Neural Comput & Applic (2011) 20:1–7
DOI 10.1007/s00521-010-0452-y

ORIGINAL ARTICLE

Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation

Yunong Zhang · Chenfu Yi · Dongsheng Guo · Jinhuan Zheng

Received: 21 July 2008 / Accepted: 16 September 2010 / Published online: 6 October 2010

Abstract Different from gradient-based neural dynamics, a special kind of recurrent neural dynamics has recently been proposed by Zhang et al. for solving online time-varying problems. Such a recurrent neural dynamics is designed based on an indefinite error-monitoring function instead of a usually norm- or square-based energy function. In addition, Zhang neural dynamics (ZND) are depicted generally in implicit dynamics, whereas gradient-based neural dynamics (GND) are associated with explicit dynamics. In this paper, we generalize the ZND design method to solving online nonlinear time-varying equations in the form of $f(x, t) = 0$. For comparative purposes, the GND model is also employed for such time-varying equations' solving. Computer-simulation results via power-sigmoid activation functions substantiate the theoretical analysis and efficacy of the ZND model for solving online nonlinear time-varying equations.

Keywords Recurrent neural networks · Gradient-based neural dynamics · Time-varying nonlinear equations · Exponential convergence

Y. Zhang (✉) · C. Yi · D. Guo
School of Information Science and Technology, Sun Yat-sen University, 510006 Guangzhou, China
e-mail: zhynong@mail.sysu.edu.cn

C. Yi
School of Information Engineering, Jiangxi University of Science and Technology, 341000 Ganzhou, China

J. Zheng
School of Software, Sun Yat-sen University, 510006 Guangzhou, China

1 Introduction

The solution of nonlinear equations in the form of $f(x) = 0$ is widely encountered in science and engineering fields. Many numerical algorithms have thus been presented and investigated [1–3]. However, it may not be efficient enough for most numerical algorithms because of their serial-processing nature performed on digital computers [4]. In recent years, due to the in-depth research in neural networks, the dynamic system approach using recurrent neural networks is one of the important parallel-processing methods for solving optimization and equation problems [5–19]. For example, many studies have been reported on the real-time solution of algebraic equations, including matrix inversion and Sylvester equation [5–8, 11, 14, 18, 19]. Generally speaking, their schemes as reported are related to the gradient-descent method or other methods intrinsically designed for constant/stationary problems (or to say, time-invariant problems) solving. In other words, the coefficients of nonlinear equation $f(x) = 0$ are not the functions of time argument t (or simply put, not time-variant).

Different from gradient-based neural networks [5–7, 20], a special kind of recurrent neural networks has recently been proposed by Zhang et al. [8–12, 16, 18, 19] for solving online time-varying problems. In this paper, we develop and generalize Zhang et al.'s design method to solving time-varying nonlinear equations in the form of $f(x,t) = 0$. The resultant Zhang neural dynamics (ZND) are elegantly introduced by defining an indefinite error-monitoring function (being the time-varying nonlinear function itself) so as to make it decrease to zero exponentially. For comparative purposes, the conventional gradient-based neural dynamics (GND) are also investigated, of which the design is based on a nonnegative energy function. For superior convergence, different kinds of activation functions (linear, sigmoid, power functions, and/or their variants, e.g., power-sigmoid function) are investigated in this paper as well. Moreover, theoretical and simulative results are provided to demonstrate the efficacy of such a ZND model for nonlinear time-varying equations solving.

The remainder of this paper is organized as follows. Section 2 presents the general frameworks of GND and ZND models. ZND convergence properties are discussed in Sect. 3. In Sect. 4, an illustrative example of solving a nonlinear equation is presented. Section 5 concludes the paper with some final remarks. Before ending this section, it is worth mentioning the main contributions of this paper as follows.

(1) The time-varying nonlinear equation solving is analyzed, modeled, and simulated in this paper, which is different from the usually investigated time-invariant problem solving [1–3]. In other words, conventional stationary/constant nonlinear equation $f(x) = 0$ could be viewed as a specific case of our time-varying nonlinear equation $f(x, t) = 0$.

(2) In order to obtain the solutions of nonstationary problems, Zhang neural dynamics are designed based on an indefinite error-monitoring function, rather than the generally used nonnegative energy function associated with gradient-based neural dynamics. Furthermore, the proposed ZND model fully utilizes the time derivative of coefficients of equation $f(x, t) = 0$ and could thus efficiently solve the nonstationary problems, while GND could only approximately approach its theoretical time-varying solutions with relatively large residual errors owing to no use of the important derivative information.

