Low-Computational-Complexity Zeroing Neural Network Model for Solving Systems of Dynamic Nonlinear Equations

Kangze Zheng, Shuai Li, *Senior Member, IEEE*, and Yunong Zhang, *Member, IEEE*

***Abstract*— Nonlinear equation systems are ubiquitous in a variety of fields, and how to tackle them has drawn much attention, especially dynamic ones. As a particular class of recurrent neural network, zeroing neural network (ZNN) takes time-derivative information into consideration, and thus, is a competent approach to dealing with dynamic problems. Hitherto, two kinds of ZNN models have been developed for solving systems of dynamic nonlinear equations. One of them is explicit, involving the computation of a pseudoinverse matrix, and the other is of implicit dynamics essentially. To address these two issues at once, a low-computational-complexity ZNN (LCCZNN) model is proposed. It does not need to compute any pseudoinverse matrix, and is in the form of explicit dynamics. Additionally, a novel activation function is presented to endow the LCCZNN model with finite-time convergence and certain robustness, which is proved rigorously by Lyapunov theory. Numerical experiments are conducted to validate the results of theoretical analyses, including the competence and robustness of the LCCZNN model. Finally, a pseudoinverse-free controller derived from the LCCZNN model is designed for a UR5 manipulator to online accomplish a trajectory-following task.**

***Index Terms*— Low computational complexity, activation function, zeroing neural network, dynamic nonlinear equation systems, trajectory following.**

I. Introduction

NONLINEARITY is the inherent characteristic of most systems in nature. Nonlinear equation systems are encountered with relatively high frequency in distinct fields, such as manipulator control, multi-agent systems, signal processing, and other industrial applications [1]–[8]. For instance, in [1], the constrained quadratic programming problem with dynamic parameters is converted into a set of dynamic nonlinear equations, which is then solved for the optimal solution. In [8], the time-varying containment problem for a multi-agent system is transformed into a dynamic nonlinear equation system (DNES) describing the system error, and then a control strategy is exploited to make the system variable converge to the solution of the DNES. Therefore, it is of great significance to solve nonlinear equation systems. Since most systems in nature are dynamic rather than static, it is much more important to deal with DNESs.

Considerable efforts have been devoted to solutions of nonlinear equation systems. There is a series of numerical methods, including dichotomy, Newton-Raphson's method, secant method, Steffensen's method, and other variation methods [9]–[11]. According to [10], these numerical methods, or so-called root-finders, are divided into four categories, namely one-point iterative methods with or without memory, and multi-point iterative methods with or without memory. Generally, multi-point methods perform better than one-point methods, since they surmount one-point methods' theoretical shortcomings associated with convergence order and rate [10]. It is worth mentioning that the aforementioned numerical methods secure solutions by iteration, and they may not be efficient enough for online solutions on account of their serial-processing nature [12]. Additionally, these traditional methods have a common major drawback, that is, strong sensitiveness to the initial point [5].

Lately, recurrent neural network (RNN) has been developed rapidly, and utilized to address various problems online, due to its parallel-processing capability and potential of analog-circuit implementation [6], [13]–[18]. Gradient neural network (GNN) is one kind of the RNN based on the elimination of a scalar-valued nonnegative error. It updates a candidate solution recursively along the negative-gradient direction of the predefined objective function. For finding the root of a static nonlinear equation, the GNN is sensitive to the initial point and may yield an inaccurate or misleading solution when the root is multiple or does not exist [17]. Furthermore, the GNN tends to generate solutions with nonnegligible lagging errors, when dealing with DNESs [18]. The reason why there exist lagging errors is that the GNN does not make use of time-derivative information of DNESs. Aiming at eliminating lagging errors, Zhang *et al.* proposed zeroing neural network (ZNN) in [19], which is another particular class of the RNN taking time-derivative information into consideration. Basically, the ZNN surmounts the aforementioned weakness of the GNN, including the existence of lagging errors. Overall, the ZNN outperforms the GNN in dynamic problem solving [18], [20]. Afterwards, on the basis of the design scheme in [19], literatures [6], [12], [21], [22] construct different ZNN models

This work is aided by the National Natural Science Foundation of China (with number 61976230), the Project Supported by Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (with number 2018). *(Corresponding author: Yunong Zhang.)*

Kangze Zheng and Yunong Zhang are with School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China (e-mails: zhengkz@mail2.sysu.edu.cn; zhynong@mail.sysu.edu.cn).

