Hindawi
Complexity
Volume 2017, Article ID 1895897, 14 pages
https://doi.org/10.1155/2017/1895897

Review Article

A Brief Review of Neural Networks Based Learning and Control and Their Applications for Robots

Yiming Jiang,[1] Chenguang Yang,[1] Jing Na,[2] Guang Li,[3] Yanan Li,[4] and Junpei Zhong[5]

[1] *Key Laboratory of Autonomous Systems and Networked Control, College of Automation Science and Engineering, South China University of Technology, Guangzhou, China*
[2] *Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming, China*
[3] *School of Engineering and Material Sciences, Queen Mary University of London, London, UK*
[4] *School of Engineering and Informatics, University of Sussex, Brighton, UK*
[5] *National Institute of Advanced Industrial Science and Technology, Tokyo, Japan*

Correspondence should be addressed to Chenguang Yang; cyang@ieee.org

Received 15 June 2017; Accepted 1 October 2017; Published 31 October 2017

Academic Editor: Thierry Floquet

As an imitation of the biological nervous systems, neural networks (NNs), which have been characterized as powerful learning tools, are employed in a wide range of applications, such as control of complex nonlinear systems, optimization, system identification, and patterns recognition. This article aims to bring a brief review of the state-of-the-art NNs for the complex nonlinear systems by summarizing recent progress of NNs in both theory and practical applications. Specifically, this survey also reviews a number of NN based robot control algorithms, including NN based manipulator control, NN based human-robot interaction, and NN based cognitive control.

1. Introduction

In recent years, the research of neural network (NN) has attracted great attention. It is well known that, mammals' brain, which consists of billions of interconnected neurons, has the ability to deal with complex and computationally demanding tasks, such as face recognition, body motion planning, and muscles activities control. Figure 1 shows a cellular structure of a mammalian neuron. Inspired by the neuron structure, artificial NN (ANN) was developed to emulate the learning ability of the biological neurons system [1–3]. The concept of artificial NNs was initially investigated by McCulloch and Pitts in the 1940s [3], where the network is established with a parallel structure. The basically mathematical model of NN consists of three layers, that is, input layer, hidden layer, and output layer, which are of simple parallel computational structure but with appealing learning ability and computational power to predict nonlinear dynamic patterns.

In past decades, the NN technique has been studied extensively in areas such as control engineering, aerospace, medicine, automotive, psychology, economics, energy science, and many other fields [4–7]. It has been reported that NN can approximate any unknown continuous nonlinear function by overlapping the outputs of each neuron. Moreover, the approximation errors could be made arbitrarily small by choosing sufficient neurons. This enables us to deal with control problems for complex nonlinear systems [8–13]. In addition to system modeling and control, NN has also been successfully applied in various fields such as learning [14–17], pattern recognition [18], and signal processing [19]. And NN has been extensively used for functions approximation, such as to compensate for the effect of unknown dynamics in nonlinear systems [20–31]. The NN control has been proved to be effective for controlling uncertain nonlinear systems and demonstrated superiority in many aspects.

Recently, the researchers have focused on the study of robotics for its increasing importance in both industrial

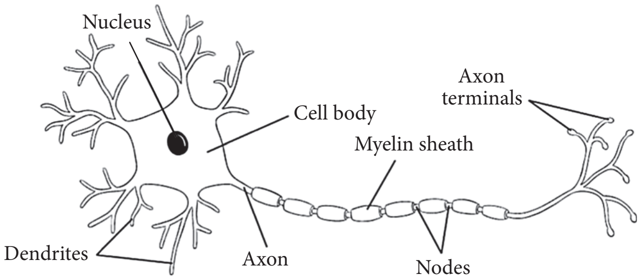


Figure 1: An example of mammalian neuron (modified from [32]).

applications and daily life [33–38]. Many advanced robots such as YuMi made by ABB, Baxter made by Rethink, and Rolins' Justin developed by German Aerospace Agency (DLR) have also been widely allocated. The robots manipulator system is characterized with high-nonlinearity, strong coupling, and time-varying dynamics, thus controlling a robot with not only positioning accuracy, but also enough flexibility to complete a complex task became an interesting yet challenge work. To achieve a high performance control, dynamics of the robot should be known in advance. However, in practice, the robot dynamic model is often rarely known due to the complex robot mechanism, let alone various uncertainties such as parametric uncertainties or modeling errors existing in the robot dynamics. Therefore, advance control algorithm is imperative for next-generation robots. Thanks to the universal approximation and learning ability, the NN has been widely applied in robot control with various applications. The combination of NN and robot controller can provide possible solutions for complex manipulation tasks, for example, robot control with unknown dynamics and robot control with unstructured environment. In this paper, we present a brief review of robot control by means of neural network. The rest of the paper is organized as follows.

After the introduction, in Section 2, we present preliminaries of several popular neural network structures, such as RBFNN and CMAC NN. Section 3 introduces a number of theoretical developments of NN in the fields of adaptive control, optimization, and evolutionary computing. In addition, Section 4 revisits the robot neural network control with the applications in manipulation, human-robot interaction, and robot cognitive control. Section 5 gives a brief discussion about the neural network control and its future research.

2. Preliminaries of Neural Networks

In this section, we will introduce several types of NN structure, which are popularly employed in the control engineering.

2.1. Radial Basis Function Neural Network (RBFNN) [14, 15]. A basic architecture of RBFNN network is shown in Figure 2, which consists of three layers, namely, input layer, hidden layer, and output layer. In the input layer, the NN inputs are applied. In hidden layer, the data is transformed from input space to hidden space, which is always with a higher dimension. The RBFNN can be used to approximate any continuous vector function, for example, $F(Z)$:

$$\widehat{F}(Z) = W^{T} S(Z), \tag{1}$$

where $\widehat{F}(Z)$ is the estimation of $F(Z)$ and Z is NN inputs vector. $\widehat{W} = [\widehat{W}_1, \widehat{W}_2, \ldots, \widehat{W}_n] \in R^{n\times l}$ is the estimation of NN optimal weight, $S(Z) = [s_1(Z), s_2(Z), \ldots, s_l(Z)]^T$ is the regressor, and l denotes the number of NN nodes. Generally, the regressor could be chosen as a Gaussian radical basis function as follows:

$$s_i(\|Z - u_i\|) = \exp\left[\frac{-(Z-u_i)^T (Z-u_i)}{\sigma_i^2}\right], \tag{2}$$

where u_i $(i = 1, \ldots, l)$ are distinct points in state space and σ_i is the width of Gaussian membership function. It has been well recognized that, using the powerful approximate ability of the RBFNN, we can approximate any continuous nonlinear function over a compact set as

$$F(Z) = W^{*T} S(Z) + \varepsilon, \tag{3}$$

where W^* is the optimal weight vector and ε is the approximate error.

2.2. Cerebellar Model Articulation Controller (CMAC) NN [39]. There has been a predominant tendency to study the learning and control techniques of robots by exploring the principles of biological systems. This is because the biological creatures, mechanisms, and underlying principles are likely to bring novel ideas to improve control performance of the robot in a complex environment. In 1972, Albus proposed a learning mechanism that imitates the structure and function of the cerebellum, called cerebellar model articulation controller (CMAC), which is designed based on a cerebellum neurophysiological model [40]. In comparison to the backpropagation neural network, the CMAC NN was adopted widely in modeling and control of robots system for its rapid learning speed, simple structure, insensitivity of data sequence, and easy implementation [39, 41].

Figure 3 shows the basic structure of the CMAC neural network. The CMAC could be used to approximate the unknown continuous function, $G(Z) = [f_1(Z), f_2(Z), \ldots, f_n(Z)]$, where $Z \in R^m$ denotes the m dimensional inputs

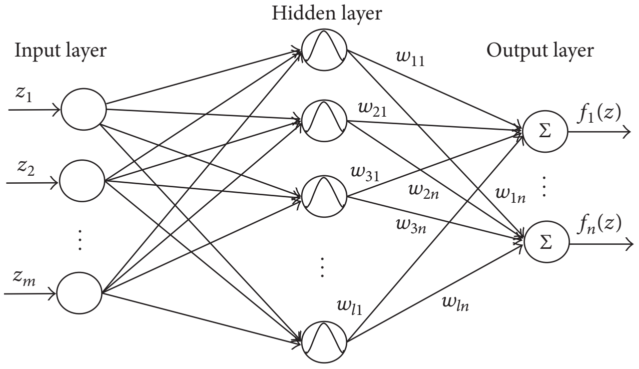


FIGURE 2: Structure of the RBFNN.

