Neurocomputing 285 (2018) 23–34

Contents lists available at ScienceDirect

Neurocomputing

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

Robot manipulator control using neural networks: A survey☆

Long Jin[a], Shuai Li[b,*], Jiguo Yu[c,*], Jinbo He[b]

[a] School of Information Science and Engineering, Lanzhou University, Lanzhou, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[c] School of Information Science and Engineering, Qufu Normal University, Rizhao, China

ARTICLE INFO

Article history:
Received 28 July 2016
Revised 21 November 2017
Accepted 4 January 2018
Available online 3 February 2018

Communicated by Bin He

Keywords:
Robot manipulators
Robot arms
Neural network
Winner-take-all
Parallel manipulators
Cable-driven manipulators
Feedforward neural network
Dual neural network
Recurrent neural network

ABSTRACT

Robot manipulators are playing increasingly significant roles in scientific researches and engineering applications in recent years. Using manipulators to save labors and increase accuracies are becoming common practices in industry. Neural networks, which feature high-speed parallel distributed processing, and can be readily implemented by hardware, have been recognized as a powerful tool for real-time processing and successfully applied widely in various control systems. Particularly, using neural networks for the control of robot manipulators have attracted much attention and various related schemes and methods have been proposed and investigated. In this paper, we make a review of research progress about controlling manipulators by means of neural networks. The problem foundation of manipulator control and the theoretical ideas on using neural network to solve this problem are first analyzed and then the latest progresses on this topic in recent years are described and reviewed in detail. Finally, toward practical applications, some potential directions possibly deserving investigation in controlling manipulators by neural networks are pointed out and discussed.

1. Introduction

In recent decades, robotics has attracted more and more attention from researchers since they have been widely used in scientific researches and engineering applications, such as space exploration, under water survey, industrial and military industries, welding, painting and assembly, and medical applications, and so on [1,2]. Much effort has been contributed to robotics, and different types of robot manipulators have thus been developed and investigated, such as serial manipulators consist of redundant manipulators [3] and mobile manipulators [4], parallel manipulators [5], and cable-driven manipulators [6]. A redundant manipulator is often designed as a series of links connected by motor-actuated joints that extends from a fixed base to an end-effector while a mobile manipulator is often designed as a robotic device composed of a mobile platform and a redundant manipulator fixed to the platform [7]. Different from these serial manipulators, a parallel manipulator is a mechanical system that usually uses several serial chains to support a single platform, or end-effector. Besides, a cable-driven manipulator is a special parallel manipulator, in which the moving platform is driven by cables instead of rigid links [8]. Using these manipulators to save labors and increase accuracies are becoming common practices in various industrial fields. As a consequence, many approaches have been proposed, investigated and employed for the control of robot manipulators [9]. Among them, thanks to many advantages in parallel distributed structure, nonlinear mapping, ability to learn from examples, high generalization performance, and capability to approximate an arbitrary function with sufficient number of neurons, the neural-network-based approach is a competitive way to control movements of robot manipulators [1]. Generally speaking, neural networks can be classified into different types according to different criterions. For example, in terms of the structure of the network, they can be classified into two categories: feedforward neural networks and recurrent neural networks [10,11]. A feedforward neural network is an artificial neural network with no cycles or feedback signal inside while a recurrent neural network allows bi-directional information

☆ This work is supported by the National Natural Science Foundation of China (with numbers 61703189 and 61401385), by the Fundamental Research Funds for the Central Universities (no. lzujbky-2017-37), by Hong Kong Research Grants Council Early Career Scheme (with no. 25214015), by Hong Kong Polytechnic University (with numbers G-YBMU, G-UA7L, 4-ZZHD, F-PP2C, 4-BCCS), by the Hunan Provincial Natural Science Foundation of China (with numbers 2017JJ3257 and 2017JJ3258), and also by the Research Foundation of Education Bureau of Hunan Province, China (nos. 17B215 and 17C1299). Kindly note that L. Jin and S. Li are jointly of the first authorship.

* Corresponding authors.
E-mail addresses: jinlong@lzu.edu.cn (L. Jin), shuaili@polyu.edu.hk (S. Li), jiguoyu@sina.com (J. Yu), adriano.he@connect.polyu.edu.hk (J. He).

https://doi.org/10.1016/j.neucom.2018.01.002
0925-2312/

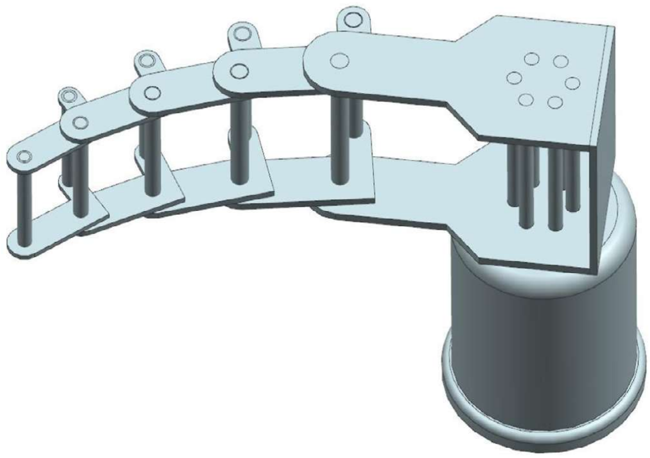

Fig. 1. An example of redundant manipulators.

flow, which means the information inside flows from a successive node to a previous one (or called feedback) or forms a closed cycle within a single node. In this paper, we make a relatively comprehensive review of research progress on controlling these robot manipulators by means of neural networks. The overall organization of the paper is as follows. After the introduction, we present preliminaries on the control of robot manipulators based on neural networks in Section 2. Section 3 presents and reviews different types of robot manipulators in detail with the corresponding schematics being illustrated. In addition, Section 4 revisits applications of different neural networks to the control of robot manipulators. Moreover, two possible future research directions on control of robot manipulators using neural networks are pointed out in Section 5. Finally, Section 6 concludes the paper with final remarks.

2. Preliminaries

The purpose of controlling manipulators is to achieve a specific task like payload carrying, trajectory tracking and so forth [12]. In order to accomplish those tasks, we have to send orders to the manipulators to let them achieve the desired velocity, acceleration or force at specific time [13]. The behavior of manipulators can be deemed as a function since the output given by a manipulator would be different with the change of inputs. Taking the redundant manipulator illustrated in Fig. 1 as an example [14], with inputs of manipulators being angles of joints often expressed as $\theta(t)$ at a specific time t, we have the following general expression [15]:

$$r(t) = f(\theta(t)), \tag{1}$$

where $r(t)$ indicates the end-effector's position and $f(\cdot)$ represents the differentiable nonlinear function. Actually, the output value could also be velocity, acceleration, and force applied on end-effectors, which just needs further calculation. The purpose is to design a controller that could send appropriate inputs when the desired output is given, sometimes with various kinds of constraints. In this paper, we mainly investigate controllers based on neural networks, which have already shown to have powerful capability in solving nonlinear problems [16–21].

An intuitive working flow of controlling a manipulator with neural network based controller is given in Fig. 2. Generally speaking, according to the extent of the knowledge on the manipulator dynamics as well as external disturbance, neural network based controllers for the motion generation and control of manipulators can be classified into three categories: full knowledge, partial knowledge, and no knowledge on the model dynamics and external disturbance of manipulators. With known structure and parameters, recurrent neural networks can be developed to control manipulators such that a performance index under extra constraints can be optimized [22–28]. A control scheme based on recurrent neural networks is presented in [26], which is able to maximize the manipulability of a robot manipulator with known model dynamics effectively in an inverse-free manner. The involved recurrent neural network solves the problem recursively and does not need to be trained in advance. In addition, under certain conditions, it has been proven that feedforward neural networks are capable of approximating various nonlinear functions to any desired degree of accuracy [29]. Thus, the adaptive neural network is designed to compensate uncertainties due to modeling errors or disturbances in the control of manipulators with partial knowledge on model dynamics [29,30]. Besides, the model-free control scheme aided with neural networks is able to address the learning and control of manipulators simultaneously in a unified framework, with the model dynamics of manipulator unknown [31].

In order to command manipulators to finish a specific task, users only need to input a desired output to the control system in practical cases [32]. Then the controller would automatically send a processed signal including commands to manipulators to achieve the final outputs. The crucial task here is to design a controller able to minimize the difference between the desired outputs and the actual outputs, in order to simulate the dynamics of target manipulators.

3. Various robot manipulators

In this section, we start a discussion from a perspective of the variety of mainstream manipulators that involved in controlling problem tackled by neural networks.

3.1. Redundant manipulators

Redundant manipulators are those manipulators that have more domain of freedom (DOF) than required by tasks, which enable some improvements on performance like avoiding collision, optimizing specific criteria like torques or velocity at joints. Different form the non-redundant manipulator, as illustrated in Fig. 1, a robot manipulator with extra redundancy could move in a wider range, have better dexterity and also work more efficient in coordinate manipulation task [33]. The optimization of redundant manipulators is frequently treated as a quadratic programming problem. To remedy the joint-angle drift phenomenon for control of two redundant manipulators, a scheme is proposed in [34] and solved by a special case of dual network termed piecewise-linear projection equation based neural network. This work can be deemed as a follow-up work on the motion planning of redundant manipulators based on neural networks. More related works done on the control of redundant manipulators include [15,35–41].