(3) An illustrative example is modeled and simulated by using ZND and GND models. The novelty and efficacy of the proposed ZND model are evidently shown via the computer-simulation results. In contrast, the GND model has a less favorable approximating performance when applied to the time-varying nonlinear equation solving.

2 Problem formulation and neural dynamic solvers

Our objective of this paper is to find $x(t) \in R$ in real time t such that the following smoothly time-varying nonlinear equation holds true:

$$f(x(t), t) = 0. \tag{1}$$

The existence of theoretical time-varying solution $x^*(t)$ at any time instant $t \in [0, +\infty)$ is assumed as a basis of discussion. The design procedures of GND and ZND are investigated comparatively for solving (1) in this section. In addition, some basic types of activation functions (such as linear, sigmoid, power, and power-sigmoid activation functions) are analyzed and discussed for the convergence of the ZND model.

2.1 Gradient-based neural dynamics

Gradient-descent method is a conventional approach frequently used, e.g., to solve the constant situation of (1). According to the gradient-based design approach, to solve the time-varying nonlinear equation (1), the gradient-descent method requires us to define a norm- or square-based error function (or termed, energy function) such as $\varepsilon(x(t), t) = f^2(x(t), t)$. Then, a typical continuous-time adaptation rule based on the negative-gradient information leads to the following differential equation (which we might term as a linear gradient dynamics and/or implement as a linear gradient neural dynamics [7, 13–15, 20]):

$$\dot{x}(t) := \frac{\mathrm{d}x}{\mathrm{d}t} = -\frac{\gamma}{2}\frac{\partial \varepsilon}{\partial x} = -\gamma f(x(t), t)\frac{\partial f}{\partial x},$$

where γ is a positive design parameter used to scale the convergence rate and $x(t)$, starting from a randomly generated initial condition $x(0) = x_0 \in R$, is the activation state corresponding to theoretical solution $x^*(t)$. As an extension to the above design approach and under the inspiration of [10], we could obtain the general nonlinear GND model by using a general nonlinear activation function $\varphi(\cdot)$ as follows:

$$\dot{x}(t) = -\gamma\varphi(f(x(t), t))\frac{\partial f}{\partial x}. \tag{2}$$

Similar to usual neural dynamic approaches, the design parameter γ, being the reciprocal of a capacitance parameter in the hardware implementation, could be set as large as the hardware permits (e.g., in analog circuits or VLSI [7, 21]) or selected appropriately (e.g., between 10^3 and 10^8) for simulative and/or experimental purposes.

2.2 Zhang neural dynamics

Following the new design method of Zhang et al. [8–12, 16, 18, 19], we could construct the following indefinite error function $e(t)$ so as to set up a ZND model to solve online the time-varying nonlinear equation (1):

$$e(t) := f(x(t), t).$$

Then, the time derivative of $e(t)$, i.e., $\dot{e}(t)$, should be chosen and forced mathematically such that the error function $e(t)$ could converge to zero. Specifically, we can describe our general choice of $e(t)$ in the following form (termed as ZND design formula [8–12, 16, 18, 19]):

$$\frac{\mathrm{d}e(t)}{\mathrm{d}t} := -\gamma\varphi(e(t)),$$

or equivalently, we have

$$\frac{\mathrm{d}f}{\mathrm{d}t} := -\gamma\varphi(f(x(t),t)), \tag{3}$$

where design parameter γ and activation function $\varphi(\cdot)$ are defined the same as before. By expanding the above ZND design formula (3), we could thus have

$$\frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial t} = -\gamma\varphi(f(x(t),t)),$$

leading to the following differential equation (which we might term and/or implement as Zhang neural dynamics [8–12, 16, 18, 19]):

$$\begin{aligned} &\frac{\partial f}{\partial x}\dot{x}(t) = -\gamma\varphi(f(x(t),t)) - \frac{\partial f}{\partial t} \quad \text{or} \\ &\dot{x}(t) = -\left(\gamma\varphi(f(x(t),t)) + \frac{\partial f}{\partial t}\right) \Big/ \left(\frac{\partial f}{\partial x}\right), \end{aligned} \tag{4}$$

where $x(t)$, starting from a randomly generated initial condition $x(0) = x_0 \in R$, is the activation state corresponding to theoretical time-varying solution $x^*(t)$ of (1).