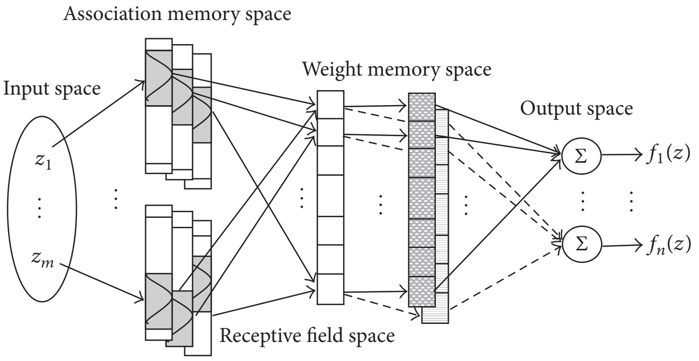


FIGURE 3: Structure of a CMAC neural network.

space. As shown in Figure 3, two components are involved in the CMAC neural network to determine the value of the approximated nonlinear function $G(Z)$:

$$\begin{aligned} R &: Z \longrightarrow C \\ P &: C \longrightarrow F, \end{aligned} \tag{4}$$

where

Z is m-dimensional input space

F is n-dimensional output space

C is N_c-dimensional association space

and $R(\cdot)$ denotes the mapping from the input vector to the association space; that is, $\alpha = R(Z)$. The outputs are computed through $P(\alpha)$, by using a projection of the association vector α onto a weights vector, such that

$$f = P(\alpha) = W^T \alpha, \tag{5}$$

It should be noted that $R(Z)$ can be represented by a multidimensional receptive filed function such that each point in input Z is assigned with an activation value. The receptive-field basis functions of the association vector could be chosen as Gaussian functions as follows:

$$h_{ik}\left(\|z_i - u_{ik}\|\right) = \exp\left[\frac{-\left(z_i - u_{ik}\right)^2}{\vartheta_{ik}^2}\right], \qquad k = 1, 2, \ldots, l, \tag{6}$$

where l is number of blocks of the associate space, h_{ik} denotes the kth block associated with the input z_i, u_{ik} denotes the receptive field's center, and ϑ_{ik} is the variance of Gaussian function. Then, the multidimensional receptive-field function can be described as

$$S(Z) = [s_1, s_2, \ldots, s_n]^T, \tag{7}$$

where $s_k(Z, u_k, \vartheta_k) = \prod_{i=1}^{m} h_{ik}(z_i)$, $u_k = [u_{1k}, u_{2k}, \ldots, u_{mk}]^T$, and $\vartheta_k = [\vartheta_{1k}, \vartheta_{2k}, \ldots, \vartheta_{mk}]^T$. The following property shows the approximation ability provided by the CMAC neural network.

Lemma 1. *For a continuous nonlinear function $F(Z)$, there exists an ideal weight value W^*, such that $F(Z)$ could be uniformly approximated by a CMAC with the multiplication of the optimal weights W^* and the associate vector $S(Z)$ as*

$$F(Z) = W^T S(Z) + \varepsilon, \tag{8}$$

where ε is the NN construction errors and satisfied $\|\varepsilon\| \leq \varepsilon_N$ and ε_N is a small bounded positive value.

3. Theoretical Developments

3.1. Adaptive Neural Control. During the past two decades, various neural networks have been incorporated into adaptive control for nonlinear systems with unknown dynamics. In [42], a multiplayer discrete-time neural network controller was constructed for a class of multi-input multioutput (MIMO) dynamical systems, where NN weights were trained using an improved online tuning algorithm. An adaptive NN output feedback control was proposed to control two classes of discrete-time systems in the presence of unknown control directions [4]. A robust adaptive neural controller was developed for a class of strict-feedback systems in [43], where a Nussbaum gain technique was employed to deal with unknown virtual control coefficients. A dynamic recurrent NN was employed for construction of an adaptive observer with online turned weights parameters in [44] and to deal with the time-delay of a class of nonlinear dynamical systems in [45]. The time-delay of strict-feedback nonlinear systems was also addressed by using NN control with proper designed Lyapunov-Krasovskii functions in [46]. For a class of unknown nonlinear affine time-delay systems, an adaptive control scheme was proposed by constructing two high-order NNs for identifying system uncertainties [47]. This idea has been further extended to affine nonlinear systems with input time-delay in [48].

It should be noticed that, piecewise continuous functions such as frictions, backlash, and dead-zone are widely existed in industrial plants. Other than continuous nonlinear function, the approximation of these piecewise functions is more challenging since the NN's universal approximation only holds for continues functions. To approximate these piecewise continuous functions, a novel NN structure was designed by involving a standard activation function and a jump approximation basis function [49]. In [47], a CMAC NN was employed for the closed-loop control of nonlinear dynamical systems with rigorous stability analysis, and in [50] a robust adaptive neural network control scheme was developed for cooperative tracking control of higher-order nonlinear systems.

The adaptive NN control scheme was also proposed for pure-feedback systems. In [51], a high-order sliding mode observer was proposed to estimate the unknown system states while two NNs were constructed to deal with approximation errors and the unknown nonlinearities, respectively. In comparison to the conventional control design for pure-feedback systems, the state-feedback control was achieved without using the backstepping technique. In [52], a neural control framework was proposed for nonlinear servo mechanism to guarantee both the steady-state and transient tracking performance. In this work, a prescribed performance function was employed in an output error transformation, such that the tracking performance can be guaranteed by the regulation control of the outputs. In [53], an adaptive neural control was also designed for a class of nonlinear systems in the presence of time-delays and input dead-zone, and high-order neural networks were employed to deal the unknown uncertainties. In this work, a salient feature lies in the fact that only the norm of the NNs' weights (a scalar) needs to be online updated, such that the computational efficiency in the online implementation could be significantly improved. In [54], the authors developed a neural network based feedforward control to compensate for the nonlinearities and uncertainties of a dynamically substructured system consisting of both numerical and physical substructures, where an adaptive law with a new leakage term of NN weights error information was developed to achieve improved convergence. An experiment on a quasi-motorcycle testing rig validated the efficacy of this control strategy. In [55], a neural dynamic control was incorporated into the strict-feedback control of a class of unknown nonlinear systems by using the dynamic surface control technique. For a class of uncertain nonlinear systems with unknown hysteresis, NN was used for compensation of the nonlinearities [56]. In [57], to deal with unknown nonsymmetrical input saturations of unknown nonaffine systems, NNs were used in the state/output feedback control based on the mean value theorem and the implicit function. To avoid using the backstepping synthesis, a dynamic surface control scheme was designed by combining the NN with a nonlinear disturbance observer [58].

3.2. NN Based Adaptive Dynamic Programming. In addition to adaptive control, neural networks have also been adopted to solve the optimization problem for nonlinear systems. In convention optimal control, the dynamic programming method was widely used. It aims to minimize a predefined cost function, such that a sequence of optimal control inputs could be derived. However, the cost function is usually difficult to online calculate due to the computation complexity in obtaining the solution of the Hamilton-Jacobi-Bellman (HJB) equation. Therefore, an adaptive/approximate dynamic programming (ADP) technique was developed in [59], where a NN was trained to estimate the cost function and then to derive solutions for the ADP. Generally, the ADP has several different synonyms, including approximate dynamic programming, heuristic dynamic programming (HDP), critic network, and reinforcement learning (RL) [60–62]. Figure 4 shows the basic framework of the HDP with a critic-actor structure. In [63], a discrete-time HJB equation was solved using an NN based HDP algorithm to derive the optimal control of nonlinear discrete-time systems. In [64], three neural networks were constructed for an iterative ADP, such that optimal feedback control of a discrete-time affine nonlinear system could be realized. In [65], a globalized dual heuristic programming was presented to address the optimal control of discrete-time systems. In each iteration, three neural networks were used to learn the cost function and the unknown nonlinear systems. In [66], a reference network combining with an action network and a critic network was introduced in the ADP architecture to derive an internal goal representation, such that the learning and optimization process could be facilitated. The reference network has also been introduced in the online action-dependent heuristic dynamic programming by employing a dual critic network framework. A policy iteration algorithm was introduced for infinite horizon optimal control of nonlinear systems using ADP in [67]. In [68], a reinforcement learning method was introduced for the stabilizing control of uncertain nonlinear

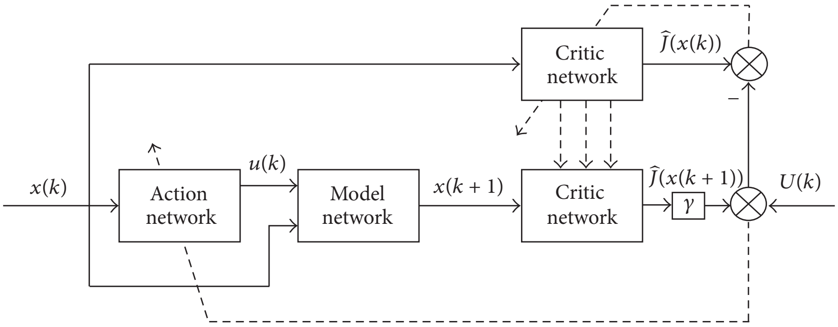


Figure 4: An overview of the HDP structure.

systems in the presence of input constraints. By using this RL-based controller, a constrained optimal control problem was solved with construction of only one critic neural network. In [69], an ADP technique for online control and learning of a generalized multiple-input-multiple-output (MIMO) system was investigated. In [70], an adaptive NN based ADP control scheme was presented for a class of nonlinear systems with unknown dynamics. The optimal control law was calculated by using a dual neural network scheme with a critic NN and an identifier NN. Particularly, parameters estimation error was used to online identify the learning weights to achieve the finite-time convergence. Optimal tracking control for a class of nonlinear systems was investigated in [71], where a new "identifier-critic" based ADP framework was proposed.