3.2. Parallel manipulators

A parallel manipulator as shown in Fig. 3 is a mechanism that an end-effector, usually a platform, is supported by several serial chains, which could be applied to the area of medication, industrial manufacturing, deep sea exploration [42,43] and flight simulators [44]. One of the most famous example is Stewart platform, consisting of six linear actuators and two platforms, one of which is the base to support actuators and the other would be the end-effector supporting by those controllable actuators [45]. Compared with serial manipulators, parallel manipulators have better stiffness and are more convenient to reconfigure. In addition, parallel manipulators may avoid the error which may be amplified by each joint

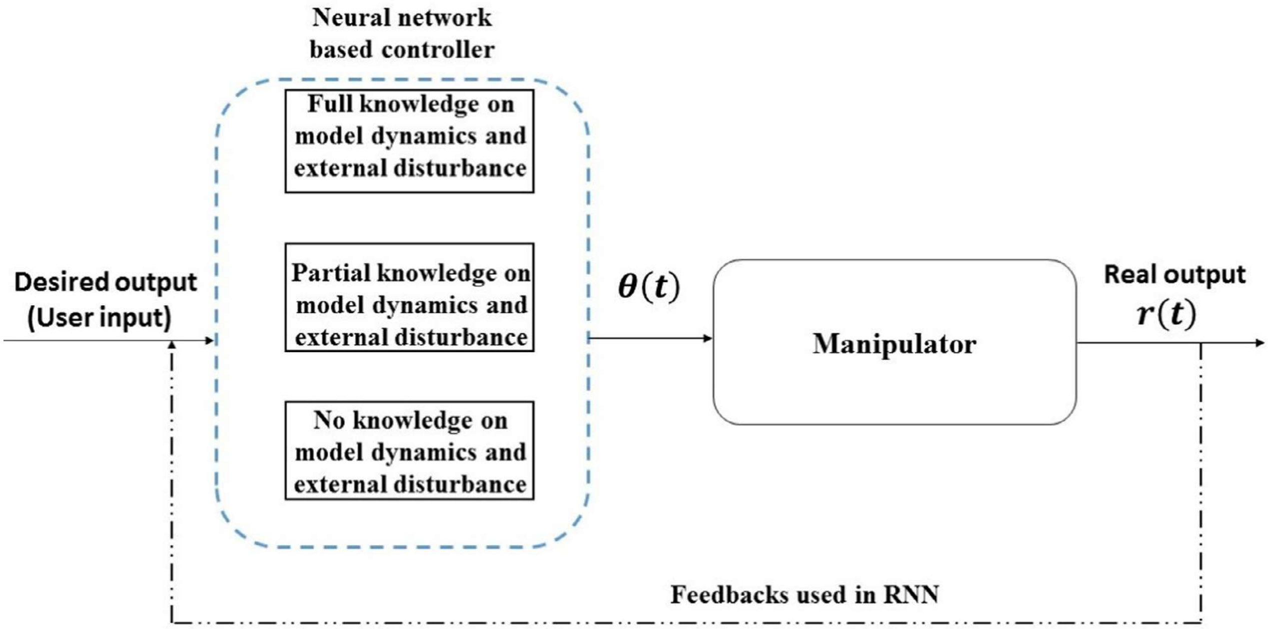

Fig. 2. Information flow of controlling a manipulator, where the dotted line from real output to user input denotes the feedback and constructs the input neural activities to recurrent neural networks.

Fig. 3. An example of parallel manipulators.

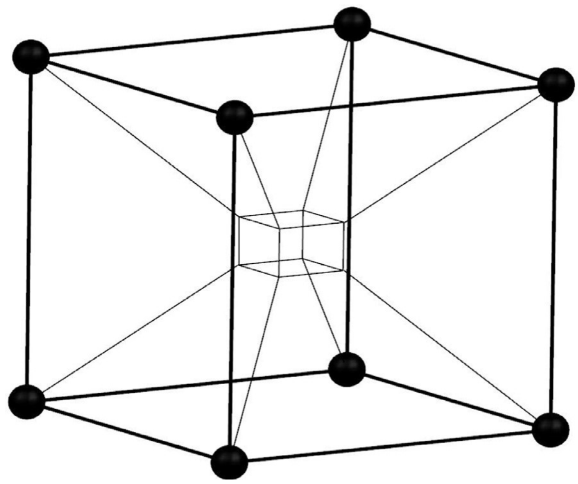

Fig. 4. An example of cable-driven manipulators.

in a serial manipulator, thus give themselves a better accuracy in positioning tasks. However, due to the structural nature of parallel manipulators, their workspaces are much more limited than those of serial manipulators. In addition, there exists one worse problem: singularity. When the mechanical system gets closer to its singular region or right at its singular point, the rigidity and precision would downgrade severely, which would makes the manipulators perform worse [46]. Forward kinematics problem of Stewart platform solved using BP-based feedforward neural network is mentioned in [47], where the authors conduct a process of optimization due the problem nature that it may have several solutions. The neural network method involved in [43] is added with an error compensation mechanism, by applying which the time to obtain the final solution could be just about 1 second, reducing the calculating time greatly in the same accuracy level. Authors in [45] formulate the kinematic control problem of Stewart platforms into a quadratic programming solved by a dynamic dual neural network. They also present theoretical analysis revealing the global convergence of the employed dynamic neural network to the optimal solution in terms of the defined criteria as well as the corresponding simulation results.

3.3. Cable-driven manipulators

Cable-driven robot manipulators, as shown in Fig. 4, are practically applied in various areas, such as carrying the payload that is too fragile to have a simple contact with the ground [48], constructing an exoskeleton to help disabled people [49], live broadcasting and so forth. In [50], the kinematics problem of cable driven robot is solved by a multi-layer perceptron based neural network trained with back propagation. Enhanced convergence and relative small errors are verified from a simulation study. Moreover, the inverse kinematics problem of a robot controlled by three cables are discussed in [51], where the authors utilize a feedforward neural network to express the relation between the manipulator tip position and the forces on those cables. Authors in [52] present a Jacobian-based method and a feedforward neural network to solve the inverse kinematics problem of cable-driven

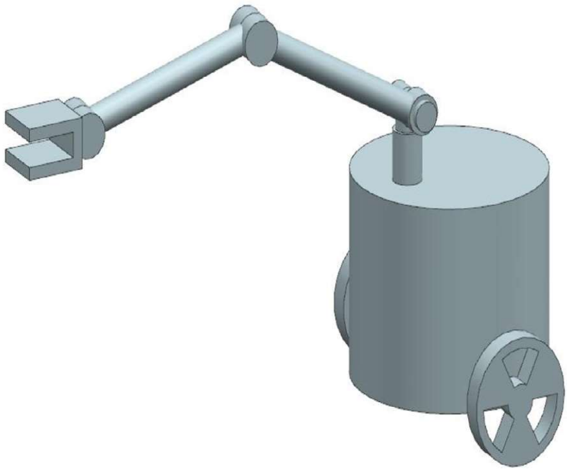

Fig. 5. An example of mobile manipulators.

soft manipulators with a comparative analysis being conducted in terms of accuracy and computational time.

3.4. Mobile manipulators

Mobile manipulators, as illustrated in Fig. 5, are those robotic arms integrated on a movable base, with which manipulators may have much more expanded workspaces thus perform better in positioning [7,53,54]. In [55], the author indicates that mobile manipulators are generally formed by a m-wheeled holonomic/non-holonomic mobile platform and an n-DOF modular manipulator mounting on the platform. Examples of practical application of mobile manipulators could be found in explosives tasks, hazardous place exploring and space operating tasks [56,57]. A robust trajectory tracking task of omnidirectional wheeled mobile manipulator is implemented and tested in [58], where a method of neural network bases sliding model control is proposed to accomplish the task. Utilized neural network is to find the unstructured dynamics inside the controlling mechanism, whose learning efficiency is enhanced by a partitioned structure of neural network.

4. Neural network methods for manipulator control

In the above section, we focus on the variety of mainstream manipulators that involved in controlling problem tackled by neural networks. In this section, we would start a discussion from a new perspective, the variety of prevalent neural network algorithms that applied to manipulators controlling problem. Artificial neural network is a learning algorithm that is inspired by the working mechanism inside humans' brains and designed to simulate the learning procedure of neurons. In an artificial neural network, basically there are three layers: input layer, hidden layer and output layer. The purpose of applying neural network is to train a set of parameters (weights), which could reflect the mappings from user inputs to the inputs sent to manipulators.

There are two main types of training algorithms of neural networks in the manipulator controlling problem, namely on-line training and off-line training respectively. These two methods could be adopted progressively in a specific task, depending on the performance. Training an off-line neural network is simpler, by which the parameters of the designed neural network would not be adjusted when applied onto corresponding manipulators [59]. There is a training procedure before the formal application, which is illustrated in Fig. 6. As the figure shows, during the training step, users receive feedbacks from manipulators and compare those to desired outputs, represented by u. A number of users' inputs would be input into the system to train the neural network. When the gap between desired inputs and real inputs is minimized, the final neural network parameters would be kept and applied into practical applications. However, those training data collected from real manipulators or simulation software may not lead the obtained results to be the real dynamics of robot manipulators, as constraints like payloads or frictions may have impacts on those data derived from ideal cases, leading training data inaccurate. Thus, a successive on-line training is needed for achieving the real dynamics. In on-line training, neural networks adopted in the controller could adjust its parameters according to the differences between expected outputs and actual outputs, with the manipulators operating simultaneously. Having this feature, training process of manipulators could deal with unexpected factors like gravity and friction that influence performance of manipulators [60]. As recurrent neural networks have feedback mechanisms, most of them adopted in real-time controlling problems do not need an off-line training [61].