In view of dynamic equations (2) and (4), different choices for $\varphi(\cdot)$ may lead to different performance of the neural dynamics. Generally speaking, any monotonically increasing odd activation function $\varphi(\cdot)$ could be used for the construction of the neural dynamics. The following four basic types of activation functions are investigated in this paper:

- linear activation function $\varphi(e) = e$;
- bipolar-sigmoid activation function (with $\xi > 2$)
  $\varphi(e) = (1 - \exp(-\xi e))/(1 + \exp(-\xi e))$;
- power activation function $\varphi(e) = e^p$ with odd integer $p \geq 3$ (note that linear activation function can be viewed as a special case of the power activation function with $p = 1$);
- power-sigmoid activation function (with $\xi \geq 1$ and $p \geq 3$)

$$\varphi(e) = \begin{cases} e^p, & \text{if } |e| \geq 1, \\ \frac{1+\exp(-\xi)}{1-\exp(-\xi)} \cdot \frac{1-\exp(-\xi e)}{1+\exp(-\xi e)}, & \text{otherwise.} \end{cases} \tag{5}$$

It is worth pointing out that other types of activation functions can be generated or extended based on the above basic types of activation functions.

3 Convergence analysis

While Sect. 2 presents the general frameworks about GND and ZND models for solving the time-varying nonlinear equation $f(x(t), t) = 0$, detailed design consideration and theoretical results are given in this section. To analyze the convergence of the ZND model, we firstly introduce the following definitions [22].

Definition 1 A neural network is said to be globally convergent, if starting from any initial point taken in the whole associated Euclidean space, every state trajectory of the network converges to an equilibrium point, which depends on the initial state of the trajectory.

Definition 2 A neural network is said to be globally exponentially convergent, if every trajectory starting from any initial condition $x(t_0)$ satisfies

$$\|x(t) - x^*\| \leq \eta\|x(t_0) - x^*\| \exp(-\lambda(t - t_0)), \quad \forall t \geq t_0 \geq 0$$

where λ and η are positive constants independent of the initial point, x^* denotes an equilibrium point (depending on the initial state), and symbol $\|\cdot\|$ denotes the Euclidean norm of a vector (which, in our present situation, denotes the absolute value of a scalar argument).

For ZND (4), we have the following propositions on its convergence properties.

Proposition 1 *Consider a solvable time-varying nonlinear equation depicted in Eq.* (1), i.e., $f(x(t), t) = 0$, *where* $f(\cdot)$ *is a continuously differentiable function. If a monotonically increasing odd function* $\varphi(\cdot)$ *is used, neural state* $x(t)$ *of ZND model* (4), *starting from a randomly generated initial state* $x(0) = x_0 \in R$, *could converge to a theoretical time-varying solution* $x^*(t)$ *of* $f(x(t),t) = 0$.

Proof We can define a Lyapunov function candidate $V(x(t), t) = f^2(x(t), t) \geq 0$ for the neural system (4) with its time derivative

$$\begin{aligned} \dot{V}(x(t),t) &:= \frac{dV(x(t),t)}{dt} = f(x(t),t)\left(\frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial t}\right) \\ &= -\gamma f(x(t),t)\varphi(f(x(t),t)). \end{aligned} \tag{6}$$

Because a monotonically increasing odd activation function is used in ZND (4), we have $\varphi(-f(x(t), t)) = -\varphi(f(x(t), t))$ and

$$\varphi(f(x(t),t)) \begin{cases} > 0, & \text{if } f(x(t),t) > 0, \\ = 0, & \text{if } f(x(t),t) = 0, \\ < 0, & \text{if } f(x(t),t) < 0. \end{cases}$$

Then, we obtain

$$f(x(t),t)\varphi(f(x(t),t)) \begin{cases} > 0, & \text{if } f(x(t),t) \neq 0, \\ = 0, & \text{if } f(x(t),t) = 0, \end{cases}$$

which guarantees the final negative-definiteness of $\dot{V}(x(t),t)$. That is, $\dot{V}(x(t),t) < 0$ for $f(x(t), t) \neq 0$ [equivalently, $x(t) \neq x^*(t)$], and $\dot{V}(x(t),t) = 0$ for $f(x(t), t) = 0$

