Neurocomputing 134 (2014) 230–238

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Mobile robots' modular navigation controller using spiking neural networks

Xiuqing Wang [a], Zeng-Guang Hou [b,*], Feng Lv [a], Min Tan [b], Yongji Wang [c]

[a] Hebei Normal University, Shijiazhuang 050031, PR China
[b] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, The Chinese Academy of Sciences, Beijing 100190, China
[c] Laboratory for Internet Technologies, State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 10 March 2013
Received in revised form
30 June 2013
Accepted 24 July 2013
Available online 24 January 2014

Keywords:
Mobile robot
Spiking neural networks
Modular navigation controller
Target-approaching
Obstacle-avoidance
Wall-following

ABSTRACT

Autonomous navigation plays an important role in mobile robots. Artificial neural networks (ANNs) have been successfully used in nonlinear systems whose models are difficult to build. However, the third generation neural networks – Spiking neural networks (SNNs) – contain features that are more attractive than those of traditional neural networks (NNs). Because SNNs convey both temporal and spatial information, they are more suitable for mobile robots' controller design. In this paper, a modular navigation controller based on promising spiking neural networks for mobile robots is presented. The proposed behavior-based target-approaching navigation controller, in which the reactive architecture is used, is composed of three sub-controllers: the obstacle-avoidance SNN controller, the wall-following SNN controller and the goal-approaching controller. The proposed modular navigation controller does not require accurate mathematical models of the environment, and is suitable to unknown and unstructured environments. Simulation results show that the proposed transition conditions for sub-controllers are feasible. The navigation controller can control the mobile robot to reach a target successfully while avoiding obstacles and following the wall to get rid of the deadlock caused by local minimum.

1. Introduction

Navigation of mobile robots refers to planning a path with obstacle avoidance to a specified goal and to execute this plan based on sensor readings and deduction in an unknown, uncertain and unstructured environment. The autonomous navigation plays an important role in mobile robots for fulfillment of given tasks [1].

Traditionally, mobile robots' navigation control requires accurate environmental models, and is effective only in structured environments. Besides, the traditional navigation controller can only fulfill some simple, repetitive tasks, and the robots are usually controlled to follow planned paths. It is very difficult to build mathematical models of unknown and unstructured environments, so the design of mobile robots' navigation controller in such circumstances is very difficult. Since neural networks (NNs) are useful tools for modeling and control of nonlinear systems, some NNs-based controllers for mobile robots have been developed successfully [2–15]. The new trend for mobile robot's controller designing is that some classical methods for controllers are usually combined with artificial neural network (ANN) methods [12–15].

Many studies have shown that the neurons in the mammalian brain use spikes, which are short electrical pulses, to communicate with other neurons. Those spike sequences can represent spatio-temporal information, and lead to a new type of neural network – spiking neural network (SNN). In SNNs, spiking neurons are employed to represent spatio-temporal information with pulse coding, like real neurons do. SNNs represent more plausible models of real biological neurons than those traditional ones. Besides that spiking neurons can be used to compute and communicate.

Some scholars believe that ANNs have developed from the first generation of artificial neural networks which consist of McCulloch–Pitts threshold neurons, the second-generation neurons which use continuous activation functions to compute their output signals, to the third generation – spiking neural networks (SNNs) [16].

SNNs, which use individual spike times to convey information, have stronger computational power than traditional neural networks (NNs). Besides that, SNNs can not only approximate arbitrary continuous functions, but also simulate any feed forward sigmoidal neural networks [17]. Because spikes are conveyed in SNNs, SNNs have better robustness to noise than other types of NNs. Moreover spikes can be modeled relatively easily by digital circuits, so SNNs are suitable to be realized by hardware. In addition, SNNs also show their good capabilities in pattern recognition and classification [18–29]. The approach used by Natschläer and Ruf [18] gives rise to a biologically plausible algorithm for finding clusters in a high-dimensional input space using SNN, even if the environment is changing constantly. In [23] new spiking neural network architecture and its corresponding

* Corresponding author. Tel./fax: +86 10 82544531.
E-mail address: zengguang.hou@ia.ac.cn (Z.-G. Hou).

0925-2312/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.07.055

learning procedures are presented to perform fast and adaptive multi-view visual pattern recognition.

Because of the attractive features of SNNs, many people are involved in the research of SNNs, and new results are constantly obtained. Various spiking neuron models are built, such as spike response model (SRM model), dynamic firing threshold model, leaky integrated-and-fire (LIF) neuron model, probabilistic spiking neuron model (PSNM). The LIF neuron model is the most famous and widely used model to simulate the SNNs. PSNM is a novel spiking neuron model, which has good robustness. There is a rigorous computational model, the liquid state machine (LSM), and there are some novel SNNs architecture: evolving SNNs, spike pattern association neuron (SPAN) architecture, the neurogenetic brain cube (NeuCube) architecture. SPAN is capable of learning input–output spike pattern association and output the desired spike train [26]. NeuCube is a novel evolving spiking model and it is used for modeling brain data specifically.

The training algorithms of SNNs can be categorized into the unsupervised methods and the supervised methods. The unsupervised spike-based learning methods include long-term potentiation (LTP) learning, long-term depression (LTD) learning, learning spike-based Hebbian learning and spike-timing-dependent plasticity (STDP). The supervised spike-based methods include statistical learning methods, spikeProp method [50], evolutionary methods, linear algebra methods, spike-based supervised-Hebbian learning (Remote Supervised Method ReSuMe) [21], SPAN method [26], and so on.

SNNs have been employed in the robotic area successfully, such as path planning [37], environment perception [28,29], and robots' behavior controllers [30–40].

Because SNNs convey temporal and spatial information, they can be used for "real" dynamic environments. While mobile robots always work in the unstructured and dynamic environments, SNNs are more suitable for robots' controller design than the traditional ANNs. The research team led by Prof. Floreano has done a lot of work on robots' controllers and they use genetic algorithm to optimize the weights and the structure of the SNNs [33,34]. In recent years there are also many new research results for robots' controllers based on SNNs: Gamez et al. [41,42] propose iSpike – a C++ library that interfaces between spiking neural network simulators and the iCub humanoid robot. iSpike converts the robot's sensory information into input spikes for the neural network simulator, and the output spikes from the network are decoded into motor signals to control the robot. Andre et al. [43] present a novel learning rule based on spike-timing-dependent plasticity for the designed SNNs which allows the SNNs to serve as a brain-like controller for the simulated robots successfully. Luque et al. [44–46] put forward a cerebellumlike spiking neural network which stores the corrective models as well-structured weight patterns distributed among the parallel fibers to Purkinje cell connections to achieve accurate control of non-stiff-joint robot-arm. The SNNs-based robot-arm controller can accomplish the given task fluently and has better robustness against noise. Alnajjar et al. [47] have designed a novel hierarchical adaptive controller, which is based on SNNs, for a real mobile robot with the goal of optimal navigation in dynamic environments. In [48], a three-layered spiking neural network with STDP learning rules as a target approaching controller for robots is used by Paolo et al.

