability analysis of a Lyapunov function candidate, which is computationally *efficient*. The control system is guaranteed to be globally asymptotically stable, and the tracking error is proved to converge to zero.

6.2 The Control Algorithm

In this section, we will briefly introduce the robot dynamics and regressor dynamics of robot manipulator. Then, the philosophy of the proposed neural network based approach to real-time fine motion control of robot manipulators is presented, including the control algorithm and the on-line learning algorithm of the neural network. The system stability analysis using a Lyapunov theory and the proof of convergence of tracking error are provided.

6.2.1 The Robot Dynamics and Regressor Dynamics

In absence of friction and other disturbances, the dynamics of an n-d.o.f. rigid robot manipulator can be modeled by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{6.1}$$

where τ is the $n \times 1$ vector of joint torques; q, $\dot{q}$ and $\ddot{q}$ are the $n \times 1$ vectors of the joint position, velocity, and acceleration, respectively; M is the $n \times n$ symmetric, positive definite mass matrix; $C\dot{q}$ is the $n \times 1$ vector of centripetal and Coriolis torques; and G is the $n \times 1$ vector of gravitational torques (Meng, 1992; Lu and Meng, 1993; Meng, 1996).

One very important property of robot dynamics, the *regressor dynamics* (Slotine and Li, 1987; Meng, 1992; Lu and Meng, 1993; Meng, 1996), is used in the proposed controller design. The highly nonlinear robot dynamics in Equation (6.1) can be expressed in a linear form in terms of the robot dynamic parameters,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Y(q,\dot{q},\ddot{q})\theta = \tau, \tag{6.2}$$

where θ is a $r \times 1$ vector consisting of the known and unknown robot dynamic parameters, such as link masses, moments of inertias, etc.; Y is an $n \times r$ coefficient matrix consisting of the known functions of joint position q, velocity $\dot{q}$ and acceleration $\ddot{q}$, which is referred to as the *manipulator regressor*.

6.2.2 The Control Algorithm

The function of a controller is to implement a mapping between the known information (e.g., the desired information and the sensory information) and the actuator commands designed to achieve the robot's task. The controller design problem can be described as: given the desired manipulator joint position $q_d(t)$, velocity $\dot{q}_d(t)$ and acceleration $\ddot{q}_d(t)$, design a control law for the actuator torques, which drive the robot to move, such that the joint position $q(t)$ precisely tracks $q_d(t)$.

The fundamental concept of the proposed controller is to take advantages from both the PD type controllers and the neural network based controllers. In addition, by incorporating the robot regressor dynamics formulation, the neural network assumes a single-layer architecture. The very simple learning algorithm is derived from the Lyapunov stability analysis, thus the global system stability and convergence are guaranteed. Therefore, in the proposed model, the actuator torque consists of two parts: the torque from a PD feedback loop and the torque from the feedforward neural network. The PD loop is used to guarantee the global stability. The neural network is used to achieve the real-time fine robot motion control through its on-line learning.

A schematic diagram of the proposed neural network based controller is shown in Figure 6.1, which consists of a feedforward neural network and a feedback PD loop. The PD feedback loop guarantees that the system is globally stable during the learning period of the neural network at the initial phase or when a sudden change in the robot dynamics happens, e.g., the robot picks up a load or drops off a load. The neural network is capable of achieving real-time fine robot motion control after a short on-line learning period and performing better and better as time goes on. Therefore the proposed controller inherits advantages from both the neural networks based controllers and the PD type controllers.

The control law for the actuator torque τ of an n-d.o.f. rigid robot manipulator is assigned as

$$\tau = \tau_{NN} + \tau_{PD}, \tag{6.3}$$

where τ_{NN} and τ_{PD} are $n \times 1$ vectors that represent the torques from the single-layer

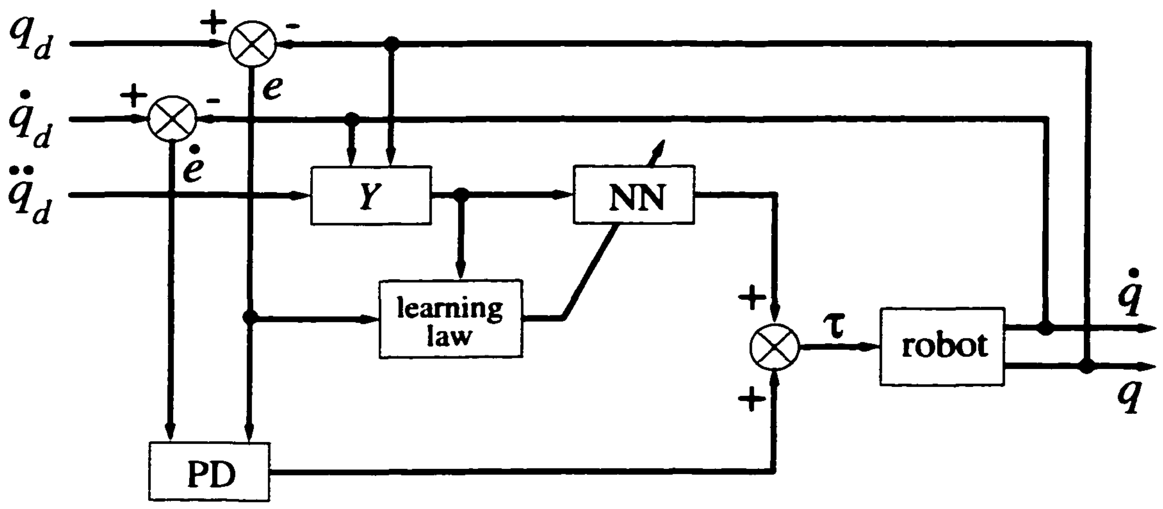


Figure 6.1: *Schematic diagram of the proposed neural network based controller.*

neural network and the PD loop, respectively. The torque τ_{PD} is given by

$$\tau_{PD} = K_p e + K_d \dot{e}, \tag{6.4}$$

where $e = q_d - q$ is the $n \times 1$ error vector of the joint position. The torque τ_{NN} is the output the single-layer neural network, which is given by

$$\tau_{NN} = Y(q, \dot{q}, \ddot{q})W, \tag{6.5}$$

where W is a $r \times 1$ vector that represents the connection weights of the neural network. The input of the neural network is the regressor Y, which is an $n \times r$ matrix.

6.2.3 The Learning Algorithm and Stability Analysis

Substituting Equations (6.3)-(6.5) into (6.2), the error dynamics is obtained as

$$K_d \dot{e} + K_p e = Y(q, \dot{q}, \ddot{q})\theta - Y(q, \dot{q}, \ddot{q})W, \tag{6.6}$$

i.e.,

$$\dot{e} + Ke = K_d^{-1} Y(\theta - W) = -K_d^{-1} Y \tilde{\theta}, \tag{6.7}$$

where $\tilde{\theta} \equiv W - \theta$, and $K = K_d^{-1}K_p$ is a positive constant matrix. Substituting $A = -K$ and $B = K_d^{-1}$, Equation (6.7) can be rewritten as

$$\dot{e} = Ae - BY\tilde{\theta}. \tag{6.8}$$

To derive the learning law of the neural network and to prove the system stability and track error convergence, the Lyapunov function candidate of the control system is chosen as,

$$v(t) = e^T Pe + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}, \tag{6.9}$$

where P and Γ are $n \times n$ and $r \times r$ positive definite constant matrices, respectively. Apparently the Lyapunov function candidate is positive definite, i.e., $v(t) \geq 0$. In addition, $v(t) = 0$ if and only if $e = 0$ and $\tilde{\theta} = 0$. By differentiation, the derivative of $v(t)$ along the trajectory of the system is given by

$$\begin{aligned} \dot{v}(t) &= \dot{e}^T Pe + e^T P\dot{e} + 2\tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \\ &= e^T(PA + AP)e - e^T PBY\tilde{\theta} - \tilde{\theta}^T Y^T B^T Pe + 2\tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \\ &= -e^T Qe + 2\tilde{\theta}^T(\Gamma^{-1}\dot{\tilde{\theta}} - Y^T B^T Pe), \end{aligned} \tag{6.10}$$

where $Q = -(PA + AP) = PK + KP$ is a positive definite symmetric matrix. By choosing the following equation,

$$\Gamma^{-1}\dot{\tilde{\theta}} - Y^T B^T Pe = 0, \tag{6.11}$$

then Equation (6.10) becomes

$$\dot{v}(t) = -e^T Qe \leq 0. \tag{6.12}$$

Equation (6.12) implies that the proposed control system is globally stable. In addition, Equation (6.12) implies that $\dot{v} = 0$ if and only if $e = 0$. Therefore, according to LaSalle's lemma, the control system is asymptotically stable, and the joint position tracking error e is guaranteed to converge to zero.

Since $\tilde{\theta} = W - \theta$, then $\dot{\tilde{\theta}} = \dot{W}$. Therefore, from Equation (6.11), the derivative of W is given as

$$\dot{W} = \Gamma Y^T B^T Pe. \tag{6.13}$$

Equation (6.13) is used as the learning law for updating the connection weight W of the neural network in the proposed model. Obviously, it is much simpler than most commonly used learning rules such as the famous least mean square learning rule and the multi-layer network back-propagation (BP) learning rule.

In summary, Equations (6.4), (6.5) and (6.13) define the proposed control algorithm.

6.3 A Case Study

The proposed neural network based controller is capable of achieving real-time fine motion control of multi-joint robot manipulators without any off-line training procedures and without any knowledge of the robot dynamics. The control law is characterized in Equation (6.3), and the corresponding learning law of the neural network is characterized in Equation (6.13). For simplification without losing generality, in this section, the proposed controller is applied to a simple two-link robot manipulator to track a desired elliptic trajectory and to compensate a sudden change in the robot dynamics in real-time with on-line learning only.

The two-link robot manipulator (shown in Figure 6.2) is modeled as two rigid links with lengths $l_1 = 1m$ and $l_2 = 2m$, and with point masses $m_1 = 2kg$ and $m_2 = 1kg$ (including the load at the tip of Link 2) at the distal ends of the links. In this case, the signal-layer neural network has 5 neurons, thus the connection weight W is a 5×1 vector. The input of the neural network is the 2×5 matrix of the robot regressor Y. The computation of Y requires the measurement of acceleration $\ddot{q}$ in addition to joint position q and velocity $\dot{q}$. To avoid this impractical requirement, $\ddot{q}_d$ is used to replace $\ddot{q}$. Thus Y is given by (Meng, 1992; Lu and Meng, 1993; Meng, 1996; Yang and Meng, 1998*b*; Meng and Yang, 1999)

$$Y = \begin{pmatrix} c_2(2\ddot{q}_{1d} + \ddot{q}_{2d}) - s_2\dot{q}_2(\dot{q}_1 + 2\dot{q}_2) & \ddot{q}_{1d} + \ddot{q}_{2d} & gc_{12} & gc_1 & \ddot{q}_{1d} \\ c_2\ddot{q}_{1d} + s_2\dot{q}_1^2 & \ddot{q}_{1d} + \ddot{q}_{2d} & gc_{12} & 0 & 0 \end{pmatrix} \tag{6.14}$$

where $c_i = \cos(q_i)$, $s_i = \sin(q_i)$ and $c_{12} = \cos(q_1 + q_2)$, and g is the gravitational acceleration constant. The output of the neural network t_{NN} is a 2×1 vector, which is the torque from the neural network. The learning rule in Equation (6.13) is used to update the weight W. Note that there are *no off-line* training procedures in the proposed controller. During the on-line learning of the neural network, the connection weight W updates once when the robot moves one step. In the simulations, the robot takes 628 steps to track the whole ellipse, therefore the neural network learns 628 times per tracking cycle. The model parameters used in the simulations are chosen as: $K_p = 100I_2, K_d = 14I_2, P = I_2$, and $\Gamma = 15I_5$, where I_k is a $k \times k$ identical matrix.