It can be seen from Fig. 7 that, for the on-line training of a neural network, there is a sensor responsible for measuring the real output and it would pass the result subtracted by user's input to the neural network. This design enables the on-line feature of this training method. Then the neural network has a mechanism which could modify its parameters until it fits training samples. Although off-line training may not achieve the dynamics that we need in practical operations, it can indeed improve the performance when on-line training is conducted.

In the next part, several prevalent neural network methods adopted in manipulators controlling problem would be reviewed and corresponding representative researches would be highlighted.

4.1. Feedforward neural network

A feedforward neural network is an artificial neural network with no cycles or feedback signal inside [62]. This type of neural networks has been widely used to solve dynamics and kinematics problem of controlling robot manipulators.

4.1.1. Feedforward neural network based on back propagation

A back propagation (BP) based feedforward neural network often uses sigmoid function as its activation function [63]. The main idea of back propagation is to adjust parameters such as the weights of connections between neurons inside the network to minimize the loss function related to the difference between the desired output and the actual output. When the loss function is optimized by the method of gradient descent, those parameters inside neural network would be fine-tuned [64]. Although back propagation may give a final solution for specific dynamics or kinematics problem, the solution may not be globally optimized due to the nature of gradient descent, which means the calculated solution is possible a local minimum [65]. As Fig. 8 shows, the algorithm would probably get stuck into the local minimum as the start point of the learning procedure is decided randomly. Also, the convergent speed to the final solution of this method could be relatively slow, resulting from the low learning rate if a relative accurate solution is required [66]. This trade-off between convergence of results and learning rate speed is also an attribute of the gradient descent.

In [58], the back propagation technique with a modification term is utilized to train a nonlinear-in-parameters neural-network-based observer. Robustness and stability of the observer are shown via simulations based on a flexible-joint manipulator. A subsequent work relating to set-point control of planar manipulators in [67] is done, in which a learning algorithm resembling back propagation is employed to obtain the weights of a radial basis function based

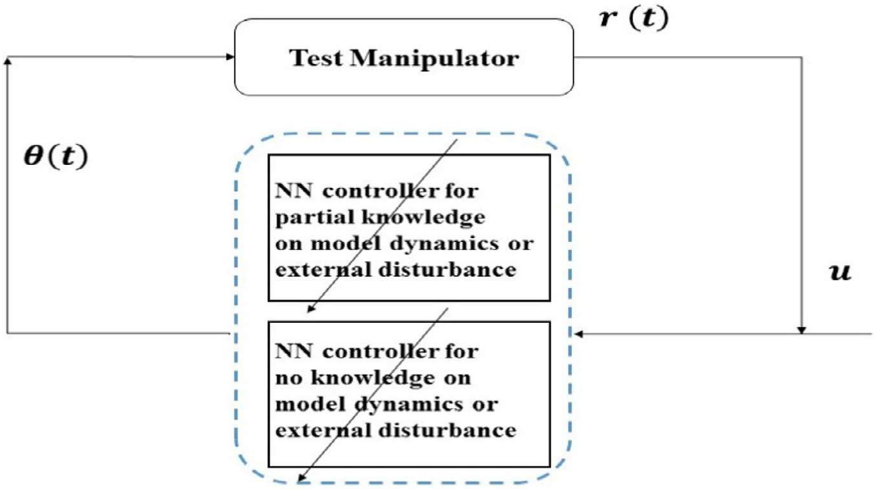

Fig. 6. Off-line training of a neural network.

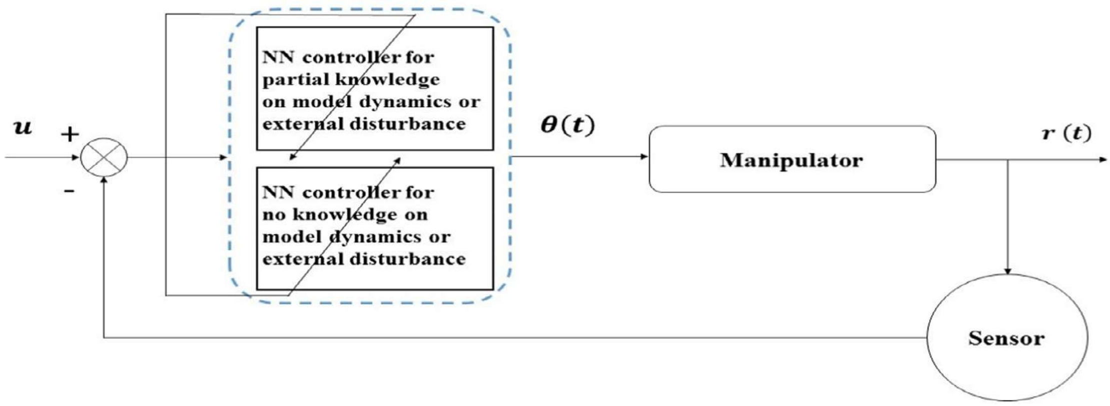

Fig. 7. On-line training of a neural network.

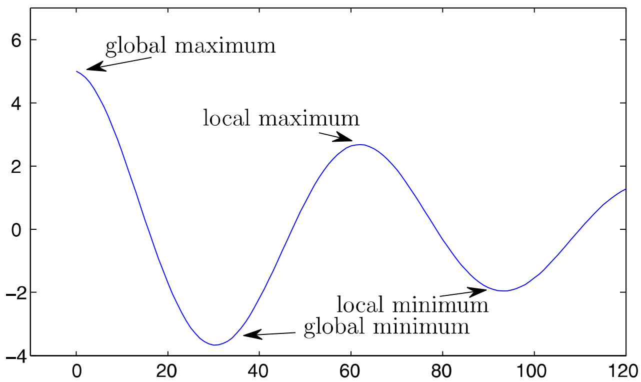

Fig. 8. Illustration for local minima.

network. The controlling method is verified by an experiment on a 2-DOF manipulator.

4.1.2. Feedforward neural network with radial basis function

Different to a BP based neural network that may have multiple hidden layers, a radial basis function (RBF) based neural network has only one hidden layer in its basic structure, which means that there are three layers in total [68,69]. The activation function adopted in hidden layers is a radial basis function, which is a kind of monotone function whose argument l is usually the Euclidean distance to a specific fixed point. With parameter $c > 0$, the following functions could be used to construct an RBF based neural network:

- Multi-quadric functions:

$$\varphi(l) = \sqrt{(l^2 + c^2)}; \tag{2}$$

- Inverse multi-quadric functions:

$$\varphi(l) = \frac{1}{\sqrt{(l^2 + c^2)}}; \tag{3}$$

- Gaussian function:

$$\varphi(l) = \exp\left(-\frac{l^2}{2c^2}\right). \tag{4}$$

The main idea of RBF based neural network is to map linearly inseparable samples to higher dimensions through nonlinear transformations so that they could be separated by linear functions. The component of output layer is a linear combination of values produced by the hidden layer. As radial basis function is influenced by Euclidean distance to specific points (the center), the change of corresponding weight would have a more significant impact on those points closer to the center, which is called local attribute. This is one of the reasons why the converging speed of RBF network is faster than that of typical BP-network when trained by a supervised learning method like gradient descent. Besides trained by gradient descent, parameters in RBF-network could be acquired in other methods: centers of different RBF function could be acquired by clustering methods like k-means classification; weights

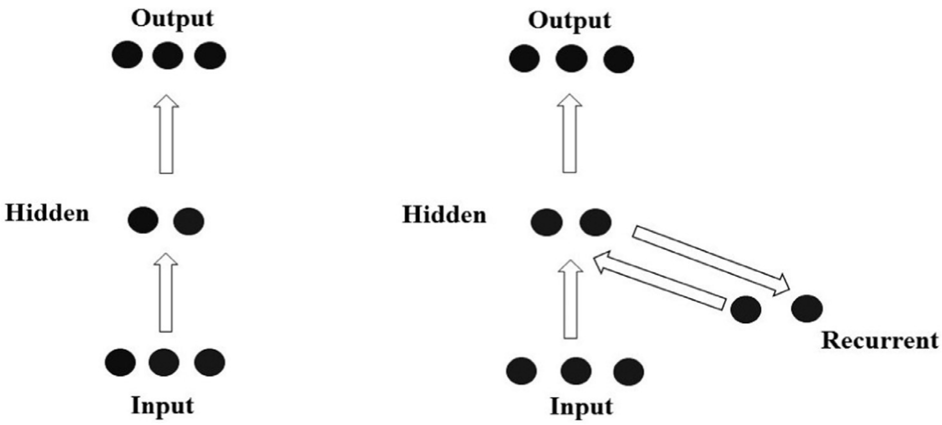

Fig. 9. Difference between feedforward neural network and recurrent neural network.

between the hidden layer and the output layer could be obtained by calculating the pseudoinverse (or inverse when the number of samples equals to the number of neurons in hidden layer) of a matrix.

The RBF based neural network is proved to be available in solving dynamic and kinematic problems of robot manipulators [70]. In [71], an RBF based neural network with a robust control strategy is applied to compensate for the nonlinear dynamics of the robot manipulator in contouring control. This work is extended to the swing-up control of a two-joint manipulator, in which an RBF neural network is adopted to cancel out the negative effect of friction [72]. Improvements of this RBF network based paradigm are observed in experimental results. In addition, an RBF based neural network with dynamic region design is proposed to control robot manipulators [73], where the stability is verified by Lyapunov-like analysis and an energy-saving feature is observed. In [74], the authors applied RBF network based terminal sliding-mode control to robotic manipulator controlling incorporated with actuator dynamics. A robust control mechanism is added in this method, which is validated from experiment results and Lyapunov theory.