Shuai Li is with the Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu, 90570, Finland, and also with VTT-Technology Research Center of Finland, Oulu, 90570, Finland (e-mail: shuaili@oulu.fi).

for solving DNESs. However, the ZNN models proposed in [6], [21], [22] either involve computing a pseudoinverse matrix or appear in implicit form. In other words, all these models need to compute a pseudoinverse, either directly or indirectly, and the computation of pseudoinverse matrices is time-consuming and inefficient during online solving processes. In this paper, we try to circumvent any pseudoinverse computation, and construct a low-computational-complexity ZNN (LCCZNN) model in the form of explicit dynamics for tackling DNESs.

Activation functions (AFs) play a powerful role in the performance of the ZNN, such as convergence and robustness. In general, an appropriate nonlinear AF outperforms a linear one in the aspect of the ZNN's convergence rate. However, not all nonlinear AFs are able to endow the ZNN with finite-time convergence [23]. Li *et al*. proposed a sign-bi-power AF with such a valuable property in [23]. From there on, a number of researches on designing proper AFs have been carried out, and the strengths of the ZNN activated by different AFs have been analyzed theoretically and verified by experiments in [22], [24]–[28]. In [22], [24]–[26], some AFs modified from the sign-bi-power AF are raised. Two more complicated AFs are developed in [27]. It is proved in [28] that nonlinear AFs enable the ZNN with certain robustness as well. On the basis of the sign-bi-power AF, a novel AF is developed to improve the LCCZNN model for better performance. Besides, comparisons among the new AF and AFs put forward in [22] are carried out, and numerical experiments are conducted to verify the relevant analysis results. At the end of this section, the major contributions of the paper are summarized as follows.

- The LCCZNN model without computing any pseudoinverse matrix is proposed for solving DNESs in high efficiency.
- The novel AF is put forward to enhance the convergence rate and robustness of the LCCZNN model. Furthermore, we prove that the performance of the proposed model in these two aspects depends on the design parameters and the AF.
- The newly proposed AF is compared with two AFs presented in [22], in terms of convergence time. Through both theoretical analyses and numerical experiments, its superiority is shown and verified.
- A pseudoinverse-free controller derived from the proposed LCCZNN model is developed for a UR5 manipulator to online finish a trajectory-following task successfully in the absence or presence of noise disturbance.

II. Preliminaries and Related Work

In this section, relevant definitions and necessary lemmas are first provided for better understanding of the paper. Then, the DNES problem is stated in mathematical form. Finally, we retrospect the related work based on the ZNN and discuss some flaws of the existing work.

A. Preliminaries

Consider the following autonomous system:

$$\dot{\boldsymbol{z}}(t)=\boldsymbol{g}(\boldsymbol{z}(t)),\quad t\in[0,+\infty),\tag{1}$$

where $\boldsymbol{g}\colon\mathbb{R}^n\mapsto\mathbb{R}^n$ is a locally Lipschitz function. Suppose that the origin $\boldsymbol{z}=\boldsymbol{0}$ is an equilibrium point of (1), that is

$$\boldsymbol{g}(\boldsymbol{0})=\boldsymbol{0}.$$

Here are definitions on the different stability of the equilibrium point.

Definition 1 [29]: The equilibrium point $\boldsymbol{z}=\boldsymbol{0}$ of the autonomous system (1) is stable if, for any $\epsilon>0$, there exists a $\sigma>0$ such that

$$\|\boldsymbol{z}(0)\|_2<\sigma\Rightarrow\|\boldsymbol{z}(t)\|_2<\epsilon,\quad\forall\ t\geq0.$$

Definition 2 [29]: The equilibrium point $\boldsymbol{z}=\boldsymbol{0}$ of the autonomous system (1) is globally asymptotically stable if it is stable and $\lim_{t\to+\infty}\boldsymbol{z}(t)=\boldsymbol{0}$ holds for any $\boldsymbol{z}(0)\in\mathbb{R}^n$.

Definition 3 [30]: The equilibrium point $\boldsymbol{z}=\boldsymbol{0}$ of the autonomous system (1) is globally finite-time-stable if it is globally asymptotically stable, and there exists a settling-time function $T_1\colon\mathbb{R}^n\backslash\{\boldsymbol{0}\}\mapsto(0,+\infty)$ such that

$$\boldsymbol{z}(t)=\boldsymbol{0},\quad\forall\ t\geq T_1(\boldsymbol{z}(0)).$$

Two lemmas are provided to lay foundation for the proofs of theorems presented in the paper.

Lemma 1 [29]: Consider the autonomous system (1) with the equilibrium point $\boldsymbol{z}=\boldsymbol{0}$. Let $V\colon\mathbb{R}^n\mapsto\mathbb{R}$ be a continuously differentiable function such that

1) $V(\boldsymbol{z})>0,\quad\forall\ \boldsymbol{z}\in\mathbb{R}^n\backslash\{\boldsymbol{0}\},$
2) $V(\boldsymbol{0})=0,$
3) $\dot{V}(\boldsymbol{z})=\dfrac{\mathrm{d}V(\boldsymbol{z})}{\mathrm{d}\boldsymbol{z}}\boldsymbol{g}(\boldsymbol{z})<0,\quad\forall\ \boldsymbol{z}\in\mathbb{R}^n\backslash\{\boldsymbol{0}\},$
4) $\dot{V}(\boldsymbol{0})=0,$
5) $\|\boldsymbol{z}\|_2\to+\infty\Rightarrow V(\boldsymbol{z})\to+\infty.$

Then, $\boldsymbol{z}=\boldsymbol{0}$ is globally asymptotically stable.

Lemma 2 [29]: Consider the following nonautonomous system:

$$\dot{\boldsymbol{z}}(t)=\boldsymbol{g}(t,\boldsymbol{z}(t)),\quad t\in[0,+\infty),\tag{2}$$

where $\boldsymbol{g}\colon[0,+\infty)\times\mathbb{R}^n\mapsto\mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in $\boldsymbol{z}$. Let $V\colon[0,+\infty)\times\mathbb{R}^n\mapsto\mathbb{R}$ be a continuously differentiable function such that

1) $\alpha\left(\|\boldsymbol{z}\|_2\right)\leq V(t,\boldsymbol{z})\leq\beta\left(\|\boldsymbol{z}\|_2\right),\quad\forall\ t\in[0,+\infty)$ and $\forall\ \boldsymbol{z}\in\mathbb{R}^n,$
2) $\dot{V}(t,\boldsymbol{z})=\dfrac{\partial V(t,\boldsymbol{z})}{\partial t}+\dfrac{\partial V(t,\boldsymbol{z})}{\partial\boldsymbol{z}}\boldsymbol{g}(t,\boldsymbol{z})\leq-W(\boldsymbol{z}),$ $\forall\ \|\boldsymbol{z}\|_2\geq\eta,$

where $\alpha(\cdot)$ and $\beta(\cdot)$ are class $\mathcal{K}_\infty$ functions, $W(\cdot)$ is a continuous positive-definite function, and η is a positive constant. Then, for any $\boldsymbol{z}(0)\in\mathbb{R}^n$, there exists a settling-time function $T_2\colon(0,+\infty)\times\mathbb{R}^n\mapsto[0,+\infty)$ such that the solution of (2) satisfies

$$\|\boldsymbol{z}\|_2\leq\alpha^{-1}(\beta(\eta)),\quad\forall\ t\geq T_2(\eta,\boldsymbol{z}(0)).$$

Lemma 3: Consider an equation described as

$$\exp(z)-z^a-1=0,\quad z\in(0,+\infty),\tag{3}$$

where $a \in (0,1)$ is a constant. Then, the above equation only has one root laying on $(0,1)$. Besides, the root decreases and approaches to 0 with the increase of a.