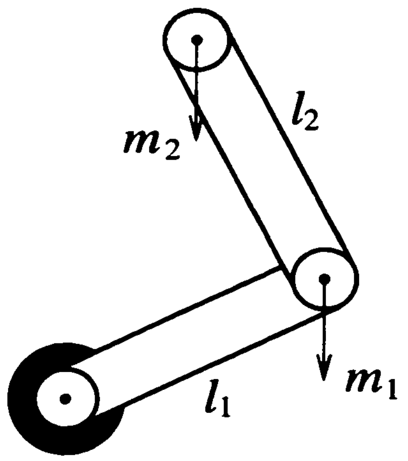


Figure 6.2: *Schematic diagram of a two-link robot manipulator. It is modeled as two rigid links with lengths $l_1 = 1m$ and $l_2 = 2m$, and with point masses $m_1 = 2kg$ and $m_2 = 1kg$ at the distal ends of the links.*

6.3.1 Motion Control of a Robot Manipulator to Track an Elliptic Trajectory

The proposed controller is used to control this two-link robot to track an elliptic trajectory (shown in Figure 6.3A with solid line). In the simulation, it is assumed that the robot dynamics is *completely unknown*. The initial connection weight W is set randomly within the interval [0,1]. The initial joint position is set to be at an position with 20% offset from the desired initial joint position, i.e., $q(0) = 0.8q_d(0)$.

A Coulomb friction term,

$$F = [0.5sign(\dot{q_1}) \quad 0.5sign(\dot{q_2})]^T, \tag{6.15}$$

is added to the robot dynamics, which is not modeled in the control algorithm in Equation (6.3). The dynamic tracking performance in the workspace is shown in Figure 6.3A. Figure 6.3B shows the error dynamics of the joint position. It shows that the tracking error sharply drops to near zero. Thus the proposed controller can achieve good real-time tracking performance through only the on-line learning of the neural network, even though there exist unmodeled disturbances. The applied torques to Joints 1 and 2 are depicted in Figures 6.4A and 6.4B, respectively. The total applied control torque, the torque from the neural network and the torque from the PD loop are plotted. It shows that at the initial period, the main contribution to the total applied torques results from the PD loop, which then quickly decreases to near zero. After the initial learning phase, the contribution from the neural network increases and becomes the dominant component.

To see the roles played by the PD loop and the neural network, the two-link robot manipulator is controlled by the PD loop alone and the neural network alone to track the same elliptic trajectory. First, the robot is controlled by the PD control loop alone without the neural network, where the parameters K_p and K_d are chosen as the same as in the previous simulation. Figure 6.5A shows the dynamic robot tracking performance in workspace, while the error dynamics of the joint position is shown in Figure 6.5B. It shows that the PD control loop alone cannot achieve a similar level of fine motion control. Then, the robot is controlled by the signal-layer neural network alone without the PD loop. Simulations show that the system is unstable and the robot moves in an uncontrolled manner. The connection weights and the tracking error quickly goes to infinity. This is because that the proposed learning algorithm for the neural network is derived from the global stability analysis of a Lyapunov function candidate with the presence of the PD loop. The neural network alone cannot perform properly without some off-line training or without the PD loop.

The ability to learn of the neural network ensures that the proposed controller

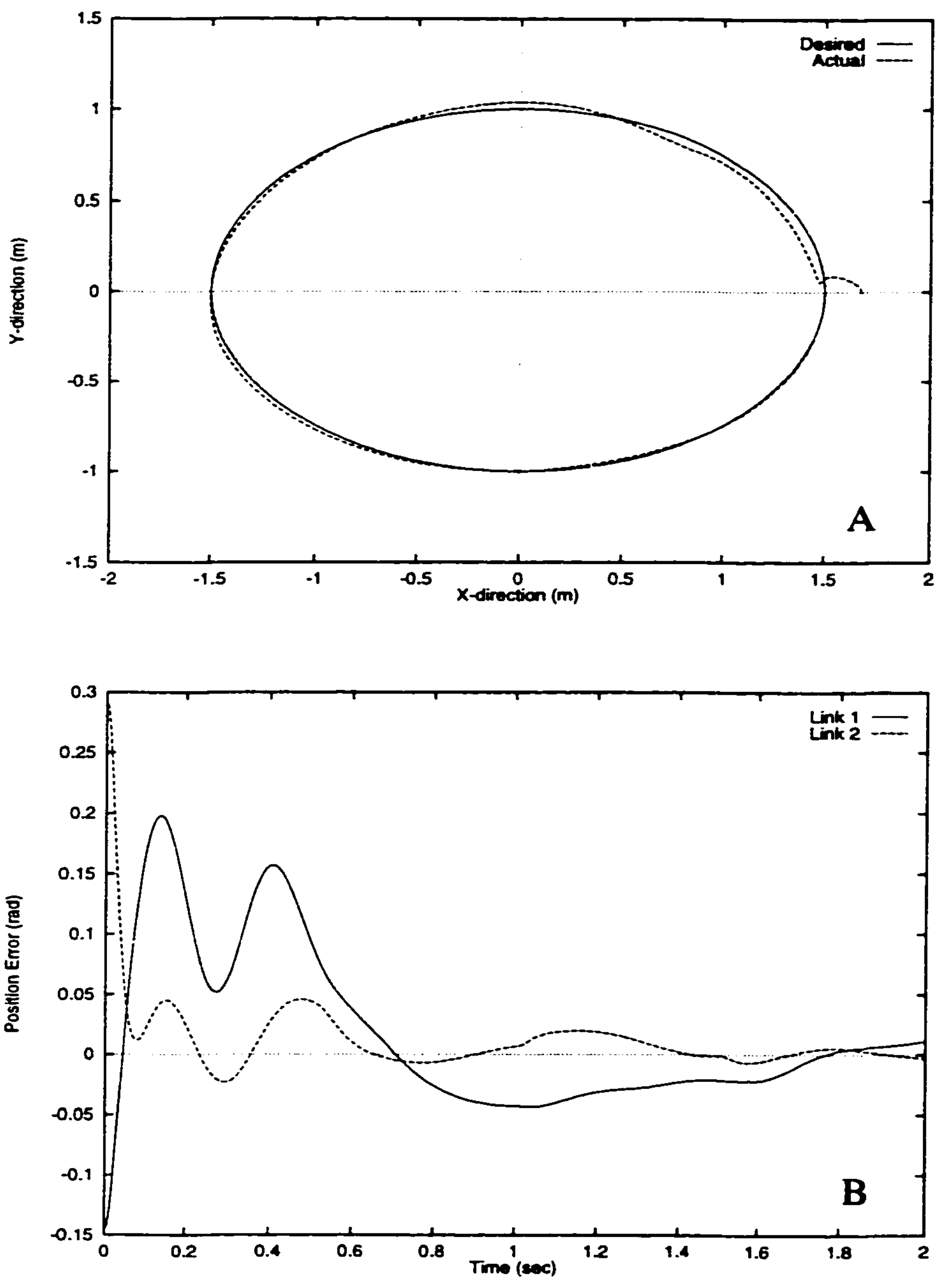
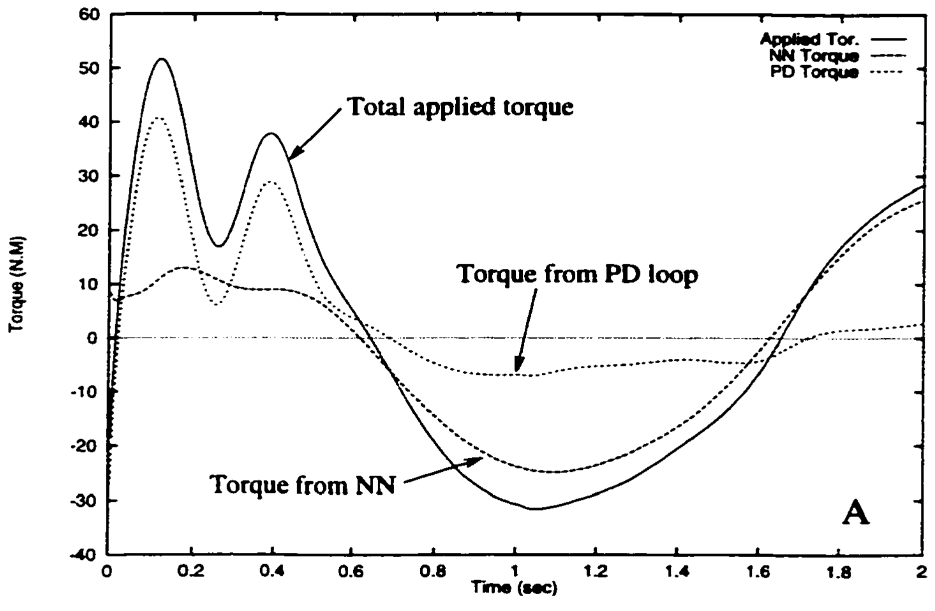


Figure 6.3: *Motion control of a two-link robot manipulator to track an elliptic trajectory using the proposed controller.* A: *the real-time tracking performance;* B: *the error dynamics of the joint position.*



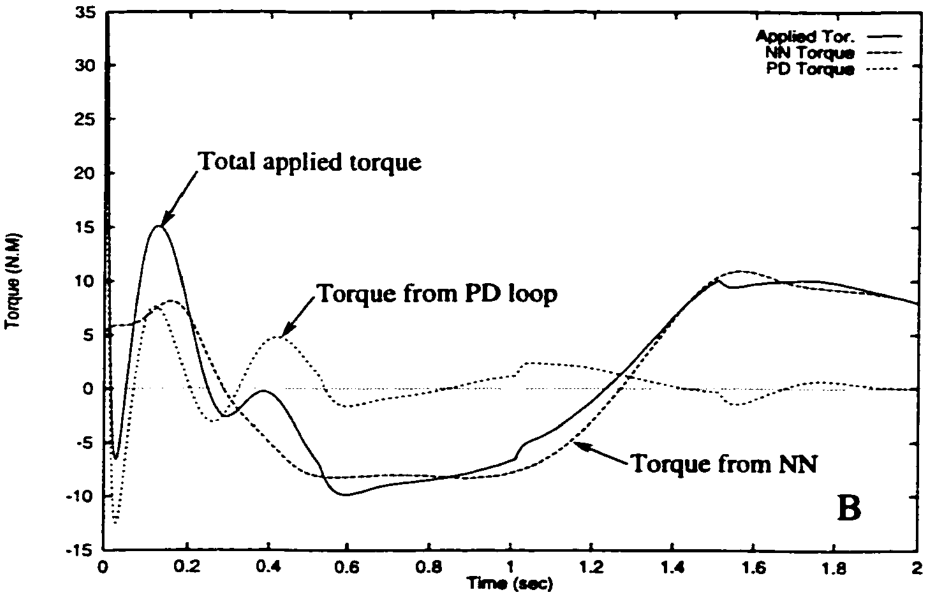
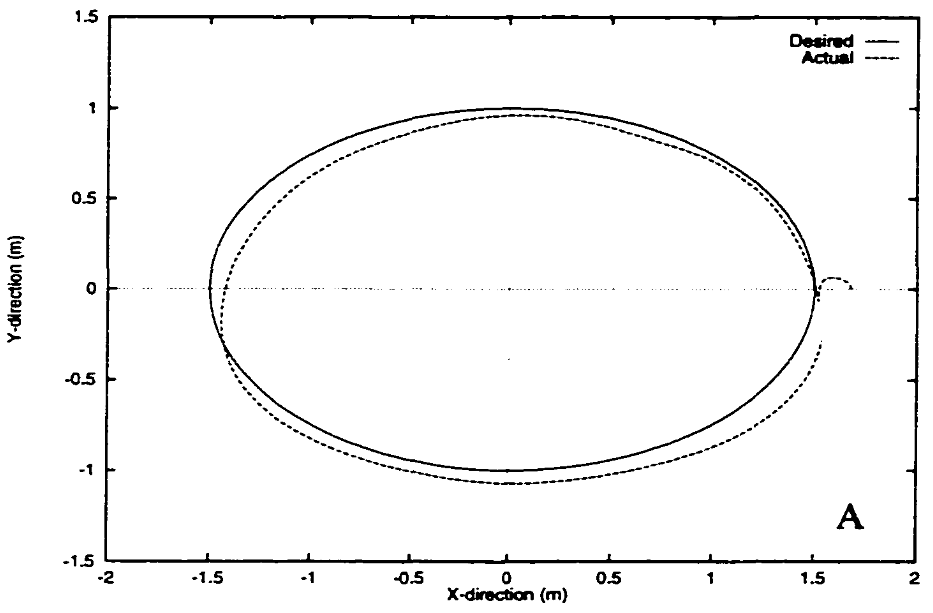