In this paper, a behavior based target-approaching controller using spiking neural networks is designed. The modular navigation controller has three sub-controllers, and the sub-controllers are partially based on the previously designed obstacle-avoidance controller [30] and the wall-following controller [32].

This paper is organized as follows: Section 2 presents the mobile robot Casia-I's kinematical model and its sensor system. Section 3 discusses the proposed modular navigation controller based on spiking neural networks. Section 4 presents the simulation results. The paper is concluded in Section 5.

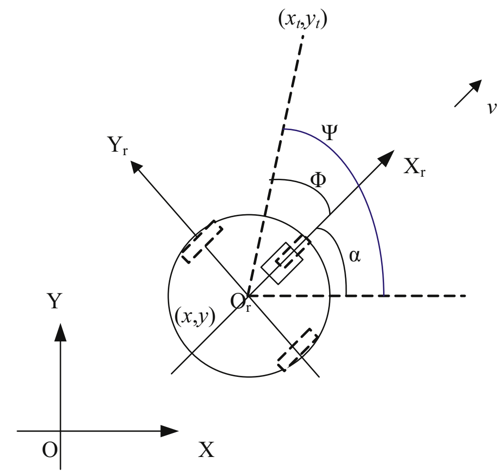

Fig. 1. Mobile robot's pose.

2. Kinematic model and sonar system of the mobile robot

2.1. The kinematic model

In this study, the mobile robot as shown in Fig. 1 is a system satisfying the nonholonomic constraints. In 2-dimensional Cartesian space, the pose of the mobile robot q is represented by

$$q = (x, y, \alpha)^T, \tag{1}$$

where $(x, y)^T$ is the position of the robot in the reference coordinate system XOY, and the heading direction α is taken counterclockwise from the positive direction of the X-axis. $X_rO_rY_r$ is the coordinate for the robot system. (x_t, y_t) is the coordination of the target for the mobile robot's navigation. The angle ϕ is taken counterclockwise from the positive direction of the X_r-axis to xx_t. The angle ψ is taken counterclockwise from the positive direction of the X-axis to xx_t. The solid line rectangle represents the camera, and the dashed line rectangles represent the robot's driving wheels and the guided wheel. If Δt is small enough, the mobile robot's trajectory can be approximated by the following equation from t to $t+m\Delta t$:

$$\begin{cases} x(m+1) = x(m) + v\cos(\alpha(m))\Delta t \\ y(m+1) = y(m) + v\sin(\alpha(m))\Delta t \\ \alpha(m+1) = \alpha(m) + \omega(m)\Delta t, \end{cases} \tag{2}$$

where m is an integer and $m = 1, 2, \ldots, [1/\Delta t]$.

2.2. The mobile robot's sonar sensor system

Ultrasonic sensors have been widely used in mobile robots because of its attractive properties, e.g. cheapness, reliability and so on. The mobile robot Casia-I used in our experiment has a peripheral ring of 16 evenly distributed Polaroid ultrasonic sensors, which are denoted by S1–S16. The sonar sensor system is shown in Fig. 2.

3. Modular navigation controller based on spiking neural networks

There are many different kinds of behavior-based controllers. According to the various given tasks, the entire task module can be divided into the goal task module and the sub-goal task module. By various classification standards, the behaviors of the mobile robots can not only be classified into the planned behaviors and the reactive behaviors, but also be classified into combined behaviors and basic behaviors. In this paper, the combined behaviors or the target-approaching behaviors have been divided into several sub-goal

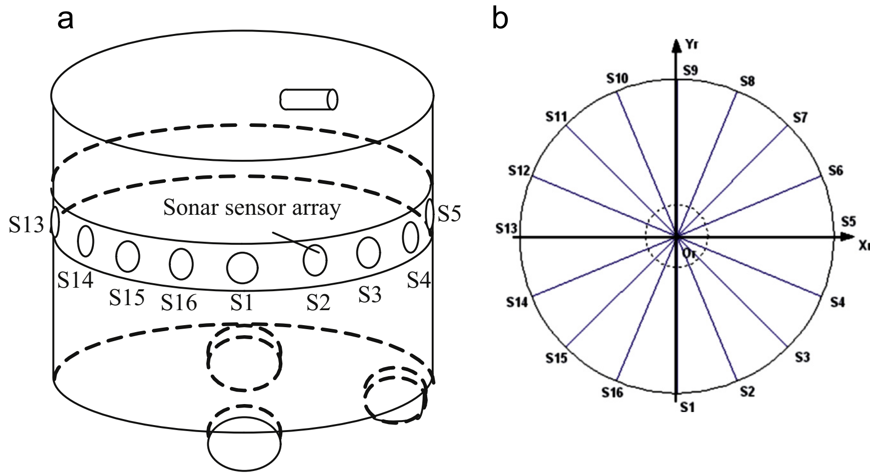

Fig. 2. Mobile robot Casia-I's sonar sensors. (a) Casia-I's sonar sensor array. (b) Layout of Casia-I's sonar sensors.

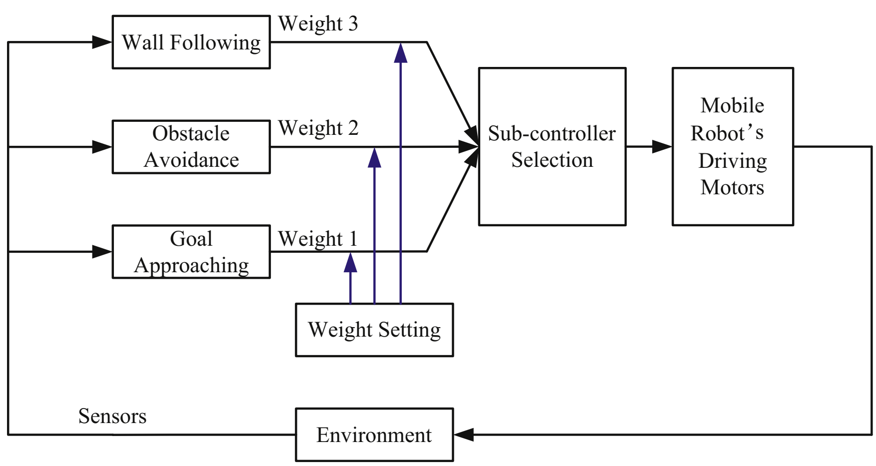

Fig. 3. Block diagram for the mobile robots' target-approaching navigation controller.

behaviors, which include the obstacle-avoidance behavior, the wall-following behavior, and the goal-approaching behavior. A target-approaching navigation controller based on SNNs is designed and the block diagram for the controller is shown in Fig. 3. The proposed navigation controller is composed of three sub-controller modules: a obstacle-avoidance controller module, a wall-following controller module and a goal-approaching controller module. The weight-setting module in the proposed navigation controller can set the weight of different sub-controllers in different situations. The details of weights' setting of the sub-controllers can be referred in Section 3.2. The sub-controller selection module chooses the sub-controller with the biggest weight to work first.