3.3. Evolutionary Computing. In addition to the capacity of approximation and optimization of the NN, there has been also a great interest in using the evolutionary approaches to train the neural networks. With the evolvement of NN architectures, learning rules, connection weights, and input features, an evolutionary artificial neural network (EANN) was designed to provide superior performance in comparison to conventional training approaches [72]. A literature review of the EANN was given in [73], where the evolution strategies such as feedforward artificial NN and genetic algorithms (GA) have been introduced for the EANNs. In [72], several EANN frameworks were introduced by embedding the evolution algorithms (EA) to evolve the NN structure. In [74], an EPNet evolution system was proposed for evolving the feedforward NN based on Fogel's evolutionary programming (EP) method, which could improve the NN's connection weights and architectures at the same time as well as decrease noise in the fitness evaluation. Good generalization ability of the evolved NN has been constructed and verified in the experiments. In [75], a GA based technique has been employed to train the NNs in direct neural control systems such that the NN architectures could be optimized. A deficiency of the EANN is that the optimization process would often result in a low training speed. To overcome this problem and facilitate adaptation processes, a hybrid multiobjective evolutionary method was developed in [76], where the singular-value-decomposition (SVD) technique was employed to choose the necessary neurons number in the training of a feedforward NN. The evolutionary approach was applied to identify a grey-box model with a multiobjective optimization between the clearly known practical systems and approximated nonlinear systems [77]. Applications of evolutionary algorithms for robotic navigation have been introduced and investigated in [78]. A survey of machine learning technique was reported in [79], where several methods to improve the evolutionary computation were reviewed.

The evolution algorithms have been employed in many aspects for evolvements of NNs, such as to train the NN connection weights or to obtain near-optimal NN architectures, as well as adapting learning rules of NNs to their environment. In a word, the evolution algorithms provide NNs with the ability of learning to learn and also to build the relationship between evolution and learning, such that the EANN could perform favorable ability to adapt to changes of the dynamic environment.

4. Applications in Robots

4.1. NN Based Robotic Manipulator Control. Generally speaking, the control methods for robot manipulators can be roughly divided into two groups, model-free control and model based control. For the model-free control approaches like proportional-integral-derivative (PID) control, satisfactory control performance may not be guaranteed. In contrast, the model based control approaches exhibit better control behavior but heavily depend on the validity of the robot model. In practice, however, a perfect robotic dynamic model is always not available due to the complex mechanisms and uncertainties. Additionally, the payload may be varied according to different tasks, which makes the accurate dynamics model hard to be obtained in advance. To solve such problems, the NN approximation-based control methods have been used extensively in applications of robot manipulator control. A basic structure of the adaptive neural network control for robot manipulator is shown in Figure 5. Consider a dynamic model of a robot manipulator given as follows [80]:

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) = \tau, \tag{9}$$

where $M(q)$, $C(\dot{q}, q)$, and $G(q)$ are the inertial matrix, Coriolis matrix, and gravity vector, respectively. Then NN control design could be given as follows:

$$\tau_d = -K_1 e_1 - K_2 e_2 + \widehat{W} S(Z), \tag{10}$$

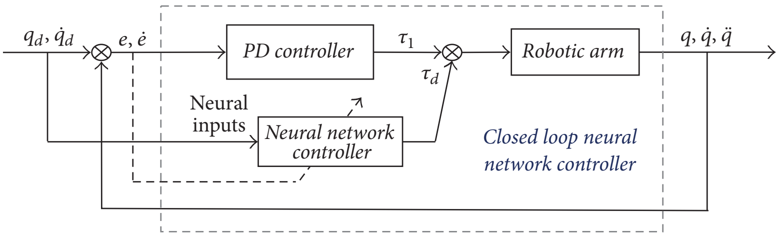


Figure 5: A basic structure of the adaptive NN robot control.

where e_1 is the tracking error, e_2 is the velocity tracking error, $\widehat{W}S(Z)$ is the NN controller with $\widehat{W}$ being the weights matrix and $S(Z)$ being the NN regressor vector, and K_1 and K_2 are control gains specified by the designer.

From (10), we can see that the robot controller consists of a PD-like controller and a NN controller. In traditional model based controllers, the dynamic model of the robot could be regarded as a feedforward to address the effect caused by the robot motion. In practice, however, $M(q)$, $C(q, q)$, and $G(q)$ may not be known. Therefore, the NNs are used to approximate the unknown dynamics $f = M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q)$ and to improve the performance of the system via the online estimation. To adapt the NN weights, adaptive laws are designed as follows:

$$\dot{\widehat{W}} = \Gamma\left(Se - \sigma\widehat{W}\right), \tag{11}$$

where Γ and σ are specified positive parameters. The last term of right-hand side of (11) is the sigma modification, which is used to enhance the convergence and robustness of the parameters adaptation.

In [80], a NN based share control method was developed to control a teleoperated robot with environmental uncertainties. In this work, the RBFNN was constructed to compensate for the unknown dynamics of the teleoperated robot. Particularly, a shared control strategy was developed into the controller to achieve the automatic obstacle avoidance combining with the information of visual camera and the robot body, such that the obstacle could be successfully avoided and the operator could focus more on the operated task rather than the environment to guarantee the stability and manipulation. In addition, error transformations were integrated into the adaptive NN control to guarantee the transient control performance. It was shown that, by using the NN technique, the control performance in both kinematic level and dynamic level of the teleoperated robot was enhanced. In [81], an extreme learning machine (ELM) based control strategy was proposed for uncertain robot manipulators to identify both the elasticity and geometry of an object. This ELM was applied to deal with the unknown nonlinearity of the robot manipulator to enhance the control performance. Particularly, by utilizing ELM, the proposed controller could guarantee that the robot dynamics follow a reference model, such that the desired set point and the feedforward force could be updated to estimate the geometry and stiffness of the object. As a result, the reference model could be exactly matched with a limited number of iterations.

In [82], the NN controller was also employed to control a wheel inverted pendulum, which has been decomposed into two subsystems, a fully actuated second-order planar moving subsystem and a passive first-order pendulum subsystem. Then the RBFNN was employed to compensate for the uncertain dynamics of the two subsystems by using its powerful learning ability, such that the enhanced control performance could be realized by using the NN learning. In [83], a global adaptive neural control was proposed for a class of robot manipulators with finite-time convergence learning performance. This control scheme employed a smooth switching mechanism combining with a nominal neural network controller and a robust controller to ensure global uniform ultimately bounded stability. The optimal weights were obtained by the finite-time estimation algorithm such that, after the learning process, the learning weights could be reused next time for repeated tasks. The global NN control mechanism has been further extended to the control of dual arm robot manipulator in [84], where knowledge of both robot manipulator and the grasping object is unavailable in advance. By integrating prescribed functions into the design of controller, the transient performance of the dual arm robot control was regularly guaranteed. The NN was also employed to deal with synchronization problem of multiple robot manipulators in [85], where the reference trajectories are only available for part of the team members. By using the NN approximation controller, the robot has shown better control performance with enhanced transient performance and enhanced robustness. A RBFNN was constructed to compensate for the nonlinear terms of a five-bar manipulator based on an error transformation function [86]. The NN control was also applied in the robot teleoperation control [87, 88]. Moreover, a NN approximation technique was employed to deal with the unknown dynamics, kinematics, and actuator properties in the manipulator tracking control [89].