Figure 6.4: *The dynamics of the applied torque in the case of Figure 6.3. The total applied torque, the torque from PD loop, and the torque from the neural network are plotted separately.* A: *the applied torque to Joint 1;* B: *the applied torque to Joint 2.*



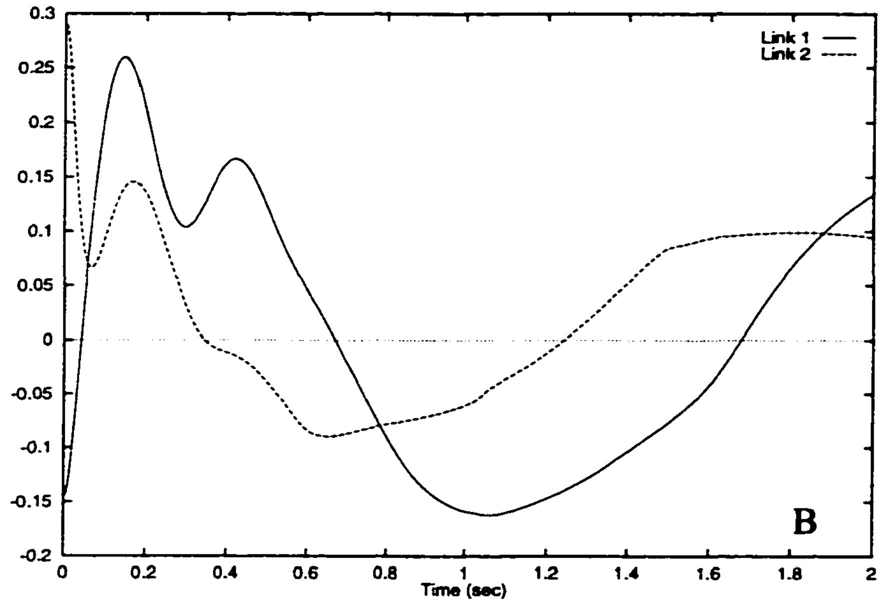


Figure 6.5: *Motion control of a two-link robot manipulator to track an elliptic trajectory using the PD control loop alone.* A: *the real-time tracking performance;* B: *the error dynamics of the joint position.*

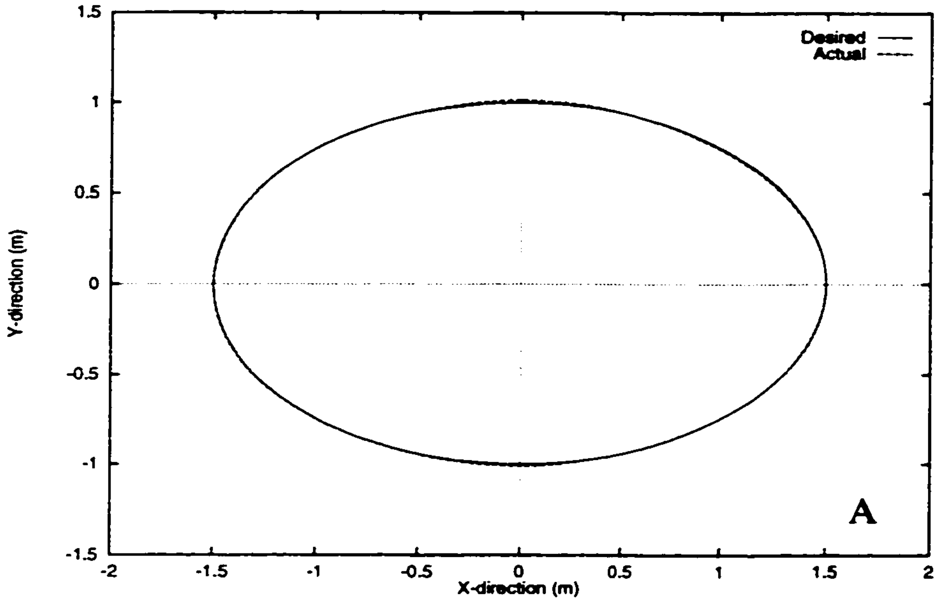
performs better and better as time goes on when there is no disturbance or changes in the robot dynamics. Figure 6.6A shows the dynamic robot tracking performance when the robot to repeatedly track the same desired elliptic trajectory at the second round. It shows that the actual and desired trajectories almost overlap each other. The error dynamics of the joint position is shown in Figure 6.6B. It shows that the position error still decreases toward zero. Note that there exist unmodeled disturbances, the Coulomb friction term in Equation (6.15). The controller performs better at the second round, even with unmodeled disturbances.

6.3.2 Motion Control of a Robot Manipulator to Compensate Sudden Changes in the Robot Dynamics

The proposed controller is capable of compensating any sudden changes in the robot dynamics. Such a property results from the fact that it inherits advantages from both the neural networks based controllers and the PD type controllers: when the sudden changes in the robot dynamics happens, the PD loop provides the dominant control torque to guarantee the system stability; the learning ability of the neural network enable it quickly compensate the sudden changes. Two simulations are carried out: the robot suddenly picks up a load and drops off a load.

In the first case, when the robot just passes a quarter of the whole elliptic trajectory at the second round, it suddenly picks up a load with point mass $m = 1kg$, equivalently, the link mass of Link 2 suddenly increases from $m_2 = 1kg$ to $m_2 = 2kg$. The dynamic robot tracking performance and the error dynamics of the joint position are shown in Figures 6.7A and 6.7B, respectively. It shows that there is a large tracking error after the presence of the sudden change in the robot dynamics. However, the tracking error sharply drops back toward zero, and the robot still perform very well in trajectory tracking.

In the second case, the robot drops off a load with point mass of $0.5kg$ when passing a quarter of the whole elliptic trajectory at the second round, that is, the link mass suddenly changes from $m_2 = 1kg$ to $m_2 = 0.5kg$. The dynamic robot tracking



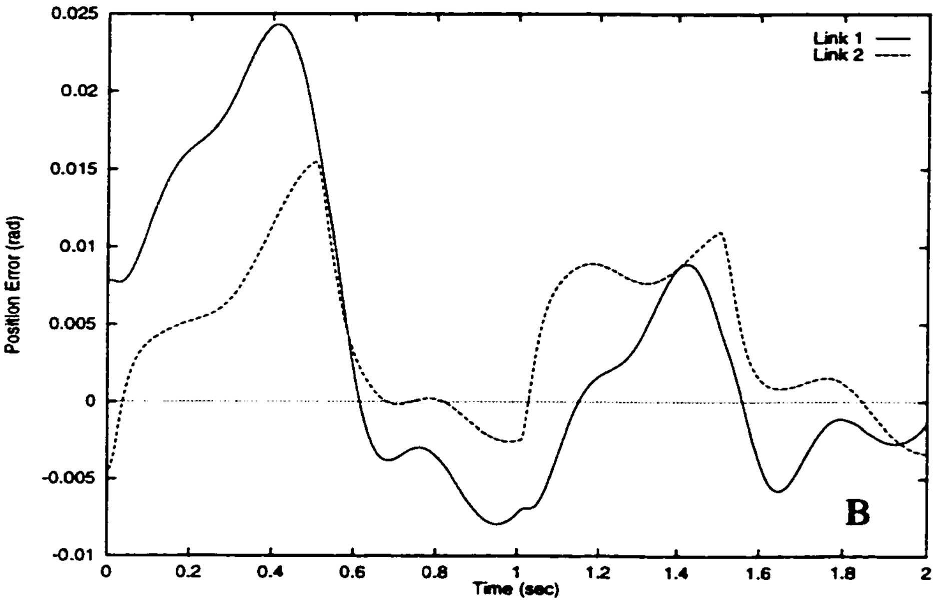
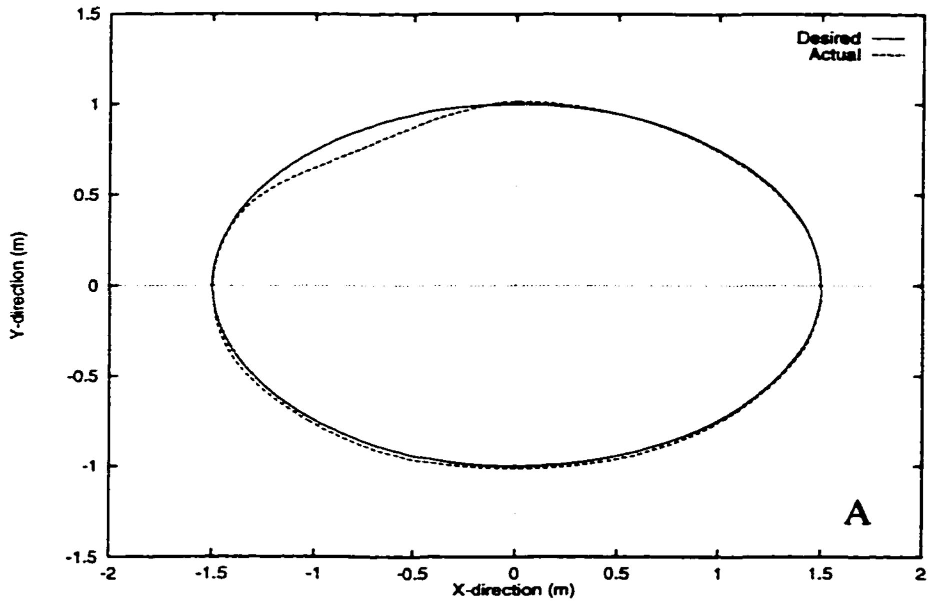


Figure 6.6: *Motion control of a two-link robot manipulator to track an elliptic trajectory using the proposed controller at the second round.* A: *the real-time tracking performance;* B: *the error dynamics of the joint position.*



