Neurocomputing 267 (2017) 597–604

Contents lists available at ScienceDirect

Neurocomputing

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

Zeroing neural networks: A survey☆

Long Jin[a,b], Shuai Li[b,*], Bolin Liao[c,*], Zhijun Zhang[d]

[a] School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[c] College of Information Science and Engineering, Jishou University, Jishou 416000, China
[d] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

ARTICLE INFO

Article history:
Received 7 April 2017
Revised 27 May 2017
Accepted 16 June 2017
Available online 29 June 2017

Communicated by Xin Luo

Keywords:
Zeroing neural network
Recurrent neural network
Stability
Numerical algorithms
Redundant manipulators
Robust stability

ABSTRACT

Using neural networks to handle intractability problems and solve complex computation equations is becoming common practices in academia and industry. It has been shown that, although complicated, these problems can be formulated as a set of equations and the key is to find the zeros of them. Zeroing neural networks (ZNN), as a class of neural networks particularly dedicated to find zeros of equations, have played an indispensable role in the online solution of time-varying problem in the past years and many fruitful research outcomes have been reported in the literatures. The aim of this paper is to provide a comprehensive survey of the research on ZNNs, including continuous-time and discrete-time ZNN models for various problems solving as well as their applications in motion planning and control of redundant manipulators, tracking control of chaotic systems, or even populations control in mathematical biosciences. By considering the fact that real-time performance is highly demanded for time-varying problems in practice, stability and convergence analyses of different continuous-time ZNN models are reviewed in detail in a unified way. For the case of discrete-time problems solving, the procedures on how to discretize a continuous-time ZNN model and the techniques on how to obtain an accuracy solution are summarized. Concluding remarks and future directions of ZNN are pointed out and discussed.

© 2017 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Approaches based on neural network for solving various knotty problems have attracted considerable attention in many fields [1–14]. For example, an adaptive fuzzy controller based on neural network is constructed for a class of nonlinear discrete-time systems with discrete-time dead zone in [1]. An adaptive decentralized scheme based on neural network is presented for multiple-input and multiple-output (MIMO) nonlinear systems with the aid of back-stepping techniques in [14]. Such a scheme guarantees the uniform ultimate boundedness of all signals in the closed-loop system with respect to mean square. To overcome the design difficulty of nonstrict-feedback structure, Ref. [3] uses variable separation technique to decompose the unknown functions of all state variables into a sum of smooth functions of each error dynamic. With the aid of radial basis function neural networks' universal approximation capability, an adaptive neural control algorithm is proposed in [3]. Authors in [8] propose a neural network model to generate winner-take-all competition, which has an explicit explanation of the competition mechanism. As a branch of artificial intelligence, recurrent neural network (RNN) models have received considerable investigation in many scientific and engineering fields, which is often exploited for computational problems [15–22] and nonlinear optimizations are solved by many methods [23,24] A gradient-based RNN model is presented in [25] for computing the inversion of a matrix online with guaranteed convergence, which can be deemed as a seminal work in this field. A simplified neural network model is presented in [26] to solve a class of linear matrix inequality problems, of which the stability and solvability are analyzed theoretically. In general, recurrent neural networks can be divided into two classes: (1) the continuous-time RNNs and (2) the discrete-time RNNs. By exploiting a numerical differential formula, a continuous-time RNN model can be discretized into a discrete-time one. However, a numerical differentiation rule does not necessarily generate a conver-

☆ This work is supported by the National Natural Science Foundation of China (with numbers 61401385 and 61563017), by the Fundamental Research Funds for the Central Universities (No. lzujbky-2017-37), by the Hunan Natural Science Foundation of China (2017JJ3257 and 2017JJ3258), by the Research Foundation of Education Bureau of Hunan Province, China (No. 17B215 and No. 17C1299), by Hong Kong Research Grants Council Early Career Scheme (with number 25214015), and also by Hong Kong Polytechnic University (with numbers G-YBMU, G-UA7L, 4-ZZHD, F-PP2C, 4-BCCS).

* Corresponding author.
E-mail addresses: jinlong@lzu.edu.cn, longjin@polyu.edu.hk (L. Jin), shuaili@polyu.edu.hk, lishuai8@gmail.com (S. Li), mulinliao8184@163.com (B. Liao), auzjzhang@scut.edu.cn (Z. Zhang).

http://dx.doi.org/10.1016/j.neucom.2017.06.030
0925-2312/© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

gent and stable discrete-time RNN model even though the original continuous-time RNN model is convergent. In addition, if the discrete-time RNN model is coded as a serial-processing program and performed on the digital computer, it can be considered as a numerical algorithm [27]. As a novel type of RNN specifically designed for solving time-varying problems, zeroing neural network (ZNN) is able to perfectly track time-varying solution by exploiting the time derivative of time-varying parameters. Then, many researchers make progresses along this direction by proposing various kinds of ZNN models for solving problems with different highlights. A detailed survey and summary are necessary for understanding the development of ZNN models as well as their applications. This paper is organized as follows. In Section 2, the descriptions and continuous-time ZNN models are presented, which include the evolution of models, activation functions, finite-time convergence, and integration-enhanced ZNN models. In Section 3, a brief review on the discrete-time ZNN models is presented. In Section 4, the applications of ZNN techniques are also analyzed. Section 5 concludes this paper with finial remarks.

2. Design formulas and various continuous-time models

Prior to the proposal of ZNN approach, many gradient-related methods had been reported on the solutions of algebraic equations and optimizations, i.e., zero-finding problems [28–31]. By constructing a performance index whose minimal point is identical to the solution to the task problem, a typical approach is to design a recurrent neural network evolving along the negative gradient descent to achieve a minimum of the performance index. However, these methods may fail to work well when exploited to the online solution of dynamic problems with time-varying coefficients, which is intrinsically due to the lacking of the compensation to the velocity components of the time-varying coefficients. Therefore, in view of the variability of coefficients, any method designed intrinsically for computing the static problem can no longer guarantee the decrease of the performance index of a time-varying problem, thereby possibly leading to a failure of the task with large residual error. For example, it is observed, investigated and analyzed in [32–34] that the residual error of gradient-based neural network (GNN) for solving a time-varying problem can not be eliminated and remains at a relative high level. Refs. [27,35,36] further point out that, when exploited to solve a time-varying problem, any traditional method that does not exploit the time-derivative information of time-varying coefficients can not converge to the theoretic solution with the residual error proportional to the value of the sampling gap.

To solve a time-varying problem in an error-free manner, Zhang et al. present a recurrent neural network for solving the time-varying Sylvester equation, which is depicted in an implicit dynamical system and can be deemed as the seminal work on ZNN [37]. They further generalize and summarize the design procedures of such a methodology, and analyze the convergence and stability of the corresponding ZNN model for time-varying matrix inversion in [38]. Specifically, for solving a time-varying matrix inversion problem depicted in the form of

$$A(t)X(t) = I, \tag{1}$$

where $A(t) \in \mathbb{R}^{(n\times n)}$ is a smooth matrix with its derivative assumed to be known, $I \in \mathbb{R}^{(n\times n)}$ is the identity matrix, $X(t) \in \mathbb{R}^{(n\times n)}$ is the unknown matrix to be obtained.