4.2. NN Based Robot Control with Input Nonlinearities. Another challenge of the robot manipulator is that the input nonlinearities such as friction, dead-zone, and actuator saturation may inevitably exist in the robot systems. These input nonlinearities may lead to larger tracking errors and degeneration of the control performance. Therefore, a number of works have been proposed to handle the nonlinearities by utilizing the neural network design. A neural adaptive controller was designed to deal with the effect of input saturation of the robot manipulator in [90] as follows:

$$\begin{aligned}\tau = {} & -z_1 + \widehat{W}_D S_D\left(Z_D\right)\alpha_1 + \widehat{W}_C S_C\left(Z_C\right)\alpha_1 \\ & + \widehat{W}_G S_G\left(Z_G\right) + K_p\left(z_2 + \xi\right) + K_r \operatorname{sgn}\left(z_2\right),\end{aligned} \tag{12}$$

where z_1 is the robot position tracking error, z_2 is the velocity tracking error, and α_1 is an auxiliary controller. $\widehat{W}_D$, $\widehat{W}_C$, and $\widehat{W}_G$ are the NN weights, $S_D(Z_D)$, $S_C(Z_C)$, and $S_G(Z_G)$ are the NN regressor vectors, and K_p and K_r are control gains specified by the designer. ξ is an auxiliary system designed to reduce the effect of the saturation with $\dot{\xi}$ defined as follows.

$$\begin{aligned}&\dot{\xi} \\ &= \begin{cases} -K_\xi \xi - \dfrac{\left|z_2^T \Delta\tau\right| + (1/2)\,\Delta\tau^T \Delta\tau}{\|\xi\|^2}\xi + \Delta\tau & \|\xi\| \geq \mu \\ 0 & \|\xi\| < \mu, \end{cases}\end{aligned} \tag{13}$$

where $\Delta\tau$ is the torque error caused by saturation, and K_ξ is a small positive value. To update the NN weights, adaptive laws are designed as follows:

$$\begin{aligned}\dot{\widehat{W}}_D &= \Gamma_{Dk}\left(S_{Dk}\alpha_1 e_{2k} - \sigma_{Dk}\widehat{W}_{Dk}\right) \\ \dot{\widehat{W}}_C &= \Gamma_{Ck}\left(S_{Ck}\alpha_1 e_{2k} - \sigma_{Ck}\widehat{W}_{Ck}\right) \\ \dot{\widehat{W}}_G &= \Gamma_{Gk}\left(S_{Gk}\alpha_1 e_{2k} - \sigma_{Gk}\widehat{W}_{Gk}\right),\end{aligned} \tag{14}$$

where Γ_{Dk}, Γ_{Ck}, and Γ_{Gk} are specified positive parameters and σ_{Dk}, σ_{Ck}, and σ_{Gk} are positive parameters.

In [91], an adaptive neural network controller was constructed to approximate the input dead-zone and the uncertain dynamics of the robotic manipulator, while the output constraint was also considered in the feedback control. In [92], the NN was applied for the estimation of the unknown model parameters of a marine surface vessel and in [93] the full-state constraint of an n-link robotic manipulator was achieved by using the NN control. The NN controller was also constructed for flexible robotic manipulators to deal with the vibration suppression based on a lumped spring-mass model [94] while in [95], two RBFNNs were constructed for flexible robot manipulators to compensate for the unknown dynamics and the dead-zone effect, respectively.

The NN has also been used in many important industrial fields, such as autonomous underwater vehicles (AUVs) and hypersonic flight vehicle (HFV). In [96], the NN has been constructed to deal with the attitude of AUVs in the presence of input dead-zone and uncertain model parameters. In [97], the adaptive neural control was employed to deal with underwater vehicle control in discrete-time domain encountered with the unknown input nonlinearities, external disturbance, and model uncertainties. Then the reinforcement learning was applied to address these uncertainties by using a critic NN and an action NN. The hypersonic flight vehicle control was investigated in [98] where the aerodynamic uncertainties and unknown disturbances were addressed by a disturbance observer based NN. In [99], a neural learning control was embedded in the HFV controller to achieve the global stability via a switching mechanism and a robust controller.

4.3. NN Based Human-Robot Interaction Control. Recently, there is a predominant tendency to employ the robots in the human-surrounded environment, such as household services or industrial applications, where humans and robots may interact with each other directly. Therefore, interaction control has become a promising research field and has been widely studied. In [100], a learning method was developed such that the dynamics of a robot arm could follow a target impedance model with only knowledge of the robotic structure (see Figure 6).

The NN was further employed in robot control in interaction with an environment [101], where impedance control was achieved with the completely unknown robotic dynamics. In [102], a learning method was developed such that the robot was able to adjust the impedance parameters when it interacted with unknown environments. In order to learn optimal impedance parameters in the robot manipulator control, an adaptive dynamic programming (ADP) method was employed when the robot interacted with unknown time-varying environments, where NNs were used for both critic and actor networks [103]. The ADP was also employed for coordination of multirobots [104], in which possible disagreement between different manipulators was handled and dynamics of both robots and the manipulated object were not required to be known.

In this work, the controller consists of two parts, a critic network which was used to approximate the cost function, and an actual NN which was designed to control the robot. The critic NN is designed as follows [104]:

$$\widehat{\Upsilon}(t) = \widehat{W}_C S_C\left(Z_C\right), \tag{15}$$

where $Z_C = \xi$, $\xi = [\dot{x}_o^T, z^T, x_o^T]^T$ with x_o being the position of the object and z being the tracking error, $\widehat{W}_C$ is the NN weight, and S_C is the regressor vector.

The critic NN is used to approximate a cost function $c(t) = \xi^T Q \xi + u^T R u$, where u denotes the control input, and Q and R are positive definite matrix. Since the control objective is to minimize the control effort, the adaptation law is designed as

$$\dot{\widehat{W}}_c = -\sigma_c \frac{\partial E_c}{\partial \widehat{W}_c} = \sigma_c\left(c - \widehat{W}_c^T \dot{S}_C\right)\dot{S}_C, \tag{16}$$

where σ_c is the learning rate and $E_c = (1/2)(c - \widehat{W}_c^T \dot{S}_C)^2$.

On the other hand, the actual NN control is designed to control the robot as

$$u = \widehat{W}_a^T S_a\left(Z_a\right) - e - K_2 e_v, \tag{17}$$

where $\widehat{W}_a^T S_a(Z_a)$ could learn the dynamics of the robot, with $\widehat{W}_a$ being the NN weight and S_a being the regressor vector. e and e_v are the position and velocity tracking errors, respectively, and K_2 is the control gain.

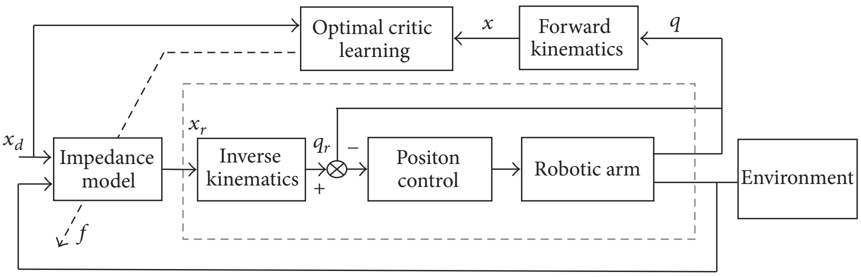


Figure 6: Block of the learning impedence control.

Since the control objective is to guarantee the estimation of both robot dynamics and the cost function $\widehat{\Upsilon}(t)$, the adaptive law is selected as follows:

$$\dot{\widehat{W}}^{T}_{a,i} = -\sigma_a \left(\widehat{W}^{T}_{a,i} S_a + k_r \widehat{\Upsilon}\right) S_a, \quad (18)$$

where σ_a and k_r are positive constants.

On the other hand, as a fundamental element of the next-generation robots, the human-robot collaboration (HRC) has been widely studied by roboticists and NN is employed in HRC with its powerful learning ability. In [105], the NNs were employed to estimate the human partner's motion intention in human-robot collaboration, such that the robot was able to actively follow its human partner. To adjust the robot's role to lead or to follow according to the human's intention, game theory was employed for fundamental analysis of human-robot interaction and an adaptation law was developed in [106]. Policy iteration combining with NN was adopted to provide a rigorous solution to the problem of the system equilibrium in human-robot interaction [107].

4.4. NN Based Robot Cognitive Control. According to the predictive processing theory [108], the human brain is always actively anticipating the incoming sensorimotor information. This process exists because the living beings exhibit latencies due to neural processing delays and a limited bandwidth in their sensorimotor processing. To compensate for such a delay, in human brain, neural feedback signals (including lateral and top-down connections) modulate the neural activities via inhibitory or excitatory connections by influencing the neuronal population coding of the bottom-up sensory-driven signals in the perception-action system. Similarly, in robotic systems, it is claimed that such a delay and a limited bandwidth also can be compensated by the predictive functions learnt by recurrent neural models. Such a learning process can be done via only visual processing [109] or in the loop of perception and action [110].