4.2. Recurrent neural network

Different from the feedforward neural network mentioned previously, it can be observed from Fig. 9 that a recurrent neural network could have bi-directional information flow, which means the information inside could flow from a successive node to a previous one (or called feedback) or form a closed cycle within a single node [75–79]. The recurrent neural work has been shown successful in the control of robot manipulators. For example, in [80], contact force and position between a manipulator and a surface are controlled by making use of recurrent neural network, which is responsible for simulating the dynamics of manipulators. A simulation on tracking force and position of manipulators is done to show the effectiveness of this method. Moreover, a predictive controller based on a recurrent neural network is designed to reduce the computational time for digital control [81]. The proposed controller is capable for quick changes taking place in inputs, whose effectiveness has been shown through simulations on both kinematics and dynamics of robotic models. Aimed at solving time-varying problems, a special recurrent neural network termed Zhang neural network is investigated in [82], which solves the redundancy resolution problem by computing the time-varying pseudoinverse of the Jacobian matrix of the robot manipulator. Theoretical analysis and simulation results therein illustrate the effectiveness of such a recurrent neural network [83–88]. By transforming nonsmooth optimization problems in multi-robot systems into a convex optimization problem, a recurrent neural network approach is proposed in [76] to solve these nonsmooth optimization problems efficiently. Authors in [89] develop two kinds of recurrent neural networks to solve the resultant redundancy resolution problem, which can achieve the joint-angle and joint-velocity drift problems in cyclic motion of redundant robot manipulators. In addition, by computing the pseudoinverse of the time-varying Jacobian matrix directly, various continues and discrete models are derived in [90] for the motion generation of a manipulator.

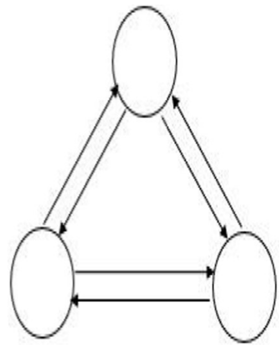

Fig. 10. A 3-neuron Hopfield neural network.

4.2.1. Hopfield network

The structure of a Hopfield network is like a fully connected graph, in which every neuron has a symmetric connection with all other neurons but no cycle with itself. A Hopfield network with 3 neurons is shown in Fig. 10. There are only two states in a neuron (e.g. 0 or 1) with the ith neuron x_i being updated asynchronously or synchronously in the following manner:

$$s_i = \phi\left(\sum_j w_{ij} s_j - \vartheta\right) \quad (i \neq j) \tag{5}$$

where s_j represents those neurons other than s_i, w_{ij} represents the weight between s_i and s_j, $\phi(\cdot)$ is the activate function and ϑ is a threshold. Although we may train the weights of Hopfield network for specific patterns by making use of Hebbian learning [10], it may lead to the local minimum, which forms an obvious drawback of Hopfield network.

It is stated in [91] that Hopfield network could be made use of to solve an arbitrary set of linear equations or constrained least squares optimization problems. In discussions about Hopfield network on practical cases, application on robotics is taken into account. In [35], an algorithm to achieve obstacle avoidance for redundant manipulators is investigated, where Hopfield network plays a role on solving kinematics control. Several experiments on a four-link planar robot arm verify the effectiveness of the presented algorithm. Additionally, a work about using Hopfield network to estimate parameters in dynamic systems is described in [92], in which Hopfield network is shown to have lower errors and less oscillations than a gradient estimator. The result may be an option to be extended to the domain of robot control. It is worth pointing out here that by comparing the weight-updating formula of BP-based neural network with the state-transition equation of Hopfield network for the generalized matrix inversion, authors in [63] show that such two derived learning-expressions turn out to be the same (in mathematics), although the BP and Hopfield-type neural networks are evidently different from each other, a great deal in terms of network architecture, physical meaning, and training patterns. In addition, they extend such an investigation to solve various mathematical problems in [93].

4.2.2. Spiking neural network

Being the third generation neural network, spiking neural network (SNN) is more related to a real neuron system compared with those networks discussed above. The input and output data are always interpreted as “spikes”, which could possibly be a delta function. One feature of spiking neural networks needs to be highlighted is that it can deal with spikes varied by time. Due to its capability of dealing with spikes in specific sequences or under accurate timing, spiking neural network is quite powerful on solving time-dependent patterns [94]. In addition, as stated in [95], SNN models have unique advantages and are good candidates for robot controllers.

In the area of controlling manipulators, some practical applications are done with spiking neural network. For instance, an SNN model is trained to control a 4-DOF manipulator in [96], where spiking timing-dependent plasticity is applied to enhance correlated synapses while weakening those synapses relatively unrelated. The effectiveness of the SNN is verified by experiments on the arm of an iCub humanoid robot. An open source interface library between the SNN and iCub humanoid robot is developed in [97], known as iSpike, which could be applied to develop intelligent robots based on an SNN. Although there are only a few abundant works about controlling manipulators with the help of the SNN, it still has great potential on solving problems in such domain due to its powerful instinct. In addition, a target tracking controller is proposed in [98] for autonomous robots, which encodes the preprocessed environmental and target information into spike trains integrated by a three-layer SNN in unknown environment. The outputs of such an SNN are generated based on the competition between the forward/backward neuron pair corresponding to each motor.

4.2.3. Central pattern generators

Central pattern generators (CPG), of which a model is shown in Fig. 11, are neural networks that produce rhythmic patterns not needing sensory feedbacks. In terms of the neural network, sensory feedbacks indicate those inputs outside the systems. As a result, rhythmic movements such as stepping or arm moving regularly within a desired trajectory could be generated and maintained regardless of the changes of surroundings. An overview about central pattern generators applied in locomotion control of robots is described in [99].

A controlling scheme based on CPG is proposed in [100] to control the locomotion of an amphibious snake robot, which is inspired from the spinal cord of lamprey. CPG helps realize that the direction and speed of locomotion could be adjusted simply in both land and water at the same time configurations could be sent to robot’s actuated joints smoothly and continuously. Similarly, a distributed CPG is applied to control a serpentine robot in [101]. This project not only closely simulates the neural control mechanism but also realizes modularizing into a specific level. In [102], a three-layered bio-inspired architecture is designed to achieve motion generation for humanoid robot iCub, in which CPG is applied as the trajectories generator. Parameters in CPG could be optimized as mentioned in [103], where full body joint trajectory generation is realized for stable bipedal walking. Quantum-inspired evolutionary algorithm is introduced to achieve the optimization. The controlling architecture is verified by simulations on a small humanoid robot.

4.2.4. Echo state network

The main idea of echo state network (ESN) is to exploit a randomly generated reservoir to replace the hidden layer in a typical neural network. To implement an ESN, a reservoir with random connections should be generated first [104]. The number of neurons inside the reservoir should depend on the scale of the problem to be solved. After configuring the reservoir, different states of reservoir at varied time should be record as the input changes. With the knowledge of captured states and desired output, we could determine the weights from reservoir to output by solving a linear regression problem, which are the only value necessary to be trained. A typical structure of echo state network is shown in Fig. 12 with only one neuron in input and output layer, respectively. As can be seen in the figure, weights form reservoir to output neuron are marked by dotted line, which are the only parameters determined form learning process. Weights from the output layer to the reservoir are allowed as a feedback. Although the process to calculate weights in an ESN is relatively easy, complexity of the reservoir appears to be much complicated as the scale of problem increases.

An improved performance on approximating uncertainties in a dynamic system is realized by applying an adaptive fuzzy wavelet echo state network. The control scheme is designed with a feedback controller and adaptive laws for predicting the uncertainties, whose boundedness and convergence are proved by Lyapunov stability and performance are ensured by experiments on a robot manipulator performing precise positional control [105]. A later work [106] about funnel dynamic surface control of prescribed performance of a nonlinear system also exploits fuzzy echo state network to enhance the effectiveness of prediction. Experiments on an MIMO nonlinear system and a manipulator have shown the validity of the control scheme.