[equivalently, $x(t) = x^*(t)$]. By Lyapunov theory [22], residual error $e(t) = f(x(t), t)$ could converge to zero, or equivalently, neural state $x(t)$ of ZND model (4) could converge to a theoretical solution $x^*(t)$ with $f(x^*(t),t) = 0$ starting from some randomly generated initial states [note that $\partial f/\partial x$ appears in the derivation of (6)]. The proof is thus complete. □

Proposition 2 *Let $x^*(t)$ denote a theoretical time-varying solution of nonlinear equation $f(x(t), t) = 0$, where $f(\cdot)$ is a continuously differentiable function (specifically, with at least first-order derivatives) at some interval containing $x^*(t)$. In addition to Proposition 1, neural state $x(t)$ of ZND model (4) could converge to $x^*(t)$, provided that initial state $x(0) = x_0$ is close enough to $x^*(t)$. In addition, ZND model (4) possesses the following properties.*

(1) If the linear activation function is used, then the exponential convergence with rate γ [in terms of residual error $e(t) = f(x(t), t) \to 0$] can be achieved for ZND model (4).
(2) If the bipolar-sigmoid activation function is used, the superior convergence can be achieved for error range $e(t) = f(x(t), t) \in [-\delta, \delta]$, $\exists \delta > 0$, when compared to the situation of using a linear activation function described in Property 1.
(3) If the power activation function is used, then the superior convergence can be achieved for error ranges $(-\infty, -1)$ and $(1, \infty)$, when compared to the situation of using a linear activation function described in Property 1.
(4) If the power-sigmoid activation function is used, the superior convergence can be achieved for the whole error range $(-\infty, \infty)$, when compared to the situation of a linear activation function described in Property 1.

Proof We now come to prove the additional convergence properties of ZND (4) to $x^*(t)$ by considering the mentioned several kinds of activation functions $\varphi(\cdot)$.

(1) For the case of using a linear activation function, it follows from (3) that $df(x(t), t)/dt = -\gamma f(x(t), t)$, which yields $f(x(t), t) = \exp(-\gamma t) f(x(0), 0)$. This proves the exponential convergence rate γ of ZND (4) in the sense of residual error $e(t) = f(x(t),t) \to 0$. Moreover, we can show that the state $x(t)$ of ZND (4) could converge to a theoretical time-varying solution $x^*(t)$ of nonlinear equation $f(x(t),t) = 0$ by the following procedure. From Taylor's theorem [23], given some α between $x(t)$ and $x^*(t)$, we have

$$\begin{aligned} f(x(t),t) &= f((x(t) - x^*(t) + x^*(t)), t) \\ &= f(x^*(t),t) + (x(t) - x^*(t)) \frac{\partial f(x(t),t)}{\partial x}\Big|_{x=x^*} \\ &\quad + \frac{(x(t) - x^*(t))^2}{2!} \frac{\partial f^{(2)}(x(t),t)}{\partial x^2}\Big|_{x=x^*} \\ &\quad + \cdots + \frac{(x(t) - x^*(t))^n}{n!} \frac{\partial f^{(n)}(x(t),t)}{\partial x^n}\Big|_{x=x^*} \\ &\quad + \frac{(x(t) - x^*(t))^{n+1}}{(n+1)!} \frac{\partial f^{(n+1)}(x(t),t)}{\partial x^{n+1}}\Big|_{x=x^*}. \end{aligned}$$

Thus, in view of $f(x(t), t) = \exp(-\gamma t) f(x(0), 0)$ and $f(x^*(t),t) = 0$, by omitting the higher-order terms, we have

$$(x(t) - x^*(t)) \frac{\partial f(x(t),t)}{\partial x}|_{x=x^*} \approx \exp(-\gamma t) f(x(0),0),$$

which, if $\partial f(x(t),t)/\partial x|_{x(t)=x^*(t)} \neq 0$, yields

$$||(x(t) - x^*(t))|| \approx \exp(-\gamma t) \left|\left| f(x(0),0) / \left(\frac{\partial f(x(t),t)}{\partial x}\Big|_{x=x^*} \right) \right|\right|.$$

More generally, even if $\partial f^{(k)}(x(t),t)/\partial x^k|_{x(t)=x^*(t)} = 0$, $\forall k = 1,2,3,\ldots,(n-1)$, and $\partial f^{(n)}(x(t),t)/\partial x^n|_{x(t)=x^*(t)} \neq 0$, we still have

$$||(x(t) - x^*(t))|| \approx \exp(-\gamma t/n) \left|\left| \left(f(x(0),0) n! / \left(\frac{\partial f^{(n)}(x(t),t)}{\partial x^n}\Big|_{x=x^*} \right) \right)^{\frac{1}{n}} \right|\right|.$$

This implies that the state $x(t)$ of ZND (4) converges to theoretical solution $x^*(t)$ in an exponential manner, provided that $x(0) = x_0$ is close enough to $x^*(t)$.