Based on the hierarchical sensorimotor integration theory, which advocates that action and perception are intertwined by sharing the same representational basis [111], the representation on different levels of sensory perception does not explicitly represent actions; instead, there is an encoding of the possible future percept which is learnt from prior sensorimotor knowledge.

In the Bayesian, once this perception and action links have been established after learning, these perception-action associations in this architecture allow the following operations.

First, these associations allow predicting the perceptual outcome of given actions by means of the forward models (e.g., Bayesian model). It can be written as

$$P(E \mid A, I) \propto P(A \mid E) P(E \mid I), \quad (19)$$

where E estimates the upcoming perception evidence given an executed action A and other prior information you have already known in I. The term $P(E \mid A, I)$ suggests that a prelearnt model representing the possibility of a motor action A will be executed given that a (possible) resulting sensory evidence E is perceived (backward computation).

Second, these associations allow selecting an appropriate movement given an intended perceptual representation. From the backward computations introduced in the following equation, a predictive sensorimotor integration occurs:

$$P(A \mid E, G) \propto P(E \mid A) P(A \mid G), \quad (20)$$

where A indicates a particular action selected given the (intended) sensory information E and a goal G. Here we assume that one's action is only determined by the current sensory input and the goal.

In terms of its hierarchical organization, it also allows this operation: with bidirectional information pathways, a low level perception representation can be expressed on a higher level, with a more complex receptive field, and vice versa $e_{\text{low}} \Leftrightarrow e_{\text{high}}$. This can be realized by the bidirectional deep architectures such as [112]. Conceptually, these operations can be achieved by extracting statistical regularity shown in Figure 7.

Since both perception and action processes can be seen as temporal sequences, from the mathematical perspective, the recurrent networks are Turing-Complete and have a learning capacity to learn time sequences with arbitrary length [113], if properly trained. Furthermore, such recurrent connections can be placed in a hierarchical way in which the prediction functions on different layers attempt to predict the nonlinear time-series in different time-scales [114]. From this point, the recurrent neural network with parametric bias units (RNNPB) [115] and multiple time-scale recurrent neural networks (MTRNN) [116] were applied to predict sequences by understanding them in various temporal levels.

The difference of the temporal levels controls the properties of the different levels of the presentation in the deep

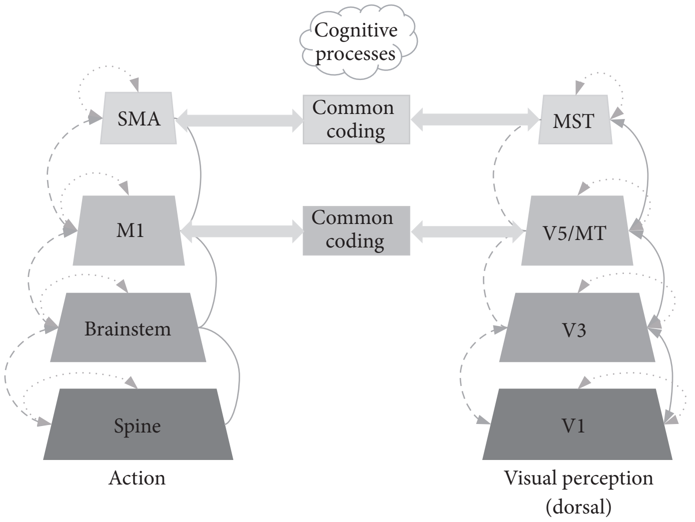


Figure 7: The process of cognitive control.

recurrent network. For instance, in the MTRNN network [112], the learning of each neuron follows the updating rule of classical firing rate models, in which the activity of a neuron is determined by the average firing rate of all the connected neurons. Additionally, the neuronal activity is also decaying over time following an updating rule of leaky integrator model. Assuming the i-th MTRNN neuron has the number of N connections, the current membrane potential status of a neuron can be defined by both the previous activation and the current synaptic inputs:

$$u_{i,t+1} = \left(1 - \frac{1}{\tau_i}\right) u_{i,t} + \frac{1}{\tau_i}\left[\sum w_{i,j} x_{j,t}\right], \qquad (21)$$

where $w_{i,j}$ represents the synaptic weight from the j-th neuron to the i-th neuron, $x_{j,t}$ is the activity of j-th neuron at t-th time-step, and τ is the time-scale parameter which determines the decay rate of this neuron: a larger τ means their activities change slowly over time compared with those with a smaller time-scale parameter τ.

In [117], the concepts of predictive coding were discussed in detail, where the learning, generation, and recognition of actions can be conducted by means of the principle of prediction error minimization. By using the predictive coding, the RNNPB and MTRNN are capable for both generating own actions and recognizing the same actions performed by others. Recently, the study on neurorobotics experiments has shown that the dynamic predictive coding scheme can be used to address fluctuations in temporal patterns when training a recurrent neural network (RNN) model [118]. This predictive coding scheme enables organisms to predict perceptual outcomes based on current intentions of actions to the external environment and to forecast perceptual sequences corresponding to given intention states [118].

Based on this architecture, two-layer RNN models were utilized to extract visual information [119] and to understand intentions [120] or emotion status [121] in social robotics; three-layer RNN models were used to integrate and understand multimodal information for a humanoid iCub robot [112, 122]. Moreover, the predictive coding framework has been extended to variational Bayes predictive coding MTRNN, which can arbitrate between deterministic model and probabilistic model by setting a metaparameter [123]. Such extension could provide significant improvement in dealing with noisy fluctuated sensory inputs which robots are expected to experience in more real world setting. In [124], a MTRNN was employed to control a humanoid robot and experimental results have shown that, by using only partial training data, the control model can achieve generalization by learning in a lower feature perception level.

The hierarchical structure of RNN exhibits a great learning capacity to store multimodal information which is beneficial for the robotic systems to understand and to predict in a complex environment. As the future models and applications, the state-of-the-art deep learning techniques or the motor actions of robotic systems can be further integrated into this predictive architecture.

5. Conclusion

In summary, great achievements for control design of nonlinear system by means of neural networks have been gained in the last two decades. Despite the impossibility in identifying or listing all the related contributions in this short review, efforts have been made to summarize the recent progress in the area of NN control and its particular applications in the robot learning control, the robot interaction control, and the robot recognition control. In this paper, we have shown that significant progress of NN has been made in control of the nonlinear systems, in solving the optimization problem, in approximating the system dynamics, in dealing with the input nonlinearities, in human-robot interaction, and in the pattern recognition. All these developments accompany not only the development of techniques in control and advanced manufactures, but also theatrical progress in constructing and developing the neural networks. Although huge efforts have been made to embed the NN in practical control systems, there is still a large gap between the theory and practice. To improve the feasibility and usability, the evolutionary computing theory has been proposed to train the NNs. It can automatically find a near-optimal NN architecture and allow a NN to adapt its learning rule to its environment. However, the complex and long training process of the evolutionary algorithms deters their practical applications. More efforts need to be made to evolve the NN architecture and NN learning technique in the control design. On the other hand, in human brain, the neural activities are modulated via inhibitory or excitatory connections by influencing the neuronal population coding of the bottom-up sensory-driven signals in the perception-action system. In this sense, how to integrate the sensor-motor information into the network to make NNs more feasible to adapt to the environment and to resemble the capacity of the human brain deserves further investigations.

In conclusion, a brief review on neural networks for the complex nonlinear systems is provided with adaptive neural control, NN based dynamic programming, evolution computing, and their practical applications in the robotic fields. We believe this area may promote increasing investigations in both theories and applications. And emerging topics, like deep learning [125–128], big data [129–131], and cloud computing, may be incorporated into the neural network control for complex systems; for example, deep neural networks could be used to process massive amounts of unsupervised data in complex scenarios, neural networks can be helpful in reducing the data dimensionality, and the optimization of NN training may be employed to enhance the learning and adaptation performance of robots.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Nature Science Foundation (NSFC) under Grant 61473120, Guangdong Provincial Natural Science Foundation, 2014A030313266, International Science and Technology Collaboration, Grant 2015A050502017, Science and Technology Planning Project of Guangzhou, 201607010006, State Key Laboratory of Robotics and System (HIT) Grant SKLRS-2017-KF-13, and the Fundamental Research Funds for the Central Universities.