The different sub-controller modules can transfer from one to another under appropriate transition conditions, and the transition conditions can be referred to in Section 3.3.

The reactive architecture, which was first proposed by Arkin [49], is used in the proposed mobile robot navigation controller. In such behavior-based architecture, environmental models are not needed. The mobile robots' activities can be divided into a series of basic behaviors. In the reactive architecture, the relationship between the sensors and the executors is built directly by the mapping between the patterns of the sensory information and that of the mobile robots' activities. By the reactive architecture, the robots' capability to respond to the environment is greatly enhanced.

The flow chart of the designed target-approaching navigation controller is shown in Fig. 4.

3.1. Control strategy for the goal-approaching module

The goal-approaching controller module includes the following sub-modules: (1) a position/orientation adjustment module; (2) a forward moving to the goal module. Different modules can transfer from one module to another under appropriate transition conditions. Angle ϕ in Fig. 1, which denotes the orientation of the robot, is calculated as

$$\psi = \begin{cases} \arctan\left(\dfrac{y_t - y}{x_t - x}\right), & x_t - x > 0 \\ \pi + \arctan\left(\dfrac{y_t - y}{x_t - x}\right), & x_t - x < 0,\ y_t - y \ge 0 \\ -\pi + \arctan\left(\dfrac{y_t - y}{x_t - x}\right), & x_t - x < 0,\ y_t - y < 0 \\ \pi/2, & x_t - x = 0,\ y_t - y > 0 \\ -\pi/2, & x_t - x = 0,\ y_t - y < 0. \end{cases} \tag{3}$$

$$\phi = \psi - \alpha \tag{4}$$

- When ϕ is larger than a given threshold value, the position/orientation adjustment module will work. In the position/orientation adjustment module, the robots' position/orientation should be adjusted as follows: control the robot's left and right wheels to turn at the same speed but in the opposite direction. By such way, the robot will rotate by ϕ.

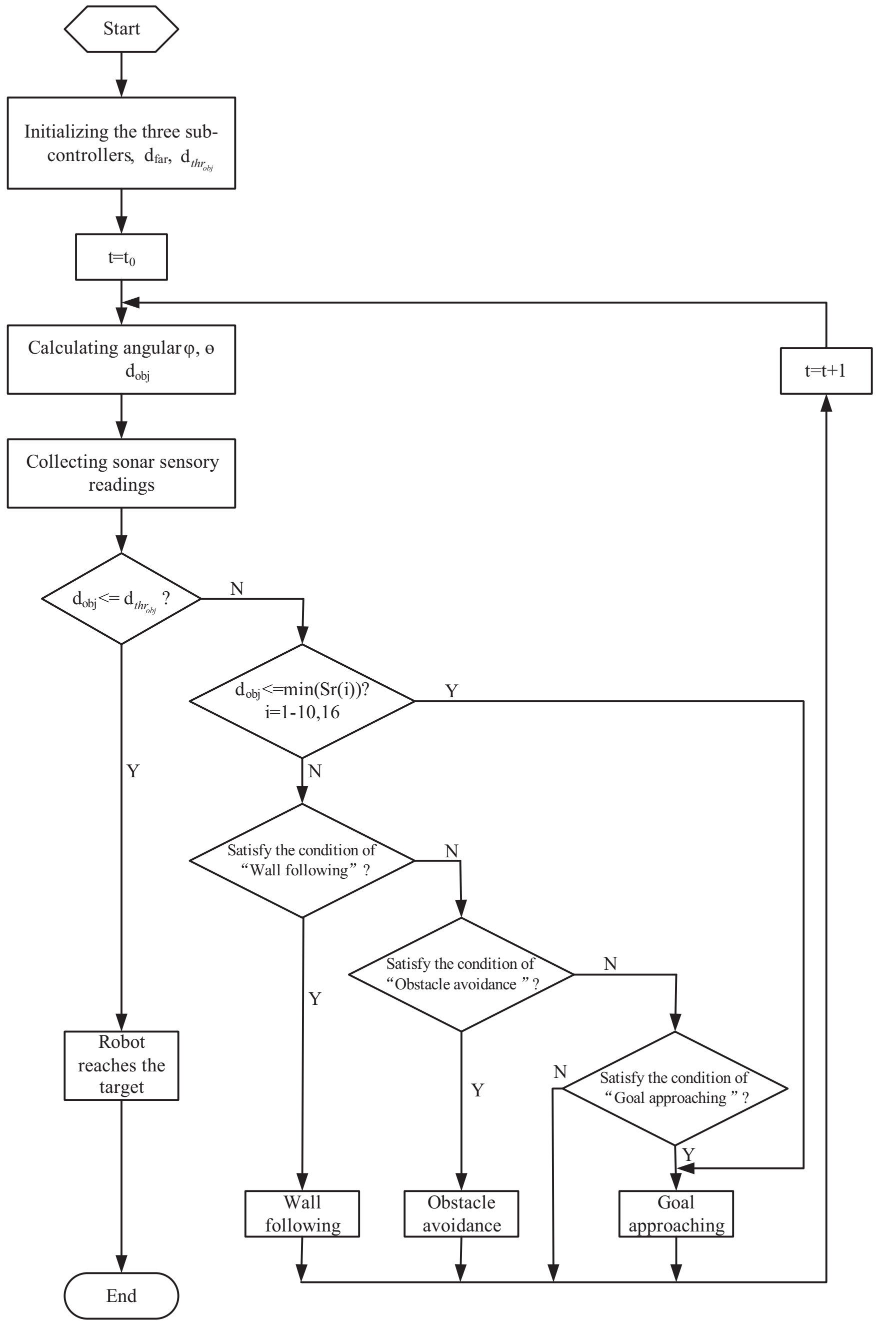

Fig. 4. Flow chart for the mobile robots' target-approaching navigation controller.

- When ϕ is less than a given threshold, position/orientation adjustment module is turned to the forward moving to the goal module; that is, the robot will move forward directly to the goal.

3.2. Weight setting for the sub-controllers

Suppose that $W_{wall-following}$, $W_{obstacle-avoidance}$ and $W_{goal-approaching}$ represent the weights of the wall-following sub-controller, the obstacle-avoidance sub-controller and the goal-approaching sub-controller respectively. The weight setting module will set the order of the weights for different sub-controllers as follows:

if $d_{obj} \le \min(S_r(i)),\ i = 1-10, 16$ **then**
$W_{goal-approaching} > W_{wall-following}$,
and $W_{wall-following} > W_{obstacle-avoidance}$.
else
$W_{wall-following} > W_{obstacle-avoidance}$,
and $W_{obstacle-avoidance} > W_{goal-approaching}$.
end if

The sub-modules' weights are different from the weights in NNs. The order of the sub-modules' weights represents the importance of different modules. The sub-controller with the biggest weight will be activated first, then the sub-controller with the bigger weight will be activated and the sub-controller with the smallest weight will be activated last.