(2) When using bipolar-sigmoid function $\varphi(e) = (1 - \exp(-\xi e))/(1 + \exp(-\xi e))$, we know that there exists an error range $e(t) = f(x(t), t) \in [-\delta, \delta]$ with $\delta > 0$, such that $||\varphi(f(x,(t),t))|| > ||f(x(t),t)||$ [7, 10]. So, by reviewing the proof of Proposition 1 [especially (6)], we know that the superior convergence is achieved by ZND (4) with a bipolar-sigmoid activation function for such an error range, when compared to Property 1.
(3) For the pth power activation function $\varphi(e) = e^p$, the solution to (3) becomes

$$f(x(t),t) = f(x(0),0)\left((p-1) f^{p-1}(x(0),0)\gamma t + 1\right)^{-\frac{1}{p-1}}.$$

Besides, for $p = 3$, residual error $e(t) = f(x(t),t) = f(x(0),0)/\sqrt{2f^2(x(0),0)\gamma t + 1}$. Evidently, as $t \to \infty$, $f(x(t), t) \to 0$. By reviewing the proof of Proposition 1, especially, the Lyapunov function candidate

$V(x, t) = f^2(x, t)/2$ and its time derivative $\dot{V}(x,t) = -\gamma f(x,t)\varphi(f(x,t))$ in (6), in the error range $\|e(t)\| = \|f(x, t)\| \gg 1$, we have $f^{p+1}(x, t) \gg f^2(x, t) \gg \|f(x, t)\|$. In other words, the deceleration magnitude of a power activation function can be much greater than that of a linear activation function. This implies that when a power activation functions is used, a much faster convergence is achieved by ZND (4) for such error ranges in comparison with Property 1 of using a linear activation function [7, 10].

(4) It follows from the above analysis that, to achieve superior convergence, a high-performance neural dynamics can be developed by switching the power activation function to a sigmoid or linear activation function at the switching points $e(t) = f(x, t) = \pm 1$. Thus, if the power-sigmoid activation function is used with suitable design parameters ξ and p, superior convergence is achieved by ZND model (4) for the whole error range $(-\infty, \infty)$, as compared to Property 1 of using the linear activation function. □

Remark 1 Nonlinearity always exists, which is one of the main motivations for us to investigate different activation functions. Even if the linear activation function is used, the nonlinear phenomenon may appear in its hardware implementation; e.g., in the form of saturation and/or inconsistency of the linear slope, and in digital realization due to truncation and round-off errors [7]. The investigation of different activation functions (such as the sigmoid function and the power function) gives us many more insights into positive and negative effects of nonlinearities existing in the implementation of linear activation functions. In addition, the ZND convergence by using nonlinear activation functions can be much faster than that by using linear functions. Because of parameter ξ (being a multiplier of the exponential convergence rate), we can have it as another effective factor to expedite the ZND convergence.

Remark 2 It is worth comparing here the two design methods of GND model (2) and ZND model (4), both of which are exploited for the online solution of time-varying nonlinear equation (1). The differences may lie in the following facts.

(1) GND model (2) is designed based on the elimination of the square-based nonnegative error function $\varepsilon(t) = f^2(x(t), t)$. In contrast, ZND model (4) is designed based on the elimination of an indefinite error function $e(x, t) = f(x(t), t)$, which can be positive, negative, bounded, or even unbounded.

(2) GND model (2) is designed based on a method intrinsically for nonlinear equations but with stationary/constant coefficients. It is thus only able to approximately approach the theoretical solution of a time-varying problem. In contrast, ZND model (4) is designed based on a new method intrinsically for time-varying problems solving. Thus, it could converge to an exact/theoretical solution of the nonlinear time-varying equation.

(3) GND model (2) has not exploited the time-derivative information of function $f(\cdot)$ [i.e., $\partial f(x(t), t)/\partial t$], and thus may not be effective enough in solving such a time-varying nonlinear-equation problem. In contrast, ZND model (4) methodically and systematically exploits the time-derivative information of function $f(\cdot)$ during its real-time solving process. This appears to be the reason why ZND model (4) could converge to an exact/theoretical solution of such a time-varying nonlinear-equation problem.

4 Simulation studies

While the previous sections present the theoretical results about GND model (2) and ZND model (4), this section substantiates them by showing the following computer-simulation results and observations (which are all based on the power-sigmoid activation functions depicted in (5) with $\xi = 4$ and $p = 3$).

For illustration and comparison purposes, both neural dynamics [i.e., GND (2) and ZND (4)] are exploited for solving online the time-varying nonlinear equation $f(x(t),t) = 0$ with the following illustrative example:

$$f(x, t) = x^2 - 2\sin(1.8t)x + \sin^2(1.8t) - 1 = 0. \qquad (7)$$

The above is actually equal to $f(x, t) = (x - \sin(1.8t) - 1)(x - \sin(1.8t) + 1) = 0$, and its time-varying theoretical solutions can be written down as $x_1^*(t) = \sin(1.8t) + 1$ and $x_2^*(t) = \sin(1.8t) - 1$. The simulation results are illustrated in Figs. 1, 2, 3.

Figure 1 shows the solution-performance comparison of GND model (2) and ZND model (4) with design parameter $\gamma = 1$ during the solution procedure of time-varying nonlinear equation (7), where dash curves correspond to theoretical solutions $x^*(t)$ and solid curves correspond to neural state $x(t)$. As seen from the left graph of Fig. 1, starting from 50 randomly generated initial states within $[-4, -4]$, neural state $x(t)$ of GND (2) does not fit well with the theoretical time-varying solutions $x_1^*(t)$ or $x_2^*(t)$. Simply put, the steady-state error of GND model (2) is considerably large, always lagging behind the theoretical solution $x^*(t)$. Moreover, when compared to the curves of theoretical solution $x^*(t)$, the curves of GND-states $x(t)$ have relatively serious distortion. In contrast, the neural state $x(t)$ of ZND model (4), also starting from 50

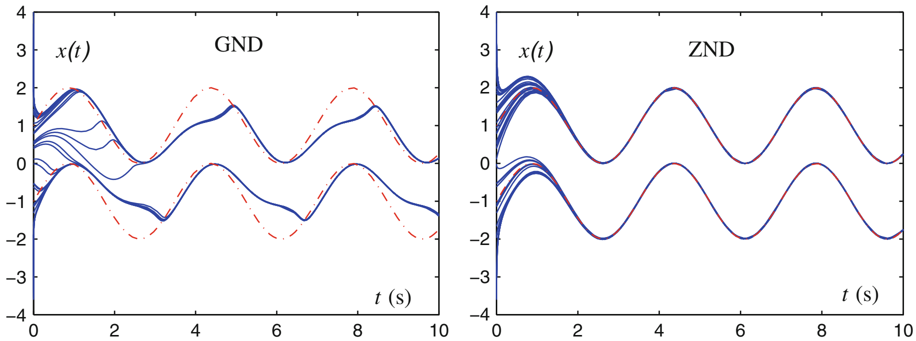

Fig. 1 Solution-performance comparison of GND (2) and ZND (4) with design parameter $\gamma = 1$ for solving nonlinear time-varying equation (7), where *dash curves* denote theoretical solutions

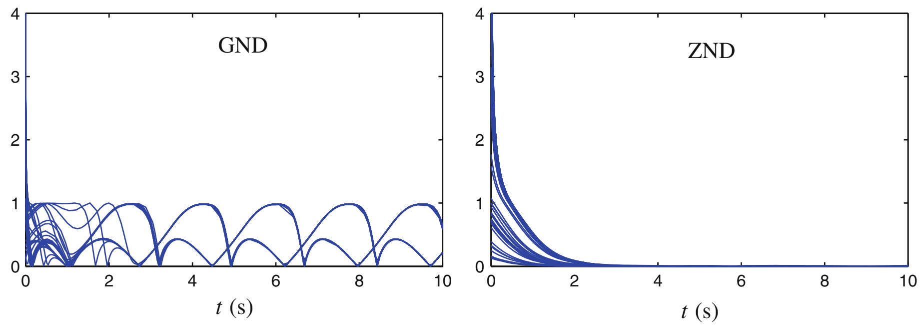

Fig. 2 Residual error $||x^2 - 2\sin(1.8t)x + \sin^2(1.8t) - 1||$ of GND (2) and ZND (4) with design parameter $\gamma = 1$ for solving nonlinear time-varying equation (7)

randomly generated initial states within $[-4, 4]$, converges to one of the theoretical solutions [i.e., $x_1^*(t)$ or $x_2^*(t)$], which is shown clearly in the right graph of Fig. 1. This is because the time-derivative information of time-varying coefficients in (7) has been fully utilized in Zhang et al.'s design method/model.

Furthermore, in order to further investigate the convergence performance, we can also monitor the residual error $||e(t)|| = ||x^2 - 2\sin(1.8t)x + \sin^2(1.8t) - 1||$ during the equation solving process by both neural models. It is seen from the left graph of Fig. 2 and other simulation data that, by applying GND (2) to online solution of time-varying nonlinear equation (7), its residual error $\|e(t)\|$ is rather large. More specifically, the steady-state residual error $\lim_{t\to\infty} ||e(t)||$ is about 0.3146 and 0.0279 (as computed at $t = 100$ s), which respectively correspond to the usage of design parameter $\gamma = 1$ and $\gamma = 10$. In contrast, as seen from the right graph of Fig. 2, by applying ZND (4) to solving time-varying nonlinear equation (7) under the same simulation conditions, its residual error $\|e(t)\|$ converges to zero within 5 s. In addition, by using different values of γ (e.g., $\gamma = 10$), the convergence performance of ZND (4), as shown in Fig. 3, is improved (when compared to the right graph of Fig. 2). More specifically, the steady-state residual error $\lim_{t\to\infty} ||e(t)||$ is about 1.0458×10^{-12} and 1.1990×10^{-13}, which respectively correspond to the usage of design parameter $\gamma = 1$ and $\gamma = 10$. Note that such a maximum steady-state residual error should theoretically be zero but are numerically nonzero because of the finite-arithmetic simulation performed on finite-memory digital computers with floating-point relative accuracy being 2.220×10^{-16} [MathWorks Inc., Version 7.0.4.365 (R14)]. It is also worth pointing out here that, as shown in Figs. 2 and 3 as well as other simulation data, the convergence time for ZND (4) can be expedited from 5 to 0.5 s and 0.05 s, as design parameter γ is increased from 1 to 10 and 100. This substantiates that ZND model (4) has an exponential convergence property, which can be expedited by increasing the value of design parameter γ.

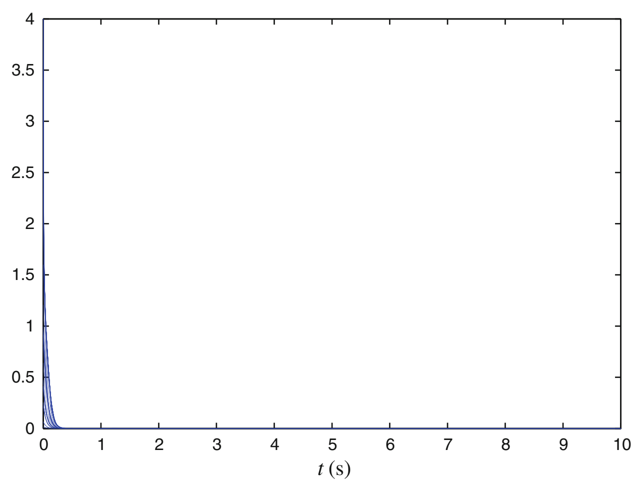

Fig. 3 Residual error $\|x^2 - 2sin(1.8t)x + sin^2(1.8t) - 1\|$ of the ZND model (4) with design parameter $\gamma = 10$ for solving nonlinear time-varying equation (7)

In summary, the simulation results (i.e., Figs. 1, 2, 3) have demonstrated that the ZND model is more effective and efficient for solving online time-varying nonlinear equations, when compared to conventional GND models.

5 Conclusions

In this paper, a general recurrent neural dynamics model (termed Zhang neural dynamics) has been developed and analyzed for solving nonlinear time-varying equations in real time. Such a neural dynamics has been elegantly introduced by defining an indefinite error-monitoring function rather than the usually used positive energy function. The ZND computation error can thus be made decreasing to zero exponentially. The conventional gradient-based neural dynamics has also been investigated for comparison purposes. Theoretical and simulative results both demonstrate the efficacy of the resultant ZND approach and model. Further efforts may be directed at the design and analysis of discrete-time neural networks, algorithms, and/or electronic circuits for solving online the nonlinear time-varying equations.

Acknowledgments This work is supported by the National Natural Science Foundation of China under Grants 61075121, 60935001 and 60775050, and also by the Fundamental Research Funds for the Central Universities of China.

References

1. Sharma JR (2005) A composite third order newton-steffensen method for solving nonlinear equations. Appl Math Comput 169:242–246
2. Chun C (2006) Construction of Newton-like iteration methods for solving nonlinear equations. Numerische Mathematik 104:297–315
3. Ujevic N (2006) A method for solving nonlinear equations. Appl Math Comput 174:1416–1426
4. Zhang Y, Leithead WE, Leith DJ (2005) Time-series Gaussian process regression based on Toeplitz computation of $O(N^2)$ operations and $O(N)$-level storage. In: Proceedings of the 44th IEEE conference on decision and control, pp 3711–3716
5. Guo B, Wang D, Shen Y, Li Z (2008) A Hopfield neural network approach for power optimization of real-time operating systems. Neural Comput Appl 17:11–17
6. Wu A, Tam PKS (1999) A neural network methodology and strategy of quadratic optimization. Neural Comput Appl 8:283–289
7. Zhang Y (2005) Revisit the analog computer and gradient-based neural system for matrix inversion. In: Proceedings of IEEE international symposium on intelligent control, Limassol, Cyprus, pp 1411–1416
8. Zhang Y, Chen K (2008) Comparison on Zhang neural network and gradient neural network for time-varying linear matrix equation $AXB = C$ solving. In: Proceedings of IEEE international conference on industrial technology, Chendu, pp 1–6
9. Zhang Y, Jiang D, Wang J (2002) A recurrent neural network for solving Sylvester equation with time-varying coefficients. IEEE Trans Neural Netw 13:1053–1063
10. Zhang Y, Ge SS (2005) Design and analysis of a general recurrent neural network model for time-varying matrix inversion. IEEE Trans Neural Netw 16:1477–1490
11. Zhang Y, Chen Z, Chen K, Cai B (2008) Zhang neural network without using time-derivative information for constant and time-varying matrix inversion. In: Proceedings of international joint conference on neural networks, Hong Kong, pp 142–146
12. Zhang Y, Guo X, Ma W (2008) Modeling and simulation of Zhang neural network for online linear time-varying equations solving based on MATLAB Simulink. In: Proceedings of the seventh international conference on machine learning and cybernetics, Kunming, pp 12–15
13. Zhang Y (2006) A set of nonlinear equations and inequalities arising in robotics and its online solution via a primal neural network. Neurocomputing 70:513–524
14. Steriti RJ, Fiddy MA (1993) Regularized image reconstruction using SVD and a neural network method for matrix inversion. IEEE Trans Signal Process 41:3074–3077
15. Tank D, Hopfield J (1986) Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit. IEEE Trans Circuits Syst 33:533–541
16. Zhang Y, Peng H (2007) Zhang neural network for linear time-varying equation solving and its robotic application. In: Proceedings of the sixth international conference on machine learning and cybernetics, Hong Kong, pp 19–22
17. Cichocki A, Unbehauen R (1993) Neural networks for optimization and signal processing. Wiley, Chichester
18. Zhang Y, Ma W, Yi C (2008) The link between Newton iteration for matrix inversion and Zhang neural network. In: Proceedings of IEEE international conference on industrial technology, Chendu, pp 1–6
19. Zhang Y, Ma W, Cai B (2009) From Zhang neural network to Newton iteration for matrix inversion. IEEE Trans Circuits Syst I Regul Pap 56:1405–1415
20. Manherz RK, Jordan BW, Hakimi SL (1968) Analog methods for computation of the generalized inverse. IEEE Trans Automat Contr 13:582–585
21. Mead C (1989) Analog VLSI and neural systems. Addison-Wesley, Reading
22. Zhang Y (2007) Dual neural networks: design, analysis, and application to redundant robotics. In: Kang GB (eds) Progress in neurocomputing research. Nova Science Publishers, New York, pp 41–81
23. Mathews JH, Fink, KD (2004) Numerical methods using MATLAB, 4th edn. Prentice Hall, New Jersey