References

[1] F. Gruau, D. Whitley, and L. Pyeatt, "A comparison between cellular encoding and direct encoding for genetic neural networks," in *Proceedings of the 1st annual conference on genetic programming*, pp. 81–89, 1996.

[2] H. de Garis, "An artificial brain ATR's CAM-Brain Project aims to build/evolve an artificial brain with a million neural net modules inside a trillion cell Cellular Automata Machine," *New Generation Computing*, vol. 12, no. 2, pp. 215–221, 1994.

[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115–133, 1943.

[4] C. Yang, S. S. Ge, C. Xiang, T. Chai, and T. H. Lee, "Output feedback NN control for two classes of discrete-time systems with unknown control directions in a unified approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 11, pp. 1873–1886, 2008.

[5] R. Kumar, R. K. Aggarwal, and J. D. Sharma, "Energy analysis of a building using artificial neural network: A review," *Energy and Buildings*, vol. 65, pp. 352–358, 2013.

[6] C. Alippi, C. De Russis, and V. Piuri, "A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 33, no. 2, pp. 259–268, 2003.

[7] E. M. Azoff, *Neural network time series forecasting of financial markets*, John Wiley & Sons, Inc, 1994.

[8] I. Fister, P. N. Suganthan, J. Fister et al., "Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution," *Nonlinear Dynamics*, vol. 84, no. 2, pp. 895–914, 2016.

[9] Y. Zhang, S. Li, and H. Guo, "A type of biased consensus-based distributed neural network for path planning," *Nonlinear Dynamics*, vol. 89, no. 3, pp. 1803–1815, 2017.

[10] X. Shi, Z. Wang, and L. Han, "Finite-time stochastic synchronization of time-delay neural networks with noise disturbance," *Nonlinear Dynamics*, vol. 88, no. 4, pp. 2747–2755, 2017.

[11] P. He and Y. Li, "H∞ synchronization of coupled reaction-diffusion neural networks with mixed delays," *Complexity*, vol. 21, no. S2, pp. 42–53, 2016.

[12] Z. Tu, J. Cao, A. Alsaedi, F. E. Alsaadi, and T. Hayat, "Global Lagrange stability of complex-valued neural networks of neutral type with time-varying delays," *Complexity*, vol. 21, no. S2, pp. 438–450, 2016.

[13] C. Wang, S. Guo, and Y. Xu, "Formation of autapse connected to neuron and its biological function," *Complexity*, vol. 2017, Article ID 5436737, 9 pages, 2017.

[14] J. D. J. Rubio, I. Elias, D. R. Cruz, and J. Pacheco, "Uniform stable radial basis function neural network for the prediction in two mechatronic processes," *Neurocomputing*, vol. 227, pp. 122–130, 2017.

[15] J. d. Rubio, "USNFIS: Uniform stable neuro fuzzy inference system," *Neurocomputing*, vol. 262, pp. 57–66, 2017.

[16] Q. Liu, J. Yin, V. C. M. Leung, J.-H. Zhai, Z. Cai, and J. Lin, "Applying a new localized generalization error model to design neural networks trained with extreme learning machine," *Neural Computing and Applications*, pp. 1–8, 2014.

[17] R. J. de Jesús, "Interpolation neural network model of a manufactured wind turbine," *Neural Computing and Applications*, pp. 1–12, 2016.

[18] C. Mu and D. Wang, "Neural-network-based adaptive guaranteed cost control of nonlinear dynamical systems with matched uncertainties," *Neurocomputing*, vol. 245, pp. 46–54, 2017.

[19] Z. Lin, D. Ma, J. Meng, and L. Chen, "Relative ordering learning in spiking neural network for pattern recognition," *Neurocomputing*, 2017.

[20] J. Yu, J. Sang, and X. Gao, "Machine learning and signal processing for big multimedia analysis," *Neurocomputing*, vol. 257, pp. 1–4, 2017.

[21] T. Zhang, S. S. Ge, and C. C. Hang, “Adaptive neural network control for strict-feedback nonlinear systems using backstepping design,” *Automatica*, vol. 36, no. 12, pp. 1835–1846, 2000.

[22] S. S. Ge and T. Zhang, “Neural-network control of nonaffine nonlinear system with zero dynamics by state and output feedback,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 4, pp. 900–918, 2003.

[23] S. S. Ge, C. Yang, and T. H. Lee, “Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 9, pp. 1599–1614, 2008.

[24] F. W. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*, CRC Press, 1998.

[25] S. Jagannathan and F. L. Lewis, “Identification of nonlinear dynamical systems using multilayered neural networks,” *Automatica*, vol. 32, no. 12, pp. 1707–1712, 1996.

[26] D. Vrabie and F. Lewis, “Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems,” *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.

[27] C. Yang, S. S. Ge, and T. H. Lee, “Output feedback adaptive control of a class of nonlinear discrete-time systems with unknown control directions,” *Automatica*, vol. 45, no. 1, pp. 270–276, 2009.

[28] C. Yang, Z. Li, and J. Li, “Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models,” *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 24–36, 2013.

[29] Y. Jiang, C. Yang, and H. Ma, “A review of fuzzy logic and neural network based intelligent control design for discrete-time systems,” *Discrete Dynamics in Nature and Society*, Article ID 7217364, Art. ID 7217364, 11 pages, 2016.

[30] Y. Jiang, C. Yang, S.-L. Dai, and B. Ren, “Deterministic learning enhanced neutral network control of unmanned helicopter,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, pp. 1–12, 2016.

[31] Y. Jiang, Z. Liu, C. Chen, and Y. Zhang, “Adaptive robust fuzzy control for dual arm robot with unknown input deadzone nonlinearity,” *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1301–1314, 2015.

[32] https://stemcellthailand.org/neurons-definition-function-neurotransmitters/.

[33] M. Defoort, T. Floquet, A. Kökösy, and W. Perruquetti, “Sliding-mode formation control for cooperative autonomous mobile robots,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 11, pp. 3944–3953, 2008.

[34] X. Liu, C. Yang, Z. Chen, M. Wang, and C. Su, “Neuro-adaptive observer based control of flexible joint robot,” *Neurocomputing*, 2017.

[35] F. Hamerlain, T. Floquet, and W. Perruquetti, “Experimental tests of a sliding mode controller for trajectory tracking of a car-like mobile robot,” *Robotica*, vol. 32, no. 1, pp. 63–76, 2014.

[36] R. J. de Jesús, “Discrete time control based in neural networks for pendulums,” *Applied Soft Computing*, 2017.

[37] Y. Pan, M. J. Er, T. Sun, B. Xu, and H. Yu, “Adaptive fuzzy PD control with stable H∞ tracking guarantee,” *Neurocomputing*, vol. 237, pp. 71–78, 2017.

[38] R. J. de Jesús, “Adaptive least square control in discrete time of robotic arms,” *Soft Computing*, vol. 19, no. 12, pp. 3665–3676, 2015.

[39] S. Commuri, S. Jagannathan, and F. L. Lewis, “CMAC neural network control of robot manipulators,” *Journal of Robotic Systems*, vol. 14, no. 6, pp. 465–482, 1997.

[40] J. S. Albus, “Theoretical and experimental aspects of a cerebellar model,” *Developmental Disabilities Research Reviews*, vol. 17, pp. 93–101, 1972.

[41] B. Yang, R. Bao, and H. Han, “Robust hybrid control based on PD and novel CMAC with improved architecture and learning scheme for electric load simulator,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5271–5279, 2014.

[42] S. Jagannathan and F. L. Lewis, “Multilayer discrete-time neural-net controller with guaranteed performance,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 7, no. 1, pp. 107–130, 1996.

[43] S. S. Ge and J. Wang, “Robust adaptive neural control for a class of perturbed strict feedback nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 6, pp. 1409–1419, 2002.

[44] Y. H. Kim, F. L. Lewis, and C. T. Abdallah, “A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems,” *Automatica*, vol. 33, no. 8, pp. 1539–1543, 1997.

[45] J.-Q. Huang and F. L. Lewis, “Neural-network predictive control for nonlinear dynamic systems with time-delay,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 2, pp. 377–389, 2003.

[46] S. S. Ge, F. Hong, and T. H. Lee, “Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 499–516, 2004.

[47] J. Na, X. M. Ren, and H. Huang, “Time-delay positive feedback control for nonlinear time-delay systems with neural network compensation,” *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 34, no. 9, pp. 1196–1202, 2008.

[48] J. Na, X. Ren, C. Shang, and Y. Guo, “Adaptive neural network predictive control for nonlinear pure feedback systems with input delay,” *Journal of Process Control*, vol. 22, no. 1, pp. 194–206, 2012.

[49] R. R. Selmic and F. L. Lewis, “Neural-network approximation of piecewise continuous functions: application to friction compensation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 3, pp. 745–751, 2002.