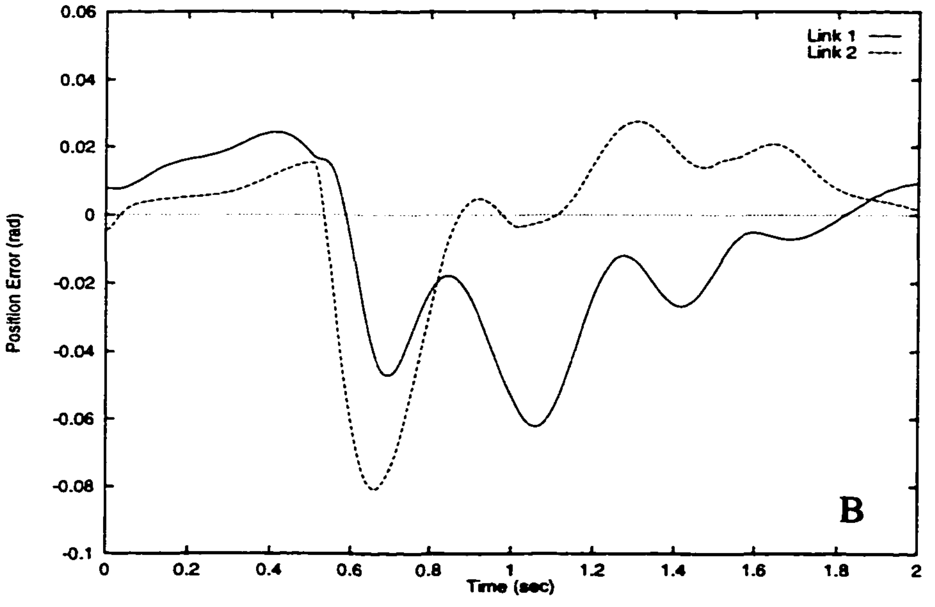


Figure 6.7: *Motion control of trajectory tracking when the robot suddenly picks up a load.* A: *the real-time trajectory tracking performance;* B: *the error dynamics of the joint position.*

performance and the error dynamics of the joint position are shown in Figures 6.8A and 6.8B, respectively. It shows that the robot can quickly compensate such a sudden change in the robot dynamics as well. The applied torque to Joint 2 is shown in Figure 6.8C, where the total applied torque, the torque from the neural network and the torque from the PD loop are plotted separately. It shows that initially the dominant contribution of the total applied torque to Joint 2 is the torque from the neural network, because this is at the second round. when the change in the robot dynamics occurs, the torque from the PD loop has a sharp change. However, the quick learning of the neural network results in that the contribution from the PD loop quickly drops toward zero, and the torque from the neural network becomes dominant again.

6.4 An Alternative Control Algorithm

The basic idea of the proposed controller is to take advantages from both the PD control and the neural network based control. By incorporating the robot regressor dynamics formulation, the neural network assumes a single-layer architecture. The very simple learning algorithm is derived from the Lyapunov stability analysis, thus the global stability of the control system and the convergence of the tracking error are guaranteed. In addition to the learning algorithm of the neural network in presented Section 6.2, an alternative control algorithm is presented in this section based on the same philosophy, which does not require the measurement of the acceleration $\ddot{q}$ in its computation.

Alternatively, the control torque from the neural network τ_{NN} is given by

$$\tau_{NN} = Y(q, \dot{q}, \ddot{q}_r)W, \tag{6.16}$$

where q_r is defined as

$$\dot{q}_r = \dot{q}_d + Ke, \tag{6.17}$$

$$\ddot{q}_r = \ddot{q}_d + K\dot{e}, \tag{6.18}$$

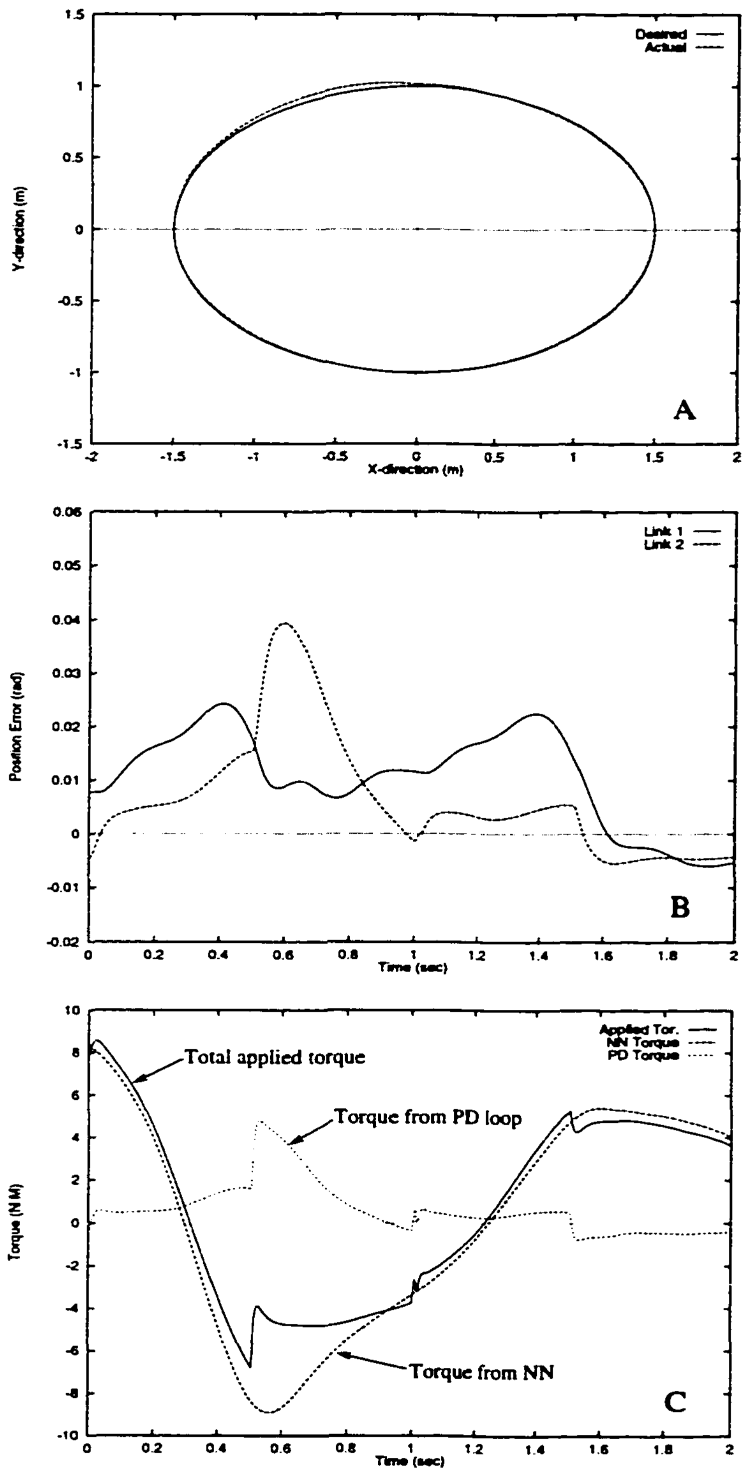


Figure 6.8: *Motion control of trajectory tracking when the robot suddenly drops off a load.* A: *the real-time trajectory tracking performance;* B: *the error dynamics of the joint position;* C: *the dynamics of the applied torque to Joint 2.*

where $K \equiv K_d^{-1}K_p$. The control torque from the PD loop described in Equation (6.4) can be rewritten as

$$\tau_{PD} = K_d(\dot{e} + K_d^{-1}K_p e) = K_d(\dot{e} + Ke) = -K_d s, \tag{6.19}$$

with the definition

$$s = -\dot{e} - Ke = -(\dot{q}_d - \dot{q}) - (\dot{q}_r - \dot{q}_d) = \dot{q} - \dot{q}_r. \tag{6.20}$$

Thus the control torque is now given by

$$\tau = \tau_{NN} + \tau_{PD} = Y(q, \dot{q}, \ddot{q}_r)W - K_d s. \tag{6.21}$$

To prove the system stability and convergence of the tracking error, and to derive the learning law of the neural network, the Lyapunov function candidate of the control system is chosen as

$$v(t) = \frac{1}{2}(s^T M s + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}), \tag{6.22}$$

where Γ is a $r \times r$ positive constant matrix. Thus have $v(t) \geq 0$, and $v(t) = 0$ if and only if $s = 0$ and $\tilde{\theta} = 0$. By Differentiation, the derivative of $v(t)$ along the trajectory of the system is given as

$$\begin{aligned} \dot{v}(t) &= s^T M \dot{s} + \frac{1}{2} s^T \dot{M} s + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \\ &= s^T (M\ddot{q} - M\ddot{q}_r) + \frac{1}{2} s^T \dot{M} s + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \\ &= s^T (M\ddot{q} - M\ddot{q}_r + Cs) + \frac{1}{2} s^T (\dot{M} - 2C) s + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \\ &= s^T (M\ddot{q} - M\ddot{q}_r + Cs) + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}}, \end{aligned} \tag{6.23}$$

where the property that $\dot{M} - 2C$ is skew symmetric (Meng, 1992; Lu and Meng, 1993; Meng, 1996) has been used. From Equations (6.1), (6.2) and (6.21), the closed-loop robot dynamics is given as

$$\begin{aligned} M\ddot{q} + C\dot{q} + G &= Y(q, \dot{q}, \ddot{q}_r)W - K_d s \\ &= \hat{M}\ddot{q}_r + \hat{C}\dot{q} + \hat{G} - K_d s, \end{aligned} \tag{6.24}$$

i.e.,

$$
\begin{aligned}
M\ddot{q} - M\ddot{q}_r &= \tilde{M}\ddot{q}_r + \tilde{C}\dot{q} + \tilde{G} - K_d s \\
&= Y(q, \dot{q}, \ddot{q}_r)\tilde{\theta} - K_d s, \qquad (6.25)
\end{aligned}
$$

where $(\tilde{*}) = (\hat{*}) - (*)$, $(\hat{*})$ is the estimated value of $(*)$. Thus Equation (6.23) becomes

$$
\begin{aligned}
\dot{v}(t) &= s^T\left(Y(q, \dot{q}, \ddot{q}_r)\tilde{\theta} - K_d s + Cs\right) + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}} \\
&= -s^T(K_d - C)s + s^T Y(q, \dot{q}, \ddot{q}_r)\tilde{\theta} + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}} \\
&= -s^T(K_d - \bar{C})s + \tilde{\theta}^T\left(Y^T(q, \dot{q}, \ddot{q}_r)s + \Gamma^{-1}\dot{\tilde{\theta}}\right), \qquad (6.26)
\end{aligned}
$$

where $\bar{C} = \frac{1}{2}(C + C^T)$ is a symmetric matrix. By choosing the following equation

$$
Y^T(q, \dot{q}, \ddot{q}_r)s + \Gamma^{-1}\dot{\tilde{\theta}} = 0, \qquad (6.27)
$$

