1

Zeroing Neural Networks for Control

1.1 Introduction

In addition to the remarkable features such as parallelism, distributed storage, and adaptive self-learning capability, neural networks can be readily implemented by hardware, and have thus been applied widely in many fields [1–6]. The zeroing neural network (ZNN) as well as its variant (i.e. zeroing dynamic), as a systematic approach to the online solution of time-varying problems with scalar situation included, has been applied to online matrix inversion [7], motion generation and control of redundant robot manipulators [3], and tracking control of nonlinear chaotic systems [8]. For example, a ZNN model with a nonlinear function activated is applied to the kinematic control of redundant robot manipulators via Jacobian matrix pseudoinversion in [9], which achieves high accuracy but cannot handle the bound constraints existing in the robots. In [10], present a finite-time convergent ZNN model is presented for solving dynamic quadratic programs with application to robot tracking, which requires convex activation functions and cannot remedy the issue of joint-limit avoidance. Such a ZNN method is further discretized to compute the solution to time-varying nonlinear equations based on a new three-step formula, which can be implemented on a digital computer directly. In addition, for the applications, ZNN is exploited in [3] to remedy the joint-angle drift phenomenon of redundant robot manipulators by minimizing the difference between the desired joint position and the actual one.

It is worth pointing out here that although these existing models differ in choosing different error functions or using different activation functions, all of them follow similar design procedures: the ZNN method usually formulates a time-varying problem into a regulation problem in control. Specifically, the residual error of a ZNN model for the task function to be solved is to be regulated to zero. Then, a monotonically increasing and odd function activated ZNN model with its equilibrium identical to the solution of this time-varying problem is devised to solve the latter recursively. In addition, the design parameter in the ZNN method should be larger than zero. To the best of the authors' knowledge, all existing results on ZNN assume that the set for projection of the activation function is a convex one, which evidently excludes the nonconvex set from consideration. General conclusions relaxing the convex constraint on activation functions remain unexplored.

Kinematic Control of Redundant Robot Arms Using Neural Networks, First Edition.
Shuai Li, Long Jin and Mohammed Aquil Mirza.

In this chapter, we make progress in this direction by proposing new results on ZNN to remedy these weaknesses. The presented ZNN models in this chapter are able to deal with nonconvex projection set Ω in the activation functions, while the existing solutions require the projection set to be convex. Additionally, this is the first work on ZNN for solving a time-varying optimization problem with inequality and bound constraints, which opens a door to the research on solving time-varying constrained optimization problems in an error-free manner. In short, there are two limitations in the existing research on ZNN, i.e. lacking the technique for handling inequality and bound constraints when solving dynamic optimization problems and requiring the activation function to be odd and monotonically increasing. This chapter overcomes these limitations by proposing ZNN models, allowing nonconvex sets for projection operations in activation functions and incorporating new techniques for handling inequality constraint.

1.2 Scheme Formulation and ZNN Solutions

In this section, a ZNN model for dynamic quadratic programming subject to equality and inequality constraints is presented. Then, new results are derived using the ZNN model with the aid of a nonconvex activation function.

1.2.1 ZNN Model

Consider the convex dynamic quadratic programming subject to equality and inequality constraints in the form of

$$\begin{aligned} \text{minimize} \quad & x^{\mathrm{T}}(t)P(t)x(t)/2 + q^{\mathrm{T}}(t)x(t), \\ \text{subject to} \quad & A(t)x(t) = b(t), \\ & C(t)x(t) \leq d(t), \end{aligned} \tag{1.1}$$

where superscript $^{\mathrm{T}}$ denotes the transpose operation over a vector or a matrix; smoothly time-varying matrix $P(t) \in \mathbb{R}^{n\times n}$ is positive-definite; $q(t) \in \mathbb{R}^n$, $A(t) \in \mathbb{R}^{m\times n}$ being of full-row-rank, $C(t) \in \mathbb{R}^{p\times n}$ and $b(t) \in \mathbb{R}^m$, $d(t) \in \mathbb{R}^p$ are all smoothly time-varying.

By adding a time-varying nonnegative term to the inequality constraint, the convex dynamic quadratic programming problem (1.1) is converted into

$$\begin{aligned} \text{minimize} \quad & x^{\mathrm{T}}(t)P(t)x(t)/2 + q^{\mathrm{T}}(t)x(t), \\ \text{subject to} \quad & A(t)x(t) - b(t) = 0, \\ & C(t)x(t) - d(t) + \overline{\sigma^2}(t) = 0, \end{aligned} \tag{1.2}$$

where $\overline{\sigma^2}(t) \in \mathbb{R}^p$ is defined as $\overline{\sigma^2}(t) = \sigma(t)\circ\sigma(t) = [\sigma_i^2(t)]$. Define a Lagrange function as follows:

$$\begin{aligned} L(x(t), \rho_1(t), \rho_2(t), \sigma(t), t) = {} & x^{\mathrm{T}}(t)P(t)x(t)/2 \\ & + q^{\mathrm{T}}(t)x(t) + \rho_1^{\mathrm{T}}(t)(A(t)x(t) - b(t)) \\ & + \rho_2^{\mathrm{T}}(t)(C(t)x(t) - d(t) + \overline{\sigma^2}(t)). \end{aligned} \tag{1.3}$$

By using the related Karush–Kuhn–Tucker condition [11], we have

$$
\begin{aligned}
&P(t)x(t) + q(t) + A^{\mathrm{T}}(t)\rho_1(t) + C^{\mathrm{T}}(t)\rho_2(t) = 0,\\
&A(t)x(t) - b(t) = 0,\\
&C(t)x(t) - d(t) + \overline{\sigma^2}(t) = 0,\\
&\rho_2(t)\circ\sigma(t) = 0.
\end{aligned}
\tag{1.4}
$$

Letting $y(t) = [x(t), \rho_1(t), \rho_2(t), \sigma(t)]^{\mathrm{T}}$, the above equation can be rewritten as

$$
f(y(t), t) = 0 \in \mathbb{R}^{m+n+2p}, \tag{1.5}
$$

where mapping function $f(\cdot)$ is used to denote the left-hand side of (1.4). By defining $\varepsilon(t) = -f(y(t), t)$, we adopt the following evolution for $\varepsilon(t)$, i.e. the ZNN design formula:

$$
\dot{\varepsilon}(t) = -\gamma\varepsilon(t). \tag{1.6}
$$

Substituting (1.5) into (1.6) yields a dynamic equation:

$$
J(y(t), t)\dot{y}(t) = -\gamma(f(y(t), t)) - \frac{\partial f(y(t), t)}{\partial t}, \tag{1.7}
$$

where

$$
J(y(t), t) = \frac{\partial f(y(t), t)}{\partial y(t)} = \begin{bmatrix} P(t) & A^{\mathrm{T}}(t) & C^{\mathrm{T}}(t) & 0 \\ A(t) & 0 & 0 & 0 \\ C(t) & 0 & 0 & \overline{\sigma}(t) \\ 0 & 0 & 2\overline{\sigma}(t) & \overline{\rho}_2(t) \end{bmatrix} \in \mathbb{R}^{(m+n+2p)\times(m+n+2p)},
$$

with

$$
\overline{\sigma}(t) = \begin{bmatrix} \sigma_1(t) & 0 & \cdots & 0 \\ 0 & \sigma_2(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma_p(t) \end{bmatrix}, \quad \overline{\rho}_2(t) = \begin{bmatrix} \rho_1(t) & 0 & \cdots & 0 \\ 0 & \rho_2(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \rho_p(t) \end{bmatrix}.
$$

For the situation of Jacobian matrix $J(x(t), t)$ being nonsingular, the above equation is further rewritten as

$$
\dot{y}(t) = -\gamma J^{-1}(y(t), t)\left(f(y(t), t) - \frac{\partial f(y(t), t)}{\partial t}\right), \tag{1.8}
$$

where $y(t)$, starting from a given initial condition, denotes the neural state as well as the output corresponding to theoretical solution $y^*(t)$, with its first n elements constituting the optimal solution $x^*(t)$ to (1.1).

Remark 1.1 As stated in [12], the only systematic approach presented so far for the solution of time-varying problems is the ZNN-based technique. Under suitable assumptions, it is shown that the generated solution for solving a time-varying problem synthesized by the ZNN-based technique converges to the exact theoretical time-varying solution globally and exponentially [12]. A gradient-based (or gradient-related) technique is widely used for the online solving of various problems, which is taken as an example to compare with the ZNN-based technique in this remark. The differences between these two techniques are listed in Table 1.1.

Table 1.1 Comparison of ZNN-based and gradient-based techniques for solving $f(y(t), t) = 0$.

	Error function	**Design formula**	**Dynamic equation**
ZNN-based technique	$\varepsilon(t) = -f(y(t), t)$	$\dot{\varepsilon}(t) = -\gamma \varepsilon(t)$	$\dot{y}(t) = -\gamma J^{-1}(y(t), t)(f(y(t), t) - \frac{\partial f(y(t),t)}{\partial t})$
Gradient-based technique	$e(t) = \Vert f(y(t), t) \Vert_2^2 /2$	$\dot{y}(t) = -\gamma \frac{\partial e(t)}{\partial y}$	$\dot{y}(t) = -\gamma J^{\mathrm{T}}(y(t), t) f(y(t), t)$

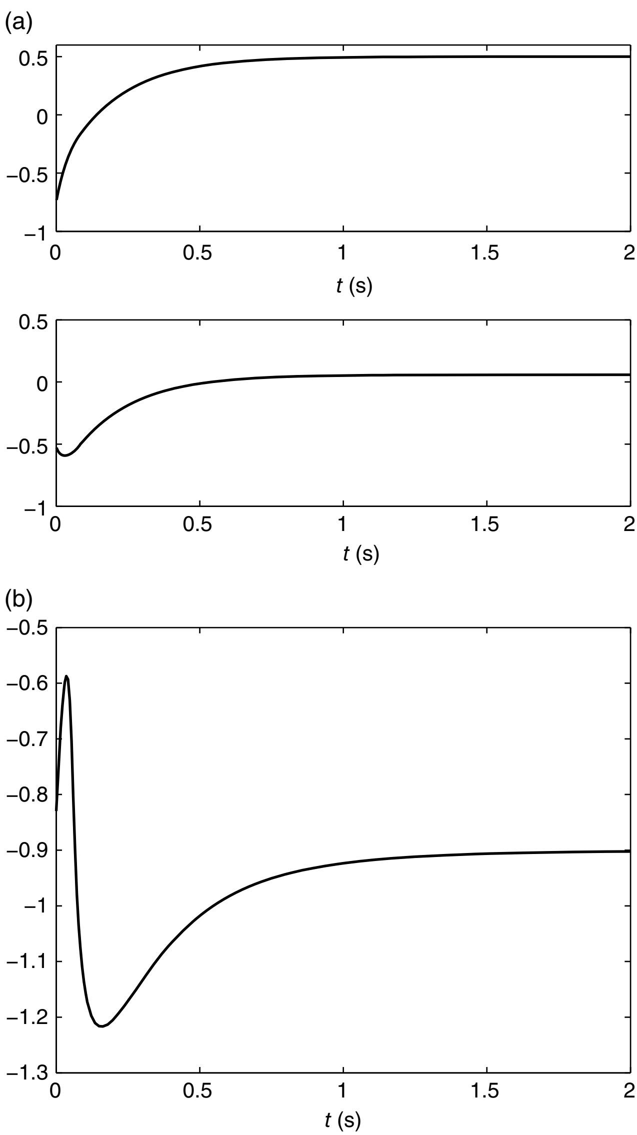

Figure 1.1 State vector $x(t)$, $\rho_1(t)$ of ZNN model (1.8) for solving (1.13) at $t = 1$. (a) Profiles of $x(t)$ and (b) profile of $\rho_1(t)$.

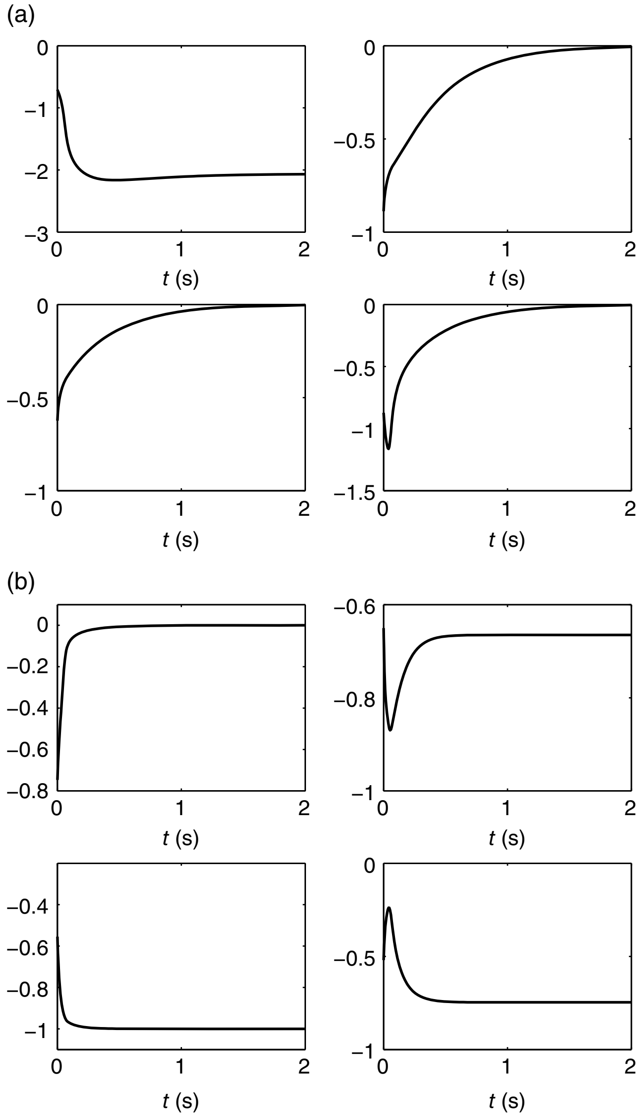

Figure 1.2 State vector $\rho_2(t)$, $\sigma(t)$ of ZNN model (1.8) for solving (1.13) at $t = 1$. (a) Profiles of $\rho_2(t)$ and (b) profile of $\sigma(t)$.

In addition, it is revealed in [13] that a controller designed by the ZNN-based technique is stable naturally as long as design parameter $\gamma > 0$, while a controller designed by other techniques cannot have a guaranteed stability. This can be deemed as another advantage of the ZNN-based technique compared with other existing techniques.

A disadvantage of the ZNN-based technique compared with other existing techniques is that, as shown in Table 1.1, the matrix inversion operation required in the model may lead to the failure of the solving task when encountering a singularity.

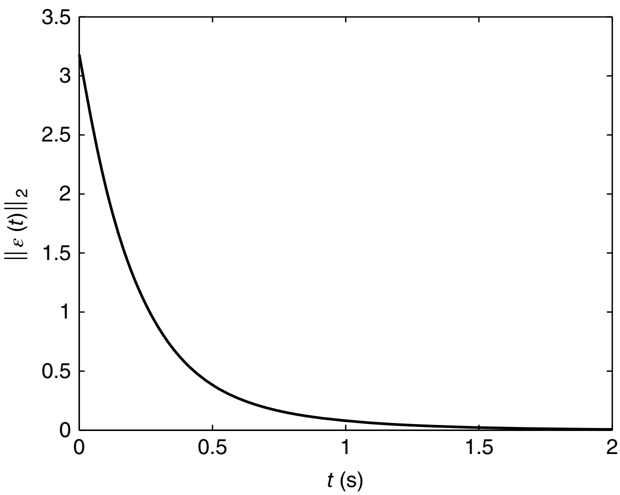

Figure 1.3 Residual error of ZNN model (1.8) for solving (1.13) at $t = 1$.

Remark 1.2 It is revealed in [1, 9] that any monotonically increasing odd function can be used as the activation function $\phi(\cdot)$ to accelerate the convergence speed of ZNN models. Then, the ZNN design formula with activation function $\phi(\cdot)$ can be rewritten as

$$\dot{\varepsilon}(t) = -\gamma\Phi(\varepsilon(t)), \tag{1.9}$$

where $\Phi(\cdot)$ denotes a vector with each element being $\phi(\cdot)$. Accordingly, the ZNN model (1.8) with activation function can be rewritten as

$$\dot{y}(t) = -\gamma J^{-1}(y(t), t)\left(\Phi(f(y(t), t)) - \frac{\partial f(y(t), t)}{\partial t}\right). \tag{1.10}$$

Note that, by exploiting the linear activation function $\Phi(\varepsilon(t)) = \varepsilon(t)$, (1.10) reduces to (1.8). That is to say, Equation (1.8) can be deemed as a linear function activated ZNN model.

1.2.2 Nonconvex Function Activated ZNN Model

As reviewed in Section 1.1, different ZNN models as well as their variants have been extensively studied and exploited for solving dynamic problems over the past 15 years. Although these existing models differ in choosing different error functions or using different activation functions, all of them follow similar design procedures and share the same convergence condition. For example, the ZNN model is often designed as a dynamic system with its equilibrium identical to the solution of the problem to be solved and then solves the latter recursively. In addition, the design parameter γ should be larger than zero and the activation function used to accelerate the convergence speed should be monotonically increasing and odd. To the best of the authors' knowledge, all existing results on ZNN assume that the set for projection of activation function is a convex one, which evidently excludes a nonconvex set from consideration. General conclusions relaxing the convex constraint on activation functions remain unexplored.

Let $\Upsilon_\Omega(U)$ be the projection from a set U to a set Ω such that $\Upsilon_\Omega(U) = \arg\min_{Y\in\Omega} \| Y - U\|_2$ with $0 \in \Omega$, we show the new design formula as follows:

$$\dot{\varepsilon}(t) = -\Upsilon_\Omega(\varepsilon(t)). \tag{1.11}$$

Expanding (1.11) leads to a new nonconvex function activated ZNN model:

$$\dot{y}(t) = -J^{-1}(y(t), t)\left(\Upsilon_\Omega(\gamma f(y(t), t)) - \frac{\partial f(y(t), t)}{\partial t}\right). \tag{1.12}$$

It can be concluded from the definition of $\Upsilon_\Omega(\cdot)$ that $\Upsilon_\Omega(\cdot)$ incorporates the existing ZNN activation functions as special cases. That is to say, any monotonically increasing odd activation function $\phi(\cdot)$ can be deemed as a subcase of $\Upsilon_\Omega(\cdot)$. In addition, it also can be generalized that, different from the existing results, the following special set can be used as the activation function of ZNN.

$\Omega = \{U \in \mathbb{R}^{(n+m+2p)}, -c_1 \le U_i \le c_1 \quad \text{or} \quad U_i = \pm c_2\}$, where c_1 and c_2 are two constants and $c_1 < c_2$.

1.3 Theoretical Analyses

In this section, we conduct analyses on the convergence of the presented ZNN model (1.8) and (1.12) via the following theorems.

Theorem 1.1 The presented ZNN model (1.8) is stable and is exponentially and globally convergent to an equilibrium point $y^*(t)$, of which the first n elements constitute the optimal solution $x^*(t)$ to (1.1).

Proof: In terms of (1.1), ZNN model (1.8) is an equivalent expansion of $\dot{\varepsilon}(t) = -\gamma\varepsilon(t)$. By selecting a Lyapunov function candidate $v(t) = \varepsilon^2(t)$ and by using the Lyapunov theory, one can readily derive that ZNN model (1.8) is stable with $\dot{v}(t) = -2\gamma\varepsilon(t)$.

In addition, solving the dynamic system $\dot{\varepsilon}(t) = -\gamma\varepsilon(t)$ directly, we have $\varepsilon(t) = \varepsilon(0)\exp(-\gamma t)$ with $\varepsilon(0)$ denoting the initial value of $\varepsilon(t)$. Therefore, we come to the conclusion that, starting from any initial condition $\varepsilon(0)$, the residual error $\varepsilon(t)$ of ZNN model (1.8) for solving (1.1) globally and exponentially converges to zero. That is, the state vector $y(t)$ globally and exponentially converges to an equilibrium point $y^*(t)$, of which the first n elements constitute the optimal solution to (1.1). The proof is thus completed. ■

Theorem 1.2 ZNN model (1.12) globally converges to the theoretical solution of convex dynamic quadratic programming problem (1.1) subject to equality and inequality constraints.

Proof: A Lyapunov function candidate can be defined for (1.11) as

$$V(t) = \varepsilon^{\mathrm{T}}(t)\varepsilon(t)/2,$$

which is positive definite in view of the facts that $V(t) > 0$ for any $\varepsilon(t) \neq 0$, and that $V(t) = 0$ only for $\varepsilon(t) = 0$. In addition, the time derivative of $V(t)$ can be derived as

$$\dot{V}(t) = -\varepsilon^{\mathrm{T}}(t)\Upsilon_\Omega(\varepsilon(t)).$$

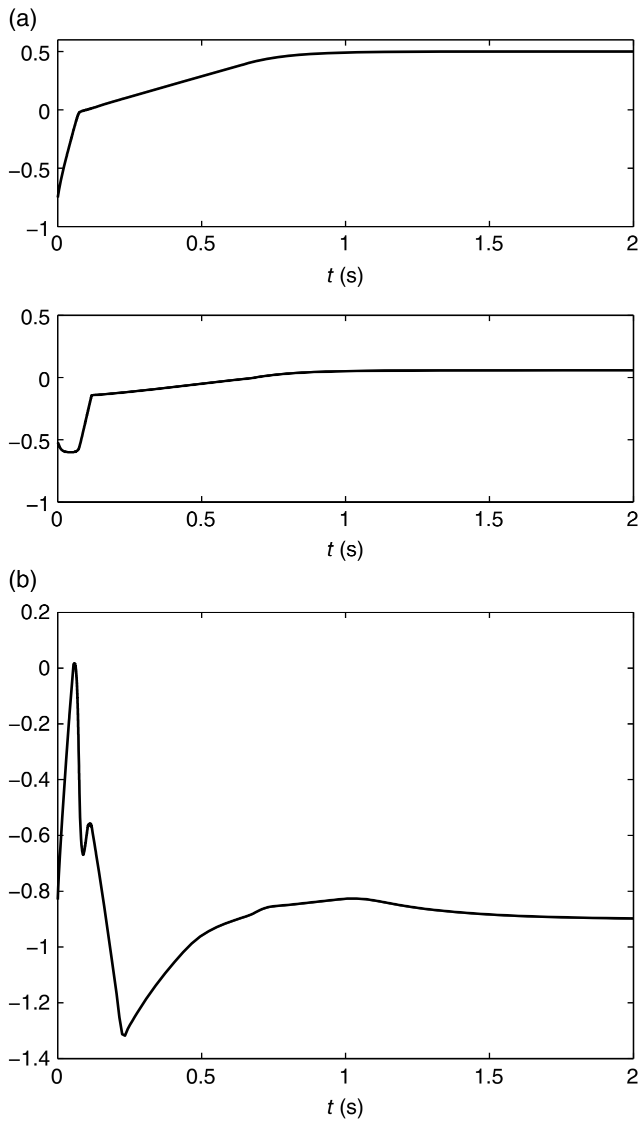

Figure 1.4 State vector $x(t)$, $\rho_1(t)$ of ZNN model (1.8) for solving (1.13) at $t = 1$. (a) Profiles of $x(t)$ and (b) profile of $\rho_1(t)$.

According to the definition of $\Upsilon_\Omega(\varepsilon(t))$, we have

$$\| \Upsilon_\Omega(\varepsilon(t)) - \varepsilon(t) \|_2^2 \leq \| Y - \varepsilon(t) \|_2^2, \quad \forall Y \in \Omega.$$

Choosing $Y = 0$ leads to

$$\| \Upsilon_\Omega(\varepsilon(t)) - \varepsilon(t) \|_2^2 \leq \| \varepsilon(t) \|_2^2,$$

which is equivalent to

$$\Upsilon_\Omega^{\mathrm{T}}(\varepsilon(t))\Upsilon_\Omega(\varepsilon(t)) - 2\Upsilon_\Omega^{\mathrm{T}}(\varepsilon(t))\varepsilon(t) \leq 0.$$