[50] H. Zhang and F. L. Lewis, “Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics,” *Automatica*, vol. 48, no. 7, pp. 1432–1439, 2012.

[51] J. Na, X. Ren, and D. Zheng, “Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 3, pp. 370–382, 2013.

[52] J. Na, Q. Chen, X. Ren, and Y. Guo, “Adaptive prescribed performance motion control of servo mechanisms with friction compensation,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 486–494, 2014.

[53] J. Na, X. Ren, G. Herrmann, and Z. Qiao, “Adaptive neural dynamic surface control for servo systems with unknown deadzone,” *Control Engineering Practice*, vol. 19, no. 11, pp. 1328–1343, 2011.

[54] G. Li, J. Na, D. P. Stoten, and X. Ren, “Adaptive neural network feedforward control for dynamically substructured systems,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 944–954, 2014.

[55] B. Xu, Z. Shi, C. Yang, and F. Sun, "Composite neural dynamic surface control of a class of uncertain nonlinear systems in strict-feedback form," *IEEE Transactions on Cybernetics*, 2014.

[56] M. Chen and S. Ge, "Adaptive neural output feedback control of uncertain nonlinear systems with unknown hysteresis using disturbance observer," *IEEE Transactions on Industrial Electronics*, 2015.

[57] M. Chen and S. S. Ge, "Direct adaptive neural control for a class of uncertain nonaffine nonlinear systems based on disturbance observer," *IEEE Transactions on Cybernetics*, vol. 43, no. 4, pp. 1213–1225, 2013.

[58] M. Chen, G. Tao, and B. Jiang, "Dynamic surface control using neural networks for a class of uncertain nonlinear systems with input saturation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2086–2097, 2015.

[59] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, 1992.

[60] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, 2009.

[61] F. L. Lewis, D. Vrabie, and K. . Vamvoudakis, "Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[62] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[63] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 4, pp. 943–949, 2008.

[64] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 20, no. 9, pp. 1490–1503, 2009.

[65] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.

[66] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, 2012.

[67] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 621–634, 2014.

[68] D. Liu, X. Yang, D. Wang, and Q. Wei, "Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints," *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1372–1385, 2015.

[69] J. Fu, H. He, and X. Zhou, "Adaptive learning and control for MIMO system based on adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, no. 7, pp. 1133–1148, 2011.

[70] Y. Lv, J. Na, Q. Yang, X. Wu, and Y. Guo, "Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics," *International Journal of Control*, vol. 89, no. 1, pp. 99–112, 2016.

[71] J. Na and G. Herrmann, "Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 412–422, 2014.

[72] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[73] S. F. Ding, H. Li, C. Y. Su, J. Z. Yu, and F. X. Jin, "Evolutionary artificial neural networks: a review," *Artificial Intelligence Review*, vol. 39, no. 3, pp. 251–260, 2013.

[74] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 3, pp. 694–713, 1997.

[75] Y. Li and A. Häußler, "Artificial evolution of neural networks and its application to feedback control," *Artificial Intelligence in Engineering*, vol. 10, no. 2, pp. 143–152, 1996.

[76] C.-K. Goh, E.-J. Teoh, and K. C. Tan, "Hybrid multiobjective evolutionary design for artificial neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 9, pp. 1531–1548, 2008.

[77] K. C. Tan and Y. Li, "Grey-box model identification via evolutionary computing," *Control Engineering Practice*, vol. 10, no. 7, pp. 673–684, 2002.

[78] L. Wang, C. K. Tan, and C. M. Chew, *Evolutionary robotics: from algorithms to implementations*, Chew C M. Evolutionary robotics, from algorithms to implementations[M]. NJ, 2006.

[79] J. Zhang, Z.-H. Zhang, Y. Lin et al., "Evolutionary computation meets machine learning: a survey," *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 68–75, 2011.

[80] C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-learning-based telerobot control with guaranteed performance," *IEEE Transactions on Cybernetics*, 2017.

[81] C. Yang, K. Huang, H. Cheng, Y. Li, and C. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, 2017.

[82] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[83] C. Yang, T. Teng, B. Xu, Z. Li, J. Na, and C. Su, "Global adaptive tracking control of robot manipulators using neural networks with finite-time learning convergence," *International Journal of Control, Automation, and Systems*, vol. 15, no. 4, pp. 1916–1924, 2017.

[84] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Transactions on Industrial Informatics*, 2017.

[85] R. Cui and W. Yan, "Mutual synchronization of multiple robot manipulators with unknown dynamics," *Journal of Intelligent & Robotic Systems*, vol. 68, no. 2, pp. 105–119, 2012.

[86] L. Cheng, Z.-G. Hou, M. Tan, and W. J. Zhang, "Tracking control of a closed-chain five-bar robot with two degrees of freedom by integration of an approximation-based approach and mechanical design," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 5, pp. 1470–1479, 2012.

[87] C. Yang, X. Wang, Z. Li, Y. Li, and C. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[88] C. Yang, J. Luo, Y. Pan, Z. Liu, and C. Su, “Personalized variable gain control with tremor attenuation for robot teleoperation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2017.

[89] L. Cheng, Z.-G. Hou, and M. Tan, “Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model,” *Automatica*, vol. 45, no. 10, pp. 2312–2318, 2009.

[90] W. He, Y. Dong, and C. Sun, “Adaptive Neural Impedance Control of a Robotic Manipulator with Input Saturation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 334–344, 2016.

[91] W. He, A. O. David, Z. Yin, and C. Sun, “Neural network control of a robotic manipulator with input deadzone and output constraint,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2015.

[92] W. He, Z. Yin, and C. Sun, “Adaptive Neural Network Control of a Marine Vessel With Constraints Using the Asymmetric Barrier Lyapunov Function,” *IEEE Transactions on Cybernetics*, 2016.

[93] W. He, Y. Chen, and Z. Yin, “Adaptive neural network control of an uncertain robot with full-state constraints,” *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.

[94] C. Sun, W. He, and J. Hong, “Neural Network Control of a Flexible Robotic Manipulator Using the Lumped Spring-Mass Model,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 1863–1874, 2017.

[95] W. He, Y. Ouyang, and J. Hong, “Vibration Control of a Flexible Robotic Manipulator in the Presence of Input Deadzone,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 48–59, 2017.

[96] R. Cui, X. Zhang, and D. Cui, “Adaptive sliding-mode attitude control for autonomous underwater vehicles with input nonlinearities,” *Ocean Engineering*, vol. 123, pp. 45–54, 2016.

[97] R. Cui, C. Yang, Y. Li, and S. Sharma, “Adaptive Neural Network Control of AUVs With Control Input Nonlinearities Using Reinforcement Learning,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.

[98] B. Xu, D. Wang, Y. Zhang, and Z. Shi, “DOB based neural control of flexible hypersonic flight vehicle considering wind effects,” *IEEE Transactions on Industrial Electronics*, vol. PP, no. 99, p. 1, 2017.

[99] B. Xu, C. Yang, and Y. Pan, “Global neural dynamic surface tracking control of strict-feedback systems with application to hypersonic flight vehicle,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2563–2575, 2015.

[100] Y. Li, S. S. Ge, and C. Yang, “Learning impedance control for physical robot-environment interaction,” *International Journal of Control*, vol. 85, no. 2, pp. 182–193, 2012.

[101] Y. Li, S. S. Ge, Q. Zhang, and T. . Lee, “Neural networks impedance control of robots interacting with environments,” *IET Control Theory & Applications*, vol. 7, no. 11, pp. 1509–1519, 2013.

[102] Y. Li and S. S. Ge, “Impedance learning for robots interacting with unknown environments,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1422–1432, 2014.

[103] C. Wang, Y. Li, S. S. Ge, and T. . Lee, “Optimal critic learning for robot control in time-varying environments,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2301–2310, 2015.

[104] Y. Li, L. Chen, K. P. Tee, and Q. Li, “Reinforcement learning control for coordinated manipulation of multi-robots,” *Neurocomputing*, vol. 170, pp. 168–175, 2015.

[105] Y. Li and S. S. Ge, “Human—robot collaboration based on motion intention estimation,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2013.

[106] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, “Continuous Role Adaptation for Human-Robot Shared Control,” *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 672–681, 2015.

[107] Y. Li, K. P. Tee, R. Yan, W. L. Chan, and Y. Wu, “A Framework of Human-Robot Coordination Based on Game Theory and Policy Iteration,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1408–1418, 2016.

[108] A. Clark, *Surfing uncertainty: Prediction, action, and the embodied mind*, Oxford University Press, 2015.