The core in the design of ZNN model is to construct an error function $E(t) = A(t)X(t) - I$, which is evidently different from the performance index of gradient-related methods. Then, the ZNN design formula is used to enforce the corresponding $E(t)$ to converge to zero:

$$\dot{E}(t) = -\gamma\Phi(E(t)), \tag{2}$$

Table 1
Continuous-time ZNN models constructed for solving time-varying problems.

	Dynamic problem	Error function	ZNN model
[41]	4th root finding $x^4(t) = a(t)$	$e(t) = x^4(t) - a(t)$	$\dot{x}(t) = \frac{\dot{a}(t)-\gamma\phi(x^4(t)-a(t))}{4x^3(t)}$
[44]	Linear system $A(t)\mathbf{x}(t) = \mathbf{b}(t)$	$\mathbf{e}(t) = A(t)\mathbf{x}(t) - \mathbf{b}(t)$	$A(t)\dot{\mathbf{x}}(t) = -\dot{A}(t)\mathbf{x}(t) + \dot{\mathbf{b}}(t) - \gamma\Phi(A(t)\mathbf{x}(t) - \mathbf{b}(t))$
[38]	Matrix inversion $A(t)X(t) = I$	$E(t) = A(t)X(t) - I$	$A(t)\dot{X}(t) = -\dot{A}(t)X(t) - \gamma\Phi(A(t)X(t) - I)$
[45]	Matrix square roots finding $X^2(t) = A(t)$	$E(t) = X^2(t) - A(t)$	$X(t)\dot{X}(t) + \dot{X}(t)X(t) = -\gamma\Phi(X^2(t) - A(t)) - \dot{A}(t)$
[46]	Nonlinear equations $\mathbf{f}(\mathbf{x}(t), t) = 0$	$\mathbf{e}(t) = \mathbf{f}(\mathbf{x}(t), t)$	$\dot{\mathbf{x}}(t) = -J^{-1}(\mathbf{x}(t), t)\left(\gamma\Phi(\mathbf{f}(\mathbf{x}(t), t)) + \frac{\partial\mathbf{f}(\mathbf{x}(t),t)}{\partial t}\right)$

where $\gamma > 0$ and $\Phi(\cdot)$ is a matrix array of activation function $\phi(\cdot)$. Similarly, for the vector-valued time-varying problems [39], e.g., the system of linear equation $A(t)\mathbf{x}(t) = \mathbf{b}(t)$, with $A(t) \in \mathbb{R}^{(m\times n)}$, $\mathbf{x}(t) \in \mathbb{R}^n$, and $\mathbf{b}(t) \in \mathbb{R}^m$, the error function can be designed as $\mathbf{e}(t) = A(t)\mathbf{x}(t) - \mathbf{b}(t)$. Even for the scalar-valued time-varying problems [40–43], e.g., the time-varying 4th root finding problem $x^4(t) = a(t)$, with $a(t) \in \mathbb{R}$ and $x(t) \in \mathbb{R}$, the error function can be designed as $e(t) = x^4(t) - a(t)$. Note that the matrix-valued (or vector-valued) error function is a decoupled system and thus its ijth (or ith) subsystem, i.e., the scale-valued dynamical system $\dot{e}(t) = -\gamma\phi(e(t))$, can be used to analyze the corresponding convergence and stability. By exploiting ZNN design formula to solve different time-varying problems, various ZNN models that exploit the time-derivative information of coefficients can be constructed with their formulations shown in Table 1. In short, any ZNN model for solving any time-varying problem can be deemed as an equivalently expansion of the ZNN design formula.

Since Zhang et al. proposed ZNN models in the 2000s, modified models have been frequently proposed by considering different internal and external factors. Especially, when nonlinear activation functions are incorporated into the network models, stability research has gained significant progress. A brief review on the design of continuous-time ZNN models for various problems solving is presented in [47]. However, with the rapid development of the theory on ZNN, new variations have taken place [48] and in the ensuing part, we will briefly review some basic models of ZNN from different perspective.

2.1. Convergence and stability

In the research of neural networks, the key issues are convergence and stability. Broadly speaking, there are three ways for proving the convergence of ZNN models, i.e., proof based on Lyapunov theory, ordinary differential equation (ODE), or Laplace transform.

(1) *Proof based on Lyapunov theory [49]*. For example, for the time-varying nonlinear minimization problem solving with the task function being $f(\mathbf{x}(t), t) \in \mathbb{R}$ and $\mathbf{x}(t) \in \mathbb{R}^n$ in [36], the error function can be designed as

$$\mathbf{e}(t) = \frac{\partial f(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}.$$

By constructing a Lyapunov function candidate:

$$V(t) = \frac{1}{2}\mathbf{e}^{\mathrm{T}}(t)\mathbf{e}(t),$$

it can be concluded that $V(t)$ is evidently of the positive-definiteness. Then, computing its time derivative leads to

$$\dot{V}(t) = -\gamma\mathbf{e}^{\mathrm{T}}(t)\mathbf{e}(t),$$

which is of negative-definiteness and we draw the conclusion that the residual error of the corresponding ZNN model globally converges to zero. It is worth noting that, by replacing the definition of error function $\mathbf{e}(t)$, the global convergence of all the existing continuous-time ZNN models can be analyzed from the similar way. This way is the dominant approach in the analysis of continuous-time ZNN models and has been widely studied in [36,50–52].

(2) *Proof based on ODE.* In addition to the convergent range, ODE-based approach can be used to prove the convergent speed of linear function activated ZNN models. For example, for the same problem shown in [36], by solving the ith subsystem of design formula, i.e., $\dot{e}(t) = -\gamma e(t)$, one can readily have

$$e(t) = e(0)\exp(-\gamma t),$$

where $e(0)$ is the initial value of $e(t)$. Then, we have the conclusion that the residual error of the ZNN model globally and exponentially to zero. This way has been widely studied in [36,38,39].

(3) *Proof based on Laplace transform.* Using Laplace transform to $\dot{e}(t) = -\gamma e(t)$ produces

$$se(s) - e(0) = -\gamma e(s),$$

and we further have

$$e(s) = \frac{e(0)}{s+\gamma}. \tag{3}$$

In view of $\gamma > 0$, it can be readily concluded that the final value theorem applies. Based on the final value theorem, we have

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} se(s) = \lim_{s\to 0} \frac{se(0)}{s+\gamma} = 0,$$

which completes the proof. This way is a new approach and has been preliminarily studied in [53–56].

2.2. Nonlinear activation functions

In the design and construction of ZNN models, nonlinear activation functions are used to accelerate the convergent speed. Typically, the following ones are often employed to construct ZNN models [45,57,58]:

- the power-sum activation function:

$$\phi(e_i) = \sum_{0}^{N} e_i^{2N-1},$$

where $N > 1$.