4.3. Dual network

Technically, dual network is one kind of recurrent neural networks. As much works related to robotics controlling have done with the help of dual neural network, we would like to treat this part as an important component and discuss in more detail. The main difference between dual network and other neural networks is that it adopts the notion of dual space [107]. The main idea of applying dual space is to convert an optimization problem from primal space to its dual space. In primal space, the convex optimization problem could be represented as follows:

$$\begin{aligned} &\text{minimize} && g(x), \\ &\text{subject to} && c_i(x) \le 0, \quad \text{with } i = 1, 2, \ldots, m, \\ & && d_j(x) = 0, \quad \text{with } j = 1, 2, \ldots, p, \end{aligned} \tag{6}$$

where $g(x)$ is the criteria needing optimized and $c(x)$ and $d(x)$ are inequality and equality constraints respectively. By changing space, equality and inequality constraints could be converted into a form only represented by corresponding dual variables via constructing a Lagrangian function [108,109]:

$$L(x, \lambda_0, \ldots, \lambda_m) = \lambda_0 g(x) + \lambda_1 c_1(x) + \ldots + \lambda_m c_m(x), \tag{7}$$

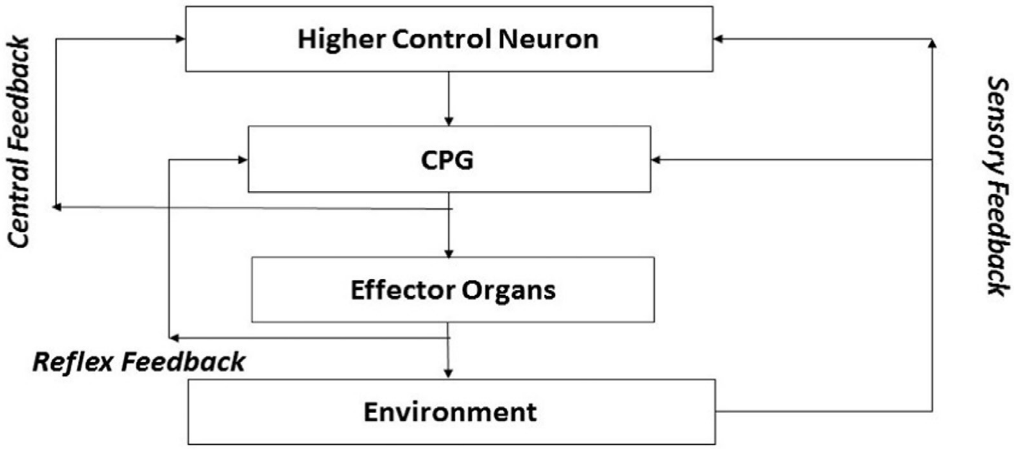

Fig. 11. A model of central pattern generators.

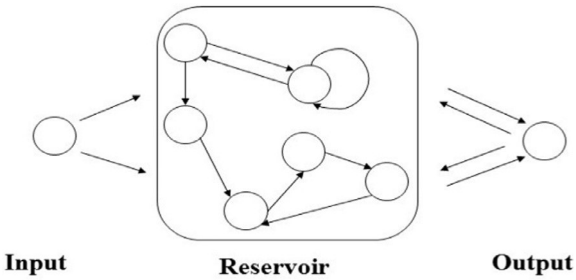

Fig. 12. A model of echo state network.

where λ_i for $i \in \{0, 1, \ldots, m\}$ denotes the Karush–Kuhn–Tucker (KKT) multipliers. Then a neural network could be applied to obtain the optimal solution in an iterative way, as the analytical answers are always hard to give. Various controlling problems could be formulated as quadratic programming (QP) optimization problems which could be solved by applying dual network. This method is common in controlling problems of redundant manipulators that some specific criterion needed to be optimized. An advantage of utilizing dual network to solve QP problem is that an accurate optimized solution could be given even if there exist constraints with inequalities.

A considerable amount of works on controlling robot manipulators with dual network have been done in recent twenty years. For example, in [36], infinity-norm acceleration minimization is realized by an LVI-based primary dual network on redundant manipulators. With the dual network proposed, matrix-matrix multiplication could be avoided thus computational load is saved. Practical simulations done on PUMA 560 robot have validated the algorithm. Authors in [37] treat inverse kinematics problem in robotics as a time-varying quadratic optimization problem and utilized a dual neural network that is globally exponentially stable to perform kinematics control for redundant robot manipulators. A subsequent work about bi-criteria kinematics control for redundant manipulators is done in [38], in which the bi-criteria indicates infinity and Euclidean norms respectively. The bi-criteria adopted in this paper could eliminate the discontinuity of minimum infinity-norm solutions. In addition, a dual network proven to be globally convergent under new criteria is proposed and applied to the control of the PA10 robot manipulators. This work is further extended in [15], in which physical constraints including joint limits, joint velocity limits and an attribute called drift-free are taken into consideration for the optimization problem solved by a dual neural network. The proposed network is shown to be convergent and verified by experiments done on PA10 manipulators. A QP-based solver with simpler piecewise linear dynamics and faster calculation by neglecting matrix inverting is proposed in [39], which is examined on PUMA 560 robot arms and works smoothly. In [33], the optimized criteria chosen include kinetic energy of the system and two-norm of generalized forces added on objects with the limits of torques at joints and applied forces are considered. The proposed dual network is validated in multi-robot coordinate manipulation task.

4.4. Modern control theories and advanced techniques

The model uncertainties and external disturbances of manipulators may cause performance degradation as well as safety problems. The proportional-integral-derivative controllers are a conventional way for handling external disturbances in the control of manipulators. However, the adjustment of the control parameters to the optimum values for the desired control response is rather complicated [110]. Recent progresses have shown advantages of using neural networks based on modern control theories and advanced techniques, e.g., sliding mode, T–S fuzzy mode, adaptive dynamic programming, and reinforcement learning to handle these intractable problems in the control of manipulators [111–117]. However, pure sliding mode has several limitations such as chattering and sensitive problems [118]. In addition, pure fuzzy mode sometimes cannot guarantee stability and acceptable performance [118]. Thanks to many advantages in parallel distributed structure, nonlinear mapping, ability to learn from examples, high generalization performance, and capability to approximate an arbitrary function with sufficient number of neurons, the neural-network-based approach is a competitive way to control movements of robot manipulator. Therefore, hybrid techniques combining together with neural networks are often adopted to control manipulators. A scheme is presented in [118] in order to design high performance nonlinear controller in the presence of uncertainties, which combines sliding control, adaptive dynamic programming, fuzzy control as well as PID control. A concise discussion is presented in [117], which finds that the core of reinforcement learning is identical to that of adaptive optimal control (or adaptive dynamic programming). An adaptive control scheme is provided in [115] for robot manipulator systems with unknown functions and dead-zone input by using a reinforcement learning scheme, of which the parameters of the dead zone are assumed to be unknown but bounded. A survey on the applications of reinforcement learning in robots is presented in [119], in which the authors discuss the equivalency of reinforcement learning and adaptive dynamic programming. The tracking control problem for an uncertain n-link manipulator is investigated in [29], of which the manipulator is formulated as a multi-input and multi-output (MIMO) system. Then, adaptive neural network is designed to handle system uncertainties and disturbances. Then, such a technique is further employed to handle output constraint [120], backlash-like hysteresis [121] and the system uncertainties of biped robots [122]. In the case of performing complicated tasks via coordinated dual manipulators, accurate coordination of motions is required to achieve effective cooperation between the two manipulators. In addition, apart from the external forces, the forces applied on the object grasped must be considered. Yang et al. design an adaptive neural control with the aid of RBF neural network in [123] for controlling the Baxter robot in the presence of unknown dynamics and the manipulated object. Then, they further employ RBF neural network to the control of Baxter robot at both kinematic and dynamic levels [124] as well as the design of controller for a teleoperation system [125]. Besides, the extreme learning machine (ELM) is also used to construct control scheme for uncertain robot manipulators to perform haptic identification in [126], where ELM is used to compensate for the unknown nonlinearity in the manipulator dynamics. It is worth mentioning that, for the situation that state variables of robot manipulators required by the controller are not measurable, RBF neural network based observer can be designed to handle the unmeasurable problem [127,128].

5. Discussion on future directions

This section presents the related discussion on possible future research directions of the combinations of neural networks and controlling manipulators.

5.1. Winner-take-all for manipulator control

In past years, consensus has attracted intensive research attentions and finds its applications in [129–132], which is mostly limited to the modeling of dynamic cooperation. However, research in many fields confirms the same importance of competition as that of cooperation in the emergence of complex behaviors [133]. Winner-take-all (WTA) is an operation that outputs the largest value from the input signals, which is used to capture the competitive nature in the interaction of multi-agent systems. Mathematically, The WTA problem can be formulated as a function:

$$x_i = \eta(\nu_i) = \begin{cases} 1, & \text{if } \nu_i \text{ is the largest element of } \nu, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

which can be further extended to the following so called k-WTA problem [134,135]:

$$x_i = \eta(\nu_i) = \begin{cases} 1, & \text{if } \nu_i \in \ \{\text{the k largest elements of } \nu_i\}, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

The learning phase in the WTA network can also be interpreted by the recursive weight updating formulation, where only the weights associated with the winning neuron are updated and all the other weights remain unchanged [136]. Such a feature of WTA network can be further used for the cooperation and competition of multi robots. For example, equipped with the WTA network and based on the distances between each robot, multi robots can assemble vehicles on an assembly line in sequence.

5.2. Long short term memory for manipulator control

Recurrent neural networks can use their feedback connections to store representations of recent inputs in form of activations, which is termed short term memory [137,138]. Recurrent neural networks based short term memory has been successfully used in time series prediction problems, such as machine translation, natural language process and music composition [139]. In all these tasks, the outputs of the networks are time series. When the minimal time lags between inputs and the output signals are long, the short-term memory consumes too much time or even do not work well at all. To remedy the weakness of short term memory, authors in [137] propose the so-called long short term memory (LSTM) based on gradient method. Different from traditional recurrent neural networks, an LSTM network is well-suited to learn from experience to classify, process and predict time series when there are very long time lags of unknown size between important events [140]. The LSTM has successful applications in font recognition on single Chinese characters [141], unsegmented connected handwriting recognition [142]. In view of the advantages of the LSTM network, it can be expected that the control of manipulators based on long short term memory networks to remedy the existing weaknesses. For example, equipped with the LSTM network, the motion generations of manipulators can be processed in a prediction manner.