[109] J. Zhong, C. Weber, and S. Wermter, “Learning features and predictive transformation encoding based on a horizontal product model,” *Neural Networks and Machine LearningICANN*, pp. 539–546, 2012.

[110] J. Zhong, C. Weber, and S. Wermter, “A predictive network architecture for a robust and smooth robot docking behavior,” *Journal of Behavioral Robotics*, vol. 3, no. 4, pp. 172–180, 2012.

[111] W. Prinz, “Perception and Action Planning,” *European Journal of Cognitive Psychology*, vol. 9, no. 2, pp. 129–154, 1997.

[112] J. Zhong, M. Peniak, J. Tani, T. Ogata, and A. Cangelosi, “Sensorimotor Input as a Language Generalisation Tool: A Neurorobotics Model for Generation and Generalisation of Noun-Verb Combinations with Sensorimotor Inputs,” Tech. Rep., arXiv, 1605.03261, 2016, arXiv:1605.03261.

[113] H. T. Siegelmann and E. D. Sontag, “On the computational power of neural nets,” *Journal of Computer and System Sciences*, vol. 50, no. 1, pp. 132–150, 1995.

[114] J. Zhong, Artificial Neural Models for Feedback Pathways for Sensorimotor Integration,.

[115] J. Tani, M. Ito, and Y. Sugita, “Self-organization of distributedly represented multiple behavior schemata in a mirror system: Reviews of robot experiments using RNNPB,” *Neural Networks*, vol. 17, no. 8-9, pp. 1273–1289, 2004.

[116] W. Hinoshita, H. Arie, J. Tani, H. G. Okuno, and T. Ogata, “Emergence of hierarchical structure mirroring linguistic composition in a recurrent neural network,” *Neural Networks*, vol. 24, no. 4, pp. 311–320, 2011.

[117] J. Tani, *Exploring robotic minds: actions, symbols, and consciousness as self-organizing dynamic phenomena*, Oxford University Press, 2016.

[118] A. Ahmadi and J. Tani, “How can a recurrent neurodynamic predictive coding model cope with fluctuation in temporal patterns? Robotic experiments on imitative interaction,” *Neural Networks*, vol. 92, pp. 3–16, 2017.

[119] J. Zhong, A. Cangelosi, and S. Wermter, “Toward a self-organizing pre-symbolic neural model representing sensorimotor primitives,” *Frontiers in Behavioral Neuroscience*, vol. 8, article no. 22, 2014.

[120] H. K. Abbas, R. M. Zablotowicz, and H. A. Bruns, “Modeling the colonization of maize by toxigenic and non-toxigenic *Aspergillus flavus* strains: implications for biological control,” *World Mycotoxin Journal*, vol. 1, no. 3, pp. 333–340, 2008.

[121] J. Zhong and L. Canamero, “From continuous affective space to continuous expression space: Non-verbal behaviour recognition and generation,” in *Proceedings of the 4th Joint IEEE*

International Conference on Development and Learning and on Epigenetic Robotics, IEEE ICDL-EPIROB 2014, pp. 75–80, ita, October 2014.

[122] J. Zhong, A. Cangelosi, and T. Ogata, "Toward Abstraction from Multi-modal Data: Empirical Studies on Multiple Time-scale Recurrent Models," in *Proceedings of the International Joint Conference on Artificial Neural Networks (IJCNN)*, 2017.

[123] G. Park and J. Tani, "Development of compositional and contextual communicable congruence in robots by using dynamic neural network models," *Neural Networks*, vol. 72, pp. 109–122, 2015.

[124] A. Ahmadi and J. Tani, "Bridging the gap between probabilistic and deterministic models: a simulation study on a variational Bayes predictive coding recurrent neural network model," 2017, https://arxiv.org/abs/1706.10240.

[125] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[126] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

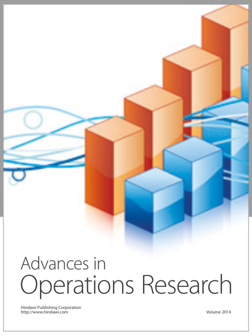
[127] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

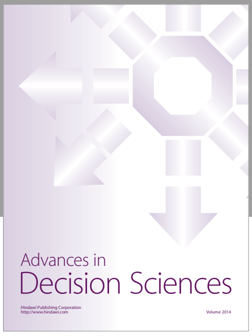
[128] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

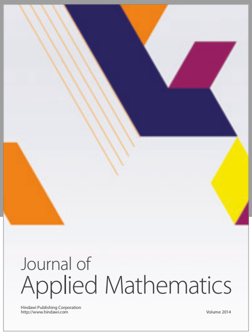
[129] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *American Association for the Advancement of Science: Science*, vol. 313, no. 5786, pp. 504–507, 2006.

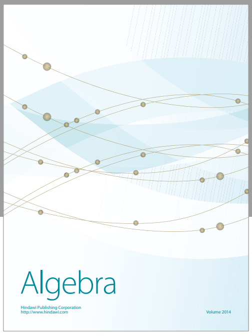
[130] B. C. Pijanowski, A. Tayyebi, J. Doucette, B. K. Pekin, D. Braun, and J. Plourde, "A big data urban growth simulation at a national scale: configuring the GIS and neural network based Land Transformation Model to run in a High Performance Computing (HPC) environment," *Environmental Modelling & Software*, vol. 51, pp. 250–268, 2014.

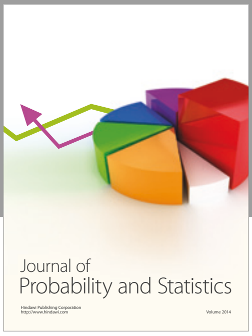
[131] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.


Advances in
Operations Research

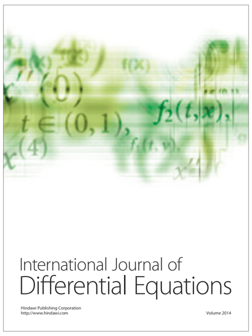

Advances in
Decision Sciences

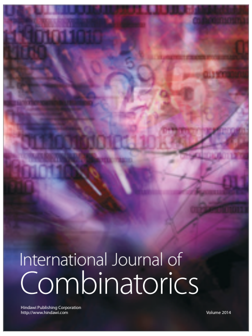

Journal of
Applied Mathematics

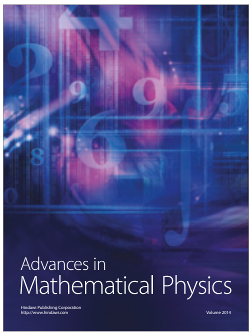

Algebra

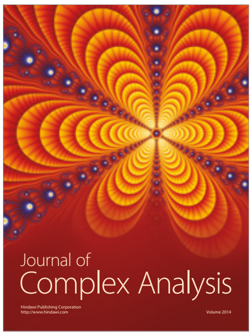

Journal of
Probability and Statistics

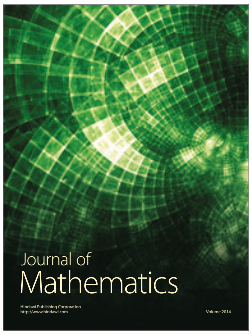
The Scientific
World Journal

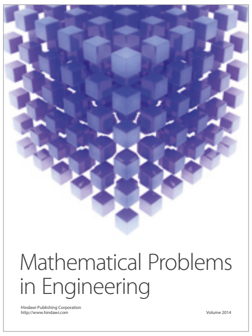

International Journal of
Differential Equations

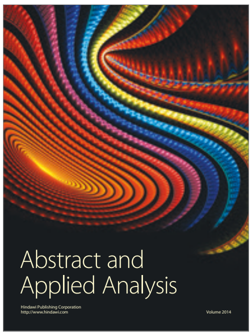

International Journal of
Combinatorics

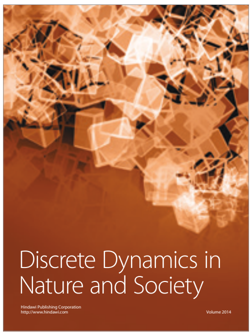

Advances in
Mathematical Physics

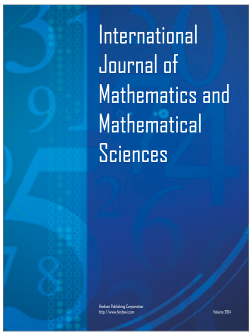

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

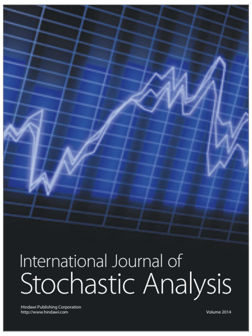

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization