Robustness analysis of the Zhang neural network for online time-varying quadratic optimization

IOP Publishing Journal of Physics A: Mathematical and Theoretical

J. Phys. A: Math. Theor. **43** (2010) 245202 (19pp) doi:10.1088/1751-8113/43/24/245202

Robustness analysis of the Zhang neural network for online time-varying quadratic optimization

Yunong Zhang, Gongqin Ruan, Kene Li and Yiwen Yang

School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510006, People's Republic of China

E-mail: zhynong@mail.sysu.edu.cn and ynzhang@ieee.org

Received 18 January 2010, in final form 15 April 2010
Published 19 May 2010
Online at stacks.iop.org/JPhysA/43/245202

Abstract

A general type of recurrent neural network (termed as Zhang neural network, ZNN) has recently been proposed by Zhang *et al* for the online solution of time-varying quadratic-minimization (QM) and quadratic-programming (QP) problems. Global exponential convergence of the ZNN could be achieved theoretically in an ideal error-free situation. In this paper, with the normal differentiation and dynamics-implementation errors considered, the robustness properties of the ZNN model are investigated for solving these time-varying problems. In addition, linear activation functions and power-sigmoid activation functions could be applied to such a perturbed ZNN model. Both theoretical-analysis and computer-simulation results demonstrate the good ZNN robustness and superior performance for online time-varying QM and QP problem solving, especially when using power-sigmoid activation functions.

PACS number: 07.05.Mh

(Some figures in this article are in colour only in the electronic version)

1. Introduction

The online solution of (equality-constrained) quadratic programs (including the quadratic-minimization (QM) problem solving as its special case) appears to be ubiquitous in science and engineering fields. It is usually an essential part of the solution of many problems, e.g. optimal controller design [1], power scheduling [2], robot-arm motion planning [3] and digital signal processing [4]. A well-accepted approach to solving such quadratic programs is the numerical algorithms/methods performed on digital computers (e.g. nowadays computers). However, the minimal arithmetic operations of a numerical quadratic-programming (QP) algorithm are proportional to the cube of the related Hessian matrix's dimension. Consequently, such numerical algorithms may not be efficient enough for large-scale online applications [4].

Due to the in-depth research on neural networks, numerous dynamic and analog solvers based on recurrent neural networks have been developed and investigated [5–12] recently. The neural-dynamic approach is now viewed as a powerful alternative to online computation and optimization because of its parallel distributed nature and convenience of hardware/circuits implementation [13–17]. Specifically, the neural approach is also an efficient method for solving static QM and QP problems [7, 11, 18–20].

The conventional gradient (or gradient-based) neural networks (GNN) have been proven to be very useful for static (or to say, time-invariant, stationary) matrix/vector problem solving [7, 13, 19, 21, 22]. However, when applied to time-varying problems, gradient neural networks (GNN) only work approximately, lagging behind with appreciable solution errors [13]. In view of this, a special class of recurrent neural networks has recently been proposed by Zhang *et al* for the real-time solution of time-varying problems [13, 23–26]. Different from GNN [13, 17, 19], the design of Zhang neural network (ZNN) models is based on matrix- or vector-valued indefinite error functions, instead of scalar-valued positive or lower bounded energy functions. They are depicted generally in implicit dynamics, instead of explicit dynamics. Furthermore, ZNN modeling and simulation have been investigated as well, which substantiate the efficacy of this new neural-dynamic method [25–27].

As shown in [28, 29], ZNN models could converge exactly to the time-varying optimal theoretical solution when exploited for time-varying QM and QP problem solving. However, in the hardware implementation of a recurrent neural network, there always exist some realization errors, which are more complicated than the proved ideal situation. Generally speaking, an electronic circuit of a neural network is composed of multipliers and amplifiers [19, 30–32], and each neuron can be implemented by three operational amplifiers: a summing amplifier, an integrating amplifier and an inverting amplifier [19, 32]. In practice, these components of the circuit are not under ideal conditions. For example, the tolerance of electronic components, the finite gain and bandwidth of operational amplifiers and multipliers are discussed in [30]. The incapacity of electronic components would limit the performance of the ZNN, and generate various errors, such as differentiation error and model-implementation error. Due to these realization errors, the solution of the circuit-implementation ZNN might not be accurate. In this case, robustness analysis of the proposed neural network would be important and necessary. By analyzing the robustness of ZNN solving time-varying QM and QP problems, we would know how the design parameters might affect the performance of the ZNN, and how to improve the robustness of the ZNN. For these purposes, this paper investigates the robustness properties of the ZNN models with both differentiation and dynamics-implementation errors considered. Robustness analysis is carried out by theoretically estimating the upper bound of steady-state errors of the ZNN model exploited for time-varying QM and QP problem solving. Such upper-bound errors could be decreased by adjusting the design parameter γ of the ZNN model. Furthermore, using power-sigmoid activation functions has the superior robustness compared with linear activation functions.

The remainder of this paper is organized into four sections. Section 2 gives problem formulation about time-varying QM and QP problems, and describes the situation of using the ZNN for solving these problems. In section 3, we analyze the robustness performance of the ZNN when applied to time-varying QM and QP problems; specifically, the solution-error bound and the superior performance of using power-sigmoid activation functions. Several illustrative computer-simulation examples are presented in section 4, which solve online time-varying QM and QP problems via the ZNN. Finally, we conclude this paper with section 5.

2. Problem formulation and ZNN solver

To lay a basis for further discussion, the procedure of solving time-varying QM and QP problems using the ZNN is investigated and developed in this section.

2.1. Time-varying quadratic minimization

Consider the following time-varying QM problem:

$$\text{minimize } f(x) := x^T(t)P(t)x(t)/2 + q^T(t)x(t) \in \mathbb{R}, \tag{1}$$

where the Hessian matrix $P(t) \in \mathbb{R}^{n\times n}$ is smoothly time-varying, positive-definite and symmetric for any time instant $t \in [0, +\infty) \subset \mathbb{R}$, and the coefficient vector $q(t) \in \mathbb{R}^n$ is smoothly time varying as well. In expression (1), the unknown vector $x(t) \in \mathbb{R}^n$ is to be solved to obtain the minimum value of $f(x)$ at any time instant $t \in [0, +\infty)$.

One simple method of solving the time-varying QM problem (1) could be performed by zeroing the partial derivative $\nabla f(x)$ of $f(x)$ [4] at every time instant *t*; in mathematics,

$$\nabla f(x) := \frac{\partial f(x)}{\partial x} = P(t)x(t) + q(t) = \mathbf{0} \in \mathbb{R}^n, \qquad \forall\, t \in [0, +\infty). \tag{2}$$

More specifically, it follows from the above that the theoretical time-varying solution $x^*(t) \in \mathbb{R}^n$ to (1), being the minimum point of $f(x)$ at any time instant *t*, satisfies $x^*(t) = -P^{-1}(t)q(t)$. The theoretical minimum value $f^* := f(x^*)$ of the time-varying quadratic function $f(x)$ is thus achieved as $f^* = x^{*T}(t)P(t)x^*(t)/2 + q^T(t)x^*(t)$.

2.2. Time-varying quadratic programming

Evidently, minimizing quadratic function (1) may not be enough to describe practical problems in some fields. For instance, in the case of robot-arm motion planning [3, 33–36], the consideration of end-effector trajectory and joint physical limits is also necessary in a unified QP formulation, which is subject to linear constraints.

Let us consider the following time-varying convex QP problem subject to a time-varying linear-equality constraint:

$$\text{minimize} \quad x^T(t)P(t)x(t)/2 + q^T(t)x(t), \tag{3}$$

$$\text{subject to} \quad A(t)x(t) = b(t), \tag{4}$$

where the time-varying decision vector $x(t) \in \mathbb{R}^n$ is unknown and to be solved at any time instant $t \in [0, +\infty)$. In addition to the coefficients' description in subsection 2.1, in equality constraint (4), the coefficient matrix $A(t) \in \mathbb{R}^{m\times n}$ (being of full row rank) and vector $b(t) \in \mathbb{R}^m$ are also smoothly time varying.

Based on the knowledge of QP problem solving [28, 29, 37, 38], we can similarly transform the time-varying QP problem (3)–(4) into the following time-varying linear matrix–vector equation:

$$\tilde{P}(t)\tilde{x}(t) = -\tilde{q}(t), \tag{5}$$

where

$$\tilde{P}(t) := \begin{bmatrix} P(t) & A^T(t) \\ A(t) & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)},$$

$$\tilde{x}(t) := \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \in \mathbb{R}^{n+m}, \qquad \tilde{q}(t) := \begin{bmatrix} q(t) \\ -b(t) \end{bmatrix} \in \mathbb{R}^{n+m},$$

where $\lambda(t) \in \mathbb{R}^m$ denotes the Lagrange-multiplier vector. With the positive-definite matrix $P(t) \in \mathbb{R}^{n\times n}$ and the full-row-rank matrix $A(t) \in \mathbb{R}^{m\times n}$ at any time instant $t \in [0, +\infty)$, $\tilde{P}(t) \in \mathbb{R}^{(n+m)\times(n+m)}$ must be nonsingular at any time instant $t \in [0, +\infty)$, which guarantees the solution uniqueness in equation (5) [37, 38]. Moreover, for the purposes of better understanding and comparison, we know that the time-varying theoretical solution could be written as

$$\tilde{x}^*(t) = [x^{*T}(t), \lambda^{*T}(t)]^T := -\tilde{P}^{-1}(t)\tilde{q}(t) \in \mathbb{R}^{n+m}.$$

2.3. Zhang neural network

As mentioned in the above sections, the minimization of QM and QP problems could be transformed into solving linear equations (2) and (5), respectively. For the purpose of further discussion and convenience, the problems of QM and QP could be formulated in a unified manner as

$$W(t)y(t) = u(t), \tag{6}$$

where

$$W(t) := \begin{cases} P(t), & \text{in QM,} \\ \tilde{P}(t), & \text{in QP,} \end{cases} \qquad y(t) := \begin{cases} x(t), & \text{in QM,} \\ \tilde{x}(t), & \text{in QP,} \end{cases} \qquad u(t) := \begin{cases} -q(t), & \text{in QM,} \\ -\tilde{q}(t), & \text{in QP.} \end{cases}$$

The time-varying theoretical optimal solution could be written as $y^*(t) = W(t)^{-1}u(t)$. In order to describe more conveniently, assuming $W(t) \in \mathbb{R}^{n\times n}$, $y(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^n$. After the above equation (6) is given, the design procedure of the ZNN could be formalized as follows.

Step 1. We could define the following vector-valued error function $e(t) \in \mathbb{R}^n$; in mathematics,

$$e(t) := W(t)y(t) - u(t). \tag{7}$$

Step 2. The time derivative $\dot{e}(t)$ of the error function $e(t)$ could be constructed better as (or termed the general ZNN-design formula)

$$\dot{e}(t) := \frac{\mathrm{d}e(t)}{\mathrm{d}t} = -\gamma\mathcal{F}\big(e(t)\big), \tag{8}$$

where, being the reciprocal of a capacitance parameter, the design parameter $\gamma > 0 \in \mathbb{R}$ should be implemented as large as possible or selected appropriately for simulative purposes. In addition, $\mathcal{F}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ denotes an activation-function (vector) array; or in other words, the array $\mathcal{F}(\cdot)$ is made of n monotonically increasing odd activation functions $f(\cdot)$ [13, 24]:

- linear activation function $f(e_i) = e_i$ (with e_i being the ith element of the residual-error vector);
- power-sigmoid activation function

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

with design parameters $\xi \geqslant 2$ and odd integer $p \geqslant 3$.

Step 3. By expanding the ZNN-design formula (8), we have

$$W(t)\dot{y}(t) = -\dot{W}(t)y(t) - \gamma\mathcal{F}\big(W(t)y(t) - u(t)\big) + \dot{u}(t). \tag{9}$$

In summary, the following lemma guarantees the excellent convergence of the ZNN model (9) under ideal conditions, e.g. with no errors involved and using linear or power-sigmoid activation functions [24–26].

Lemma 1. *Given a smoothly time-varying nonsingular matrix $W(t)$ and vector $u(t)$, if a monotonically increasing odd activation-function array $\mathcal{F}(\cdot)$ is used, then the state vector $y(t)$ of the ZNN model (9) starting from any initial state $y(0)$ globally (exponentially) converges to the time-varying theoretical solution $y^*(t) = W^{-1}(t)u(t)$.*

In practice, however, realization errors always exist in the hardware implementation. Thus, in the ensuing sections, the robustness properties of the ZNN model (9) are investigated with normal model-implementation errors involved.

3. Robustness analysis

In the hardware implementation of the ZNN model (9), the differentiation errors about matrix $W(t)$ and/or vector $u(t)$ as well as the dynamics-implementation error (collectively termed as the model-implementation error) appear most frequently [31]. Therefore, let us consider the following dynamic equation which might depict such a perturbed ZNN model (with argument t omitted sometimes for presentation convenience):

$$W\dot{y} = -(\dot{W} + \Delta_{\mathrm{D}})y - \gamma\mathcal{F}(Wy - u) + \dot{u} + \Delta_{\mathrm{m}}, \tag{10}$$

where $\Delta_{\mathrm{D}}(t) \in R^{n\times n}$ denotes the differentiation error of the matrix $W(t)$, and $\Delta_{\mathrm{m}}(t) \in R^n$ denotes the dynamics-implementation error (including the differentiation error of the vector $u(t)$ as a part). These errors may result from truncating/roundoff errors in digital realization and/or high-order residual errors of circuit components in analog realization (see [13, 19, 31] and references therein). For the perturbed ZNN model with the differentiation error $\Delta_{\mathrm{D}}(t)$ and the dynamics-implementation error $\Delta_{\mathrm{m}}(t)$ involved in (10), we could have the following theoretical results on robustness.

Theorem 1. *If $\|\Delta_{\mathrm{D}}(t)\|_F \leqslant \varepsilon_{\mathrm{D}}$, $\|\Delta_{\mathrm{m}}(t)\|_2 \leqslant \varepsilon_{\mathrm{m}}$, $\|W^{-1}(t)\|_F \leqslant \varphi_{\mathrm{W}}$ and $\|u(t)\|_2 \leqslant \varphi_{\mathrm{u}}$ for any time instant $t \in [0, \infty)$, and $0 < \varepsilon_{\mathrm{D}}, \varepsilon_{\mathrm{m}}, \varphi_{\mathrm{W}}, \varphi_{\mathrm{u}} < +\infty$, then the computational error $\|y(t) - W^{-1}(t)u(t)\|_2$ of the perturbed ZNN model (10) using linear or power-sigmoid activation functions is upper bounded with the maximal steady-state error around $(\sqrt{n} + n)\varphi_{\mathrm{W}}(\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/2(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ under the design-parameter requirement $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}/\rho$, where $\|\cdot\|_F$ and $\|\cdot\|_2$ denote, respectively, the Frobenius norm of a matrix and the two-norm of a vector, and there exists $\rho \geqslant 1$. In addition, as the design parameter γ tends to be positive infinity, the steady-state computational error can be decreased to zero.*

The proof of theorem 1 can be found in the appendix.

Theorem 1 presents the upper bound of the computational error $\|y(t) - W^{-1}(t)u(t)\|_2$ of the perturbed ZNN model (10), which means that the error $\|y(t) - W^{-1}(t)u(t)\|_2$ could not be beyond this upper bound. However, in practice, we would like to know whether the perturbed ZNN model (10) could exponentially converge to an error bound, and how fast could it reach to this error bound. For completeness of the analysis, the perturbed ZNN (10) is investigated further, which leads to the following results on global exponential convergence rate and finite convergence time of ZNN (10) to an error bound of $(\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ with $0 < \alpha < 1$ chosen by ZNN users.

Theorem 2. *As the same condition in theorem 1, starting from any initial state $y(0) \in R^n$, the solution error $\|e(t)\|_2$ of a perturbed ZNN (10) is globally exponentially convergent to or* *staying within the error bound* $(\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ *under the design-parameter requirement* $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}/\rho$*, where the exponential convergence rate is* $(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ *and the convergence time is*

$$t_c = \ln\left(\frac{\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(0)\|_2}{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}\right)\Big/((1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})),$$

for any $\alpha \in (0, 1)$.

The proof of theorem 2 can be found in the appendix.

Theorem 3. *In addition to the robustness results in theorems 1 and 2, the perturbed ZNN model (10) possesses the following properties.*

- *If linear activation functions are used, the steady-state entry residual error is upper bounded around* $(1+\sqrt{n})(\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/2(\gamma - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ *under requirement* $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}$*. Moreover, the solution error* $\|e(t)\|_2$ *of a perturbed ZNN (10) is globally exponentially convergent to or staying within the error bound* $(\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/\alpha(\gamma - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$*, where the exponential convergence rate is* $(1-\alpha)(\gamma - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ *and the convergence time is*
$$t_c = \ln\left(\frac{\alpha(\gamma - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(0)\|_2}{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}\right)\Big/((1-\alpha)(\gamma - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})),$$
for any $\alpha \in (0, 1)$.
- *If power-sigmoid activation functions are used, then we can remove the design-parameter requirement of* γ *being large enough, and, in addition, superior robustness properties (e.g. faster convergence and smaller steady-state error) exist on the whole error range* $e_i(t) \in (-\infty, +\infty)$*, as compared to the linear-activation case.*

The proof of theorem 3 can be found in the appendix.

4. Simulative verification

The robustness analysis of the perturbed ZNN solver (10) has been presented in the above section. As shown by the theorems in section 3, the solution errors $e(t)$ of the ZNN could converge to a steady-state error bound in the context of the model-implementation errors. Observed from the formula of the maximal steady-state error in theorem 1, it is clearly found that the design parameter γ is inversely proportional to the maximal steady-state error, so we could increase the value of γ to decrease the upper bound of the steady-state solution error $e(t)$. In addition, theorem 3 indicates that using power-sigmoid activation functions could have a superior robustness in the perturbed ZNN model (10) as compared with using linear activation functions.

For the validity of the above theoretical results, let us consider the following time-varying coefficients of QM (1) and QP (3):

$$P(t) = \begin{bmatrix} 0.5\cos t + 2 & \sin t \\ \sin t & 0.5\sin t + 2 \end{bmatrix}, \qquad q(t) = \begin{bmatrix} \sin 3t \\ \cos 3t \end{bmatrix},$$

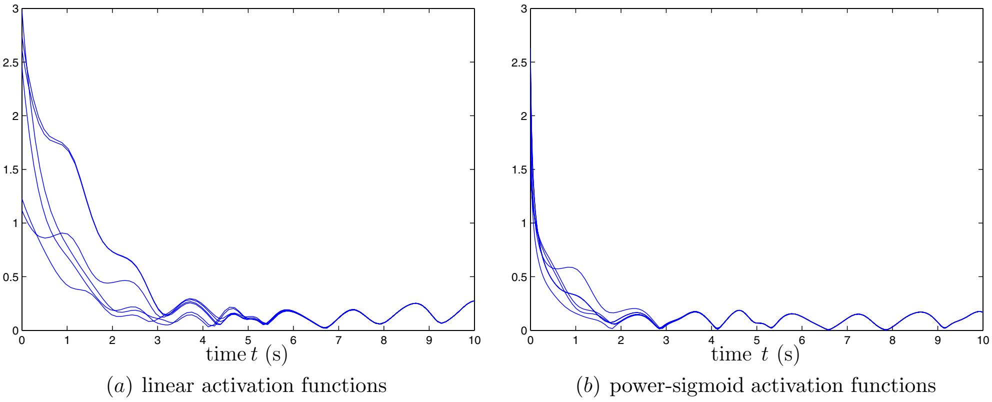
and the following time-varying coefficients of linear-equality constraint (4) in QP:

$$A(t) = [\sin 4t \quad \cos 4t], \qquad b(t) = \cos 2t.$$

It follows from equation (5) that we have

$$\tilde{P}(t) = \begin{bmatrix} 0.5\cos t + 2 & \sin t & \sin 4t \\ \sin t & 0.5\sin t + 2 & \cos 4t \\ \sin 4t & \cos 4t & 0 \end{bmatrix},$$

$$\tilde{q}(t) = [\sin 3t, \quad \cos 3t, \quad -\cos 2t]^T,$$



(a) linear activation functions

(b) power-sigmoid activation functions

Figure 1. Computational error $\|x(t) + P^{-1}(t)q(t)\|_2$ of the perturbed ZNN model (10) with $\gamma = 1, \xi = 4$ and $p = 3$.

and the perturbed ZNN solver (10) with time-varying model-implementation errors as below (with $\varepsilon_{\rm D} = \varepsilon_{\rm m} = 0.5$).

(1) In the time-varying QM case:

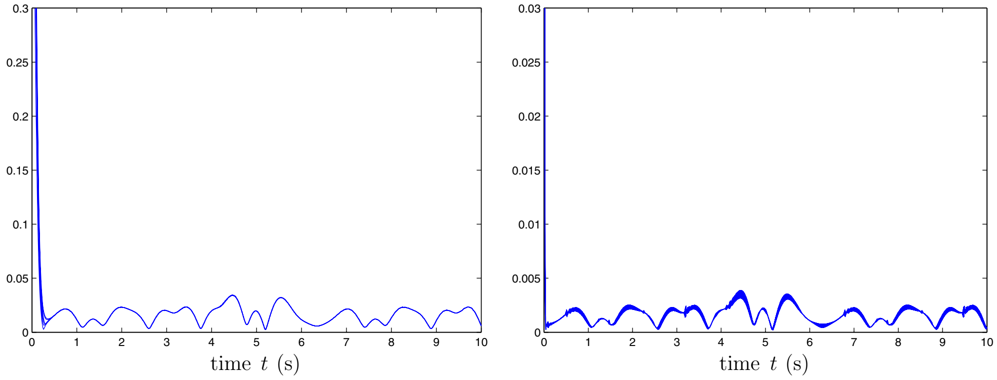
$$\Delta_{\rm D}(t) = \varepsilon_{\rm D}\begin{bmatrix}\cos 4t & -\sin 4t\\ \sin 4t & \cos 4t\end{bmatrix}, \qquad \Delta_{\rm m}(t) = \varepsilon_{\rm m}\begin{bmatrix}\sin 4t\\ \cos 4t\end{bmatrix}.$$

(2) In the time-varying QP case:

$$\Delta_{\rm D}(t) = \varepsilon_{\rm D}\begin{bmatrix}\cos 4t & -\sin 4t & \sin 6t\\ \sin 4t & \cos 4t & \cos 6t\\ \sin 6t & \cos 6t & 0\end{bmatrix}, \qquad \Delta_{\rm m}(t) = \varepsilon_{\rm m}\begin{bmatrix}\sin 4t\\ \cos 4t\\ \sin 4t\end{bmatrix}.$$

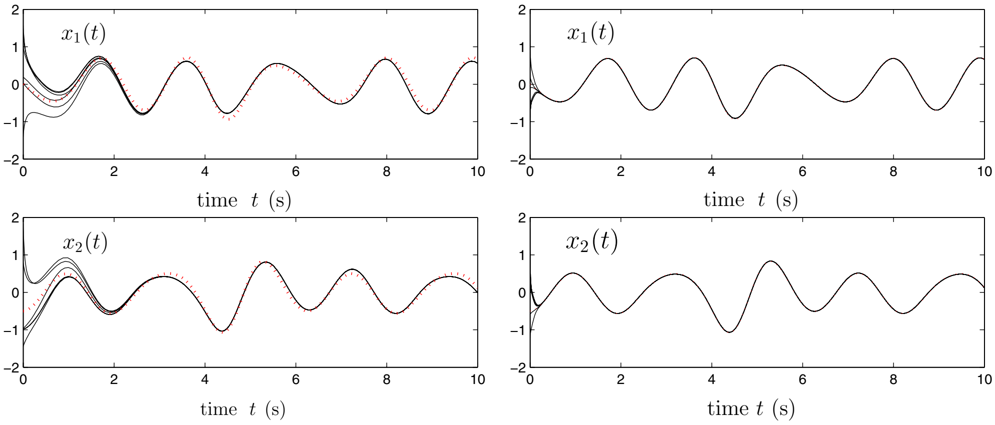
Starting from six randomly generated initial state $x(0)$, the computer-simulation results of the time-varying QM problem are shown in figures 1–3. As seen from figures 1 and 2, with the model-implementation errors, the computational error $\|x(t) + P^{-1}(t)q(t)\|_2$ of the perturbed ZNN (10) is still bounded and very small. In addition, with $\gamma = 1$, the convergence time of the perturbed ZNN (10) using power-sigmoid activation functions is faster than that using linear activation functions, and using power-sigmoid activation functions has smaller steady-state residual error than using linear activation functions. Moreover, comparing figures 1–3, we can see that, as the design parameter γ increases from 1 to 10 and then to 100, the convergence is evidently expedited and the upper bound of the steady-state solution error is decreased substantially (from around 0.17 to 0.03 and then to 0.003). From figure 3(b), we see little difference between the expected target output and the ZNN output when $\gamma = 10$. These computer-simulation results substantiate well the theoretical results.

The computer-simulation results of the time-varying QP problem solving have been shown in figures 4 through 6, which are similar to the time-varying QM problem. Figure 4 also indicates the better convergence and robustness of the perturbed ZNN (10) using power-sigmoid activation functions, compared with the linear case. As the same situation of time-varying QM problems, increasing γ from 1 to 10 and then to 100, the upper bound of the steady-state solution error decreases from roughly 0.9 to 0.12, and then to 0.013. Moreover, from figure 6(b), we see little difference between the expected target output and the ZNN output when $\gamma = 10$. In summary, the above simulation results and observations have agreed well with the theoretical results presented in section 3.



(a) synthesized with $\gamma = 10$ (b) synthesized with $\gamma = 100$

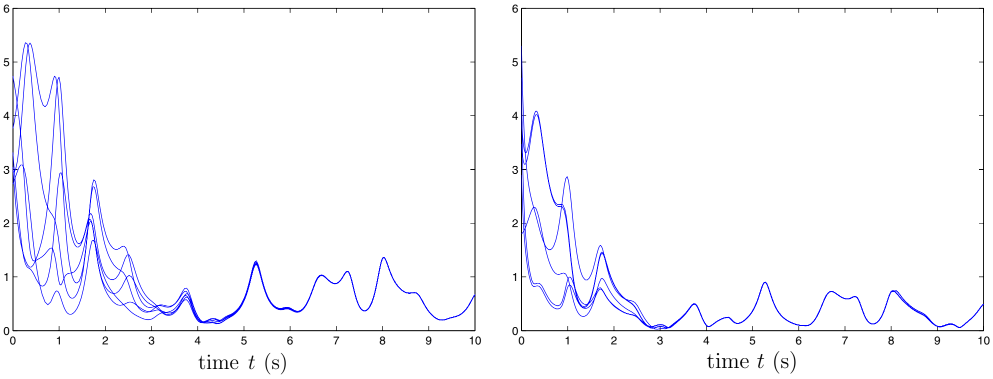
Figure 2. Computational error $\|x(t) + P^{-1}(t)q(t)\|_2$ of the perturbed ZNN model (10) using power-sigmoid activation functions with $\xi = 4$ and $p = 3$.



(a) synthesized with $\gamma = 1$ (b) synthesized with $\gamma = 10$

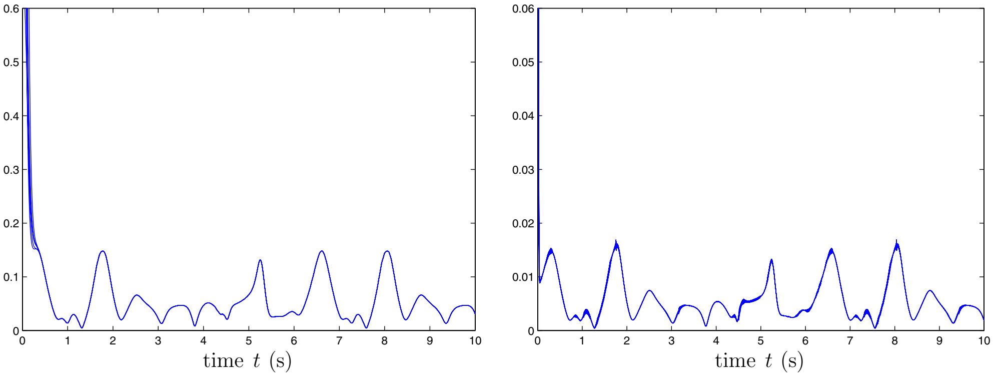
Figure 3. The γ-related robustness of the perturbed ZNN model (10) using power-sigmoid activation functions, where the dashed–dotted curves correspond to the theoretical solution $-P^{-1}(t)q(t)$ and the solid curves correspond to the ZNN-computed solutions starting with randomly-generated initial states.

As presented in theorem 3, when an entry computational error $|e_i(t)|$ is larger than 1, the perturbed ZNN (10) using power-sigmoid activation functions could further reduce $|e_i(t)|$ into a smaller error bound, which guarantees the superior property of the ZNN model solving time-varying QM and QP problems, e.g. faster convergence and smaller steady-state error. Specially, in the situation involving a large model-implementation error, superior robustness can be achieved by using power-sigmoid activation functions exploited by the perturbed ZNN model (10). Now, consider the above time-varying QM and QP problems again but perturbed with relatively large errors (with $\varepsilon_{\mathrm{D}} = \varepsilon_{\mathrm{m}} = 20$). The robustness simulation results are shown in figures 7 and 8. From figure 7, we can see that, even with very large model-implementation errors, the computational error



(a) linear activation functions (b) power-sigmoid activation functions

Figure 4. Computational error $\|x(t)+\tilde{P}^{-1}(t)\tilde{q}(t)\|_2$ of the perturbed ZNN model (10) with $\gamma = 1, \xi = 4$ and $p = 3$.



(a) synthesized with $\gamma = 10$ (b) synthesized with $\gamma = 100$

Figure 5. Computational error $\|x(t)+\tilde{P}^{-1}(t)\tilde{q}(t)\|_2$ of the perturbed ZNN model (10) using power-sigmoid activation functions with $\xi = 4$ and $p = 3$.

$\|x(t)+P^{-1}(t)q(t)\|$ synthesized by the perturbed ZNN dynamic system (10) using power-sigmoid activation functions is still bounded and relatively very small (e.g. roughly 7.7). In contrast, using linear activation functions for (10) in the context of very large model-implementation errors still generates very large computational error (e.g. more than 2.1×10^4 as depicted in figure 7(a)). Evidently, compared to the situation of using linear activation functions, superior performance can be achieved by using power-sigmoid activation functions under the same simulation conditions. The reason for this is that in this larger error case, $\gamma = 1$ cannot satisfy the design parameter requirement on γ when using linear activation functions, so the upper bound of $\dot{v}$ in theorem 1 is always a positive scalar, which might generate the worst situation $\dot{v} > 0$ and increase the computational error of the perturbed ZNN model (10) continuously. Correspondingly, using a power-sigmoid activation function can remove the requirement on γ presented in theorem 3. To sum up, the results of figure 7 have agreed well with theorem 3. Furthermore, figure 8 shows the influence of increasing the value of p. When increasing p from 3 to 5, and to 7, the

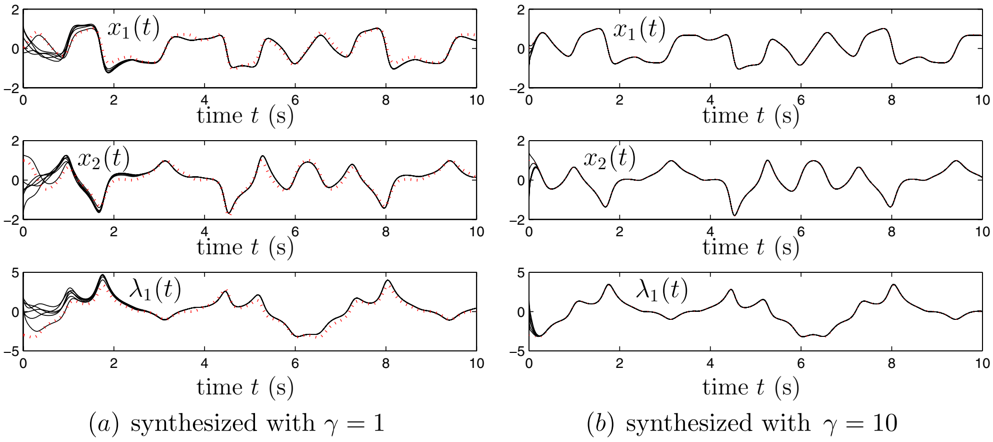


Figure 6. The γ-related robustness of the perturbed ZNN model (10) using power-sigmoid activation functions, where the dashed-dotted curves correspond to theoretical solution $-\tilde{P}^{-1}(t)\tilde{q}(t)$ and the solid curves correspond to the ZNN-computed solutions starting with randomly-generated initial states.

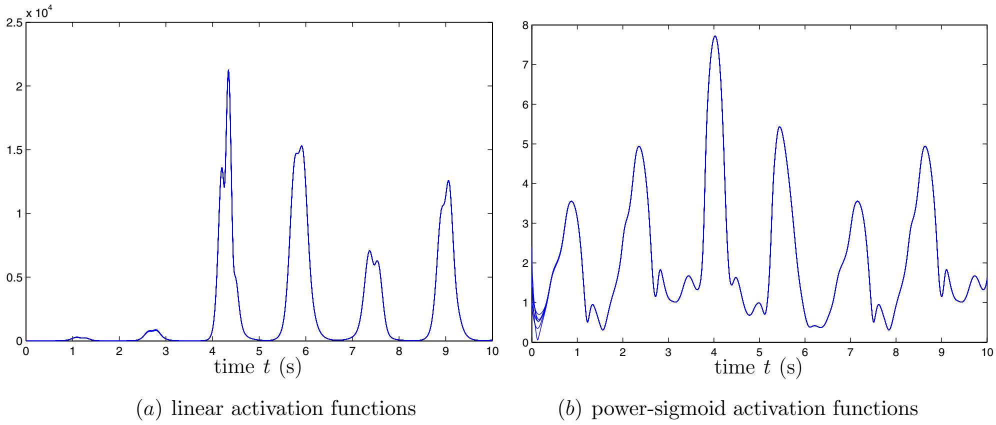
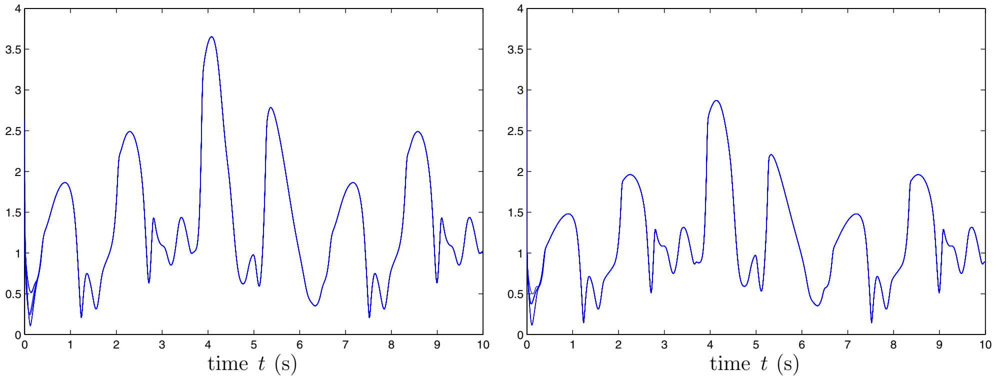


Figure 7. Computational error $\|x(t)+P^{-1}(t)q(t)\|_2$ of the perturbed ZNN model (10) using power-sigmoid activation functions with $\gamma = 1, \xi = 4$ and $p = 3$ and with very large implementation error.

upper bound of the steady-state solution error decreases from roughly 7.7 to 3.7, and then to 2.8. These simulation results have also substantiated well the theoretical results presented before.

One advantage of recurrent neural networks is the nature of parallel and distributed processing. Compared with the serial-processing algorithms, this kind of approach provides a powerful alternative in real-time applications, especially in the large-scale cases. For investigating this computational advantage of the ZNN, we would like to give an example with different dimensions n of time-varying coefficients $W(t)$ and $u(t)$. Let us consider the



(a) synthesized with $p = 5$

(b) synthesized with $p = 7$

Figure 8. Computational error $\|x(t) + P^{-1}(t)q(t)\|_2$ of the perturbed ZNN model (10) using power-sigmoid activation functions with $\gamma = 1$ and $\xi = 4$ and with very large implementation error.

following time-varying Toeplitz matrix $W(t)$:

$$W(t) = \begin{bmatrix} w_1(t) & w_2(t) & w_3(t) & \cdots & w_n(t) \\ w_2(t) & w_1(t) & w_2(t) & \cdots & w_{n-1}(t) \\ w_3(t) & w_2(t) & w_1(t) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & w_2(t) \\ w_n(t) & w_{n-1}(t) & \cdots & w_2(t) & w_1(t) \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where $w(t) = [w_1(t), w_2(t), w_3(t), \ldots, w_n(t)]^T$ denotes the first column vector of the matrix $W(t)$. Let $w_1(t) = 5 + \sin(t)$ and $w_k(t) = \cos(t)/(k-1)(k = 2, 3, \ldots, n)$. The time-varying vector $u(t)$ is

$$u(t) = \left[\sin(3t), \sin\left(3t + \frac{\pi}{2}\right), \cdots, \sin\left(3t + \frac{(n-1)\pi}{2}\right)\right]^T \in \mathbb{R}^{n\times 1}.$$

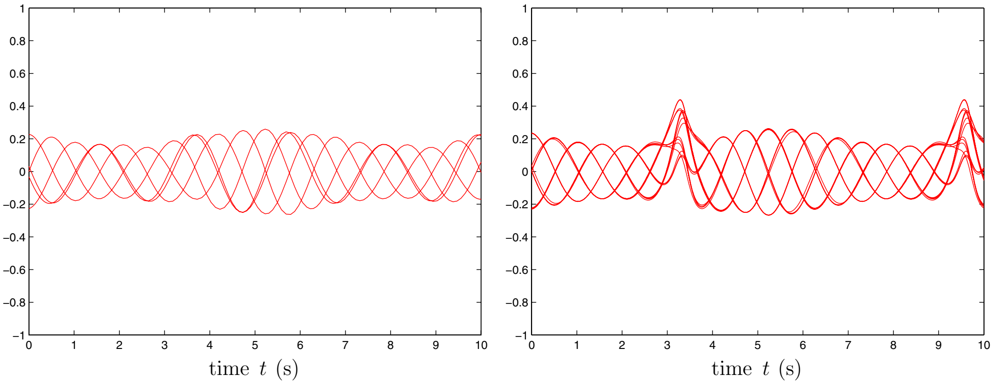
In addition, we have the following model-implementation errors (with $\varepsilon_{\mathrm{D}} = \varepsilon_{\mathrm{m}} = 0.5$):

$$\Delta_{\mathrm{D}}(t) = \varepsilon_{\mathrm{D}} \begin{bmatrix} d_1(t) & d_2(t) & d_3(t) & \cdots & d_n(t) \\ d_2(t) & d_1(t) & d_2(t) & \cdots & d_{n-1}(t) \\ d_3(t) & d_2(t) & d_1(t) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & d_2(t) \\ d_n(t) & d_{n-1}(t) & \cdots & d_2(t) & d_1(t) \end{bmatrix} \in \mathbb{R}^{n\times n},$$

$$\Delta_{\mathrm{m}}(t) = \varepsilon_{\mathrm{m}} \left[\cos(4t), \cos\left(4t - \frac{\pi}{2}\right), \cdots, \cos\left(4t - \frac{(n-1)\pi}{2}\right)\right]^T \in \mathbb{R}^{n\times 1},$$

where $d(t) = [d_1(t), d_2(t), d_3(t), \ldots, d_n(t)]^T$ denotes the first column vector of the matrix $\Delta_{\mathrm{D}}(t)$. Let $d_k(t) = \cos\left(4t + \frac{(k-1)\pi}{2}\right)(k = 1, 2, 3, \ldots, n)$.

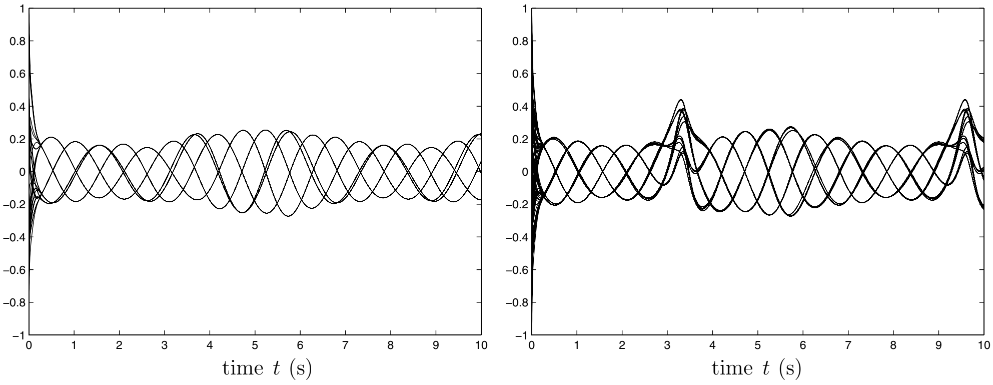
When the design parameter $\gamma = 10$ and power-sigmoid activation functions (with $\xi = 4$ and $p = 3$) are used, the robustness performance of the perturbed ZNN model (10) can be seen in figures 10. For clearer comparison, figure 9 shows entry trajectories of the theoretical



(a) $n = 5$

(b) $n = 15$

Figure 9. Entry trajectories of the theoretical solution $W^{-1}(t)u(t)$.



(a) $n = 5$

(b) $n = 15$

Figure 10. Entry trajectories of ZNN-computed solutions using power-sigmoid activation functions with $\gamma = 10$, $\xi = 4$ and $p = 3$.

solution $W^{-1}(t)u(t)$. As shown in figure 10, starting from six randomly generated initial states, the state matrix $y(t)$ of the perturbed ZNN model (10) always converges to the theoretical time-varying solution, even if the dimension n of the given problem increases from 5 to 15 (which have the small computational error $\|y(t) - W^{-1}(t)u(t)\|_2$, i.e. roughly 0.01 and roughly 0.06, respectively).

The time-varying coefficients of QM (1) and QP (3) in the above simulation are based on the sinusoidal form. Now, let us consider the following exponential form in the time-varying QM problem (with the same time-varying model-implementation errors and $\varepsilon_{\mathrm{D}} = \varepsilon_{\mathrm{m}} = 0.5$):

$$P(t) = \begin{bmatrix} \exp(0.5t) + 1 & \exp(0.2t) \\ \exp(0.2t) & \exp(0.5t) + 1 \end{bmatrix}, \qquad q(t) = \begin{bmatrix} \exp(0.2t) \\ \exp(-0.1t) \end{bmatrix}.$$

Similarly, the computational error $\|x(t) + P^{-1}(t)q(t)\|_2$ of the perturbed ZNN model (10) also has a very small value (with the related figures omitted). The results are similar to the above time-varying problem, which shows that the perturbed ZNN model (10) also has a good robustness performance when exponential functions are considered.

5. Conclusions

A new type of recurrent neural networks has recently been proposed by Zhang *et al* for time-varying problem solving. As the time-derivative information of problem coefficients is exploited, global exponential convergence of such ZNN models for time-varying QM and QP problems could be achieved readily under ideal conditions. The main contributions of this paper lie in the following facts. Firstly, by considering the differentiation and dynamics-implementation errors, we have investigated the robustness properties of the ZNN model for time-varying QM and QP problem solving. As a result, even with relatively large model-implementation errors, the computational error of the perturbed ZNN model is still upper bounded. Secondly, we have proven that the convergence time and steady-state residual error can be reduced obviously by increasing the design parameter γ, and using power-sigmoid activation functions, the perturbed ZNN model can have better robustness than that using linear activation functions. Thirdly, computer-simulation results have been presented, which fitted well with the theoretical analysis and further demonstrated the excellent robustness of the ZNN models for time-varying linear equation solving.

Appendix

The proof of theorem 1. As the time derivative of the residual-error vector $e(t) = W(t)y(t) - u(t)$ is $\dot{e}(t) = W(t)\dot{y}(t) + \dot{W}(t)y(t) - \dot{u}(t)$, the perturbed ZNN model (10) can be reformulated as follows:

$$\dot{e} = -\gamma\mathcal{F}(e) - \Delta_{\mathrm{D}}W^{-1}e + \Delta_{\mathrm{m}} - \Delta_{\mathrm{D}}W^{-1}u. \tag{A.1}$$

We could then define a Lyapunov function candidate $v = \|e\|_2^2/2 = e^Te/2 = \sum_{i=1}^n e_i^2(t)/2 \geqslant 0$ for dynamic equation (A.1). Evidently, v is positive definite in the sense that $v > 0$ for any $e \neq 0$ and $v = 0$ only for $e = 0$. In addition, $v \to \infty$ as $\|e\|_2 \to \infty$. Furthermore, the time derivative of v is

$$\begin{aligned}\dot{v} &= e^T\dot{e} = e^T(-\gamma\mathcal{F}(e) - \Delta_{\mathrm{D}}W^{-1}e + \Delta_{\mathrm{m}} - \Delta_{\mathrm{D}}W^{-1}u)\\ &= -\gamma e^T\mathcal{F}(e) + e^TQe + e^T\Delta_{\mathrm{m}} + e^T(-\Delta_{\mathrm{D}}W^{-1}u)\\ &= -\gamma e^T\mathcal{F}(e) + e^T\frac{Q+Q^T}{2}e + e^T\Delta_{\mathrm{m}} + e^T(-\Delta_{\mathrm{D}}W^{-1}u)\end{aligned} \tag{A.2}$$

with $Q := -\Delta_{\mathrm{D}}W^{-1}$. For the second term of the above equation, it follows from $\max_{1\leqslant i\leqslant n}|\lambda_i(W)| \leqslant \|W\|_F$ that

$$\begin{aligned}e^T\frac{Q+Q^T}{2}e &\leqslant e^Te\max_{1\leqslant i\leqslant n}\left|\lambda_i\left(\frac{Q+Q^T}{2}\right)\right|\\ &= e^Te\max_{1\leqslant i\leqslant n}\left|\lambda_i\left(\frac{\Delta_{\mathrm{D}}W^{-1} + (\Delta_{\mathrm{D}}W^{-1})^T}{2}\right)\right|\\ &\leqslant e^Te\left\|\frac{\Delta_{\mathrm{D}}W^{-1} + (\Delta_{\mathrm{D}}W^{-1})^T}{2}\right\|_F\\ &\leqslant e^Te\|\Delta_{\mathrm{D}}\|_F\|W^{-1}\|_F \leqslant e^Te\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}.\end{aligned}$$

In addition, for the third and fourth terms of the above $\dot{v}$ equation, it follows from $\max_{1\leqslant i\leqslant n}|u_i| \leqslant \|u\|_2$ that

$$\begin{aligned} e^T\Delta_{\mathrm{m}} &\leqslant \sum_{i=1}^{n}|e_i|\max_{1\leqslant i\leqslant n}|[\Delta_{\mathrm{m}}]_i| \\ &\leqslant \sum_{i=1}^{n}|e_i|\|\Delta_{\mathrm{m}}\|_2 \leqslant \sum_{i=1}^{n}|e_i|\varepsilon_{\mathrm{m}}, \end{aligned}$$

and from the vector–matrix norms' relation $\|u\|_2 = \|u\|_F$ [39] that

$$\begin{aligned} e^T(-\Delta_{\mathrm{D}}W^{-1}u) &\leqslant \sum_{i=1}^{n}|e_i|\max_{1\leqslant i\leqslant n}|[\Delta_{\mathrm{D}}W^{-1}u]_i| \\ &\leqslant \sum_{i=1}^{n}|e_i|\|\Delta_{\mathrm{D}}W^{-1}u\|_2 \leqslant \sum_{i=1}^{n}|e_i|\|\Delta_{\mathrm{D}}W^{-1}\|_F\|u\|_2 \\ &\leqslant \sum_{i=1}^{n}|e_i|\|\Delta_{\mathrm{D}}\|_F\|W^{-1}\|_F\|u\|_2 \leqslant \sum_{i=1}^{n}|e_i|\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}. \end{aligned}$$

Hence, in view of the above inequalities, we have

$$\begin{aligned} \dot{\upsilon} &\leqslant -\gamma e^T\mathcal{F}(e) + e^Te\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}} + \sum_{i=1}^{n}|e_i|\varepsilon_{\mathrm{m}} + \sum_{i=1}^{n}|e_i|\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}} \\ &= -\sum_{i=1}^{n}|e_i|\left(\gamma f(|e_i|) - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}\right). \end{aligned} \tag{A.3}$$

The following two situations can now be analyzed.

(i) For the time interval $[t_0, t_1)$, if $\gamma f(|e_i|) - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}} \geqslant 0$, $\forall i \in \{1, 2, \ldots, n\}$, then $\dot{\upsilon} \leqslant 0$ which implies that the residual-error vector $e(t)$ of (A.1) converges toward zero (correspondingly, the state $y(t)$ of the perturbed ZNN model (10) converges toward the time-varying theoretical solution $y^*(t) = W^{-1}(t)u(t)$). As time t evolves, if $\dot{\upsilon} = 0$ at a certain time instant t_2, the residual-error vector $e(t)$ of (A.1) achieves the steady state; otherwise, it would fall into the following situation as the decreasing of $\gamma f(|e_i|) - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}$.

(ii) For any time instant t, if $\gamma f(|e_i|) - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}} < 0$, $\exists i \in \{1, 2, \ldots, n\}$, then the upper bound of $\dot{\upsilon}$, that is $-\sum_{i=1}^{n}|e_i|\left(\gamma f(|e_i|) - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}\right)$, might be a positive scalar, which means that $\dot{\upsilon}$ contains two cases $\dot{\upsilon} \leqslant 0$ and $\dot{\upsilon} > 0$, and thus $e(t)$ may not converge toward zero (correspondingly, the state $y(t)$ of the perturbed ZNN model (10) may not converge toward the time-varying theoretical solution $y^*(t) = W^{-1}(t)u(t)$ in this situation). Now let us analyze the worst case, i.e. $\dot{\upsilon} > 0$: it is readily known that $e(t)$ diverges outward, and $|e_i(t)|$ increases, which decreases the upper bound of $\dot{\upsilon}$ as well (under requirement $\gamma f(|e_i(t)|) - \varepsilon_D\varphi_A|e_i(t)| \geqslant 0$), as time t evolves. So, there must exist a certain time instant t_3 such that $\dot{\upsilon} = 0$ (or < 0), which makes $e(t)$ achieve the steady state (or decreases again).

From the above theoretical analysis, the residual-error vector $e(t)$ of the perturbed ZNN model (10) could not always diverge, and is limited into a certain upper bound. For the convenience of theoretical estimate, we rewrite inequality (A.3) as

$$\begin{aligned} \dot{\upsilon} &\leqslant -\sum_{i=1}^{n}|e_i|\left(\gamma\rho|e_i| - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}\right) \\ &= -(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\sum_{i=1}^{n}|e_i|\left(|e_i| - \frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}}\right), \end{aligned} \tag{A.4}$$

where $\rho \geqslant 1$ exists. Now, assuming $\forall i \in \{1, 2, \ldots, n-1\}$,

$$|e_i| = \frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{2(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}$$

under the design parameter requirement $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}/\rho$, so the summation of the former $n-1$ elements of equation (A.4) can achieve the maximum value. In this situation, calculating $|e_n|$ to zero (A.4),

$$-\sum_{i=1}^{n} |e_i| \left(|e_i| - \frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}} \right)$$
$$= -|e_n|^2 + \frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}}|e_n| + \frac{n-1}{4}\left(\frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}}\right)^2 = 0,$$

and the solution of the above equation is

$$|e_n| = \frac{1}{2}(1+\sqrt{n})\frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}}.$$

So the following inequality always holds true for the perturbed ZNN model (10):

$$\max_{1\leqslant i\leqslant n} |e_i(t)| \leqslant \frac{1}{2}(1+\sqrt{n})\frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}}. \tag{A.5}$$

The above inequality indicates that all error terms, $|e_i(t)|, \forall i \in \{1, 2, \ldots, n\}$, could not go beyond the upper bound. Because once a residual error $|e_j(t)|$, $j \in \{1, 2, \ldots, n\}$, is beyond the upper bound, the upper bound of $\dot{v}$ must be a negative value, which would force $\|e(t)\|_2$ to decrease in view of $v = \|e\|_2^2/2$. In this situation, this residual error $|e_j(t)|$ would decease into the bound to stop v deceasing. On the other hand, we have

$$\begin{aligned}\|y(t) - W^{-1}(t)u(t)\|_2 &= \|W^{-1}(t)\,(W(t)y(t) - u(t))\,\|_2 \\ &\leqslant \|W^{-1}(t)\|_F\|e(t)\|_2 \leqslant \varphi_{\mathrm{W}}\sqrt{\Sigma_i^n e_i^2(t)} \\ &\leqslant \sqrt{n}\varphi_{\mathrm{W}} \max_{1\leqslant i\leqslant n} |e_i(t)|.\end{aligned}$$

Thus, it follows from the above estimations that

$$\lim_{t\to\infty} \|y(t) - W^{-1}(t)u(t)\|_2 \lesssim \frac{(\sqrt{n}+n)\varphi_{\mathrm{W}}(\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})}{2(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}.$$

Evidently, the steady-state computational error of the perturbed ZNN model (10) can be made arbitrarily small by increasing the value of the design parameter γ: as $\gamma \to \infty$, such a steady-state error of the perturbed ZNN model (10) decreases to zero. □

To prove theorem 2, we first give a lemma.

Lemma 2. *Given vectors $a \in R^n$ and $b \in R^n$, the following inequality holds:*

$$a^T b \leqslant \|a\|_2\|b\|_2.$$

The proof of theorem 2. Following the proof of theorem 1, for the perturbed ZNN (10), we define the solution error $e(t) = W(t)y(t) - u(t)$. The perturbed ZNN (10) is then transformed into the following error dynamic equation (with $e(0) = W(0)y(0) - u(0)$):

$$\dot{e} = -\gamma\mathcal{F}(e) - \Delta_{\mathrm{D}}W^{-1}e + \Delta_{\mathrm{m}} - \Delta_{\mathrm{D}}W^{-1}u.$$

Define a Lyapunov-like positive-definite function $v(t) = \|e(t)\|_2^2/2$. The time derivative of $v(t)$ along the state trajectory of (A.1) is derived as (A.2) and repeated below for the readers' convenience:

$$\dot{v} = -\gamma e^T\mathcal{F}(e) + e^T\frac{Q+Q^T}{2}e + e^T\Delta_{\mathrm{m}} + e^T(-\Delta_{\mathrm{D}}W^{-1}u). \tag{A.6}$$

With the same results on the first two terms of the above equation, we rederive the following two inequalities from the remainder term (according to lemma 2):

$$e^T\Delta_{\mathrm{m}} \leqslant \|e(t)\|_2\|\Delta_{\mathrm{m}}\|_2 \leqslant \varepsilon_{\mathrm{m}}\|e(t)\|_2, \tag{A.7}$$

and

$$\begin{aligned} e^T(-\Delta_{\mathrm{D}}W^{-1}u) &\leqslant \|e(t)\|_2\|\Delta_{\mathrm{D}}W^{-1}u\|_2 \\ &\leqslant \|e(t)\|_2\|\Delta_{\mathrm{D}}\|_F\|W^{-1}\|_F\|u\|_2 \\ &\leqslant \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}\|e(t)\|_2. \end{aligned} \tag{A.8}$$

Then, following (A.4) and substituting (A.7) and (A.8) into (A.6) yields the following:

$$\begin{aligned} \dot{v} &\leqslant -\gamma\rho\|e(t)\|_2^2 + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\|e(t)\|_2^2 + \varepsilon_{\mathrm{m}}\|e(t)\|_2 + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}\|e(t)\|_2 \\ &= -(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(t)\|_2^2 + (\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})\|e(t)\|_2 \\ &= -(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(t)\|_2^2 \\ &\quad + [-\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(t)\|_2^2 + (\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})\|e(t)\|_2], \end{aligned} \tag{A.9}$$

where the weighting parameter $\alpha \in (0, 1)$ could be termed as 'a loosing ratio'.

Evidently, observed from (A.9), the first term $-(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(t)\|_2^2 \leqslant 0$ under the design-parameter requirement $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}/\rho$. So, for the solution error $e(t)$ satisfying

$$\|e(t)\|_2 \geqslant \frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})},$$

the second term of (A.9) could be dropped. It follows from (A.9) that

$$\begin{aligned} \dot{v}(t) &\leqslant -(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(t)\|_2^2 \\ &= -2(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})v(t); \end{aligned}$$

thus, we have the following exponential convergence about $\|e(t)\|_2$:

$$\begin{aligned} &v(t) \leqslant \exp\left(-2(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})t\right) v(0) \\ &\|e(t)\|_2 \leqslant \exp\left(-(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})t\right) \|e(0)\|_2, \qquad \forall t \in [0, t_c], \end{aligned} \tag{A.10}$$

with the exponential convergence rate $(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ and the convergence time

$$t_c = \ln\left(\frac{\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(0)\|_2}{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}\right)\Big/((1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}))$$

in view of

$$\begin{aligned} &\exp\left(-(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})t_c\right) \|e(0)\|_2 = \frac{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}, \\ &((1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}))t_c = \ln\left(\frac{\alpha(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(0)\|_2}{\varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}\right). \end{aligned}$$

Thus, from (A.10) and the above analysis, defining the loosing ratio $\alpha \in (0, 1)$ and convergence time t_c, we could have

$$\|e(t)\|_2 \begin{cases} \begin{cases} \leqslant \exp(-(1-\alpha)(\gamma\rho - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})t)\|e(0)\|_2, & \forall t \in [0, t_c], \\ \leqslant \frac{\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\alpha(\gamma\rho-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}, & \forall t \in [t_c, \infty), \end{cases} & \|e(0)\|_2 \geqslant \frac{\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\alpha(\gamma\rho-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}; \\ \leqslant \frac{\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\alpha(\gamma\rho-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}, \quad \forall t \in [0, \infty), & \|e(0)\|_2 \leqslant \frac{\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\alpha(\gamma\rho-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})}; \end{cases}$$

The proof is thus completed. □

The proof of theorem 3. For the linear activation function, the parameter $\rho \equiv 1$. From the proof of theorems 1 and 2, we readily have $\max_{1\leqslant i\leqslant n}|e_i(t)| \leqslant (1+\sqrt{n})(\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/2(\gamma-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ under the design-parameter requirement $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}$. Moreover, the solution error $\|e(t)\|_2$ of the perturbed ZNN (10) is globally exponentially convergent to or staying within the error bound $(\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/\alpha(\gamma-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$, where the exponential convergence rate is $(1-\alpha)(\gamma-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ and the convergence time is

$$t_c = \ln\left(\frac{\alpha(\gamma-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})\|e(0)\|_2}{\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}\right)\Big/((1-\alpha)(\gamma-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})),$$

for any $\alpha \in (0,1)$.

For the power-sigmoid activation function, the parameter $\rho \geqslant 1$ (where the sign of equality is taken only for $|e_i(t)| = 1$). From the results of theorem 2, we easily find that, compared to the linear-activation-function case $\rho \equiv 1$, the error bound $(\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/\alpha(\gamma\rho-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ is much smaller (in view of that the parameter ρ is inversely proportional to the error bound), and the exponential convergence rate $(1-\alpha)(\gamma\rho-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$ is also faster (in view of the exponential decay with the parameter ρ). In addition, according to theorem 1, we have the same results, for which the (steady-state) analysis has two parts, i.e. $|e_i(t)| \leqslant 1$ and $|e_i(t)| > 1$.

(i) For a small error $|e_i(t)| \leqslant 1$, the sigmoid part of $f(\cdot)$ is activated with $f(|e_i(t)|) \geqslant |e_i(t)|$ and $\rho \geqslant 1$. Compared to the linear-activation-function case $\rho \equiv 1$, $\dot{v}$ becomes more negative according to (A.4) (which implies a faster convergence). Meanwhile, the upper bound of $|e_i(t)|$ becomes smaller according to (A.5). In addition, the design-parameter requirement $\gamma > \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}/\rho$ is relaxed with $\rho \geqslant 1$, as compared to the linear-activation case.

(ii) For a large error $|e_i(t)| > 1$, the power part of $f(\cdot)$ is activated with $f(|e_i(t)|) = \left|e_i^p(t)\right| \geqslant |e_i(t)|$ and $\rho \geqslant 1$. From the proof of theorem 1, the first situation becomes $\gamma|e_i|^p - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| - \varepsilon_{\mathrm{m}} - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}} \geqslant 0, \forall i \in \{1,2,\ldots,n\}$, under the requirement $\gamma|e_i|^p - \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| > 0$, which is a sufficient condition for ensuring $\dot{v} \leqslant 0$. So, it has the following relationship:

$$\begin{aligned}\gamma|e_i|^p &\geqslant \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}|e_i| + \varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}\\ &> \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}} + \varepsilon_{\mathrm{m}} + \varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}.\end{aligned}$$

Thus, following the above inequality and the design-parameter requirement, $|e_i(t)|$ is required as

$$|e_i| > \max\left(\sqrt[p-1]{\frac{\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}}{\gamma}}, \sqrt[p]{\frac{\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}+\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}}}{\gamma}}\right). \tag{A.11}$$

Correspondingly, in the linear-activation-function case, the requirement of $|e_i(t)|$ for the sufficient condition of $\dot{v} \leqslant 0$ is larger than $(\varepsilon_{\mathrm{m}}+\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}}\varphi_{\mathrm{u}})/(\gamma-\varepsilon_{\mathrm{D}}\varphi_{\mathrm{W}})$. Clearly, the requirement of $|e_i(t)|$ in the power-sigmoid activation function case is much smaller than that of the linear-function case, in view of the $(p-1)$st/pth root (which means easier satisfaction of the sufficient condition of $\dot{v} \leqslant 0$ in this error range and smaller $|e_i(t)|$ in power-sigmoid function cases). In addition, (A.11) indicates that the parameter requirement on γ can be removed in this case. Furthermore, when the model-implement errors are larger, the amplifying effect of the power activation function in this error range, $\left|e_i^p(t)\right| \geqslant |e_i(t)|$, makes $\dot{v}$ more negative (which implies a faster convergence again).

Summarizing the above analysis, we know that, for the whole error range $e_i(t) \in (-\infty, +\infty)$ (except the isolated points $|e_i(t)| = 1, \forall i$, with equal performance), the perturbed ZNN model (10) using power-sigmoid activation functions has superior robustness, as compared to the case of using linear activation functions. Furthermore, the results of theorem 2 indicate how a faster and lower bound could be obtained when using the power-sigmoid function, compared with the linear function. □

References

[1] Johansen T A, Fsosen T I and Berge S P 2004 Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming *IEEE Trans. Control Syst. Technol.* **12** 211–6

[2] Grudinin N 1998 Reactive power optimization using successive quadratic programming method *IEEE Trans. Power Syst.* **13** 1219–25

[3] Wang J and Zhang Y 2004 Recurrent neural networks for real-time computation of inverse kinematics of redundant manipulators *Machine Intelligence Quo Vadis?* (Singapore: World Scientific)

[4] Leithead W E and Zhang Y 2007 O(N^2)-operation approximation of covariance matrix inverse in Gaussian process regression based on quasi-Newton BFGS method *Commun. Stat. Simul. Comput.* **36** 367–80

[5] Ichiki A and Shiino M 2007 The Thouless–Anderson–Palmer equation for an analogue neural network with temporally fluctuating white synaptic noise *J. Phys. A: Math. Theor.* **40** 9201–11

[6] Zhang Y, Cai B, Zhang L and Li K 2008 Bi-criteria velocity minimization of robot manipulators using a linear variational inequalities-based primal-dual neural network and PUMA560 example *Adv. Robot.* **22** 1479–96

[7] Zhang Y, Wang J and Xu Y 2002 A dual neural network for bi-criteria kinematic control of redundant manipulators *IEEE Trans. Robot. Automat.* **18** 923–31

[8] Xu X 2008 Complicated dynamics of a ring neural network with time delays *J. Phys. A: Math. Theor.* **41** 035102

[9] Metz F L and Theumann W K 2008 Instability of frozen-in states in synchronous Hebbian neural networks *J. Phys. A: Math. Theor.* **41** 265001

[10] Metz F L and Theumann W K 2009 Symmetric sequence processing in a recurrent neural network model with a synchronous dynamics *J. Phys. A: Math. Theor.* **42** 385001

[11] Zhang Y 2009 *Recurrent Neural Networks* (Saarbrücken: LAP LAMBERT Academic)

[12] Zhang Y, Ma W and Cai B 2009 From Zhang neural network to Newton iteration for matrix inversion *IEEE Trans. Circuits Syst.* **56** 1405–15

[13] Zhang Y and Ge S S 2005 Design and analysis of a general recurrent neural network model for time-varying matrix inversion *IEEE Trans. Neural Netw.* **16** 1477–90

[14] Neves F S and Timme M 2009 Controlled perturbation-induced switching in pulse-coupled oscillator networks *J. Phys. A: Math. Theor.* **42** 345103

[15] Wang Y, Wang Z D and Liang J L 2009 Global synchronization for delayed complex networks with randomly occurring nonlinearities and multiple stochastic disturbances *J. Phys. A: Math. Theor.* **42** 135101

[16] Yoon B G, Choi J and Choi M Y 2008 Effects of neuronal loss in the dynamic model of neural networks *J. Phys. A: Math. Theor.* **41** 385102

[17] Zhang Y, Jiang D and Wang J 2002 A recurrent neural network for solving Sylvester equation with time-varying coefficients *IEEE Trans. Neural Netw.* **13** 1053–63

[18] Yi C and Zhang Y 2008 Analogue recurrent neural network for linear algebraic equation solving *Electron. Lett.* **44** 1078–9

[19] Wang J 1992 Electronic realization of recurrent neural network for solving simultaneous linear equations *Electron. Lett.* **28** 493–5

[20] Zhang Y 2006 Towards piecewise-linear primal neural networks for optimization and redundant robotics *Proc. IEEE Int. Conf. on Networking, Sensing and Control (Ft Lauderdale, FL, 23–25 April)* pp 374–9

[21] Zhang Y and Wang J 2002 Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment *IEEE Trans. Neural Netw.* **13** 633–44

[22] Steriti R J and Fiddy M A 1993 Regularized image reconstruction using SVD and a neural network method for matrix inversion *IEEE Trans. Signal Process.* **41** 3074–7

[23] Zhang Y and Ge S S 2003 A general recurrent neural network model for time-varying matrix inversion *Proc. 42nd IEEE Conf. on Decision and Control (HI, USA)* pp 6169–74

[24] Zhang Y and Peng H 2007 Zhang neural network for linear time-varying equations solving and its robotic application *Proc. 6th Int. Conf. on Machine Learning and Cybernetics* pp 3543–8

[25] Zhang Y, Chen K and Ma W 2007 MATLAB simulation and comparison of Zhang neural network and gradient neural network for online solution of linear time-varying equations *Proc. Int. Conf. on Life System Modeling and Simulation (Shanghai, 14–17 September)* pp 450–4

[26] Zhang Y, Guo X and Ma W 2008 Modeling and simulation of Zhang neural network for online linear time-varying equations solving based on MATLAB Simulink *Proc. 7th Int. Conf. on Machine Learning and Cybernetics (Kunming, 12–15 July)* pp 805–10

[27] Zhang Y, Yi C and Ma W 2009 Simulation and verification of Zhang neural network for online time-varying matrix inversion *Simul. Model. Pract. Theory* **17** 1603–17

[28] Zhang Y and Li Z 2009 Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints *Phys. Lett.* A **373** 1639–43

[29] Zhang Y, Li X and Li Z 2009 Modeling and verification of Zhang neural networks for online solution of time-varying quadratic minimization and programming *Proc. 4th Int. Symp. on Intelligence Computation and Applications (Huangshi, 23–25 October)* pp 101–10

[30] Carneiro N C F and Caloba L P 1995 A new algorithm for analog matrix inversion *Proc. 38th Midwest Symp. on Circuits Syst. (Rio de Janeiro, 13–16 August)* vol 1 pp 401–4

[31] Mead C 1989 *Analog VLSI and Neural Systems* (Reading, MA: Addison-Wesley)

[32] Wang J 1992 Recurrent neural network for solving quadratic programming problems with equality constraints *Electron. Lett.* **28** 1345–7

[33] Zhang Y, Wang J and Xia Y 2003 A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits *IEEE Trans. Neural Netw.* **14** 658–67

[34] Zhang Y, Ma W, Li X, Tan H and Chen K 2009 MATLAB Simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs *Neurocomputing* **72** 1679–87

[35] Zhang Y, Yin J and Cai B 2009 Infinity-norm acceleration minimization of robotic redundant manipulators using the LVI-based primal-dual neural network *Robot. Cim. Int. Manuf.* **25** 358–65

[36] Zhang Y and Li K 2009 Bi-criteria velocity minimization of robot manipulators using LVI-based primal-dual neural network and illustrated via PUMA560 robot arm *Robotica* **0** 1–13

[37] Boyd S and Vandenberghe L 2004 *Convex Optimization* (New York: Cambridge University Press)

[38] Nocedal J and Wright S J 1999 *Numerical Optimization* (Berlin: Springer)

[39] Chen Z and Sheng J 1998 *An Introduction to Matrix Theory* (Beijing: Beijing University of Aeronautics and Astronautics Press)