3.3. Module transition conditions for the navigation controller

Suppose that the readings of the ith sonar sensor are $S_r(i)$ (The sonar sensor model is shown in Fig. 2.). When $S_r(i) > d_{far}$ (d_{far} is the threshold), it is supposed that there is no obstacle in the direction of sensor $S_r(i)$. The obstacles in different directions affect the robot differently, thus the threshold values are different for the sensors in different directions. The threshold value for the front obstacle of the sensor is $d_{thr_{obstacle}}$, and the threshold value for the obstacles at two sides is d_{far}. The distance between the center of the robot and the target is denoted by d_{obj}, and $d_{thr_{obj}}$ is the threshold value for the distance between the center of the robot and the target point. If $d_{obj} \le d_{thr_{obj}}$, it is believed that the robot reaches the target point and should stop moving.

3.3.1. Wall-following module

Wall-following behavior is a commonly used control strategy in mobile robot control. Wall-following behavior can help the robot to get rid of the deadlocking caused by the local minimum. Besides, the robot can navigate smoothly in the structured corridor scene by the wall-following strategy.

The transition condition for starting the wall-following module is A: when the robot moves forward and the front sensors $S_r(i)$ ($i=4,5,6$) satisfy the following inequality:

$$\cup S_r(i) < d_{thr_{obstacle}}, \quad i=4,5,6. \tag{5}$$

The transition condition for exiting the wall-following module is B:

$$\left(\phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right) \bigcap_{i=4,5,6} (S_r(i) > d_{thr_{obstacle}}). \tag{6}$$

The condition B means that if the target lies in the front range of the robot and the wall (or obstacle) is far enough away from the robot, the robot could exit the wall-following module.

3.3.2. Obstacle-avoidance module

The transition condition for starting the obstacle-avoidance module is C:

$$(\cup S_r(i) < d_{far}, \ i=1,2,3,7,8,9,10,16) \cap \overline{A}. \tag{7}$$

3.3.3. Goal-approaching module

The transition conditions for starting the goal-approaching module will be the following cases:

(1) The first case is D:

$$(\cap S_r(i) > d_{far}, \ i=1-10,16) \cap \overline{A} \cap \overline{C}. \tag{8}$$

If Eq. (8) is satisfied, the goal-approaching module starts.

(2) The second case is E:
If $d_{obj} \le \min(S_r(i))$, $i=1$–$10, 16$, the weight setting module will let the goal-approaching module have the biggest weight, and then the goal-approaching module also starts.

So to activate the goal-approaching module, the condition F must satisfy

$$\mathrm{F} = \mathrm{D} \cup \mathrm{E}. \tag{9}$$

3.4. Principle of obstacle-avoidance module

The block diagram of the SNNs for the obstacle-avoidance controller is shown in Fig. 5, and the detailed structure of the SNNs in the obstacle-avoidance controller could be found in [30]. In the obstacle-avoidance SNN controller, the readings of the sonar sensors from S1 to S9 are considered (layout of the sonar sensors is shown in Fig. 2). The sensors are divided into three groups: Group 1, including sensors from S1 to S3; Group 2, from S4 to S6; Group 3, from S7 to S9.

$$S(m) = \min(S_r(i)), \tag{10}$$

where $m=1,2,3$ and i is the number of the sensors in the divided groups. In each group, the smallest reading $S(m)$ will be the testing result. Then $S(m)$ is encoded by spiking frequency coding. The information of the sonar sensors is used as the input to the corresponding sensory neuron. The approximate neuron is used to judge whether the opposite obstacle is so close that the robot need to turn around or not. The turning neurons decide which direction the mobile robot would turn to when the opposite obstacle is very close. If the angular $\phi \ge 0$, the neuron T_L fires and emits a pulse, and the mobile robot will turn left. Otherwise (if $\phi < 0$), the neuron T_R fires and emits a pulse, and the mobile robot will turn right. Spiking coincidence detecting coding is used for the hidden neurons. Spiking Integrated-and-Fire (IAF) model and spiking frequency coding are used for the motor neurons in the SNNs controller.

Unsupervised Hebbian learning is used in the obstacle-avoidance SNNs controller. In the obstacle-avoidance sub-controller, the connecting weights for the motor neurons and the sensory neurons are updated by the spike-based Hebbian learning rules. The learning window function used for Hebbian learning in SNNs controller is chosen according to [51]. The SNNs in the controller can be trained online.

Before the frequency of the output spikes of the motor neurons is used to control the speed of the mobile robot's driving wheels, the frequency of the output spikes will be adjusted by the pulse adjusting module as follows:

- In order to guarantee the speed of the robot will not be too fast to encounter the obstacle, the frequency of the output spikes is subject to certain limits, the details of the limits could be referred to in [30].
- If the speed for the robot to turn around is too great, shaking will occur. So the difference for the output spikes of the right

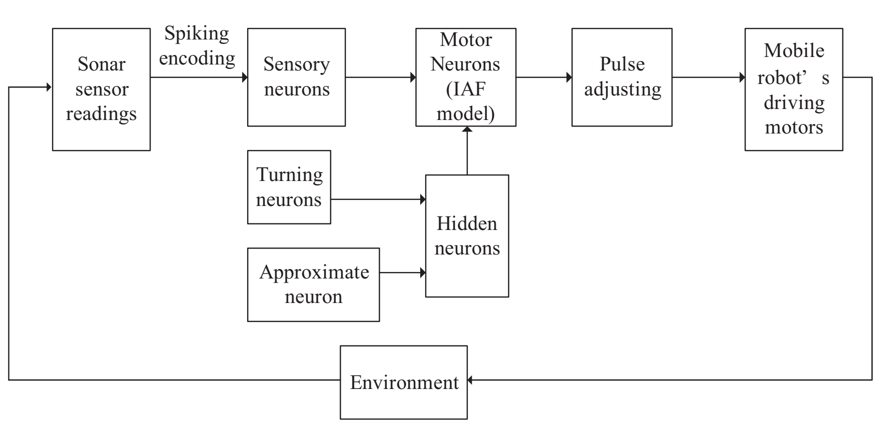

Fig. 5. Block diagram of obstacle-avoidance controller based on SNNs.

motor neurons and the left motor neurons must be below a given threshold value.

The simulation results and more details of the obstacle-avoidance controller can be referred to in [30].

3.5. Principle of wall-following module

The block diagram of the proposed SNNs wall-following controller is shown in Fig. 6. The principle of the wall-following controller is similar to that of the obstacle-avoidance controller. In the wall-following controller, too far and too close detection neurons are used to judge whether the robot is too far away or too close to the wall. If it is true, the corresponding spiking neurons will fire. Hebbian learning is also used for the SNNs wall-following controller. The wall-following controllers can control the robot with the varying initial pose to follow the wall clockwise and counter-clockwise. If the angular $\phi \geq 0$, the robot will follow the wall counter-clockwise. If the angular $\phi < 0$, the robot will follow the wall clockwise. With the wall-following controller, the robot can get rid of the deadlock problem caused by local minimum. The structure of the clockwise wall-following controller, the simulation results and the other details of the wall-following controller can be referred to in [32].

4. Simulation and discussion

4.1. Simulation

In this simulation, the robot is the same as that in [30,32]. The parameters of the robot are $r = 50$ mm, $b = 225$ mm. The simulation environment is a square area with walls, and the size of the square area is $7070 \times 7070\ \text{mm}^2$. The length of the time windows for pulse encoding of the sensory neurons and the motor neurons is 100 ms. The robot can explore the unknown environment to get the obstacle's location by sensors in real time, and then use this information to do local path planning. The simulation is carried out supposing that the robot can be localized accurately. The simulation results are shown in Figs. 7–13. "*" represents the starting point, and the target point is denoted by "⋆".

In Fig. 7, the mobile robot controlled by the navigation controller experiences the following process by use of the navigation transition conditions: goal-approaching module → obstacle-avoidance module → goal-approaching module → wall-following module → goal-approaching module, and at last the robot arrived at the target point denoted by "⋆".

Similarly, in Fig. 8 the robot experiences the following process: goal-approaching module → wall-following module → obstacle-avoidance module → goal-approaching module → obstacle-avoidance module → goal-approaching module, and then the robot arrives at the target point.

In Figs. 9 and 10 the robot experiences the following process: goal-approaching module → obstacle-avoidance module → goal-approaching module, after that the robot arrives at the target point.

In Fig. 11 the robot experiences the following process: wall-following module → goal-approaching module → wall-following module → goal-approaching module, and at last the robot arrives at the target point.

In Figs. 12 and 13 the robot follows the wall clockwise and counter-clockwise respectively according to the value of angular ϕ, which has been discussed in Section 3.5. By different types of wall-following, the robot approaches the target successfully at last.

In the simulation experiment, if it is believed that the robot has arrived at the target point, then the robot stops walking. From the simulation results it can be seen that the navigation controller can control the robot by varying initial positions to the target successfully.

4.2. Discussion

The simulation results show that not only is the sub-controller based on SNNs effectively, with the proper transition conditions for the target-approaching navigation controller, but also the whole navigation controller using SNNs can control the robot to reach the target by a sub-optimal path successfully. It shows that spiking neural networks are suitable for robot controller designs.

4.2.1. Effect of the controller

The effect of the proposed controller depends on the following:

- *The control effect of the sub-modules of the navigation controller*: If each sub-controller has higher control accuracy, the proposed navigation controller will have better control results.
- *The transition conditions*: Whether the transition conditions of the sub-module in the navigation controller are feasible or not is also a very key factor for the effectiveness of the controller.

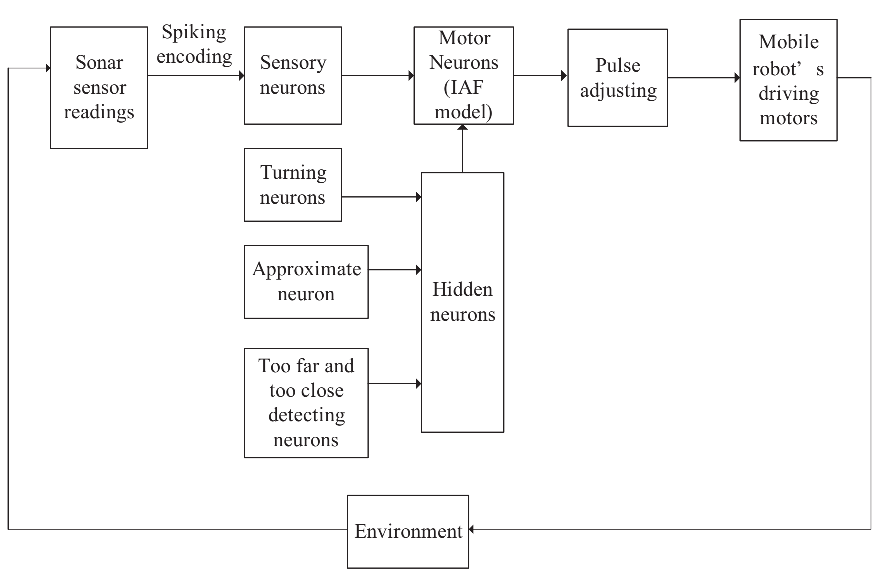

Fig. 6. Block diagram of the wall-following controller based on SNNs.

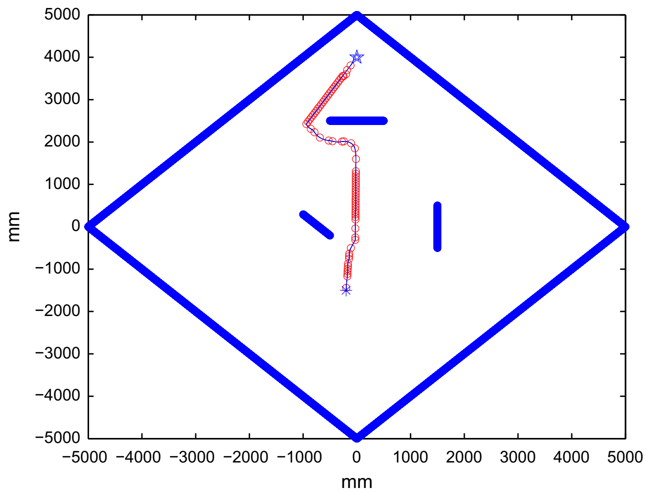

Fig. 7. Navigation results for the robot with the initial pose q: q is $(-200, -1500, 90^\circ)$.

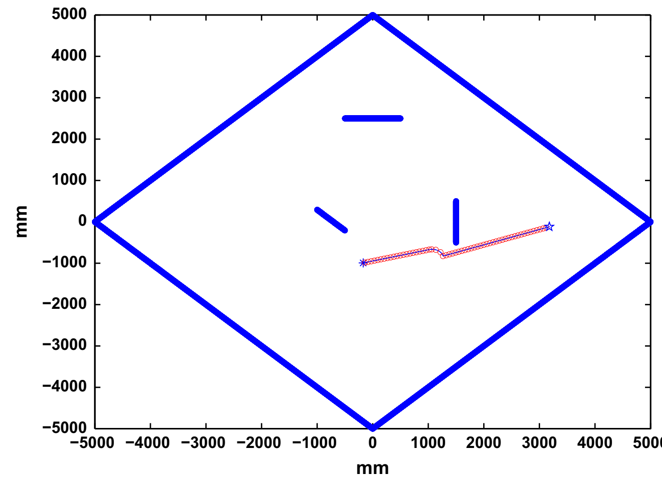

Fig. 10. Navigation results for the robot with the initial pose q: q is $(-200, -1000, 30^\circ)$.