6. Conclusions

In summary, great achievements for the control of manipulators by means of neural networks have been gained in the last two decades. However, there are still many new problems to be solved. All these future developments will accompany the development of the advanced manufacture and material for various kinds of robot manipulators as well as the mathematical theory for constructing and developing neural networks. Keeping in mind, different kinds of neural networks have their own feasible ranges, and one cannot expect that only a few existing results on neural networks can tackle all the control problems existing in different manipulators with different tasks. Every class of neural networks, for example, feedforward neural networks, recurrent neural networks, dual neural networks as well as their modifications, has their own advantages, which has considered different tradeoffs between computational complexity and efficiency for the control of robot manipulators. Finally, two possible future research directions on control of robot manipulators using neural networks are pointed out, which may open a door to the research on this topic.

References

[1] A. Zou, Z. Hou, S. Fu, M. Tan, Neural networks for mobile robot navigation: a survey, in: Advances in Neural Networks: Third International Symposium on Neural Networks (ISNN), 2006, pp. 1218–1226.

[2] Y. Zhang, L. Jin, Robot Manipulator Redundancy Resolution, Wiley, UK, 2017.

[3] S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks, Neurocomputing 91 (2012) 1–10.

[4] Z. Li, W. Chen, J. Luo, Adaptive compliant force-motion control of coordinated non-holonomic mobile manipulators interacting with unknown non-rigid environments, Neurocomputing 71 (2008) 1330–1344.

[5] T. Le, H. Kang, An adaptive tracking controller for parallel robotic manipulators based on fully tuned radial basic function networks, Neurocomputing 137 (2014) 12–23.

[6] M. Giorelli, F. Renda, G. Ferri, C. Laschi, A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 5033–5039.

[7] L. Xiao, Y. Zhang, A new performance index for the repetitive motion of mobile manipulators, IEEE Trans. Cybern. 42 (2) (2014) 280–292.

[8] S.H. Yeo, G. Yang, W.B. Lim, Design and analysis of cable-driven manipulators with variable stiffness, Mech. Mach. Theory 69 (2013) 230–244.

[9] C. Yang, J. Luo, Y. Pan, Z. Liu, C. Su, Personalized variable gain control with tremor attenuation for robot teleoperation, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2017.2694020.

[10] B. Müller, J. Reinhardt, M.T. Strickland, Neural Networks: An Introduction, Springer-Verlag, New York, 2012.

[11] L. Jin, Y. Zhang, B. Qiu, Neural network-based discrete-time z-type model of high accuracy in noisy environments for solving dynamic system

of linear equations, Neural Comput. Appl., 2018in press. doi:10.1007/s00521-016-2640-x.
[12] Y. Zhang, Z. Zhang, Repetitive Motion Planning and Control of Redundant Robot Manipulators, Springer-Verlag, New York, 2013.
[13] Y. Zhang, S.S. Ge, T.H. Lee, A unified quadratic programming based dynamical system approach to joint torque optimization of physically constrained redundant manipulators, IEEE Trans. Syst. Man Cybern. B Cybern. 34 (5) (2004) 2126–2132.
[14] Z. Zhang, Y. Zhang, Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming, IEEE-ASME Trans. Mech. 18 (2) (2013) 674–686.
[15] Y. Zhang, J. Wang, Y. Xia, A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits, IEEE Trans. Neural Netw. 14 (3) (2003) 658–667.
[16] D. Lin, X. Wang, F. Nian, Y. Zhang, Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems, Neurocomputing 73 (2010) 2873–2881.
[17] L. Xiao, R. Lu, Finite-time solution to nonlinear equation using recurrent neural dynamics with a specially-constructed activation function, Neurocomputing 151 (2015) 246–251.
[18] L. Xiao, A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation, Neurocomputing 193 (2016) 213–218.
[19] L. Jin, Y. Zhang, S. Li, Integration-enhanced Zhang neural network for real–time-varying matrix inversion in the presence of various kinds of noises, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2016) 2615–2627.
[20] W. Liu, L. Cheng, Z. Hou, J. Yu, M. Tan, An inversion-free predictive controller for piezoelectric actuators based on a dynamic linearized neural network model, IEEE-ASME Trans. Mech. 21 (1) (2016) 214–226.
[21] L. Cheng, W. Liu, Z. Hou, J. Yu, M. Tan, Neural-network-based nonlinear model predictive control for piezoelectric actuators, IEEE Trans. Ind. Electron. 62 (12) (2015) 7717–7727.
[22] Y. Zhang, S. Chen, S. Li, Z. Zhang, Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters, IEEE Trans. Ind. Electron. (2018), doi:10.1109/TIE.2017.2774720. in press
[23] Y. Zhang, S. Li, Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution, IEEE Trans. Ind. Inf. (2018). in press. doi: 10.1109/TII.2017.2737363.
[24] S. Li, H. Wang, M. Rafique, A novel recurrent neural network for manipulator control with improved noise tolerance, IEEE Trans. Neural Netw. Learn. Syst. (2018). in press. doi: 10.1109/TNNLS.2017.2672989.
[25] S. Li, Y. Zhang, L. Jin, Kinematic control of redundant manipulators using neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2018). in press. doi: 10.1109/TNNLS.2016.2574363.
[26] L. Jin, S. Li, H. La, X. Luo, Manipulability optimization of redundant manipulators using dynamic neural networks, IEEE Trans. Ind. Electron. 64 (6) (2017) 4710–4720.
[27] L. Jin, Y. Zhang, S. Li, Y. Zhang, Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators, IEEE Trans. Ind. Electron. 63 (11) (2016) 6978–6988.
[28] L. Jin, S. Li, B. Hu, RNN models for dynamic matrix inversion: a control-theoretical perspective, Ieee Trans. Ind. Inf. (2018). in press. doi: 10.1109/TII.2017.2717079.
[29] W. He, Y. Chen, Z. Yin, Adaptive neural network control of an uncertain robot with full-state constraints, IEEE Trans. Cybern. 46 (3) (2016) 620–629.
[30] W. He, Y. Dong, C. Sun, Adaptive neural impedance control of a robotic manipulator with input saturation, IEEE Trans. Syst. Man Cybern. Syst. 46 (3) (2016) 334–344.
[31] S. Li, Z. Shao, Y. Guan, A dynamic neural network approach for efficient control of manipulators, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2017.2690460.
[32] L. Cheng, Z. Hou, M. Tan, Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model, Automatica 45 (10) (2009) 2312–2318.
[33] Z.-G. Hou, L. Cheng, M. Tan, Multicriteria optimization for coordination of redundant robots using a dual neural network, IEEE Trans. Syst. Man Cybern. B Cybern. 40 (4) (2010) 1075–1087.
[34] L. Jin, Y. Zhang, G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms, IEEE Trans. Cybern. 45 (2) (2015) 153–164.
[35] H. Ding, S. Chan, A real-time planning algorithm for obstacle avoidance of redundant robots, J. Intell. Rob. Syst. 16 (3) (1996) 229–243.
[36] L. Jin, S. Li, H. Wang, Z. Zhang, Nonconvex projection activated zeroing neurodynamic models for time-varying matrix pseudoinversion with accelerated finite-time convergence, Appl. Soft Comput. (2018). in press
[37] Y. Xia, Wang, A dual neural network for kinematic control of redundant robot manipulators, IEEE Trans. Syst. Man Cybern. B Cybern. 31 (1) (2001) 147–154.
[38] Y. Zhang, Wang, Y. Xu, A dual neural network for bi-criteria kinematic control of redundant manipulators, IEEE Trans. Rob. Autom. 18 (6) (2002) 923–931.
[39] Y. Zhang, S.S. Ge, T.H. Lee, A unified quadratic-programming based dynamical system approach to joint torque optimization of physically constrained redundant manipulators, IEEE Trans. Syst. Man Cybern. B Cybern. 34 (5) (2004) 2126–2132.
[40] L. Jin, S. Li, Distributed task allocation of multiple robots: a control perspective, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2016.2627579.
[41] L. Jin, S. Li, L. Xiao, R. Lu, B. Liao, Cooperative motion generation in a distributed network of redundant robot manipulators with noises, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2017.2693400.
[42] D. Zhang, J. Lei, Kinematic analysis of a novel 3-DOF actuation redundant parallel manipulator using artificial intelligence approach, Rob. Comput. Integr. Manuf. 27 (1) (2011) 157–163.
[43] P.J. Parikh, S.S. Lam, Kinematic analysis of a novel 3-DOF actuation redundant parallel manipulator using artificial intelligence approach, Int. J. Adv. Manuf. Technol. 40 (5–6) (2009) 595–606.
[44] Z. Geng, L. Haynes, Neural network solution for the forward kinematics problem of a stewart platform, in: International Joint Conference on Robotics and Automation (ICRA), 1991, pp. 2650–2655.
[45] A. Muhammad, S. Li, Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms, IEEE Trans. Cybern. 46 (7) (2016) 1538–1550.
[46] D. Zhang, Z. Gao, Forward kinematics, performance analysis, and multi-objective optimization of a bio-inspired parallel manipulator, Rob. Comput. Integr. Manuf. 28 (4) (2012) 484–492.
[47] S. Yurt, E. Anli, I. Ozkol, Forward kinematics analysis of the 6–3 SPM by using neural networks, Meccanica 42 (2) (2007) 187–196.
[48] C. Li, J. Yi, Y. Yu, D. Zhao, Inverse control of cable-driven parallel mechanism using type-2 fuzzy neural network, Acta Autom. Sin. 36 (3) (2010) 459–464.
[49] J.C. Perry, J. Rosen, Design of a 7 degree-of-freedom upper-limb powered exoskeleton, in: International Joint Conference on RAS-EMBS, 2006, pp. 805–810.
[50] A. Ghasemi, M. Eghtesad, M. Farid, Neural network solution for forward kinematics problem of cable robots, J. Intell. Rob. Syst.2010. 60, 2, 201–215.
[51] M. Giorelli, F. Renda, G. Ferri, C. Laschi, A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three--dimensional space, in: IEEE International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 5033–5039.
[52] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, C. Laschi, Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature, IEEE Trans. Rob. 31 (4) (2015) 823–834.
[53] A.N. Asl, M.B. Menhaj, A. Sajedin, Control of leader-follower formation and path planning of mobile robots using asexual reproduction optimization (ARO), Appl. Soft Comput. 14 (2014) 563–576.
[54] Y. Zhang, W. Li, B. Liao, D. Guo, C. Peng, Analysis and verification of repetitive motion planning and feedback control for omnidirectional mobile manipulator robotic systems, J. Intell. Rob. Syst. 75 (2014) 393–411.
[55] Y. Liu, Y. Li, Sliding mode adaptive neural-network control for nonholonomic mobile modular manipulators, J. Intell. Rob. Syst. 44 (3) (2005) 203–224.
[56] S. Lin, A.A. Goldenberg, Neural-network control of mobile manipulators, IEEE Trans. Neural Netw. 12 (5) (2001) 1121–1133.
[57] Y. Chen, L. Liu, M. Zhang, H. Rong, Study on coordinated control and hardware system of a mobile manipulator, in: World Congress on Intelligent Control and Automation (WCICA), 2006, pp. 9037–9041.
[58] F. Abdollahi, H.A. Talebi, R.V. Patel, A stable neural network-based observer with application to flexible-joint manipulators, IEEE Trans. Neural Netw. 17 (1) (2006) 118–129.
[59] D. Psaltis, A. Sideris, A.A. Yamamura, A multilayered neural network controller, IEEE Control Syst. Mag. 8 (2) (1988) 17–21.
[60] O. Omidvar, P.V.D. Smagt, Neural Systems for Robotics, Academic Press, Massachusetts, 1997.
[61] J. Wang, Q. Hu, D. Jiang, A lagrangian network for kinematic control of redundant robot manipulators, IEEE Trans. Neural Netw. 10 (5) (1999) 1123–1132.
[62] J. Qiao, F. Li, H. Han, W. Li, Constructive algorithm for fully connected cascade feedforward neural networks, Neurocomputing 182 (2016) 154–164.
[63] Y. Zhang, D. Guo, Z. Li, Common nature of learning between back-propagation and Hopfield-type neural networks for generalized matrix inversion with simplified models, IEEE Trans. Neural Netw. Learn. Syst. 24 (4) (2013) 579–592.
[64] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma, Y. Uchikawa, Trajectory control of robotic manipulators using neural networks, IEEE Trans. Ind. Electron. 38 (3) (1991) 195–202.
[65] Y. Zhang, X. Yu, D. Guo, Y. Yin, Z. Zhang, Weights and structure determination of multiple-input feed-forward neural network activated by Chebyshev polynomials of class 2 via cross-validation, Neural Comput. Appl. 25 (7–8) (2014) 1761–1770.
[66] D.R. Raj, I.J. Raglend, M. Anand, Inverse kinematics solution of a five joint robot using feed forward and Elman network, in: 2015 International Conference in Circuit, Power and Computing Technologies (ICCPCT), 2015, pp. 1–5.
[67] R. Garrido, Stable neurovisual servoing for robot manipulators, IEEE Trans. Neural Netw. 17 (4) (2006) 953–965.
[68] Q. Wu, X. Wang, Q. Shen, Research on dynamic modeling and simulation of axial-flow pumping system based on RBF neural network, Neurocomputing 186 (2016) 200–206.
[69] R. Yang, P. Er, Z. Wang, K. Tan, An RBF neural network approach towards precision motion system with selective sensor fusion, Neurocomputing 199 (2016) 31–39.
[70] S.S. Ge, C. Hang, L. Woon, Adaptive neural network control of robot manipulators in task space, IEEE Trans. Ind. Electron. 44 (6) (1997) 746–752.
[71] L. Wang, T. Chai, C. Yang, Neural-network-based contouring control for robotic manipulators in operational space, IEEE Trans. Control Syst. Technol. 20 (4) (2012) 1073–1080.