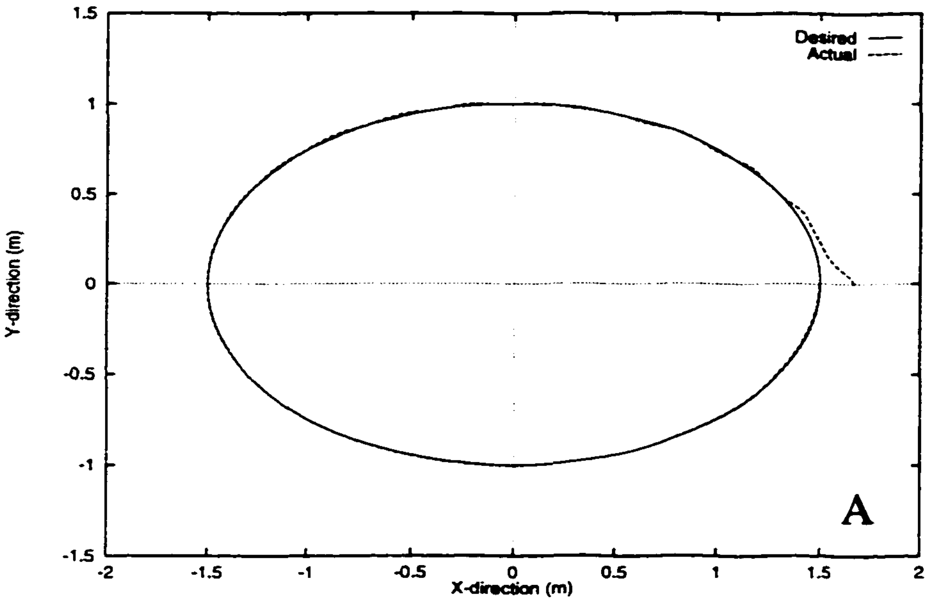
then the derivative of $v(t)$ is given as

$$
\dot{v}(t) = -s^T(K_d - \bar{C})s. \qquad (6.28)
$$

Because Equation (6.28) in conjunction with the same argument as adopted in Meng (1992) and Lu and Meng (1993), it can be proven that both position and velocity errors converge to zero (see Meng, 1992; Lu and Meng, 1993; Meng, 1996). From Equation (6.27), the learning algorithm for the neural network is obtained as

$$
\dot{W} = -\Gamma Y^T(q, \dot{q}, \ddot{q}_r)s. \qquad (6.29)
$$

In summary, Equations (6.5), (6.4) and (6.29) define the alternative control algorithm. This control algorithm is applied to the same tracking control case described in Section 6.3. By choosing $K_p = 100I_2, K_d = 20I_2$, and $\Gamma = 10I_5$, the dynamic robot tracking performance and the error dynamics of the joint position are shown in Figure 6.9A and Figure 6.9B, respectively. It shows that the proposed controller with an alternative input and learning law of neural network is capable of achieving a similar level of real-time fine motion control of robot manipulators.



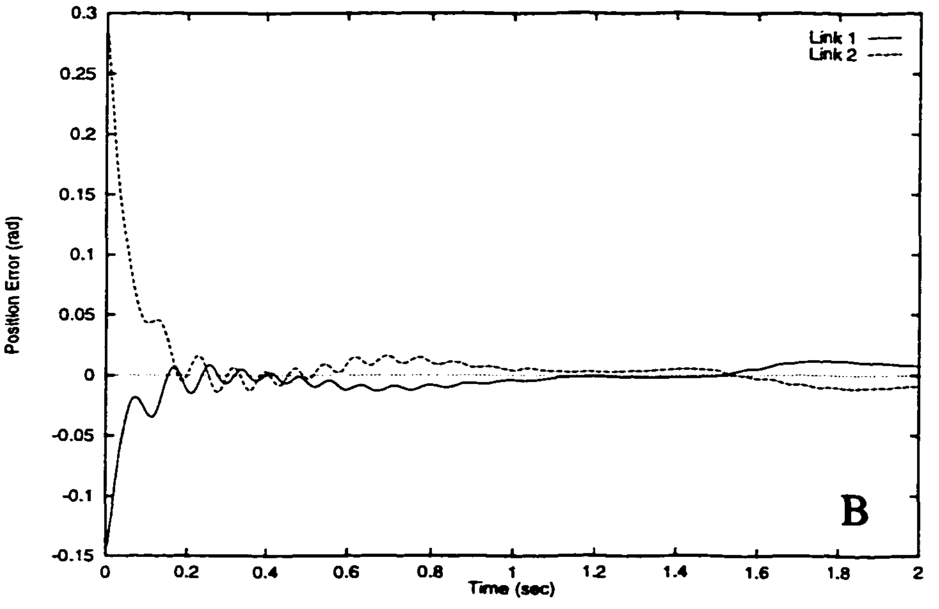


Figure 6.9: *Motion control of a two-link robot manipulator to track an elliptic trajectory using an alternative control algorithm.* A: *the real-time tracking performance;* B: *the error dynamics of the joint position.*

6.5 Conclusion

In this chapter, an efficient neural network based approach is proposed for real-time fine motion control of multi-joint robot manipulators with completely unknown robot dynamics and subject to significant uncertainties. Several feature properties of the proposed controller are worth pointing out:

- The control algorithm is computationally efficient. The neural network has signal-layer structure by taking advantage of the robot regress dynamics formulation. The learning algorithm derived from the Lyapunov stability analysis is much simpler than most commonly used neural network learning algorithms.
- No off-line training is needed. Unlike most neural network based controllers that required off-line training procedures before the controllers can perform properly, the proposed controller achieves the real-time fine motion control through the on-line learning only.
- No knowledge of the robot dynamics is needed. The proposed control can achieve real-time fine motion control with completely unknown robot dynamics parameters.
- The control system is proved to be globally, asymptotically stable and the tracking error is guaranteed to converge to zero.
- The proposed controller is capable of quickly compensating sudden changes in the robot dynamics. This property results from the fact that the proposed controller inherits the advantages from both the PD type controllers and the neural networks based control algorithms.

Chapter 7

Conclusion and Future Work

In this chapter, the results obtained in this thesis will be summarized in the conclusion section. Then the future work related to our study will be outlined.

7.1 Conclusion

In this thesis, we have investigated several aspects of motion planning and motion control of robotic systems. The results obtained in Chapters 2, 3, 4, 5, and 6 can be summarized as follows.

- A biologically inspired neural network approach is developed for real-time motion planning with obstacle avoidance of a point mobile robot in a nonstationary environment. The state space of the neural network is the Cartesian workspace of the robot. Each neuron in the topologically organized neural network is characterized by a shunting equation or an additive equation. The stability and convergence of the neural network system is guaranteed by both qualitative analysis and the Lyapunov stability theory. The proposed approach is not sensitive to model parameter variations nor sensor noise. The real-time optimal robot motion is planned through the dynamic activity landscape of the neural network without explicitly optimizing any cost functions, without explicitly searching over the free workspace or the collision paths, without any prior knowledge of the dynamic environment, and without any learning procedures. Therefore, the model algorithms are computationally efficient. There are only local connections among neurons. The computational complexity of the proposed approach linearly depends on the neural network size.

- A novel neural network approach is developed for real-time robot motion planning with safety consideration in a nonstationary environment. By employing both excitatory and inhibitory lateral connections among neurons and the threshold of neural connections, the proposed model is capable of planning a real-time "comfortable" motion path of a point mobile robot or robot manipulator, not suffering from either the "too far" nor the "too close" problems. The strength of the clearance from obstacle is adjustable. By choosing suitable model parameters, the proposed model is capable of generating either the shortest path, a comfortable path, or the safest path of a point mobile robot or robot manipulator.

- A novel neural network approach is developed for real-time motion planning with obstacle avoidance of 3-d.o.f. robots in a nonstationary 2D workspace. By extending the state space of the neural network to the configuration space of 3-d.o.f. robots, the proposed model is capable of planning real-time collision-free robot motion without any local collision checking procedures. In addition, some complicated robot motion planning problems are studied, such as real-time motion planning with sudden environmental changes, motion planning of a robot with multiple moving targets, and motion planning of multiple robots in the same workspace.

- a novel neural network approach is developed for real-time collision-free motion planning of nonholonomic car-like robots in a nonstationary environment. Based the neural network model for motion planning of 3-d.o.f. robots in 2D workspace, the proposed model employs directionally-selective lateral connections in the neural network. The neural activity propagation is subject to the kinematic constraint of the nonholonomic car-like robot. The real-time optimal robot motion is planned through the dynamic neural activity landscape, with the robot nonholonomic constraint, without any local collision checking procedures at each step of the robot movement. Therefore the model algorithm is computationally efficient. To the best of our knowledge, it is the first time that the real-time motion planning of car-like robots are studied using a non-learning based neural network approach.

- An efficient neural network based approach is proposed for real-time fine motion control of robot manipulators with completely unknown robot dynamics and subject to significant uncertainties. No off-line training procedure is needed in the proposed controller. The real-time fine motion control of robot manipulators is achieved through only on-line learning of the neural network. In addition, the proposed controller is capable of quickly compensating sudden changes in the robot dynamics. By taking advantage of the robot regressor dynamics, the neural network assumes a single-layer structure, and the learning algorithm is computationally efficient. The proposed controller inherits advantages from both the PD type controllers and the neural networks based controllers. The global asymptotic stability of the control system and the convergence of the tracking error is proved by a Lyapunov stability theory.

7.2 Future Work

The work presented in this thesis can only be considered preliminary, since many challenging and possibly more important problems have not been touched in this thesis. In this section, we would like to propose a number of problems as possible future work related to our study.

- In this thesis, we limit our scope to robot motion planning and robot motion control only and do not consider the multi-sensor fusion problems. Therefore, the future work shall study the multi-sensor fusion using biologically inspired approaches, which provides the necessary environmental and internal information to the robotic systems.

- As an continuation of our work in real-time position control of robot manipulators, it is quite natural to seek force control or hybrid position/force control strategies using neural network based approaches.

- As an continuation of our work in real-time motion planning of car-like robots, the future work shall seek motion control strategies of car-like robots using biologically inspired approaches.

- In this thesis, we limit our scope to theoretical aspects of motion planning and motion control of robotic systems. Future work shall study the practical aspects of the proposed neural network based approaches.

Bibliography

Ahmed, R. S., K. S. Rattan and O. H. Abdallah (1995). Adaptive neural network network for identification and tracking control of a robotic manipulator. In: *Proc. of IEEE Intl. Conf. on Systems, Man and Cybernetics*. Vancouver, Canada. pp. 601–609.

Al-Sultan, K. S. and D. S. Aliyu (1996). A new potential field-based algorithm for path planning. *J. of Intelligent and Robotic Systems* **17**(3), 265–282.

Alexander, J. C., J. H. Maddocks and B. A. Michalowski (1998). Shortest distance paths for wheeled mobile robots. *IEEE Trans. Robotics and Automation* **14**(5), 657-662.

Araujo, A. F. R. and M. Vieira (1998). Associative memory used for trajectory generation and inverse kinematics problem. In: *Proc. of IEEE Intl. Joint Conf. on Neural Networks*. Anchorage, USA. pp. 2057–2062.

Azarm, K. and G. Schmidt (1997). Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Albuquerque, USA. pp. 3526–3533.

Baginski, B. and M. Eldracher (1994). Path planning with neural subgoal search. In: *Proc. of IEEE Intl. Conf. on Neural Networks*. Orlando, USA. pp. 2732–2736.

Bander, J. L. and C. C. White (1998). Primal and dual neural networks for shortest path routing. *IEEE Trans. on Systems, Man, and Cybernetics, Part A* **28**(1), 131–134.

Barraquand, J. and J.-C. Latombe (1991). Robot motion planning: a distributed representation approach. *Intl. J. Robotics Research* **10**(6), 628–649.

Behera, L., M. Gopal and S. Chaudhury (1994). Trajectory tracking of robot manipulator using gaussian networks. *Robotics and Autonomous Systems* **13**(2), 107–115.