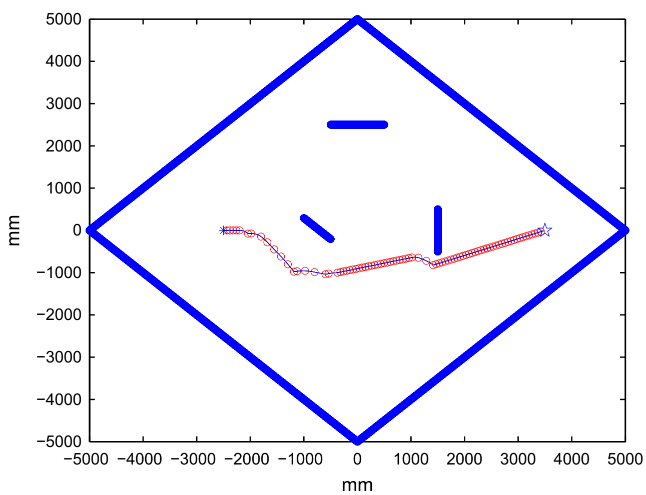

Fig. 8. Navigation results for the robot with the initial pose q: q is $(-2500, 0, 0^\circ)$.

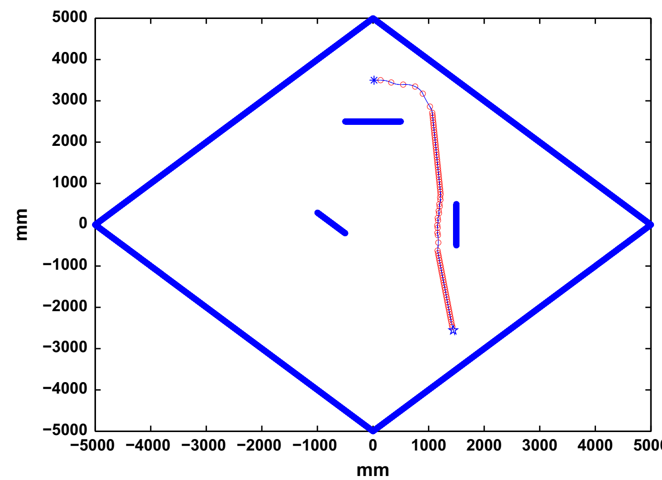

Fig. 11. Navigation results for the robot with the initial pose q: q is $(0, 3500, 0^\circ)$.

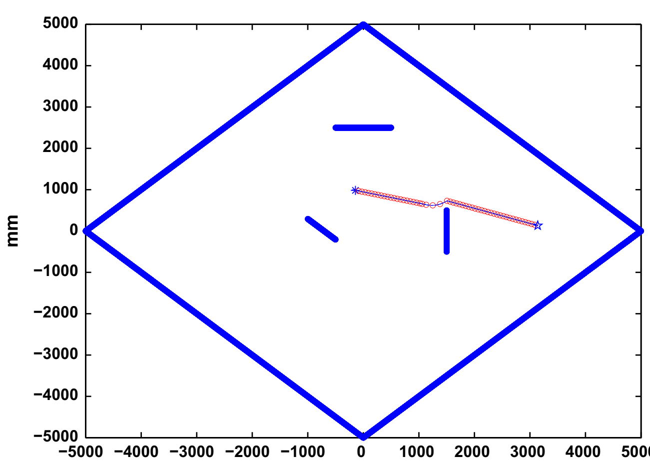

Fig. 9. Navigation results for the robot with the initial pose q: q is $(200, 1000, -30^\circ)$.

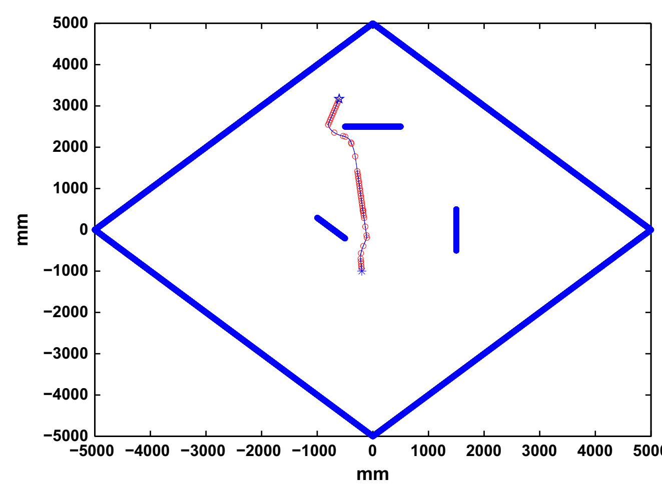

Fig. 12. Navigation results for the robot with the initial pose q: q is $(-200, -1000, 90^\circ)$.

- *The threshold value*: In the proposed controller, the threshold values such as d_{far}, $d_{thr_{obstacle}}$, and $d_{thr_{obj}}$ are set by experiences and whether they are feasible or not is also very important.

4.2.2. Comparison between SNNs controllers and traditional NNs controllers

Li et al. proposed a reinforcement learning based radial-basis function network control system and a reinforcement-based neural-fuzzy control system in [9–11]. With the NNs controller

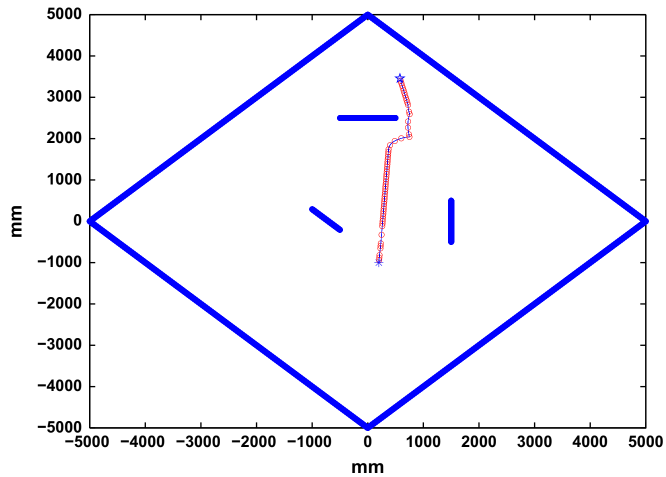

Fig. 13. Navigation results for the robot with the initial pose q: q is (200, −1000, 90°).

the robot can avoid obstacles during its goal-reaching navigation in unknown environments. Compared with the controllers designed by Li et al., although both of the controllers are effective, the proposed SNNs controller has the following advantages over the traditional NNs controllers:

- In the SNNs controller, more spatio-temporal information is incorporated than in traditional NNs controllers. Usually, that spatio-temporal information is very useful for the robot navigation controller.
- The structure of SNNs controller is much simpler than that of the traditional NNs controllers.
- The training method used in the proposed controller is Hebbian learning and it is easy to be implemented than the training methods in [9–11].