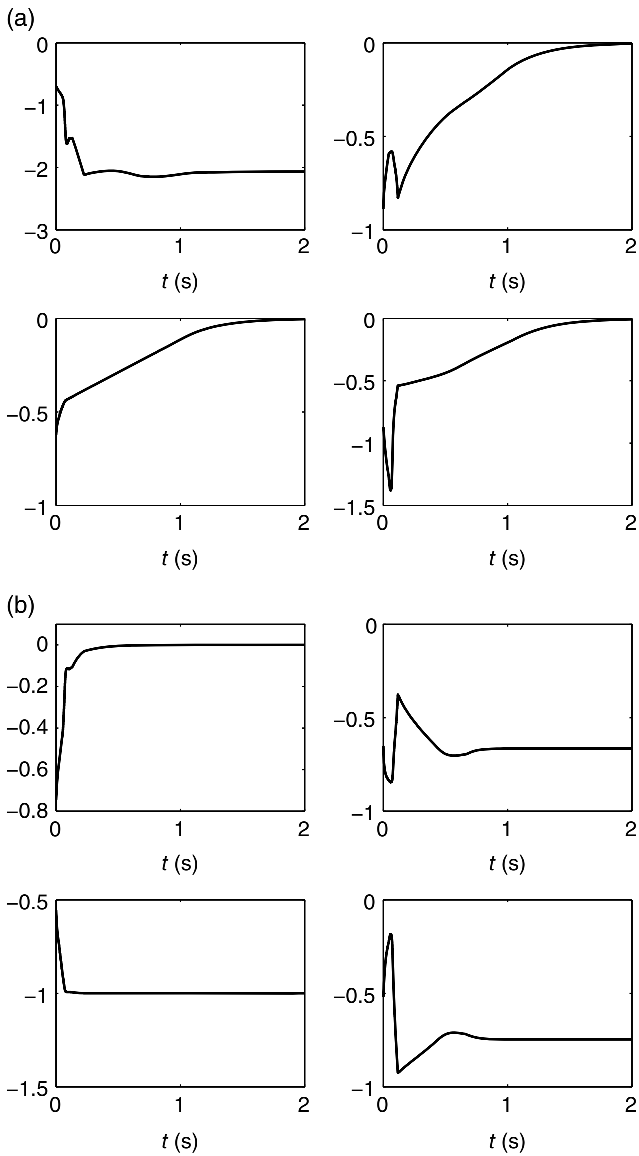

Figure 1.5 State vector $\rho_2(t)$, $\sigma(t)$ of ZNN model (1.8) for solving (1.13) at $t = 1$. (a) Profiles of $\rho_2(t)$ and (b) profiles of $\sigma(t)$.

Reformulating the above inequality produces

$$2\Upsilon_\Omega^{\mathrm{T}}(\varepsilon(t))\varepsilon(t) \geq \Upsilon_\Omega^{\mathrm{T}}(\varepsilon(t))\Upsilon_\Omega(\varepsilon(t)) \geq 0.$$

Therefore, we have

$$\dot{V}(t) \leq -\Upsilon_\Omega^{\mathrm{T}}(\varepsilon(t))\Upsilon_\Omega(\varepsilon(t))/2 \leq 0.$$

It can be readily summarized that $\varepsilon(t)$ globally converges to zero. That is, ZNN model (1.12) globally converges to the theoretical solution of convex dynamic quadratic

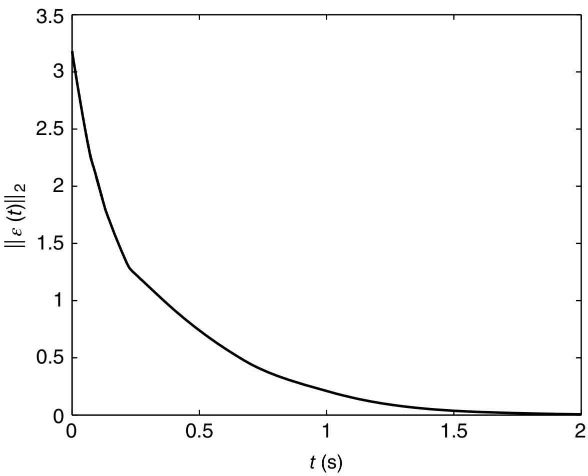

Figure 1.6 Residual error of ZNN model (1.8) for solving (1.13) at $t = 1$.

programming problem (1.1) subject to equality and inequality constraints. The proof is thus complete. ■

Remark 1.3 Theorem 1.2 extends and generalizes previous results on the activation function of ZNN models in the following three ways. (1) The activation function can be nonsymmetric and nonmonotonic. (2) The activation function can be nondifferentiable. (3) The activation function could have saturation, which is more practical.

1.4 Computer Simulations and Verifications

In this section, the following dynamic quadratic programming problem subject to equality and inequality constraints is considered for illustration and for comparison, which is modified from the problem presented in [1]:

$$\begin{aligned}\text{minimize}\quad & ((\sin t)/4 + 1)x_1^2(t) + ((\cos t)/4 + 1)x_2^2(t)\\ & +\ (\cos t)x_1(t)x_2(t) + (\sin 3t)x_1(t) + (\cos 3t)x_2(t),\\ \text{subject to}\quad & (\sin 4t)x_1(t) + (\cos 4t)x_2(t) = \cos 2t.\end{aligned} \tag{1.13}$$

In order to investigate the performance of the presented ZNN models, we consider two examples in the following subsections with different bound constraints.

1.4.1 ZNN for Solving (1.13) at $t = 1$

At $t = 1$ and with the bound constraint incorporated, (1.13) can be further rewritten as

$$\begin{aligned}\text{minimize}\quad & ((\sin 1)/4 + 1)x_1^2(t) + ((\cos 1)/4 + 1)x_2^2(t)\\ & + (\cos 1)x_1(t)x_2(t) + (\sin 3)x_1(t) + (\cos 3)x_2(t),\\ \text{subject to}\quad & (\sin 4)x_1(t) + (\cos 4)x_2(t) = \cos 2,\\ & -0.5 \le x_1 \le 0.5,\\ & -0.5 \le x_2 \le 0.5.\end{aligned} \tag{1.14}$$