Behera, L., S. Chaudhury and M. Gopal (1996). Neuro-adaptive hybrid controller for robot manipulator tracking control. *IEE Proc.: Control Theory and Applications* **143**(3), 270–275.

Bemporad, A., A. L. De and G. Oriolo (1996). Local incremental planning for a car-like robot navigating among obstacles. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Minneapolis, USA. pp. 1205–1211.

Beom, H. R. and H. S. Cho (1995). Sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *IEEE Trans. on Systems, Man, and Cybernetics* **25**(3), 464–477.

Bicchi, A., G. Casalino and C. Santilli (1995). Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Nagoya, Japan. pp. 1349–1354.

Bicchi, A., G. Casalino and C. Santilli (1998). Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. *J. of Intelligent and Robotic Systems* **16**(4), 387–405.

Bourbakis, N. G., D. Goldman, R. Fematt, I. Vlachavas and L. H. Tsoukalas (1997). Path planning in a 2-d known space using neural networks and skeletonization. In: *Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics.* Orlando, USA. pp. 2001–2005.

Brooks, R. A. and T. Lozano-Pérez (1985). A subdivision algorithm in configuration space for findpath with rotation. *IEEE Trans. Systems, Man, and Cybernetics* **15**(2), 224–233.

Bullock, D. and S. Grossberg (1988*a*). Neural dynamics of planned arm movements: emergent invariant and speed-accuracy properties during trajectory formation. *Psychological Review* **95**, 49–90.

Bullock, D. and S. Grossberg (1988*b*). *The VITE Model: A Neural Command Circuit for Generating Arm and Articulatory Trajectories*. pp. 305–326. World Scientific Publishers. Singapore.

Bullock, D. and S. Grossberg (1989). *VITE and FLETE: neural modules for trajectory formation and postural control*. pp. 253–298. North-Holland-Elsevier. Amsterdam.

Castellano, G., E. Stella, G. Attolico and A. Distante (1995). Optimal path planning for robot navigation by the hopfield net. In: *Proc. of SPIE Conf. on Intelligent Robots and Computer Vision*. Philadelphia, USA. pp. 566–575.

Chakravarthy, A. and D. Ghose (1998). Obstacle avoidance in a dynamic environment: a collision cone approach. *IEEE Trans. on Systems, Man, and Cybernetics, Part A* **28**(5), 562–574.

Chan, R. H., P. K. Tam and D. N. Leung (1993). Neural network approach for solving the path planning problem. In: *Proc. of IEEE Intl. Symp. on Circuits and Systems*. Chicago, USA. pp. 2454–2457.

Chang, C. C. and K. T. Song (1996). Dynamic motion planning based on real-time obstacle prediction. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Minneapolis, USA. pp. 2402–2407.

Chang, C. C. and K. T. Song (1997). Environment prediction for a mobile robot in a dynamic environment. *IEEE Trans. on Robotics and Automation* **13**(6), 862–872.

Chang, T. Y., S. W. Kuo and J. Y. Hsu (1994). Two-phase navigation system for mobile robots in dynamic environments. In: *Proc. of IEEE/RSJ/GI Intl. Conf. on Intelligent Robots and Systems*. Munich, Germany. pp. 306–313.

Chen, P. C. and Y. K. Hwang (1998). Sandros: A dynamic graph search algorithm for motion planning. *IEEE Trans. on Robotics and Automation* **14**(3), 390–403.

Chen, Y. M. and K. F. Gill (1996). Fuzzy neural controller design for use with a non-linear system. *Proc. of IME, Part I: J. of Systems and Control Engineering* **210**(2), 141–150.

Chiaverini, S., B. Siciliano and L. Villani (1997). An adaptive force/position control scheme for robot manipulators. *Applied Mathematics and Computer Science* **7**(2), 293–303.

Chohra, A. and C. Benmehrez (1998). Neural navigation approach for intelligent autonomous vehicles (iav) in partially structured environments. *Applied Intelligence* **8**, 219–233.

Ciliz, M. and C. Isik (1997). On-line learning control of manipulators based on artificial neural network models. *Robotica* **15**, 293–304.

Craig, J. J. (1988). *Adaptive Control of Mechanical Manipulators.* Addison-Wesley. Massachusetts.

Crowley, J. L. (1985). Navigation for an intelligent mobile robot. *IEEE J. Robotics Automation* **1**(1), 31–41.

de Lamadrid, J. G. and M. L. Gini (1990). Path tracking through uncharted moving obstacles. *IEEE Trans. Systems, Man, and Cybernetics* **20**(6), 1408–1422.

de Leon, J. L. D. and J. H. Sossa (1998). Automatic path planning for a mobile robot among obstacles of arbitrary shape. *IEEE Trans. on Systems, Man, and Cybernetics, Part B* **28**(3), 467–472.

Desaulniers, G. (1996). On shortest paths for a car-like robot maneuvering around obstacles. *Robotics and Autonomous Systems* **17**(3), 139–148.

Desaulniers, G. and F. Soumis (1995). Efficient algorithm to find a shortest path for a car-like robot. *IEEE Trans. on Robotics and Automation* **11**(6), 819–828.

Dixon, W. E., M. S. de Queiroz, F. Zhang and D. M. Dawson (1999). Tracking control of robot manipulators with bounded torque inputs. *Robotica* **17**, 121–129.

Donald, B. R. (1987). A search algorithm for motion planning with six degrees of freedom. *Artificial Intelligence* **31**(3), 295–353.

ElDeeb, Y. and W. H. ElMaraghy (1998). Robust adaptive control of a robotic manipulator including motion dynamics. *J. of Robotic Systems* **15**(11), 661–669.

Ertugrul, M. and O. Kaynak (1997). Neural network adaptive sliding mode control and its application to scara type robot manipulator. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Albuquerque, USA. pp. 2932–2937.

Fang, G. and M. W. M. G. Dissanayake (1998). A neural network-based method for time-optimal trajectory planning. *Robotica* **16**, 143–158.

Ferrari, C., E. Pagello, J. Ota and T. Arai (1996). Framework for robust multiple robots motion planning. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems.* Osaka, Japan. pp. 1684–1690.

Fiorini, P. and Z. Shiller (1998). Motion planning in dynamic environments using velocity obstacles. *Intl. J. Robotics Research* **17**(7), 760–774.

Fraichard, T. and A. Scheuer (1994). Car-like robots and moving obstacles. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* San Diego, USA. pp. 64–69.

Fraichard, T. and R. Mermond (1998). Path planning with uncertainty for car-like robots. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Leuven, Belgium. pp. 27–32.

Fujii, T., Y. Arai, H. Asama and I. Endo (1998). Multilayered reinforcement learning for complicated collision avoidance problems. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Leuven, Belgium. pp. 2186–2186.

Gambardella, L. M. and C. Versino (1994). Robot motion planning integrating planning strategies and learning methods. In: *Proc. of 2nd Intl. Conf. on AI Planning Systems*. Chicago, USA.

Gambardella, L. M. and M. Maex (1993). Incorporating learning in motion planning techniques. In: *Proc. of Intl. Conf. on Intelligent Robots and Systems*. Yokohama, Japan. pp. 712–715.

Gaudiano, P. and S. Grossberg (1991). Vector associative maps: unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks* **4**, 147–183.

Gaudiano, P., E. Zalama and J. López Coronado (1996). An unsupervised neural network for low-level control of a mobile robot: noise resistance, stability, and hardware implementation. *IEEE Trans. Systems, Man, and Cybernetics, Part B* **26**(3), 485–496.

Glasius, R., A. Komoda and S. C. A. M. Gielen (1994). Population coding in a neural net for trajectory formation. *Network: Computation in Neural Systems* **5**, 549–563.

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.

Glasius, R., A. Komoda and S. C. A. M. Gielen (1996). A biologically inspired neural net for trajectory formation and obstacle avoidance. *Biological Cybernetics* **74**, 511–520.

Goh, C. J., N. J. Edwards and A. Y. Zomaya (1993). Feedback control of minimum-time optimal control problems using neural networks. *Optimal Control Applications and Methods* **14**(1), 1–16.

Gomi, H. and M. Kawato (1993). Neural network control for a closed-loop system using feedback-error-learning. *Neural Networks* **6**, 933–946.

Grolinger, K., B. Jerbic and B. Vranjes (1997). Autonomous robot behavior based on neural networks. In: *Proc. SPIE Intl. Conf. on Applications and Science of Artificial Neural Networks.* Orlando, USA. pp. 459–470.

Grossberg, S. (1973). Contour enhancement, short term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics* **52**, 217–257.

Grossberg, S. (1982). *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.* Reidel Press. Boston.

Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. Systems, Man, and Cybernetics* **13**(5), 815–926.

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

Haddad, H., M. Khatib, S. Lacroix and R. Chatila (1998). Reactive navigation in outdoor environments using potential fields. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Leuven, Belgium. pp. 1232–1237.

Hamavand, Z. and H. M. Schwartz (1995). Trajectory control of robotic manipulators by using a feedback-error-learning neural network. *Robotica* **13**(5), 449–459.

Hodgkin, A. L. (1964). *The Conduction of the Nervous Impulse.* Liverpool University. Liverpool.

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.

Hong, S. G., S. W. Kim, K. B. Park and J. J. Lee (1996). Local motion planner for nonholonomic mobile robots in the presence of the unknown obstacles. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Minneapolis, USA. pp. 1212–1217.

Hyun, W. K. and I. H. Suh (1995). Hierarchical collision-free path planning algorithm for robotics. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. Pittsburgh, USA. pp. 488–495.

Ilari, J. and C. Torras (1990). 2d path planning: a configuration space heuristic approach. *Intl. J. Robotics Research* **9**(1), 75–91.

Jagannathan, S., S. Q. Zhu and F. L. Lewis (1994). Path planning and control of a mobile base with nonholonomic constraints. *Robotica* **12**(6), 529–539.

Jarvis, R. A. and J. C. Byrne (1986). Robot navigation: touching, seeing and knowing. In: *Proc. Australian Conf. on Artificial Intelligence*. Melbourne, Australia.

Jeong, K. S., Y. S. Hong and C. K. Park (1997). Neuro-fuzzy control of a robot manipulator for a trajectory design. In: *Proc. of 6th IEEE Intl. Workshop on Robot and Human Communication*. Sendai, Japan. pp. 260–265.

Jerez, C., Y. Hashimoto, T. Matsuda and T. Tsuchiya (1998). Environment representation using enclosed obstacles and minimum-turns path planning. *Advanced Robotics* **12**(4), 351–371.

Jiang, K., L. D. Seneviratne and S. W. E. Earles (1996). Motion planning with reversal manoeuvres for a non-holonomic constrained robot. *Proc. of the Institution of Mechanical Engineers, Part B* **210**(B5), 487–497.

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.

Jung, S. and T. C. Hsia (1996*a*). Neural network reference compensation technique for position control of robot manipulators. In: *Proc. of IEEE Intl. Conf. on Neural Networks*. Washington, USA. pp. 1765–1770.

Jung, S. and T. C. Hsia (1996*b*). Study of neural network control of robot manipulators. *Robotica* **14**(1), 7–15.

Jung, S. and T. C. Hsia (1997). On an effective design approach of cartesian space neural network control for robot manipulators. *Robotica* **15**(3), 305–312.

Jung, S. and T. C. Hsia (1998). Analysis of nonlinear neural network impedance force control for robot manipulators. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Leuven, Belgium. pp. 1731–1736.