- the power-sigmoid activation function:

$$\phi(e_i) = \begin{cases} \dfrac{1+\exp(-\xi)}{1-\exp(-\xi)}\dfrac{1-\exp(-\xi e_i)}{1+\exp(-\xi e_i)}, & \text{if } |e_i| < 1, \\ e_i^p, & \text{if } |e_i| \geqslant 1, \end{cases}$$

where p is an odd integer and $\xi > 0$.

- and the hyperbolic sine activation function:

$$\phi(e_i) = \frac{\exp(e_i m)}{2} - \frac{\exp(-e_i m)}{2},$$

where m is an odd integer.

It is worth noting that, to prove the convergence of these nonlinear function activated ZNN models, a common approach is to construct a Lyapunov function candidate $V(t) = \mathbf{e}^{\mathrm{T}}(t)\mathbf{e}(t)/2$, and then compute its time derivative $\dot{V}(t)$, which is smaller than that of linear function activated ZNN model. Then, we draw the conclusion that nonlinear activation function can be used to accelerate the convergence speed. Many existing results on the ZNN concern the case that activation functions should be continuous, and strictly monotonically increasing, which is a limitation and should be remedied in the future.

2.3. Finite-time convergence

Due to in-depth research on the ZNN model and inspired by the study on finite-time convergence in continuous autonomous system [59–61], Li et al. present a nonlinear function to accelerate the continuous-time ZNN to finite-time convergence for solving time-varying Sylvester equation in [62] and then extend to the online solution of dual neural networks for solving quadratic programming problems in [63]. After that, many different finite-time activation functions have been proposed and employed to various ZNN models, e.g, time-varying matrix pseudoinversion in complex domain [64], Lyapunov equation [65], equality-constrained quadratic optimization [66], linear complex matrix equation [67] and so on [43,68–70]. Note that the residual error of a traditional ZNN model exponentially converges to zero, which means that the smaller the residual error is, the slower the convergent speed is. Therefore, for achieving finite-time convergence, an effective is to amplify the value of $\dot{e}(t)/e(t)$ to large enough when $e(t)$ approaches to zero. In addition, it is investigated in [71] that various ZNN models can be designed by exploiting different error functions for a time-varying problem solving, which can be accelerated to finite-time convergence.

2.4. Complex-valued ZNN models

In the past years, different ZNN models have received considerable studies in many scientific and engineering fields, which successfully tackles the estimation error problem in the real domain. In contrast to these ZNN models defined in the real domain, the research on complex-valued ZNN models shows advantage over conventional real-valued neural networks in complex problems solving. The first complex-valued ZNN model is proposed to solve the time-varying matrix-inversion problems in complex domain in [52]. To guarantee the global convergence of the neural network, only linear activation functions are considered in it. As pointed previously, many nonlinear function can be used to accelerate the convergence speed of ZNN model, which inspires Li et al. to explore nonlinear complex-valued activation functions to accelerate the convergence of ZNN with guaranteed global convergence in [72]. They find two classes of activation functions to achieve the global convergence of the complex-valued ZNN for solving the complex-valued time-varying Sylvester equation and then accelerate it to finite-time convergence. Liao et al. propose five ZNN models by exploiting different techniques to compute the time-varying complex-valued pseudoinversion problem in [73], which is further generalized to complex-valued matrix inversion in [74]. Qiao et al. present two complex-valued ZNN models for computing the Drazin inverse, which is accelerated to finite-time with proven upper bounds of the convergence time in [69]. In addition, as a comparison, online solution of complex-valued systems of linear equations is investigated in the complex domain via a GNN model in [75].

2.5. Noise-tolerant ZNN models

Many computational models for solving time-varying problems (including the conventional ZNN models) usually assume that the solving task is free of noises or that the noise reduction operation has been completed before the computation [76]. Note that, for the online computation of time-varying problems, time

Table 2
Continuous-time noise-tolerant ZNN models constructed for solving time-varying problems.

Dynamic problem	Noise-tolerant ZNN model
4th root finding $x^4(t)=a(t)$	$\dot{x}(t)=\dot{a}(t)-\gamma(x^4(t)-a(t))$ $-\lambda\int_0^t(x^4(\delta)-a(\delta)\mathrm{d}\delta/(4x^3(t))$
Linear system $A(t)\mathbf{x}(t)=\mathbf{b}(t)$	$A(t)\dot{\mathbf{x}}(t)=-\dot{A}(t)\mathbf{x}(t)+\dot{\mathbf{b}}(t)-\gamma(A(t)\mathbf{x}(t)-\mathbf{b}(t))$ $-\lambda\int_0^t(A(\delta)\mathbf{x}(\delta)-\mathbf{b}(\delta))\mathrm{d}\delta$
Matrix inversion $A(t)X(t)=I$	$A(t)\dot{X}(t)=-\dot{A}(t)X(t)-\gamma(A(t)X(t)-I)$ $-\lambda\int_0^t(A(\delta)X(\delta)-I)\mathrm{d}\delta$
Matrix square roots finding $X^2(t)=A(t)$	$X(t)\dot{X}(t)+\dot{X}(t)X(t)=-\gamma(X^2(t)-A(t))-\dot{A}(t)$ $-\lambda\int_0^t(X^2(\delta)-A(\delta))\mathrm{d}\delta$
Nonlinear equations $\mathbf{f}(\mathbf{x}(t),t)=0$	$\dot{\mathbf{x}}(t)=-J^{-1}(\mathbf{x}(t),t)\big(\gamma\Phi(\mathbf{f}(\mathbf{x}(t),t))+\frac{\partial\mathbf{f}(\mathbf{x}(t),t)}{\partial t}$ $\lambda\int_0^t(\mathbf{f}(\mathbf{x}(\delta),\delta))\mathrm{d}\delta\big)$

is precious and the preprocessing for denoising may consume extra time, thereby breaking the requirement for online computation. Therefore, an integration-enhanced ZNN design formula is proposed in [55] for time-varying matrix inversion, which leverages the integration-control technique in control theoretical and is able to handle simultaneously the noises. Then, such a noise-tolerant ZNN design formula is used to construct continuous-time ZNN model in [56] with nonlinear activation function exploited for accelerating the noise-tolerant model. In [54], the authors analyze and design different ZNN models via a systematic approach from the control perspective for solving time-varying problems. The essence of the noise-tolerant ZNN models is to leverage the error integration information to eliminate the constant bias errors. A challenging topic existing in this approach is how to determine the suitable activation functions for accelerating its convergence. By exploiting noise-tolerant ZNN design formula to solve different time-varying problems, various noise-tolerant ZNN models that exploit the time-derivative information of coefficients as well as the error integration can be constructed with their formulations shown in Table 2.

3. Numerical formulas and various discrete-time models

As reviewed previously, various continuous-time ZNN models have been presented for solving different problems. However, due to the fact that step size in simulating continuous-time systems is variable and that digital computer often requires constant time step, it is different for digital computer to implement continuous-time ZNN models directly [77–80]. In addition, for continuous-time ZNN models, it is always assumed that neurons communicated and responded instantaneously and without any delay. However, in digital circuits, time delay is unavoidable due to the existing of the sampling gap. Thus, researchers devote their effects to propose and investigate discrete-time ZNN models for online time-varying problems solving. In addition, some challenges of developing the discrete-time ZNN models from the continuous-time ones for the time-varying problems solving are listed as follows.

1. From the perspective of time, any time-varying problem can be deemed as a causal system and thus, the related computation should be conducted based on the existing data, i.e., the present and/or previous data, due to the unavailability of future data but for computing the future solution. For example, for solving the time-varying matrix inversion problem 1 in a discrete manner, at the time instant t_k, we can use the known information, e.g., $A(t_k)$ and $\dot{A}(t_k)$, not the unknown information, e.g., $A(t_{k+1})$ and $\dot{A}(t_{k+1})$ for computing the inverse of $A(t_{k+1})$, i.e., $X(t_{k+1})$. Thus, a fundamental requirement for constructing the discrete-time ZNN model is that we can not use future data.
2. Accordingly, a feasible numerical differentiation formula for discretizing continuous-time ZNN model should have and only have one point ahead of the target point. Consequently, when the numerical differentiation formula is used to discretize the continuous-time ZNN model, the constructed discrete-time ZNN model has one and only one unknown point $X(t_{k+1})$ to be computed via the known data (e.g., $A(t_k)$ and $\dot{A}(t_k)$). Therefore, the backward and multiple-point central differentiation rules can not be used to construct discrete-time ZNN models, no matter how tiny the truncation error of each formula is. Moreover, only the forward differentiation formulas with one step ahead can be considered for the discretization of ZNN.
3. Computation consumes time inevitably at each time instant and time is precious for the time-varying problems solving in practice. So, how to design a very simple discrete-time ZNN model with less calculation time is important. In other words, as Leonardo da Vinci said, ‘simplicity is the ultimate sophistication’.

The first discrete-time ZNN model is proposed in [81] for constant matrix inversion, which bridges the gap between discrete-time ZNN model and the traditional Newton iteration. At the early stage, Euler forward difference is used to construct the discretize the continuous-time ZNN model [82,83]. The discrete-time ZNN models generated by Euler forward difference are of the error pattern of $O(\tau^2)$, where τ denotes the sampling gap. For example, the Euler-type discrete-time ZNN model for time-varying matrix inversion is directly given as [84]:

$$X_{k+1}=X_k-\tau X_k\dot{A}_kX_k-hX_k(A_kX_k-I), \tag{4}$$

where k denotes the iteration index and $h=\tau\gamma>0$. In addition, by omitting the term $\tau X_k\dot{A}_kX_k$ and letting $h=1$, the Euler-type discrete-time ZNN model (4) reduces to

$$X_{k+1}=X_k-X_k(A_kX_k-I), \tag{5}$$

which is the Newton iteration. In other words, the traditional Newton iteration can be deemed as a special case of Euler-type discrete-time ZNN model (4). Meanwhile, the link between Getz–Marsden dynamic system and discrete-time ZNN models is found in [85]. To achieving high accuracy in the discretization of ZNN models, a Taylor-type numerical differentiation formula is proposed in [84] for the first-order derivative approximation, which has a truncation error of $O(\tau^2)$ and formulated as

$$f'(t_k)=\frac{2f(t_{k+1})-3f(t_k)+2f(t_{k-1})-f(t_{k-2})}{2\tau}+O(\tau^2). \tag{6}$$

A new Taylor-type discrete-time ZNN model can be developed for time-varying matrix inversion as

$$X_{k+1}=-\tau X_k\dot{A}_kX_k-hX_k(A_kX_k-I)+\frac{3}{2}X_k-X_{k-1}+\frac{1}{2}X_{k-2}. \tag{7}$$

The above Taylor-type discrete-time ZNN model converges to the theoretical solution of the time-varying problem with the residual error being $O(\tau^3)$. Recently, a numerical difference rule is established in [86] for first-order derivative approximation with the truncation error of $O(\tau^3)$. Based on such a new formula, a five-step discrete-time ZNN model is proposed for time-varying matrix inversion, of which the residual error is $O(\tau^4)$. Note that, the discrete-time ZNN models can be deemed as time delay systems, and thus, a large value of h may lead the model to oscillate. To remedy the instability, a direct way is to lessen the value of h. However, lessening the value of h would significantly slow the convergence of discrete-time ZNN models.

4. Applications

As a systematic approach for solving time-varying zero-finding problems, ZNN design formula as well as the derived models has been applied to the motion generation and control of redundant manipulators, the tracking control of chaotic systems or even the population control in biosciences. Along the above lines, we will give a detailed review on the application research of ZNN in this section.

4.1. Robotic applications

The joint-drift problem existing in the motion generation and control of redundant manipulators is that, after completing a closed path in the workspace, the joint variables do not go back to their initial values [87]. Researchers exploit the ZNN approach to remedy such a weakness. For a redundant robot manipulator, we have [88–92]:

$$\mathbf{f}(\theta) = \mathbf{r}, \tag{8}$$

$$J(\theta)\dot{\theta} = \dot{\mathbf{r}}, \tag{9}$$

where $\mathbf{r} \in \mathbb{R}^m$ is the orientation vector to be achieved; $\mathbf{f}(\theta)$ denotes the end-effector position; Differentiating Equation (9) with respect to time t leads to (8). To eliminate the joint displacement $\theta(T) - \theta(0)$, where $\theta(T)$ and $\theta(0)$ denote the values of joint variables at the final state and initial state, respectively, a direction based on ZNN design formula is to construct an error function $\epsilon(t) = \theta(t) - \theta(0)$ and then employ the ZNN design formula to force $\epsilon(t)$ converging to zero. Then, we have $\dot{\theta}(t) = -\gamma(\theta(t) - \theta(0))$. For considering the equality and inequality constraints, it is better to minimize the performance index $\|\dot{\theta}(t) + \gamma(\theta(t) - \theta(0))\|_2^2/2$, rather than using $\dot{\theta}(t) + \gamma(\theta(t) - \theta(0)) = 0$ directly. Thus, the repetitive motion generation scheme based on ZNN method can be formulated as

$$\text{minimize} \quad \|\dot{\theta}(t) + \gamma(\theta(t) - \theta(0))\|_2^2/2 \tag{10}$$

$$\text{subject to} \quad J(\theta)\dot{\theta} = \dot{\mathbf{r}}, \tag{11}$$

$$\theta_l^- \leqslant \theta_l \leqslant \theta_l^+, \tag{12}$$

$$\dot{\theta}_l^- \leqslant \dot{\theta}_l \leqslant \dot{\theta}_l^+. \tag{13}$$

It is investigated in [93] that the above scheme is reformulated as a quadratic programming problem. Moreover, different RNN models are investigated comparatively for solving such a problem. In general, the redundancy-resolution problem arising in the motion planning and control of redundant manipulators can be solved at the joint-velocity level or at the joint-acceleration level, resulting in the corresponding velocity-level and acceleration-level redundancy-resolution schemes. Then, to solve the joint-angle drift problems in repetitive motion control of redundant robot manipulators at the level of acceleration, a scheme called acceleration-level drift-free (ALDF) scheme subject to a linear equality constraint is presented in [94] with guaranteed effectiveness. Reference [95] point out that there existing equivalent relationship between joint-velocity level and joint-acceleration level via the ZNN-related techniques. Miao et al. present ZNN models for solving time-varying quadratic program problems and applied to robot tracking [96], where the models are of finite-time convergence. A noise-tolerant ZNN model is designed and presented in [97] for the motion generation of multiple redundant robot manipulators in a distributed manner, which can be deemed as a seminal work on ZNN with application to distributed systems. The Jacobian equality constraints for multiple manipulators in [97] are compacted into an equality based on consensus with limited communications, and the scheme could coordinately control all manipulators involved to complete a given task with the aid of ZNN, as long as the communication topological graph is connected.

4.2. Chaos applications

For the tracking-control problems of two chaotic systems, many methods fail to solve it due to the existence of singularities. For demonstration, the Lu system equipped with a single control input u is presented. Let us consider the following chaotic system with input u:

$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = -xz + cy + u, \\ \dot{z} = xy - bz, \end{cases} \tag{14}$$

where $a = 36,\ b = 3$ and $c = 20$; $\vartheta = z$ denotes the output of system (14). The aim is to construct a controller such that ϑ tracks time-varying desired path ϑ_d, with $e = \vartheta - \vartheta_\mathrm{d}$ approaching zero.

To adopt ZNN design formula, the first error function is designed as

$$e_1 = \vartheta - \vartheta_\mathrm{d} = z - \vartheta_\mathrm{d},$$

where ϑ_d denotes the desired trajectory. In addition, the following ZNN design formula is provided:

$$\dot{e}_1 = -\gamma e_1. \tag{15}$$

Expanding the above design formula leads to

$$xy + (\gamma - b)z - \dot{\vartheta}_\mathrm{d} - \gamma\vartheta_\mathrm{d} = 0. \tag{16}$$

To get an expression of u in (16) explicitly, the following second error function is constructed: $e_2 = xy + (\gamma - b)z - \dot{\vartheta}_\mathrm{d} - \gamma\vartheta_\mathrm{d}$. Then, we have

$$\dot{x}y + x\dot{y} + (\gamma - b)\dot{z} - \ddot{\vartheta}_\mathrm{d} - \gamma\dot{\vartheta}_\mathrm{d} = -\gamma(xy + (\gamma - b)z - \dot{\vartheta}_\mathrm{d} - \gamma\vartheta_\mathrm{d}). \tag{17}$$

With the aid of (14), we have

$$x^2z + (a + b - c - 2\gamma)xy - ay^2 - xu + g_2 = 0, \tag{18}$$

where $g_2 = (2\gamma b - b^2 - \gamma^2)z + \ddot{\vartheta}_\mathrm{d} + 2\gamma\dot{\vartheta}_\mathrm{d} + \gamma^2\vartheta_\mathrm{d}$. Thus, a controller based on ZNN design formula is obtained:

$$u = \frac{1}{x}\left(x^2z + (a + b - c - 2\gamma)xy - ay^2 + g_2\right), \tag{19}$$

which has a singularity plane $x = 0$. To conquer it, the idea based on ZNN-related approach is to eliminate the division operation by transforming the direct control (19) into a minimization problem. By defining $h = x^2z + (a + b - c - 2\mu)xy - ay^2 - xu + g_2$ and based on the gradient method, a tracking controller in the form of $\dot{u}$ can be designed as

$$\dot{u} = \gamma xh, \tag{20}$$

which is free of division operation. In view of the fact that such a controller-design method is consisted of ZNN method and GNN method, it is termed ZG neural dynamics in [98]. Such a method is extended to the singularity-conquering of inverted pendulum on a cart system in [99], the tracking control of chaotic systems with the mixture of additive and multiplicative inputs [100].

4.3. Other applications

In addition to the robotics as well as chaos, ZNN is also applied to the multi-dimensional spectral estimation. For avoiding the computation of direct inverse of covariance matrix, Benchabane et al. use discrete-time ZNN models to compute the inverse of covariance matrix online in [79]. To handle the output tracking control of general-form single-input single- output nonlinear system, controllers based on ZNN are designed in [101], which can avoid the division-by-zero problem. In [102], discrete-time ZNN models are used for motion estimation for computing 2D optical flow. In addition, new fractals are yielded by using the discrete-time complex-valued ZNN for solving time-varying nonlinear equations in the complex domain in [103–105]. The population control of the Lotka-Volterra model in mathematical ecology based on the ZNN-related techniques is investigated in [106]. The presented controller is able to drive the prey population and/or predator population to a desired state.

5. Conclusion

In summary, the studies for ZNN have achieved a great deal in the last 16 years. However, there are still many new problems to be solved. For future directions of the study on zeroing neural networks as well as their applications, we now provide some prospective suggestions.

1. Continue to apply and find some useful one-step ahead numerical differential formulas to discretize the continuous-time ZNN models, especially to further expand the step-size in the existing discrete ZNN results while keeping a high computational accuracy. This direction is closely related to the development of applied mathematics and computational mathematics.
2. How to construct nonlinear activation functions to accelerate the convergence speed of zeroing neural networks for solving complex-valued problems is still an open problem. For the case of real-valued problems, a great deal of nonlinear activations have been proposed and investigated. Moreover, how to obtain the convergence conditions is also meaningful in the development of zeroing neural networks.
3. In addition to the global stability property, how to construct and analyze the stability criteria for noise-tolerant and nonconvex-allowed zeroing neural networks still needs more efforts. In general, these modified zeroing neural networks are highly related to the control techniques [107,108].

All these future developments will accompany the development of mathematical theory, especially applied mathematics and computational mathematics. For example, new models should be derived for solving the inequality constraint in time-varying optimization problems. All these future developments will accompany the development of the computational mathematics for constructing and developing neural networks as well as the techniques for various kinds of robot manipulators. Keeping in mind, different kinds of neural networks e.g., discrete-time ZNN models or even the gradient neural network models, have their own feasible ranges, and one cannot expect that only a few existing results on neural networks can tackle all the computational problems.

References

[1] Y.-J. Liu, S. Tong, D.-J. Li, Y. Gao, Fuzzy adaptive control with state observer for a class of nonlinear discrete-time systems with input constraint, IEEE Trans. Fuzzy Syst. 24 (5) (2016) 1147–1158.
[2] Y.-J. Liu, S. Tong, Adaptive fuzzy identification and control for a class of nonlinear pure-feedback MIMO systems with unknown dead zones, IEEE Trans. Fuzzy Syst. 23 (5) (2015) 1387–1398.
[3] H. Wang, B. Chen, K. Liu, X. Liu, C. Lin, Adaptive neural tracking control for a class of nonstrict-feedback stochastic nonlinear systems with unknown backlash-like hysteresis, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 947–958.
[4] Y.-J. Liu, S. Tong, Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems, IEEE Trans. Cybern. 46 (11) (2016) 2670–2680.
[5] Y.-J. Liu, J. Li, S. Tong, C.P. Chen, Neural network control-based adaptive learning design for nonlinear systems with full-state constraints, IEEE Trans. Neural Netw. Learn. Syst. 27 (7) (2016) 1562–1571.
[6] S. Li, J. He, Y. Li, M.U. Rafique, Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 415–426.
[7] 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.
[8] S. Li, B. Liu, Y. Li, Selective positive–negative feedback produces the winner-take-all competition in recurrent neural networks, IEEE Trans. Neural Netw. Learn. Syst. 24 (2) (2013) 301–309.
[9] Y. Zhang, D. Guo, Z. Luo, K. Zhai, H. Tan, CP-activated WASD neuronet approach to asian population prediction with abundant experimental verification, Neurocomputing 198 (2016) 48–57.
[10] X. Luo, M. Shang, Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications, in: Proceedings of the IEEE 16th International Conference on Data Mining, IEEE, 2016, pp. 311–319.
[11] Y.-A. Huang, Z.-H. You, X. Li, X. Chen, P. Hu, X. Luo, Construction of reliable protein–protein interaction networks using weighted sparse representation based classifier with pseudo substitution matrix representation features, Neurocomputing 218 (2016) 131–138.
[12] X. Luo, M. Zhou, Regularizaed extraction of non-negative latent factors from high-dimensional sparse matrices, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2016, pp. 001221–001226.
[13] H. Wang, P.X. Liu, S. Liu, Adaptive neural synchronization control for bilateral teleoperation systems with time delay and backlash-like hysteresis, IEEE Trans. Cybern. (2017).
[14] H. Wang, W. Liu, P.X. Liu, H. Lam, Adaptive fuzzy decentralized control for a class of interconnected nonlinear system with unmodeled dynamics and dead zones, Neurocomputing 214 (2016) 972–980.
[15] P.S. Stanimirović, I.S. Živković, Y. Wei, Recurrent neural network approach based on the integral representation of the Drazin inverse, Neural Comput. 27 (10) (2015) 2107–2131.
[16] I.S. Živković, P.S. Stanimirović, Y. Wei, Recurrent neural network for computing outer inverse, Neural Comput. 28 (5) (2016) 970–998.
[17] X.-Z. Wang, H. Ma, P.S. Stanimirović, Nonlinearly activated recurrent neural network for computing the Drazin inverse, Neural Process. Lett. (2017) 1–23.
[18] Z. Zhang, S. Li, X. Zhang, Simulink comparison of varying-parameter convergent-differential neural-network and gradient neural network for solving online linear time-varying equations, in: Proceedings of the 12th World Congress on Intelligent Control and Automation, IEEE, 2016, pp. 887–894.
[19] M. Di Marco, M. Forti, P. Nistri, L. Pancioni, Discontinuous neural networks for finite-time solution of time-dependent linear equations, IEEE Trans. Cybern. 46 (11) (2016) 2509–2520.
[20] X. Wang, M. Che, Y. Wei, Recurrent neural network for computation of generalized eigenvalue problem with real diagonalizable matrix pair and its applications, Neurocomputing 216 (2016) 230–241.
[21] X. Luo, M. Zhou, Z. You, Y. Xia, Q. Zhu, A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method, IEEE Trans. Neural Netw. Learn. Syst. 27 (3) (2016) 579–592.
[22] Z. Zhang, L. Zheng, J. Yu, Y. Li, Z. Yu, Three recurrent neural networks and three numerical methods for solving repetitive motion planning scheme of redundant robot manipulators, IEEE/ASME Trans. Mechatron. (2017).
[23] X. Luo, M. Zhou, Y. Xia, Z.-H. You, Q. Zhu, H. Leung, Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data, IEEE Trans. Cybern. (2017).
[24] X. Luo, M. Zhou, S. Li, Y. Xia, Z. You, Q. Zhu, H. Leung, An efficient second-order approach to factorize sparse matrices in recommender systems, IEEE Trans. Ind. Inform. 11 (4) (2015) 946–956.
[25] J. Wang, A recurrent neural network for real-time matrix inversion, Appl. Math. Comput. 55 (1) (1993) 89–100.
[26] L. Cheng, Z.-G. Hou, M. Tan, A simplified neural network for linear matrix inequality problems, Neural Process. Lett. 29 (3) (2009) 213–230.
[27] 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. (2016) 1–16.
[28] J. Wang, G. Wu, Recurrent neural networks for synthesizing linear control systems via pole placement, Int. J. Syst. Sci. 26 (12) (1995) 2369–2382.
[29] Y. Zhang, J. Wang, Recurrent neural networks for nonlinear output regulation, Automatica 37 (8) (2001) 1161–1173.
[30] Y. Zhang, Y. Shi, K. Chen, C. Wang, Global exponential convergence and stability of gradient-based neural network for online matrix inversion, Appl. Math. Comput. 215 (3) (2009) 1301–1306.
[31] K. Chen, Z. Zhang, An improved recurrent network for online equality-constrained quadratic programming, in: Proceedings of the 8th International Conference on Advances in Brain Inspired Cognitive Systems: Beijing, China, November 28–30, 2016, Springer, 2016, pp. 1–10.
[32] Y. Zhang, K. Chen, H.-Z. Tan, Performance analysis of gradient neural network exploited for online time-varying matrix inversion, IEEE Trans. Automatic Control 54 (8) (2009) 1940–1945.
[33] Y. Zhang, Y. Yang, G. Ruan, Performance analysis of gradient neural network exploited for online time-varying quadratic minimization and equality-constrained quadratic programming, Neurocomputing 74 (10) (2011) 1710–1719.
[34] Y.-N. Zhang, C. Zeng-Hai, C. Ke, Convergence properties analysis of gradient neural network for solving online linear equations, Acta Automatica Sinica 35 (8) (2009) 1136–1139.
[35] L. Jin, Y. Zhang, Discrete-time Zhang neural network of $O(\tau^3)$ pattern for time-varying matrix pseudoinversion with application to manipulator motion generation, Neurocomputing 142 (2014) 165–173.
[36] L. Jin, Y. Zhang, Continuous and discrete Zhang dynamics for real-time varying nonlinear optimization, Numer. Algorithms 73 (1) (2016) 115–140.