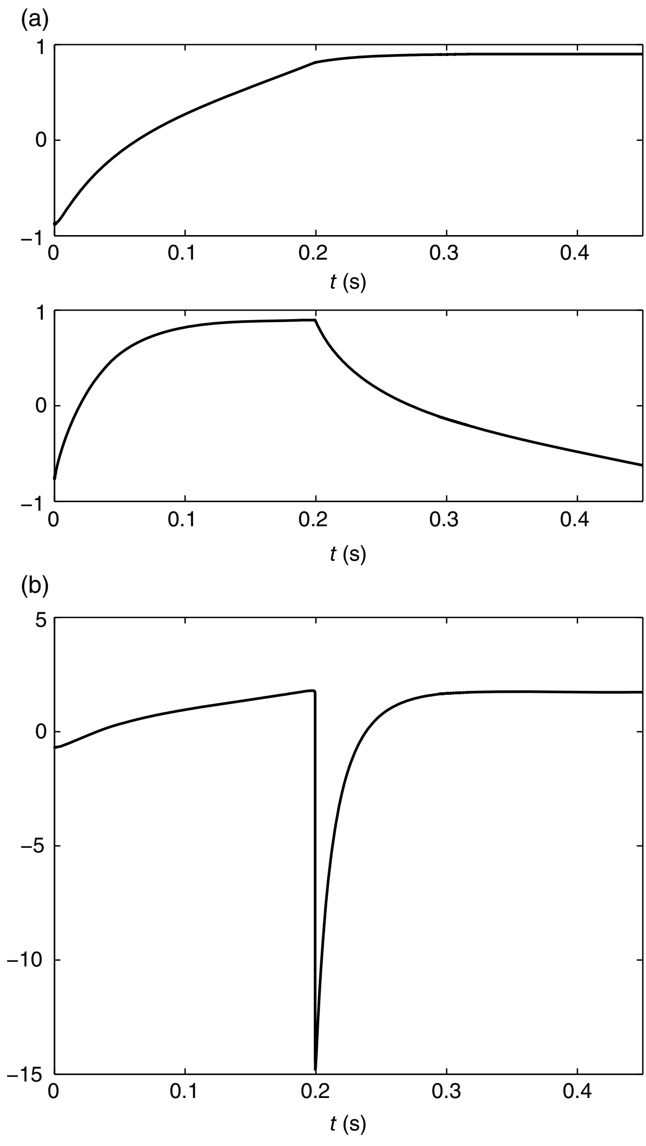

Figure 1.7 State vector $x(t)$, $\rho_1(t)$ of ZNN model (1.12) for solving (1.16). (a) Profiles of $x(t)$ and (b) profile of $\rho_1(t)$.

Starting with a randomly generated initial state, the corresponding computer simulation results are shown in Figures 1.1–1.3. Specifically, the element trajectories of the state $x(t)$, $\rho_1(t)$, $\rho_2(t)$ and $\sigma(t)$ are shown in Figures 1.1 and 1.2, from which we could observe that the solution of ZNN model (1.8) satisfies the given bound constraint. In addition, the corresponding residual error shown in Figure 1.3 further illustrates the effectiveness of the presented ZNN model (1.8).

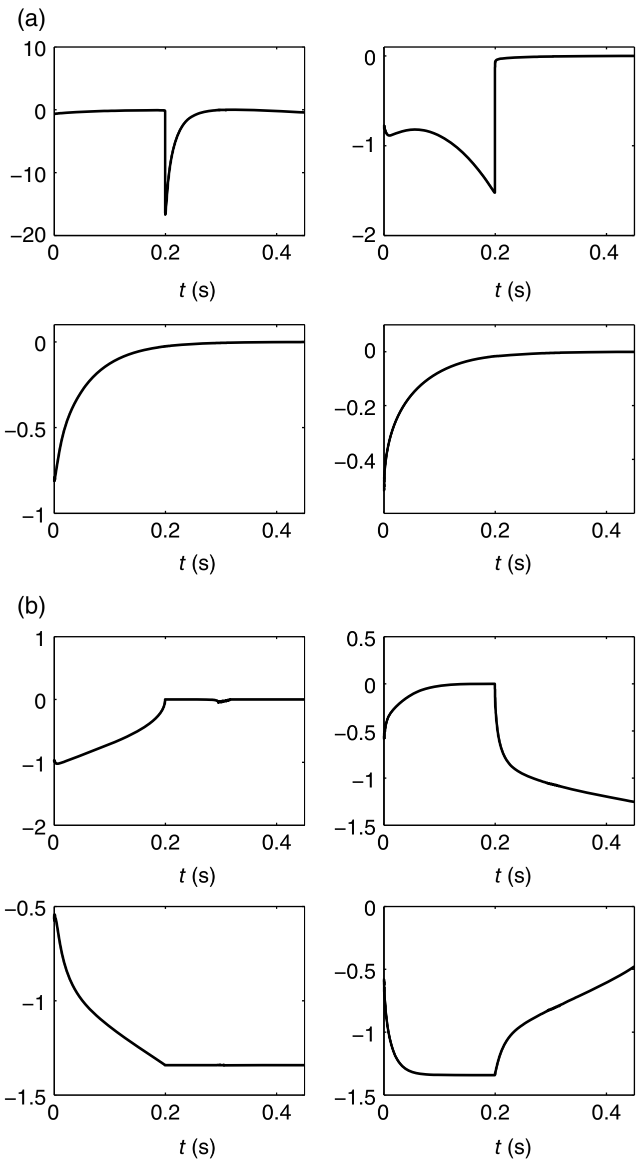

Figure 1.8 State vector $\rho_2(t), \sigma(t)$ of ZNN model (1.12) for solving (1.16). (a) Profiles of $\rho_2(t)$ and (b) profiles of $\sigma(t)$.