[72] D. Xia, L. Wang, T. Chai, Neural-network-friction compensationbased energy swing-up control of pendubot, IEEE Trans. Ind. Electron. 61 (3) (2014) 1411–1423.
[73] X. Li, C.C. Cheah, Adaptive neural network control of robot based on a unified objective bound, IEEE Trans. Control Syst. Technol. 22 (3) (2014) 1032–1043.
[74] L. Wang, T. Chai, L. Zha, Neural-network-based terminal slidingmode control of robotic manipulators including actuator dynamics, IEEE Trans. Ind. Electron. 56 (9) (2009) 3296–3304.
[75] L. Cheng, Z. Hou, Y. Lin, M. Tan, W. Zhang, F. Wu, Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks, IEEE Trans. Neural Netw. 22 (5) (2011) 714–726.
[76] Y. Wang, L. Cheng, Z. Hou, J. Yu, M. Tan, Optimal formation of multirobot systems based on a recurrent neural network, IEEE Trans. Neural Netw. Learn. Syst. 27 (2) (2016) 322–333.
[77] B. Cai, Y. Zhang, Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient descent and Zhang et a's neural-dynamic methods, IEEE Trans. Ind. Electron. 59 (8) (2012) 3146–3155.
[78] Q. Liu, J. Wang, A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints, IEEE Trans. Neural Netw. Learn. Syst. 24 (5) (2013) 812–824.
[79] Z. Guo, Q. Liu, J. Wang, A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints, IEEE Trans. Neural Netw. 22 (12) (2011) 1892–1900.
[80] L. Tian, J. Wang, Z. Mao, Constrained motion control of flexible robot manipulators based on recurrent neural networks, IEEE Trans. Syst. Man Cybern. B Cybern. 34 (3) (2004) 1541–1552.
[81] R. Köker, Design and performance of an intelligent predictive controller for a six-degree-of-freedom robot using the Elman network, Inf. Sci. 176 (12) (2006) 1781–1799.
[82] D. Guo, Y. Zhang, Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time-varying Jacobian matrix pseudoinversion, Appl. Soft Comput. 24 (2014) 158–168.
[83] L. Jin, S. Li, Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints, Neurocomputing 267 (2017) 107–113.
[84] L. Jin, S. Li, B. Liao, Z. Zhang, Zeroing neural networks: a survey, Neurocomputing 267 (2017) 597–604.
[85] L. Jin, Y. Zhang, T. Qiao, M. Tan, Y. Zhang, Tracking control of modified Lorenz nonlinear system using ZG neural dynamics with additive input or mixed inputs, Neurocomputing. 196 (2016) 82–94.
[86] L. Jin, Y. Zhang, S. Li, Y. Zhang, Noise-tolerant ZNN models for solving time–varying zero-finding problems: a control-theoretic approach, IEEE Trans. Autom. Control 62 (2) (2017) 992–997.
[87] L. Jin, Y. Zhang, Continuous and discrete Zhang dynamics for real-time varying nonlinear optimization, Numer. Algorithms 73 (2016) 115–140.
[88] L. Jin, Y. Zhang, Discrete-time Zhang neural network for online time- varying nonlinear optimization with application to manipulator motion generation. IEEE trans., Neural Netw. Learn. Syst. 27 (6) (2015) 1525C1531.
[89] Z. Zhang, Y. Zhang, Design and experimentation of acceleration-level drift-free scheme aided by two recurrent neural networks, IET Control Theory A 7 (1) (2013) 25–42.
[90] D. Guo, Y. Zhang, Zhang neural network, Getz-Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots' kinematic control, Neurocomputing 97 (2012) 22–32.
[91] G.G. Lendaris, K. Mathia, R. Saeks, Linear Hopfield networks and constrained optimization, IEEE Trans. Syst. Man Cybern. B Cybern. 29 (1) (1999) 114–118.
[92] M. Atencia, G. Joya, F. Sandoval, Hopfield neural networks for parametric identification of dynamical systems, Neural Process. Lett. 21 (2) (2005) 143–152.
[93] D. Guo, Y. Zhang, Z. Xiao, M. Mao, J. Liu, Common nature of learning between BP-type and Hopfield-type neural networks, Neurocomputing 167 (2015) 578–586.
[94] S. Ghosh-Dastidar, H. Adeli, Spiking neural networks, Int. J. Neural Syst. 19 (4) (2009) 295–308.
[95] X. Wang, Z. Hou, A. Zou, M. Tan, L. Cheng, A behavior controller based on spiking neural networks for mobile robots, Neurocomputing 71 (2008) 655–666.
[96] A. Bouganis, M. Shanahan, Training a spiking neural network to control a 4-DOF robotic arm based on spike timing-dependent plasticity, in: International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1–8.
[97] D. Gamez, A.K. Fidjeland, E. Lazdins, iSpike: a spiking neural interface for the icub robot, Bioinspiration Biomimetics 7 (2) (2012) 025008.
[98] Z. Cao, L. Cheng, C. Zhou, N. Gu, X. Wang, M. Tan, Spiking neural network-based target tracking control for autonomous mobile robots, Neural Comput. Appl. 26 (8) (2015) 1839–1847.
[99] A.J. Ijspeert, Central pattern generators for locomotion control in animals and robots: a review, Neural Netw. 21 (4) (2008) 642–653.
[100] A.J. Ijspeert, A. Crespi, Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model, in: IEEE International Conference on Robotics and Automation (ICRA), 2007, pp. 262–268.
[101] J. Conradt, P. Varshavskaya, Distributed central pattern generator control for a serpentine robot, in: International Joint Conference on Artificial Neural Networks (ICANN), 2003, pp. 338–341.
[102] S. Degallier, L. Righetti, L. Natale, F. Nori, G. Metta, A. Ijspeert, A modular bio-inspired architecture for movement generation for the infant-like robot icub, in: IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, 2008, pp. 795–800.
[103] C.-S. Park, Y.-D. Hong, J.-H. Kim, Full-body joint trajectory generation using an evolutionary central pattern generator for stable bipedal walking, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 160–165.
[104] S. Lun, X. Yao, H. Qi, H. Hu, A novel model of leaky integrator echo state network for time-series prediction, Neurocomputing 159 (2016) 58–66.
[105] S. Han, J.M. Lee, et al., Precise positioning of nonsmooth dynamic systems using fuzzy wavelet echo state networks and dynamic surface sliding mode control, IEEE Trans. Ind. Electron. 60 (11) (2013) 5124–5136.
[106] S. Han, J.M. Lee, et al., Fuzzy echo state neural networks and funnel dynamic surface control for prescribed performance of a nonlinear dynamic system, IEEE Trans. Ind. Electron. 61 (2) (2014) 1099–1112.
[107] P. Miao, Y. Shen, X. Xia, Finite time dual neural networks with a tunable activation function for solving quadratic programming problems and its application, Neurocomputing 143 (2014) 80–89.
[108] L. Cheng, Z. Hou, M. Tan, A delayed projection neural network for solving linear variational inequalities, IEEE Trans. Neural Netw. 20 (6) (2009) 915–925.
[109] L. Cheng, Z. Hou, M. Tan, A neutral-type delayed projection neural network for solving nonlinear variational inequalities, IEEE Trans. Circuits Syst. II 55 (8) (2008) 806–810.
[110] R. Kelly, A tuning procedure for stable PID control of robot manipulators, Robotica 13 (2) (1995) 141–148.
[111] Y. Zhang, S. Li, Predictive suboptimal consensus of multiagent systems with nonlinear dynamics, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2017.2668440.
[112] Y. Zhang, S. Li, Distributed biased min-consensus with applications to shortest path planning, IEEE Trans. Autom. Control (2018). in press. doi: 10.1109/TAC.2017.2694547.
[113] H. Wang, P. Shi, H. Li, Q. Zhou, Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties, IEEE Trans. Cybern., 2018, in press, doi:10.1109/TCYB.2016.2607166.
[114] H. Wang, X.P. Liu, K.F. Liu, Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems, IEEE Trans. Neural Netw. Learn. Syst. 27 (3) (2016) 510–523.
[115] L. Tang, Y. Liu, S. Tong, Adaptive neural control using reinforcement learning for a class of robot manipulator, Neural Comput. Appl. 25 (2014) 135–141.
[116] F. Sun, L. Li, H. Li, H. Liu, Neuro-fuzzy dynamic inversion-based adaptive control for robotic manipulators discrete time case, IEEE Trans. Ind. Electron. 54 (3) (2014) 1342–1351.
[117] R.S. Sutton, A.G. Barto, R.J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Syst. 12 (2) (1992) 19–22.
[118] F. Piltan, N. Sulaiman, S. Soltani, M. Marhaban, R. Ramli, An adaptive sliding surface slope adjustment in PD sliding mode fuzzy control for robot manipulator, Int. J. Control Autom. 4 (3) (2011) 65–76.
[119] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: a survey, Int. J. Rob. Res. 32 (11) (2015) 1238–1274.
[120] S. Zhang, M. Lei, Y. Dong, W. He, Adaptive neural network control of coordinated robotic manipulators with output constraint, IET Control Theory A 10 (17) (2016) 2271–2278.
[121] W. He, D. Amoateng, C. Yang, D. Gong, Adaptive neural network control of a robotic manipulator with unknown backlash-like hysteresis, IET Control Theory A 11 (4) (2017) 567–575.
[122] W. He, Y. Dong, C. Sun, Adaptive neural network control of biped robots, IEEE Trans. Syst. Man Cybern. Syst. 47 (2) (2017) 315–326.
[123] C. Yang, Y. Jiang, Z. Li, W. He, C. Sun, Neural control of bimanual robots with guaranteed global stability and motion precision, IEEE Trans. Ind. Inf. (2018). in press. doi: 10.1109/TII.2016.2612646.
[124] C. Yang, X. Wang, L. Cheng, H. Ma, Neural-learning-based telerobot control with guaranteed performance, IEEE Trans. Cybern. (2018). in press. doi: 10.1109/TCYB.2016.2573837.
[125] C. Yang, X. Wang, Z. Li, Y. Li, C. Sun, Teleoperation control based on combination of wave variable and neural networks, Ieee Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2016.2615061.
[126] C. Yang, K. Huang, H. Cheng, Y. Li, C. Sun, Haptic identification by ELM-controlled uncertain manipulator, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2017.2676022.
[127] W. Chen, J. Yang, L. Guo, S. Li, Disturbance-observer-based control and related methods—an overview, IEEE Trans. Ind. Electron. 63 (2) (2016) 1083–1095.
[128] X. Liu, C. Yang, Z. Chen, M. Wang, C. Su, Neuro-adaptive observer based control of flexible joint robot, Neurocomputing (2018). in press. doi: 10.1016/j.neucom.2017.05.011.
[129] L. Cheng, H. Wang, Z. Hou, M. Tan, Reaching a consensus in networks of high--order integral agents under switching directed topologies, Int. J. Syst. Sci. 47 (8) (2016) 1966–1981.
[130] Y. Wang, L. Cheng, Z. Hou, M. Tan, C. Zhou, M. Wang, Consensus seeking in a network of discrete-time linear agents with communication noises, Int. J. Syst. Sci. 46 (10) (2015) 1874–1888.
[131] Y. Wang, L. Cheng, W. Ren, Z. Hou, M. Tan, Seeking consensus in networks of linear agents: communication noises and Markovian switching topologies, IEEE Trans. Autom. Control 60 (5) (2015) 1374–1379.