[37] Y. Zhang, D. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, IEEE Trans. Neural Netw. 13 (5) (2002) 1053–1063.
[38] Y. Zhang, S.S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, IEEE Trans. Neural Netw. 16 (6) (2005) 1477–1490.
[39] D. Guo, C. Yi, Y. Zhang, Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving, Neurocomputing 74 (17) (2011) 3708–3712.
[40] Y. Zhang, Z. Ke, P. Xu, C. Yi, Time-varying square roots finding via Zhang dynamics versus gradient dynamics and the Former's link and new explanation to Newton–Raphson iteration, Inf. Process. Lett. 110 (24) (2010) 1103–1109.
[41] Y. Zhang, L. Xiao, G. Ruan, Z. Li, Continuous and discrete time Zhang dynamics for time-varying 4th root finding, Numer. Algorithms 57 (1) (2011) 35–51.
[42] C. Yi, Y. Chen, H. Wang, Simulation and verification of Zhang neural networks and gradient neural networks for time-varying stein equation solving, in: Proceedings of the International Symposium on Neural Networks, Springer, 2011, pp. 385–392.
[43] C.F. Yi, Y.H. Liu, Online solution of time-varying Lyapunov matrix equation by Zhang neural networks, Recent Pat. Comput. Sci. 6 (1) (2013) 25–32.
[44] Y. Zhang, Y. Wang, B. Mu, H. Zheng, Different ZFs leading to various ZNN models illustrated via online solution of time-varying underdetermined systems of linear equations with robotic application, in: Proceedings of the International Symposium on Neural Networks, Springer, 2013, pp. 481–488.
[45] Y. Zhang, Z. Ke, Superior performance of using hyperbolic sine activation functions in ZNN illustrated via time-varying matrix square roots finding, Comput. Sci. Inf. Syst. 9 (4) (2012) 1603–1625.
[46] Y. Zhang, C. Yi, D. Guo, J. Zheng, Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation, Neural Comput. Appl. 20 (1) (2011) 1–7.
[47] D. Guo, Y. Zhang, Novel recurrent neural network for time-varying problems solving [research frontier], IEEE Comput. Intell. Mag. 7 (4) (2012) 61–65.
[48] L. Xiao, Y. Zhang, Zhang neural network versus gradient neural network for solving time-varying linear inequalities, IEEE Trans. Neural Netw. 22 (10) (2011) 1676–1684.
[49] Y. Zhang, Z. Li, Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints, Phys. Lett. A 373 (18) (2009) 1639–1643.
[50] C. Peng, Y. Ling, Y. Wang, X. Yu, Y. Zhang, Three new ZNN models with economical dimension and exponential convergence for real-time solution of Moore-Penrose pseudoinverse, in: Proceedings of the International Joint Conference on Neural Networks, IEEE, 2014, pp. 2788–2793.
[51] D. Guo, C. Peng, Y. Ling, Y. Zhang, Different ZFs lead to different nets: examples of Zhang generalized inverse, in: Proceedings of the Chinese Automation Congress, IEEE, 2013, pp. 453–458.
[52] Y. Zhang, Z. Li, K. Li, Complex-valued Zhang neural network for online complex-valued time-varying matrix inversion, Appl. Math. Comput. 217 (24) (2011) 10066–10073.
[53] M. Liu, B. Liao, L. Ding, L. Xiao, Performance analyses of recurrent neural network models exploited for online time-varying nonlinear optimization, Comput. Sci. Inf. Syst. 13 (2) (2016) 691–705.
[54] 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. Automatic Control 62 (2) (2017) 992–997.
[55] 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.
[56] 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.
[57] Y. Yang, Y. Zhang, Superior robustness of power-sum activation functions in Zhang neural networks for time-varying quadratic programs perturbed with large implementation errors, Neural Comput. Appl. 22 (1) (2013) 175–185.
[58] C. Yi, Y. Chen, X. Lan, Comparison on neural solvers for the Lyapunov matrix equation with stationary & nonstationary coefficients, Appl. Math. Model. 37 (4) (2013) 2495–2502.
[59] S.P. Bhat, D.S. Bernstein, Finite-time stability of continuous autonomous systems, SIAM J. Control Optim. 38 (3) (2000) 751–766.
[60] M.D. Marco, M. Forti, M. Grazzini, Robustness of convergence in finite time for linear programming neural networks, Int. J. Circ. Theory Appl. 34 (3) (2006) 307–316.
[61] 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.
[62] S. Li, S. Chen, B. Liu, Accelerating a recurrent neural network to finite–time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function, Neural Process. Lett. 37 (2) (2013) 189–205.
[63] S. Li, Y. Li, Z. Wang, A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application, Neural Netw. 39 (2013) 27–39.
[64] B. Liao, Y. Zhang, From different ZFS to different ZNN models accelerated via li activation functions to finite-time convergence for time-varying matrix pseudoinversion, Neurocomputing 133 (2014) 512–522.
[65] L. Xiao, B. Liao, A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation, Neurocomputing 193 (2016) 213–218.
[66] L. Xiao, A nonlinearly-activated neurodynamic model and its finite-time solution to equality-constrained quadratic optimization with nonstationary coefficients, Appl. Soft Comput. 40 (2016) 252–259.
[67] L. Xiao, A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation, Neurocomputing 167 (2015) 254–259.
[68] Y. Shen, P. Miao, Y. Huang, Y. Shen, Finite-time stability and its application for solving time-varying Sylvester equation by recurrent neural network, Neural Process. Lett. 42 (3) (2015) 763–784.
[69] S. Qiao, X.-Z. Wang, Y. Wei, Two finite-time convergent Zhang neural network models for time-varying complex matrix Drazin inverse, Linear Algebra Appl. (2017).
[70] D. Guo, K. Li, L. Yan, Z. Nie, F. Jin, The application of li-function activated RNN to acceleration-level robots' kinematic control via time-varying matrix inversion, in: Proceedings of the Chinese Control and Decision Conference (CCDC), IEEE, 2016, pp. 3455–3460.
[71] L. Xiao, Y. Zhang, From different Zhang functions to various ZNN models accelerated to finite-time convergence for time-varying linear matrix equation, Neural Process. Lett. 39 (3) (2014) 309–326.
[72] S. Li, Y. Li, Nonlinearly activated neural network for solving time-varying complex Sylvester equation, IEEE Trans. Cybern. 44 (8) (2014) 1397–1407.
[73] B. Liao, Y. Zhang, Different complex ZFS leading to different complex ZNN models for time-varying complex generalized inverse matrices, IEEE Trans. Neural Netw. Learn. Syst. 25 (9) (2014) 1621–1631.
[74] B. Liao, L. Xiao, J. Jin, L. Ding, M. Liu, Novel complex-valued neural network for dynamic complex-valued matrix inversion, JACIII 20 (1) (2016) 132–138.
[75] L. Xiao, W. Meng, R. Lu, X. Yang, B. Liao, L. Ding, A fully complex-valued neural network for rapid solution of complex-valued systems of linear equations, in: Proceedings of the International Symposium on Neural Networks, Springer, 2015, pp. 444–451.
[76] K. Chen, C. Yi, Robustness analysis of a hybrid of recursive neural dynamics for online matrix inversion, Appl. Math. Comput. 273 (2016) 969–975.
[77] Y. Zhang, D. Guo, S. Fu, L. Xiao, Three nonlinearly-activated discrete-time ZNN models for time-varying matrix inversion, in: Proceedings of theEighth International Conference on Natural Computation, IEEE, 2012, pp. 163–167.
[78] Y. Zhang, L. Xiao, Z. Xiao, M. Mao, Zeroing Dynamics, Gradient Dynamics, and Newton Iterations, CRC Press, 2016.
[79] A. Benchabane, A. Bennia, F. Charif, A. Taleb-Ahmed, Multi-dimensional Capon Spectral estimation using discrete Zhang neural networks, Multidimens. Syst. Signal Process. 24 (2013) 1–16.
[80] Y. Zhang, Y. Zhang, D. Chen, Z. Xiao, X. Yan, From Davidenko method to Zhang dynamics for nonlinear equation systems solving, IEEE Trans. Syst. Man Cybern.: Syst. (2016).
[81] Y. Zhang, B. Cai, M. Liang, W. Ma, On the variable step-size of discrete-time Zhang neural network and newton iteration for constant matrix inversion, in: Proceedings of the Second International Symposium on Intelligent Information Technology Application, 1, IEEE, 2008, pp. 34–38.
[82] Y. Zhang, W. Ma, B. Cai, From Zhang neural network to newton iteration for matrix inversion, IEEE Trans. Circ. Syst. I: Regul. Pap. 56 (7) (2009) 1405–1415.
[83] M. Mao, J. Li, L. Jin, Y. Zhang, Enhanced discrete-time Zhang neural network for time-variant matrix inversion in the presence of bias noises, Neurocomputing 207 (2016) 220–230.
[84] Y. Zhang, D. Guo, Y. Yin, Y. Chou, Taylor-type 1-step-ahead numerical differentiation rule for first-order derivative approximation and ZNN discretization, J. Comput. Appl. Math. 273 (2015) 29–40.
[85] 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.
[86] D. Guo, Z. Nie, L. Yan, Novel discrete-time Zhang neural network for time–varying matrix inversion, IEEE Trans. Syst. Man Cybern.: Syst. (2017).
[87] Y. Zhang, D. Guo, B. Cai, K. Chen, Remedy scheme and theoretical analysis of joint-angle drift phenomenon for redundant robot manipulators, Robot. Comput.-Integr. Manuf. 27 (4) (2011) 860–869.
[88] Z. Zhang, Y. Zhang, Acceleration-level cyclic-motion generation of constrained redundant robots tracking different paths, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 42 (4) (2012) 1257–1269.
[89] L. Jin, S. Li, H.M. La, X. Luo, Manipulability optimization of redundant manipulators using dynamic neural networks, IEEE Trans. Ind. Electron. (2017).
[90] L. Jin, Y. Zhang, G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms, IEEE Trans. Cybern. 45 (2) (2015) 153–164.
[91] L. Jin, S. Li, Distributed task allocation of multiple robots: A control perspective, IEEE Trans. Syst. Man Cybern.: Syst. (2016).
[92] S. Li, Y. Zhang, L. Jin, Kinematic control of redundant manipulators using neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2016).
[93] Y. Zhang, Z. Tan, K. Chen, Z. Yang, X. Lv, Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulator's straight-line example, Robot. Autonomous Syst. 57 (6) (2009) 645–651.
[94] Z. Zhang, Y. Zhang, Design and experimentation of acceleration-level drift-free scheme aided by two recurrent neural networks, IET Control Theory Appl. 7 (1) (2013) 25–42.
[95] B. Cai, Y. Zhang, Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and Zhang's neural-dynamic methods, IEEE Trans. Ind. Electron. 59 (8) (2012) 3146–3155.