It is revealed in Theorem 1.2 that the projection operation for activation functions could be nonconvex. In this section, to exemplify the choice of Ω, we particularly consider the following set:

$$\Omega = \{\chi = [\chi_i] \in \mathbb{R}^{m+n+2p}, -c_2 \leq \chi_i \leq c_2, \text{or} \quad \chi_i = \pm c_1\}, \tag{1.15}$$

with $c_1 = 1$ and $c_2 = 0.1$ in the simulation. The choice of Ω is nonconvex due to the fact that $0 \in \Omega$ and $1 \in \Omega$ but $(0+1)/2 \notin \Omega$. Physically, Ω defined in (1.15) is generalized

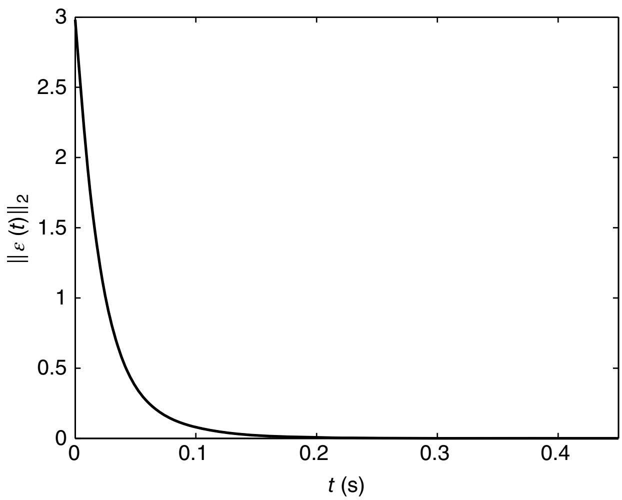

Figure 1.9 Residual error of ZNN model (1.12) for solving (1.16).

from commonly used strategies in industrial bang-bang control, where only maximum input action c_1, minus maximum input action $-c_1$, and zero input action 0 are applicable. To avoid chattering phenomena in conventional bang-bang control, it is preferable to expand zero input action into a small range $[-c_2, c_2]$, which results in the definition of Ω in (1.15). As shown in Figures 1.4–1.6, the state vector $x(t)$ is kept within its bound and the residual error converges to zero as time evolves. The convergence of the residual error in Figure 1.6 validates the effectiveness of Theorem 1.2 for the nonconvex constraint on activation function.

1.4.2 ZNN for Solving (1.13) with Different Bounds

With a new bound constraint incorporated, (1.13) can be further rewritten as

$$\begin{aligned}
\text{minimize}\quad & ((\sin t)/4+1)x_1^2(t)+((\cos t)/4+1)x_2^2(t)\\
& +(\cos t)x_1(t)x_2(t)+(\sin 3t)x_1(t)+(\cos 3t)x_2(t),\\
\text{subject to}\quad & (\sin 4t)x_1(t)+(\cos 4t)x_2(t)=\cos 2t,\\
& -0.9\le x_1\le 0.9,\\
& -0.9\le x_2\le 0.9.
\end{aligned}\tag{1.16}$$

Starting with a randomly generated initial state, the corresponding computer simulation results are shown in Figures 1.7–1.9. Specifically, the element trajectories of the state $x(t)$ and $\rho_1(t)$, and $\rho_2(t)$ and $\sigma(t)$ are shown in Figures 1.7 and 1.8, respectively, from which we could observe that the solution of ZNN model (1.8) satisfies the given bound constraint. In addition, all the states vary as time evolves. Besides, the corresponding residual error shown in Figure 1.9 further illustrates the effectiveness of the presented ZNN model (1.8).

Remark 1.4 A ZNN model is investigated in [14] for solving a quadratic programming problem with application to the repetitive motion generation of redundant robot manipulators. However, as stated in Section 1.1, the existing ZNN technique cannot solve an optimization problem with inequality constraint and thus the authors of [14] do not consider the joint physical constraints in their scheme. With the technique for handling inequality constraint presented in this chapter, the motion planning and control of redundant robot manipulators formulated as a constrained QP problem can be solved by ZNN readily and accurately, which can be deemed as one of the advantages of this chapter compared with existing results.

1.5 Summary

This chapter has pointed out two limitations in the existing ZNN results and then overcome these limitations by proposing ZNN models, allowing nonconvex sets for projection operations in activation functions and incorporating new techniques for handling the inequality constraint arising in optimizations. Theoretical analyses have been presented and shown that the presented ZNN models are of global stability with timely convergence. Finally, illustrative simulation examples have been provided and analyzed to substantiate the efficacy and superiority of the presented ZNN models for real-time dynamic quadratic programming subject to equality and inequality constraints. This chapter can be deemed as a rudiment of further investigations on constrained dynamic optimization with time-varying parameters, which can be generalized and employed for the motion planning and control of redundant robot manipulators [15] and distributed winner-take-all-based task allocation of multiple robots [16].