Kanaya, M. and M. Tanaka (1998). Path planning method for multi-robots using a cellular neural network. *Electronics and Communications in Japan, Part III* **81**(3), 10–21.

Kant, K. and S. W. Zucker (1986). Towards efficient trajectory planning: the path-velocity decomposition. *Int. J. Robotics Research* **5**, 72–89.

Kassim, A. A. and B. V. Kumar (1997). Path planning for autonomous robots using neural networks. *J. of Intelligent Systems* **7**(1-2), 33–55.

Kassim, A. A. and B. V. Kumar (1999). Path planners based on the wave expansion neural network. *Robotics and Autonomous Systems* **26**, 1–22.

Katevas, N. I., S. G. Tzafestas and C. G. Pnevmatikatos (1998). The approximate cell decomposition with local node refinement global path planning method: Path nodes refinement and curve parametric interpolation. *J. of Intelligent and Robotic Systems* **22**(3-4), 289–314.

Kawakami, T. and Y. Kakazu (1996). Reactive motion planning of robot manipulators by cs-based reinforcement learning. In: *Proc. of Japan-USA Symp. on Flexible Automation.* Boston, USA. pp. 37–40.

Kguchi, K. and T. Fukuda (1995). Fuzzy neural controller for robot manipulator force control. In: *Proc. of IEEE Intl. Conf. on Fuzzy Systems.* Yokohama, Japan. pp. 869–874.

Khatib, M., H. Jaouni, R. Chatila and J. Pl Laumond (1997). Dynamic path modification for car-like nonholonomic mobile robots. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Albuquerque, USA. pp. 2920–2925.

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

Khemaissia, S. and A. S. Morris (1993). Neuro-adaptive control of robotic manipulators. *Robotica* **11**, 465–473.

Kim, S. H., Y. H. Kim, K. B. Sim and H. T. Jeon (1993). On developing an adaptive neural-fuzzy control system. In: *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems*. Yokohama, Japan. pp. 950–957.

Kim, S. W. and J. J. Lee (1996). Filter-error-learning neural networks for stable trajectory tracking control of robot manipulators. *Mechatronics* **6**(2), 181–192.

Kimmel, R., N. Kiryati and A. M. Bruckstein (n.d.). Multivalued distance maps for motion planning on surfaces with moving obstacles. *IEEE Trans. Robotics and Automation* **14**(3), 427–436.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **73**, 49–60.

Kreczmer, B. (1998). Path planning system for car-like robot. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Leuven, Belgium. pp. 40–45.

Kruse, E., R. Gutsche and F. Wahl (1999). Acquisition of obstacle motion patterns to improve mobile robot motion planning. *Advanced Robotics* **12**(5), 565–578.

Kyriakopoulos, K. J. and G. N. Saridis (1995). Optimal and suboptimal motion planning for collision avoidance of mobile robots in non-stationary environments. *J. of Intelligent and Robotic Systems* **11**(3), 223–267.

Lambert, A. and P. N. Le-Fort (1998). Safe path planning for mobile robots. In: *Proc. of ASCE Specialty Conf. on Robotics for Challenging Environments*. Albuquerque, USA. pp. 50–56.

Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publisher. Boston.

Laumond, J. P., C. Nissoux and M. Vendittelli (1998). Obstacle distances and visibility for car-like robots moving forward. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Leuven, Belgium. pp. 33–38.

Laumond, J. P., P. E. Jacobs, M. Taix and R. M. Murray (1994). Motion planner for nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation* **10**(5), 577–593.

LaValle, S. M. and R. Sharma (1997). On motion planning in changing, partially predictable environments. *Intl. J. Robotics Research* **16**(6), 775–807.

LaValle, S. M. and S. A. Hutchinson (1998). Optimal motion planning for multiple robots having independent goals. *IEEE Trans. on Robotics and Automation* **14**(6), 912–925.

Lewis, F. L., A. Yesildirek and K. Liu (1996). Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Trans. on Neural Networks* **7**(2), 388–399.

Lewis, F. L., C. T. Abdallah and D. M. Dawson (1990). *Control of Robot Manipulators*. Macmillan Publishing. New York.

Lewis, F. L., K. Liu and A. Yesildirek (1995). Neural net robot controller with guaranteed tracking performance. *IEEE Trans. on Neural Networks* **6**(3), 703–715.

Li, L. and H. Öğmen (1994). Visually guided motor control: adaptive sensorimotor mapping with on-line visual-error correction. In: *Proc. of the World Congress on Neural Networks*. pp. 127–134.

Li, Q., S. K. Tso and W. J. Zhang (1998). Trajectory tracking control of robot manipulators using a neural-network-based torque compensator. *Proc. of IME, Part I: J. of Systems and Control Engineering* **212**(5), 361–372.

Li, W. (1995). Hybrid neuro-fuzzy system for sensor based robot navigation in unknown environments. In: *Proc. of American Control Conf.*. Seattle, USA. pp. 2749–2753.

Li, Z. X. and T. D. Bui (1998). Robot path planning using fluid model. *J. of Intelligent and Robotic Systems* **21**, 29–50.

Linsker, R. (1986). From basic network principles to neural architecture: emergence of spatial-opponents cells. *Proc. Natl. Acad. Sci. USA* **83**, 7508–7512.

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

Lu, W. S. and Q. H. Meng (1993). Regressor formulation of robot dynamics: computation and applications. *IEEE Trans. on Robotics and Automation* **9**, 323–333.

Lumelsky, V. J. and A. A. Stepanov (1986). Dynamic path planning for a mobile automation with limited information on the environment. *IEEE Trans. Automatic Control* **31**, 1058–1063.

Lumelsky, V. J. and K. R. Harinarayan (1997). Decentralized motion planning for multiple mobile robots: the cocktail party model. *Robotics and Autonomous Systems* **4**, 121–135.

Maguire, G. and X. Yang (1996). Simulation of an inner plexiform layer neural circuit in vertebrate retina leads to sustained and transient excitation. In: *Proc. of European Symposium on Artificial Neural Networks.* Brussels, Belgium.

Marcus, C. M., F. R. Waugh and R. M. Westervelt (1990). Associative memory in an analog iterated-map neural network. *Physical Review A* **41**(6), 3355–3364. this paper say that continuous signals can avoid possible oscillations related to discrete parallel network.

Marti, K. and S. Qu (1998). Path planning for robots by stochastic optimization methods. *J. of Intelligent and Robotic Systems* **22**, 117–127.

Matric, M. J. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots* **4**(1), 73–83.

Meddah, D. Y. and A. Benallegue (1997). A stable neuro-adaptive controller for rigid robot manipulators. *J. of Intelligent and Robotic Systems* **20**, 181–193.

Meeran, S. and A. Shafie (1997). Optimum path planning using convex hull and local search heuristic algorithms. *Mechatronics* **7**(8), 737–756.

Meng, M. and X. Yang (1998). A neural network approach to real-time trajectory generation. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Leuven, Belgium. pp. 1725–1730.

Meng, M. and X. Yang (1999). An efficient neural network based approach to real-time fine control of robot manipulators. *IEEE Trans. on Robotics and Automation*. Submitted.

Meng, Q.-H. M. (1992). Model-Based Adaptive Position and Force Control of Robot Manipulators. PhD thesis. University of Victoria.

Meng, Q.-H. M. (1995). Comparison study of model-based and non-model-based robot controllers. In: *Proc. of IEEE Intl. Conf. on Systems, Man and Cybernetics*. Vancouver, Canada. pp. 61–66.

Meng, Q.-H. M. (1996). Application of an efficient neural network to identification and adaptive control of unknown robot dynamics. *Intl. J. of Intelligent Control and Systems* **1**(4), 459–468.

Meng, Q.-H. M. and W.-S. Lu (1994). A unified approach to stable adaptive force/position control of robot manipulators. In: *Proceedings of the American Control Conf.*. Baltimore, USA. pp. 200–201.

Millan, J. and C. Torras (1994). Efficient reinforcement learning of navigation strategies in an autonomous robot. In: *Proc. of IEEE/RSJ/GI Intl. Conf. on Intelligent Robots and Systems*. Munich, Germany. pp. 15–22.

Miller, W. T., R. S. Sutton and P. J. Werbos (1990). *Neural Networks for Control*. MIT Press. Cambridge, USA.

Morris, A. S. and S. Khemaissia (1996). A neural network based adaptive robot controller. *J. of Intelligent and Robotic Systems* **15**, 3–10.

Moutarlier, P., B. Mirtich and J. Canny (1996). Shortest paths for a robot to manifolds in configuration space. *Intl. J. Robotics Research* **15**(1), 36–60.

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.

Muraca, P., G. Raicon and T. Varone (1996). Cooperative neural field for the path planning of a robot arm. *J. of Intelligent and Robotic Systems* **15**(1), 11–18.

Nagrath, I. J., L. Behera, K. M. Krishna and K. D. Rajasekar (1997). Real-time navigation of a mobile robot using kohonen's topology conserving neural network. In: *Proc. of the 8th Intl. Conf. on Advanced Robotics*. Monterey, USA. pp. 459–464.

Nam, B. H., S. J. Lee and S. W. Lee (1997). Neural network for the trajectory control of robotic manipulators with uncertainties. In: *Proc. of IEEE Intl. Conf. on Neural Networks*. Houston, USA. pp. 1777–1780.

Namgung, I. and J. Duffy (1997). Two dimensional collision-free path planning using linear parametric curve. *J. of Robotic Systems* **14**(9), 659–673.

Naruse, K. and M. C. Leu (1996). Nonholonomic vehicle motion planning by generating a mapping from configuration to control output with reinforcement learning. In: *Proc. of Japan-USA Symp. on Flexible Automation*. Boston, USA. pp. 609–614.

Nearchou, A. C. (1998). Path planning of a mobile robot using genetic heuristics. *Robotica* **16**(5), 575–588.

Noborio, H., T. Naniwa and S. Arimoto (1989). A feasible motion-planning algorithm for a mobile robot on a quadtree representation. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Scottsdale. pp. 327–332.

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

Oriolo, G., G. Ulivi and M. Vendittelli (1997). Fuzzy maps: a new tool for mobile robot perception and planning. *J. Robotic systems* **14**(3), 179–197.

Oriolo, G., G. Ulivi and M. Vendittelli (1998). Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Trans. Systems, Man, and Cybernetics, Part B* **28**(3), 316–333.

Ortega, R. and M. W. Spong (1988). A adaptive control of robot manipulators - a review. In: *Proc. of IEEE Conf. on Robotics and Automation.* pp. 1575–1584.

Öğmen, H. (1991). On the mechanisms underlying directional selectivity. *Neural Computation* **3**, 333–349.

Öğmen, H. (1993). A neural theory of retino-cortical dynamics. *Neural Networks* **6**, 245–273.

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

Öğmen, H. and S. Gagné (1990*b*). Neural network architecture for motion perception and elementary motion detection in the fly visual system. *Neural Networks* **3**, 487–505.

Paromtchik, I. E. and C. Laugier (1996). Motion generation and control for parking an autonomous vehicle. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Minneapolis, USA. pp. 3117–7122.

Payton, D. W. (1990). Internalised plans. *Robotics and Autonomous Systems* **6**(1-2), 89–103.

Pei, S. C. and J. H. Horng (1998). Finding the optimal driving path of a car using the modified constrained distance transformation. *IEEE Trans. Robotics and Automation* **14**(5), 663–670.