[96] P. Miao, Y. Shen, Y. Huang, Y.-W. Wang, Solving time-varying quadratic programs based on finite-time Zhang neural networks and their application to robot tracking, Neural Comput. Appl. 26 (3) (2015) 693–703.
[97] 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. (2017).
[98] Y. Zhang, Z. Xiao, D. Guo, M. Mao, Y. Yin, Singularity-conquering tracking control of a class of chaotic systems using Zhang-gradient dynamics, IET Control Theory Appl. 9 (6) (2015) 871–881.
[99] Y. Zhang, X. Yu, Y. Yin, C. Peng, Z. Fan, Singularity-conquering ZG controllers of z2g1 type for tracking control of the IPC system, Int. J. Control 87 (9) (2014) 1729–1746.
[100] 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.
[101] J. Li, M. Mao, Y. Zhang, D. Chen, Y. Yin, Zd, ZG and IOL controllers and comparisons for nonlinear system output tracking with DBZ problem conquered in different relative-degree cases, Asian J. Control (2017).
[102] F. Charif, A. Benchabane, N. Djedi, A. Taleb-Ahmed, Horn & Schunck meets a discrete Zhang neural networks for computing 2d optical flow, Matrix 1 (1) (2009) 1–10.
[103] H. Wu, F. Li, Z. Li, Y. Zhang, Zhang fractals yielded via solving nonlinear equations by discrete-time complex-valued ZD, in: Proceedings of the IEEE International Conference on Automation and Logistics, IEEE, 2012, pp. 1–6.
[104] Y. Zhang, L. Jin, Z. Zhang, L. Xiao, S. Fu, Zhang fractals yielded via solving time-varying nonlinear complex equations by discrete-time complex-valued zd, in: International Conference on Artificial Intelligence and Computational Intelligence, Springer, 2012, pp. 596–603.
[105] Y. Zhang, T. Qiao, B. Liao, J. Chen, From newton fractals to gradient fractals in addition to zhang fractals, in: Proceedings of the IEEE 17th International Conference on Computational Science and Engineering, IEEE, 2014, pp. 1237–1242.
[106] Y. Zhang, X. Yan, B. Liao, Y. Zhang, Y. Ding, Z-type control of populations for Lotka–Volterra model with exponential convergence, Math. Biosci. 272 (2016) 15–23.
[107] L. Jin, S. Li, Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints, Neurocomputing (2017).
[108] L. Jin, S. Li, B. Hu, RNN models for dynamic matrix inversion: A control-theoretical perspective, IEEE Trans. Ind. Informat. (2017).

Long Jin received the B.E. degree and the PhD degree from Sun Yat-sen University, Guangzhou, China, in 2011 and in 2016, respectively. He is currently a Postdoctoral Fellow at the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. He also works in Lanzhou University. 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.

Bolin Liao received the B.S. degree in Electronic Information Engineering from Jishou University, Jishou, China, in 2003, the M.S. degree from the Anshan University of Science and Technology, Anshan, China, in 2006 and the Ph.D. degree in Communication and Information Systems from Sun Yat-Sen University, Guangzhou, China, in 2015. He is currently an associate professor with the College of Information Science and Engineering, Jishou University. His main research interests include neural networks, robotics, and process control.

Zhijun Zhang received the Ph.D. degree in Communication and Information System from Sun Yat-sen University, Guangzhou, China, in 2012. He was a Post-Doctoral Research Fellow with the Institute for Media Innovation in Nanyang Technological University, Singapore in 2013–2015. Since 2015, he has been an Associate Professor with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include neural networks, robotics, and human-robot interaction.