5. Conclusion

The proposed navigation controller based on SNNs can control the robot by varying initial positions to the target successfully. The proposed transition conditions of the sub-modules in the navigation controller are easy to implement and feasible. The experimental results show the efficiency of the strategy of the behavior based target-reaching navigation controller using SNNs, and that SNNs are suitable for robot controllers' designs. The controller has simple structure and can be implemented easily.

Efforts are going on for handling more complex tasks such as for the neuro-rehabilitation robots. Realizing the proposed controller by hardware based on FPGA will also be part of our future research.

Acknowledgments

This work was mainly supported by the National Natural Science Foundation of China (Grants 61175059, 61005009, 61225017) and by the Nature Science Foundation of HeBei Province (Grant F2011205023), and this work was supported in part by the Open Research Project (Grant 20120103) from SKLMCCS. The authors are also thankful to Prof. N. Kasabov of Auckland University of Technology for his helpful and constructive suggestions.

References

[1] Y. Wang, M. Cartmell, Q. Tao, H. Liu, A generalized real-time obstacle avoidance method without the Cspace calculation, J. Comput. Sci. Technol. 62005 (20) (2005) 774–787.
[2] D.A. Pomerleau, Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers, Norwell, MA, USA, 1993.
[3] A.G. Barto, R.S. Sutton, C.W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Trans. Syst. Man Cybern. 13 (5) (1983) 834–846.
[4] K.H. Low, W.K. Leow, M.H. Ang, Integrated planning and control of mobile robot with self-organizing neural network, in: Proceedings of IEEE ICRA, IEEE Press, New York, 2002, pp. 3869–3875.
[5] Y.K. Na, S.Y. Oh, Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification, Auton. Robots 15 (2) (2003) 193–206.
[6] L. Cheng, Z.-G. Hou, M. Tan, Y. Lin, W. Zhang, Neural-network-based adaptive leader-following control for multi-agent systems with uncertainties, IEEE Trans. Neural Netw. 21 (8) (2010) 1351–1358.
[7] Z.-G. Hou, L. Cheng, M. Tan, Multi-criteria optimization for coordination of redundant robots using a dual neural network, IEEE Trans. Syst. Man Cybern. B Cybern. 40 (4) (2010) 1075–1087.
[8] L. Cheng, Z.-G. Hou, M. Tan, Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model, Automatica 45 (10) (2009) 2312–2318.
[9] J. Li, J. Yi, D. Zhao, G. Xi, A reinforcement learning based radial-basis function network control system, In: Proceedings of ISNN 2005, Lecture Notes in Computer Science, vol. 3496, Springer, Berlin, 2005, pp. 640–645.
[10] J. Li, J. Yi, D. Zhao, G. Xi, Cooperation learning for behavior-based neural-fuzzy controller in robot navigation, in: Proceedings of IFAC World Congress 2005, vol. 16, Part 1, IFAC, Czech Republic, pp. 1270–1275.
[11] J. Li, Soft computing method for the autonomy navigation for the mobile robots (Ph.D. thesis), Institute of Automation, The Chinese Academy of Sciences, China, 2005.
[12] J. Ye, Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot, Neurocomputing 71 (2008) 1561–1565.
[13] J. Ye, Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique, Neurocomputing 71 (2008) 3373–3378.
[14] F.G. Rossomando, C. Soria, R. Carelli, Autonomous mobile robots navigation using RBF neural compensator, Control Eng. Pract. 19 (2011) 215–222.
[15] O. Mohareri, R. Dhaouadi, A.B. Rad, Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, Neurocomputing 88 (2012) 54–66.
[16] J. Vreeken, Spiking Neural Networks, An Introduction, Technical Report UU-CS-2003-008, Institute for Information and Computing Sciences, Utrecht University, 2002. 〈http://ailab.cs.uu.nl/pubs/SNNVreeken Introduction.pdf〉.
[17] W. Maass, C.M. Bishop, Pulsed Neural Networks, MIT-Press, Cambridge, MA, 1999.
[18] T. Natschläer, B. Ruf, Spatial and temporal pattern analysis via spiking neurons, Network: Comput. Neural Syst. 9 (3) (1998) 319–332.
[19] S.M. Bohte, H.L. Poutre, J.N. Kok, Unsupervised clustering with spiking neurons by sparse temporal coding and multiplayer spike neural network, IEEE Trans. Neural Netw. 13 (2) (2002) 426–435.
[20] B. Ruf, Computing and learning with spiking neurons theory and simulations (Ph.D. thesis), Institute for Theoretical Computer Science Technische University at Graz, Austria, May 1998.
[21] F. Ponulak, A. Kasinski, Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting, Neural Comput. 22 (2) (2010) 467–510.
[22] S. Schliebs, M. Defoin-Platel, S. Worner, N. Kasabov, Integrated feature and parameter optimization for evolving Spiking neural networks: exploring heterogeneous probabilistic models, Neural Netw. 22 (5–6) (2009) 623–632.
[23] S. Wysoski, L. Benuskova, N. Kasabov, Fast and adaptive network of Spiking neurons for multi-view visual pattern recognition, Neurocomputing 71 (2008) 2563–2575.
[24] N. Kasabov, Integrative connectionist learning systems inspired by nature: current models, future trends and challenges, Nat. Comput. 8 (2) (2009) 199–218.
[25] N. Kasabov, To spike or not to spike: a probabilistic spiking neuron model, Neural Netw. 23 (1) (2010) 16–19.
[26] N. Kasabov, Evolving Spiking neural networks and neurogenetic systems for spatio- and spectro-temporal data modelling and pattern recognition, in: Proceedings of IEEE WCCI 2012, Lecture Notes in Computer Science, vol. 7311, Springer, Berlin, 2012, pp. 234–260.
[27] N. Kasabov, NeuCube evoSpike architecture for spatio-temporal modelling and pattern recognition of brain signals, in: Proceedings of ANNPR 2012, Lecture Notes in Artificial Intelligence, vol. 7477, 2012, pp. 225–243.
[28] X.Q. Wang, Z.-G. Hou, M. Tan, Y. Wang, Corridor-scene classification for mobile robot using Spiking neurons, in: Proceedings of IEEE ICNC 2008, vol. 4, IEEE Press, New York, 2008, pp. 125–129.
[29] X.Q. Wang, Z.-G. Hou, M. Tan, Improved mobile robot's corridor-scene classifier based on probabilistic Spiking neuron model, in: Proceedings of IEEE ICCICC, IEEE Press, New York, 2011, pp. 348–355.
[30] X.Q. Wang, Z.-G. Hou, M. Tan, L. Cheng, A behavior controller for mobile robot based on Spiking neural networks, Neurocomputing 71 (2008) 655–666.
[31] X.Q. Wang, Z.-G. Hou, F. Lv, M. Tan, Y. Wang, A target-reaching controller for mobile robots using spiking neural networks, in: Proceedings of ICONIP 2012, Lecture Notes in Computer Science, vol. 7666, Springer, Berlin, 2012, pp. 652–659.
[32] X.Q. Wang, Z.-G. Hou, M. Tan, Y. Wang, The wall-following controller for the mobile robot using spiking neurons, in: Proceedings of IEEE 2009 ICAICI, IEEE Press, New York, 2009, pp. 194–199.

[33] D. Floreano, C. Mattiussi, Evolution of spiking neural controllers for autonomous vision-based robots, in: Proceedings of ER 2001, Lecture Notes in Computer Science, vol. 2217, Springer, Berlin, 2001, pp. 38–61.
[34] D. Floreano, J.C. Zufferey, J.D. Nicoud, From wheels to wings with evolutionary spiking neurons, Artif. Life 11 (2005) 121–138.
[35] H. Hagras, A. Pounds-Cornish, M. Colley, Evolving spiking neural network controllers for autonomous robots, in: Proceedings of IEEE ICRA 2004, IEEE Press, New York, 2004, pp. 4620–4626.
[36] H. Soula, A. Alwan, G. Beslon, Learning at the edge of chaos: temporal coupling of spiking neurons controller for autonomous robotic, in: Proceedings of AAAI 2005 Spring Symposia on Developmental Robotic, 2005, p. 6.
[37] R.S.A. Alamdari, Unknown environment representation for mobile robot using spiking neural networks, Trans. Eng. Comput. Technol. 6 (2005) 49–52.
[38] N. Kubota, K. Nishida, The role of spiking neurons for visual perception of a partner robot, in: Proceedings of 2006 IEEE ICFS, IEEE Press, New York, 2006, pp. 530–537.
[39] X. Wang, Z.-Q. Cao, W.-W. Zhang, M. Tan, Z.-G. Hou, X.Q. Wang, Formation control for multiple mobile robots based on the Spiking neural network, in: Proceedings of IEEE ICRAM, IEEE Press, New York, 2010, pp. 447–452.
[40] X.Q. Wang, Research on environment perception and behavior control for mobile robot based on spiking neural networks (Ph.D. thesis), Institute of Automation, The Chinese Academy of Sciences, China, 2007.
[41] D. Gamez, A.K. Fidjeland, E. Lazdins, iSpike: a spiking neural interface for the iCub robot, Bioinspir. Biomim. 7 (2) (2012) 255–265.
[42] D. Gamez, Information integration based predictions about the conscious states of a spiking neural network, Conscious. Cogn. 19 (1) (2010) 294–310.
[43] C. Andre, B. Mounir, Classical conditioning in different temporal constraints: an STDP learning rule for robots controlled by spiking neural networks, Adapt. Behav. 20 (4) (2012) 257–272.
[44] N.R. Luque, J.A. Garrido, R.R. Carrillo, Adaptive cerebellar spiking model embedded in the control loop: context switching and robustness against noise, Int. J. Neural Syst. 21 (5) (2011) 385–401.
[45] N.R. Luque, J.A. Garrido, R.R. Carrillo, Cerebellar input configuration toward object model abstraction in manipulation tasks, IEEE Trans. Neural Netw. 22 (8) (2011) 1321–1328.
[46] N.R. Luque, J.A. Garrido, R.R. Carrillo, Cerebellarlike corrective model inference engine for manipulation tasks, IEEE Trans. Syst. Man Cybern. B Cybern. 41 (5) (2011) 1299–1312.
[47] F. Alnajjar, I.B.M. Zin, K. Murase, A hierarchical autonomous robot controller for learning and memory: adaptation in a dynamic environment, Adapt. Behav. 17 (3) (2009) 179–196.
[48] A. Paolo, F. Luigi, F. Mattia, Learning anticipation via Spiking networks: application to navigation control, IEEE Trans. Neural Netw. 20 (2) (2009) 202–216.
[49] R.C. Arkin, Integrating behavioral, perceptual, and world knowledge in reactive navigation, Rob. Auton. Syst. 6 (1–2) (1990) 105.
[50] S.M. Bohte, J.N. Koka, H.L. Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (2002) 17–37.
[51] R. Kempter, W. Gerstner, J.L. van Hemmen, Hebbian learning and spiking neurons, Phys. Rev. E 59 (1999) 4498–4514.

Xiuqing Wang received the B.S. degree (with honors) in precision instrument from Tianjin University, Tianjin, China, in 1992, and the M.S. degree (with honors) at control theory and control engineering from Hebei Technology University, Tianjin, China, in 2000. She received the Ph.D. degree at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. She is a Full Professor at Hebei Normal University, Shijiazhuang, China. Her research interests are neural networks, multi-sensor-fusion, and their applications in robotics.

Zeng-Guang Hou received the B.E. and M.E. degrees in electrical engineering from Yanshan University, Qinhuangdao, China, in 1991 and 1993, respectively. He received the Ph.D. degree in electrical engineering from Beijing Institute of Technology, Beijing, China, in 1997. He is a Full Professor at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. Dr. Hou is currently serving as an Associate Editor for IEEE Computational Intelligence Magazine, and an Editorial Board Member of International Journal of Intelligent Systems Technologies and Applications. He served as a Guest Editor of special issues of International Journal of Vehicle Autonomous Systems on “Computational Intelligence and Its Applications to Mobile Robots and Autonomous Systems”, and of Soft Computing on “Fuzzy-Neural Computation and Robotics”. His research interests include neural networks, optimization algorithms, robotics, and intelligent control systems.

Feng Lv is a Full Professor at Hebei Normal University, Shijiazhuang, China. Her research interests are robot control, fault testing and fault diagnosis, multisensorfusion.

Min Tan received the B.S. degree in control engineering from Tsinghua University, Beijing, China, in 1986 and the Ph.D. degree in control theory and control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1990. He is a Full professor at the Institute of Automation, Chinese Academy of Sciences. Dr. Tan is an Acting Committee Member of Robotics Technology of the National Hi-Tech Development (863) Program. He is also a Director of several academic societies, such as the Artificial Intelligence Society, and Youth Automation Society of China, etc. Dr. Tan is currently serving as an editorial board member of several journals, such as Acta Automatica Sinica, Robot, etc. He also got several academic awards from Chinese Academy of Sciences and the Beijing Science and Technical Committee. His research interests include advanced robot control, multirobot system, biomimetic robot, and manufacturing system.

Yongji Wang received the B.S. degree and M.S.(with honors) from Beijing University of Aeronautics & Astronautics, Beijing, China, in 1984 and in 1987, respectively. He received the Ph.D. degree from Edinburgh University, England, in 1995. He is a Full Professor at the Institute of Automation, Chinese Academy of Sciences, Beijing, China, at the Institute of Software, Chinese Academy of Sciences, Beijing, China. His research interests include real-time system, intelligent software engineering, control theory.