[132] L. Cheng, Z. Hou, M. Tan, A mean square consensus protocol for linear multi--agent systems with communication noises and fixed topologies, IEEE Trans. Autom. Control 59 (1) (2014) 261–267.
[133] S. Li, M. Zhou, X. Luo, Y. Zhu, Distributed winner-take-all in dynamic networks, IEEE Trans. Autom. Control (2018). in press. doi: 10.1109/TAC.2016.2578645.
[134] Q. Liu, J. Wang, Two k-winners-take-all networks with discontinuous activation functions, Neural Netw. 21 (2) (2008) 406–413.
[135] X. Hu, J. Wang, An improved dual neural network for solving a class of quadratic programming problems and its *k*-winners-take-all application, IEEE Trans. Neural Netw. 19 (12) (2008) 2022–2031.
[136] S. Kung, J. Hwang, Neural network architectures for robotic applications, IEEE Trans. Rob. Autom. 5 (5) (1989) 641–657.
[137] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[138] Y. Hua, M. Yang, Z. Zhao, R. Zhou, A. Cai, On semantic-instructed attention: from video eye-tracking dataset to memory-guided probabilistic saliency model, Neurocomputing 168 (2015) 917–929.
[139] M. Mozer, Induction of multiscale temporal structure, Adv. Neural Inf. Process. Syst. 4 (1992) 275–282.
[140] S. Naz, A.I. Umar, R. Ahma, S.B. Ahmed, S.H. Shirazi, I. Siddiqi, M.I. Razzak, Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks, Neurocomputing 177 (2016) 228–241.
[141] D. Tao, X. Lin, L. Jin, X. Li, Principal component 2-d long short-term memory for font recognition on single chinese characters, IEEE Trans. Cybern. 46 (3) (2016) 756–765.
[142] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition, IEEE Trans. Pattern Anal. 31 (5) (2009) 855–868.

Long Jin received the B.E. degree and the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2011 and in 2016, respectively. He is currently a full professor with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Before joining Lanzhou University in 2017, he was a postdoctoral fellow with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. His main research interests include neural networks, robotics and intelligent information processing.

Shuai Li received the B.E. degree in Precision Mechanical Engineering from Hefei University of Technology, China in 2005, the M.E. degree in Automatic Control Engineering from University of Science and Technology of China, China in 2008 , and the Ph.D. degree in Electrical and Computer Engineering from Stevens Institute of Technology, USA in 2014 . He is currently a research assistant professor with Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. He is on the editorial board of the International Journal of Distributed Sensor Networks. His current research interests include dynamic neural networks, wireless sensor networks, robotic networks, machine learning, and other dynamic problems defined on a graph.

Jiguo Yu received his Ph.D. degree in School of Mathematics from Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China in 2007. Currently he is a full professor in the School of Information Science and Engineering, Qufu Normal University. His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a senior member of the CCF (China Computer Federation).

Jinbo He received the B.S. degree in internet and multimedia technologies from The Hong Kong Polytechnic University, Hong Kong. He is currently a research assistant with The Hong Kong Polytechnic University, Hong Kong. His current research interests include computer graphics, artificial intelligence, neural networks, and distributed system.