Pham, D. T. and S. J. Oh (1994). Adaptive control of a robot using neural networks. *Robotica* **12**, 553–561.

Plonsey, R. and D. G. Fleming (1969). *Bioelectric Phenomena*. McGraw-Hill. New York.

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

Pruski, A. and S. Rohmer (1997). Robust path planning for non-holonomic robots. *J. of Intelligent and Robotic Systems* **18**, 329–350.

Pulakka, K. and V. Kujanpaa (1998). Rough level path planning method for a robot using sofm neural network. *Robotica* **16**(4), 415–423.

Ratering, S. and M. Gini (1995). Robot navigation in a known environment with unknown moving obstacles. *Autonomous Robots* **1**(2), 149–165.

Ritter, H. J., T. M. Martinetz and K. J. Schulten (1989). Topology-conserving maps for learning visuo-motor-coordination. *Neural Networks* **2**, 159–189.

Saab, Y. and M. VanPutte (1999). Shortest path planning on topographical maps. *IEEE Trans. on Systems, Man, and Cybernetics, Part A* **29**(1), 139–150.

Sadegh, N. and R. Horowitz (1990). Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. *Intl. J. Robotics Research* **9**(3), 74–92.

Samuel, S. and S. S. Keerth (1993). Numerical determination of optimal non-holonomic paths in the presence of obstacles. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Atlanta, USA. pp. 826–830.

Sanger, T. D. (1994). Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE Trans. on Robotics and Automation* **10**(3), 323–333.

Scheuer, A. and T. Fraichard (1997). Collision-free and continuous-curvature path planning for car-like robots. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation*. Albuquerque, USA. pp. 867–873.

Schlegel, C. (1998). Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. Victoria, Canada. pp. 594–599.

Schmitt, P. R. and W. J. Book (1996). Reactive path shaping: local path planning for autonomous mobile robots. In: *Proc. of ASME Intl. Mechanical Engineering Congress and Exposition*. Atlanta, USA. pp. 639–646.

Sekhavat, S., P. Svestka, J. P. Laumond and M. H. Overmars (1998). Multilevel path planning for nonholonomic robots using semiholonomic subsystems. *Intl. J. Robotics Research* **17**(8), 840–857.

Seshadri, C. and A. Ghosh (1993). Optimum path planning for robot manipulators amid static and dynamic obstacles. *IEEE Trans. Systems, Man, and Cybernetics* **23**, 576–584.

Simeon, T., S. Leroy and J. P. Laumond (1997). Computing good holonomic collision-free paths to steer nonholonomic mobile robots. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. Grenoble, France. pp. 1004–1009.

Simon, D. (1993). Application of neural networks to optimal robot trajectory planning. *Robotics and Autonomous Systems* **11**(1), 23–34.

Slotine, J. E. and W. Li (1987). On the adaptive control of robot manipulators. *Intl. J. of Robotics Research* **6**, 49–59.

Slotine, J. E. and W. Li (1988). Adaptive manipulator control: a case study. *IEEE Trans. on Automic Control* **33**(11), 995–1003.

Song, K. T. and L. H. Sheen (1995). Fuzzy-neuro control design for obstacle avoidance of a mobile robot. In: *Proc. of IEEE Intl. Conf. on Fuzzy Systems*. Yokohama, Japan. pp. 71–76.

Song, K. T. and W. Y. Sun (1998). Robot control optimization using reinforcement learning. *J. of Intelligent and Robotic Systems* **21**(3), 221–238.

Stentz, A. (1995). Optimal and efficient path planning for unknown and dynamic environments. *Intl. J. of Robotics and Automation* **10**(3), 89–100.

Sun, F. and Z. Sun (1997). Stable discrete-time neural control for robots using adaptive bounding techniques. In: *Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics*. Orlando, USA. pp. 4086–4091.

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

Szczerba, R. J., D. Z. Chen and J. J. Uhran (1998). Planning shortest paths among 2d and 3d weighted regions using framed-subspaces. *Intl. J. Robotics Research* **17**(5), 531–546.

Thorpe, C. E. (1984). FIDO: Vision and Navigation for A Robot Rover. PhD thesis. Carnegie Mellon University.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* **99**(1), 21–71.

Tomiyama, K., T. Furuta and T. Noguchi (1998). Three hybrid control schemes consisting of neural network and adaptive controllers for robot manipulators. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. Victoria, Canada. pp. 1987–1992.

Touzet, C. and J. M. Santos (1998). Reinforcement function design and bias for efficient learning in mobile robots. In: *Proc. of IEEE Intl. Conf. on Fuzzy Systems*. Archorage, USA. pp. 153–158.

Tso, S. K. and N. L. Lin (1995). Adaptive neural network controller for robot manipulator systems. In: *Proc. of IEEE Intl. Conf. on Neural Networks*. Perth, Australia. pp. 2320–2324.

Tsoularis, A. and C. Kambhampati (1998). On-line planning for collision avoidance on the nominal path. *J. of Intelligent and Robotic Systems* **21**(4), 327–371.

Tsoularis, A. and C. Kambhampati (1999). Avoiding moving obstacles by deviation from a mobile robot's nominal path. *Intl. J. Robotics Research* **18**(5), 454–465.

van der Smagt, P. P. and B. J. A. Kröse (1991). A real-time learning neural robot controller. In: *Proc. of the 1991 Intl. Conf. on Artificial Neural Networks.* Espoo, Finland. pp. 351–356.

Vandorpe, J., H. Van Brussel and H. Xu (1996). Lias: a reflexive navigation architecture for an intelligent mobile robot system. *IEEE Trans. on Industrial Electronics* **43**(3), 432–440.

Vasseur, H. A., F. G. Pin and J. R. Taylor (1992). Navigation of car-like mobile robots in obstructed environments using convex polygonal cells. *Robotics and Autonomous Systems* **10**(2-3), 133–146.

Wada, Y. and M. Kawato (1993). A neural network model for arm trajectory formation using forward and inverse dynamics models. *Neural Networks* **6**, 919–932.

Wang, J. (1998). Primal and dual neural networks for shortest path routing. *IEEE Trans. on Systems, Man, and Cybernetics, Part A* **28**(6), 864–869.

Wang, Y. (1996). Nonholonomic motion planning: a polynomial fitting approach. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Minneapolis, USA. pp. 2956–2961.

Wang, Y., J. A. Linnett and J. Roberts (1994). Kinematics, kinematic constraints and path planning for wheeled mobile robots. *Robotica* **12**(5), 391–400.

Watanabe, K., S. H. Jin and S. G. Tzafestas (1995). Learning multiple fuzzy control for robot manipulators. *J. of Artificial Neural Networks* **2**(1-2), 119–136.

Wu, C. J. (1995). Learning fuzzy algorithm for motion planning of mobile robots. *J. of Intelligent and Robotic Systems* **11**(3), 209–221.

Wyard-Scott, L. and Q.-H. M. Meng (1995). A potential maze solving algorithm for a micromouse robot. In: *Proc. of IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing.* Victoria, Canada. pp. 614–618.

Xu, H., F. Sun and Z. Sun (1996). Adaptive sliding mode control based on a fuzzy neural network for manipulators. In: *Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics.* Beijing, China. pp. 1942–1946.

Yamaura, M. and T. Onozuka (1998). Reinforcement learning with knowledge by using a stochastic gradient method on a bayesian network. In: *Proc. of IEEE Intl. Joint Conf. on Neural Networks.* Anchorage, USA. pp. 2045–2050.

Yan, L. L. and C. J. Li (1997). Robot learning control based on recurrent neural network inverse model. *J. of Robotic Systems* **14**(3), 199–212.

Yang, X. (1996). A neural network architecture for visual information processing in vertebrate retina. Master's thesis. University of Houston.

Yang, X. and H. Öğmen (1995). A neural network architecture for control of wing beat in locust. In: *13th Annual Houston Conf. on Biomedical Engineering Research.* Houston, USA.

Yang, X. and M. Meng (1998*a*). Dynamical trajectory generation with collision free using neural networks. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems.* Victoria, Canada. pp. 1634–1639.

Yang, X. and M. Meng (1998*b*). An efficient neural control method for robot manipulators. In: *Proc. of 2nd IEEE Intl. Conf. on Intelligent Processing Systems.* Gold Coast, Australia. pp. 709–713.

Yang, X. and M. Meng (1998*c*). A neural network approach to real-time path planning with safety consideration. In: *Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics.* San Diego, USA. pp. 3412–3417.

Yang, X. and M. Meng (1999*a*). An efficient neural network approach to dynamic robot motion planning. *Neural Networks.* Submitted.

Yang, X. and M. Meng (1999*b*). An efficient neural network method for real-time motion planning with safety consideration. *Robotics and Autonomous Systems.* Accepted.

Yang, X. and M. Meng (1999*c*). An efficient neural network model for path planning of car-like robots in dynamic environment. *Intl. J. of Advanced Computational Intelligence.* Submitted.

Yang, X. and M. Meng (1999*d*). A neural network approach to real-time collision-free navigation of 3-d.o.f. robots in 2d. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Detroit, USA. pp. 23–28.

Yang, X. and M. Meng (1999*e*). Neural network approaches to dynamic collision-free trajectory generation. *IEEE Trans. on Systems, Man, and Cybernetics.* Submitted.

Yang, X. and M. Meng (1999*f*). Real-time motion planning of car-like robots. In: *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robotic Systems.* Kyungju, Korea. Accepted.

Yang, X. and M. Meng (1999*g*). Real-time tracking control of robot manipulators with on-line learning based approach. In: *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering.* Edmonton, Canada. pp. 1374–1379.

Yang, X. and M. Meng (1999*h*). Real-time trajectory generation with clearance from obstacles. In: *Proc. of 14th World Congress of International Federation of Automatic Control.* Beijing, China. Accepted.

Yih, C. C. and P. I. Ro (1998). Nonlinear optimization-based motion planning of nonholonomic systems using an iterative algorithm. *Intelligent Automation and Soft Computing* 4(3), 227–239.

Yung, N. H. C. and C. Ye (1998*a*). An adaptive fuzzy approach to obstacle avoidance. In: *Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics.* San Diego, USA. pp. 3418–3423.

Yung, N. H. C. and C. Ye (1998*b*). Avoidance of moving obstacles through behavior fusion and motion prediction. In: *Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics.* San Diego, USA. pp. 3424–3429.

Zalama, E., P. Gaudiano and J. López Coronado (1995). A real-time, unsupervised neural network for the low-level control of a mobile robot in a nonstationary environment. *Neural Networks* **8**, 103–123.

Zelek, J. S. (1995). Dynamic path planning. In: *Proc. of IEEE Intl. Conf. on Systmes, Man, and Cybernetics.* Vancouver, Canada. pp. 1285–1290.

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

Zhang, H. and B. Zhang (1995). Improved trajectory generation schemes based on resolved motion rate control. *Intl. J. Robotics and Automation* **10**(2), 70–77.

Zhang, M., S. Peng and Q. Meng (1997). Neural network and fuzzy logic techniques based collision avoidance for a mobile robot. *Robotica* **15**(6), 627–632.

Zhu, D. and J. C. Latombe (1991). New heuristic for efficient hierarchical path planning for mobile robot. *IEEE Trans. Robotics and Automation* **7**(1), 9–20.

Zomaya, A. Y. and A. S. Morris (1992). Direct neuro-adaptive control of robot manipulators. In: *Proc. of IEEE Intl. Conf. on Robotics and Automation.* Nice, France. pp. 1902–1907.