proof: Please see Appendix for the rigorous proof. ■

B. Problem Statement

Consider the following DNES:

$$\begin{cases} f_1(t, x_1(t), x_2(t), \cdots, x_n(t)) = 0, \\ f_2(t, x_1(t), x_2(t), \cdots, x_n(t)) = 0, \\ \qquad\qquad\qquad\vdots \\ f_m(t, x_1(t), x_2(t), \cdots, x_n(t)) = 0, \end{cases} \tag{4}$$

where symbols t, $x_i \in \mathbb{R}$ $(i \in \{1, 2, \cdots, n\})$, and $f_j\colon [0, +\infty) \times \mathbb{R}^n \mapsto \mathbb{R}$ $(j \in \{1, 2, \cdots, m\})$ respectively denote time, the ith dynamic state variable, and the jth smooth nonlinear function, with $n \geq m$. To acquire a more compact form of the above problem, a vector $\boldsymbol{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^{\mathrm{T}}$ and a function $\boldsymbol{f}(t, \boldsymbol{x}(t)) = [f_1(t, \boldsymbol{x}(t)), f_2(t, \boldsymbol{x}(t)), \cdots, f_m(t, \boldsymbol{x}(t)]^{\mathrm{T}}$ are defined. Then, a concise modality of (4) is presented as

$$\boldsymbol{f}(t, \boldsymbol{x}(t)) = \mathbf{0} \in \mathbb{R}^m, \quad \boldsymbol{x}(t) \in \mathbb{R}^n, \quad n \geq m. \tag{5}$$

In consideration that each f_j is nonlinear, an assumption that system (5) has at least one solution $\boldsymbol{x}^*(t)$ is made in this paper. Our primary objective is to solve (5) with more efficiency (e.g., lower computational complexity and shorter convergence time) than previous works. To pursue the brevity (i.e., reduce the redundancy of equations), when some variables and functions are referred, the variables that they depend on may be omitted in the remainder of the paper, since the dependences have been already introduced or are negligible. For instance, function $\boldsymbol{f}(t, \boldsymbol{x}(t))$ and variable $\boldsymbol{x}(t)$ in (5) are abbreviated as $\boldsymbol{f}$ and $\boldsymbol{x}$, respectively.

C. Related Work on ZNN

The ZNN has stronger convergence capability because of its parallel-processing feature, and thus, it is proper for large-scale dynamic problem solving in real time [31]–[33].

With the development of the ZNN, there are mainly two ZNN models proposed in [12], [21], [22]. In literatures [21], a ZNN model is presented as

$$\dot{\boldsymbol{x}} = -J^{\dagger}\left(\mu\boldsymbol{\phi}(\boldsymbol{f}) + \frac{\partial \boldsymbol{f}}{\partial t}\right). \tag{6}$$

Thereinto, $\mu > 0$ is a design parameter, $\partial\boldsymbol{f}/\partial t = [\partial f_1/\partial t, \partial f_2/\partial t, \cdots, \partial f_m/\partial t]^{\mathrm{T}}$ is the partial derivative of $\boldsymbol{f}$ with respect to t, $J^{\dagger} \in \mathbb{R}^{n\times m}$ is the pseudoinverse of Jacobian matrix J with the form as

$$J = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 & \cdots & \partial f_1/\partial x_n \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 & \cdots & \partial f_2/\partial x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial f_m/\partial x_1 & \partial f_m/\partial x_2 & \cdots & \partial f_m/\partial x_n \end{bmatrix},$$

and $\boldsymbol{\phi}(\boldsymbol{f}) = [\phi(f_1), \phi(f_2), \cdots, \phi(f_m)]^{\mathrm{T}}$ is a function array, where ϕ denotes a monotonically increasing and odd AF. It should be noted that, model (6) needs to compute the pseudoinverse $J^{\dagger}$, leading to high computational complexity.

In order to fix the above weakness, the other ZNN model is raised as [22]:

$$\dot{\boldsymbol{x}} = -\left(\mu\boldsymbol{\phi}(\boldsymbol{f}) + \frac{\partial \boldsymbol{f}}{\partial t}\right) + (I - J)\dot{\boldsymbol{x}}, \tag{7}$$

where I denotes an n-dimensional identity matrix. Note that model (7) works only when $m = n$. In addition, to endow model (7) with the ability of finite-time convergence, authors in [22] exploited the sign-bi-power AF and another modified one, namely

$$\phi(z) = \operatorname{sig}^r(z) \tag{8}$$

and

$$\phi(z) = \operatorname{sig}^r(z) + z, \tag{9}$$

where $\operatorname{sig}^r(z)$ is defined as

$$\operatorname{sig}^r(z) = \begin{cases} |z|^r, & \text{if } z > 0, \\ 0, & \text{if } z = 0, \\ -|z|^r, & \text{if } z < 0, \end{cases}$$

with a constant parameter $r \in (0,1)$. It seems that model (7) gets rid of the pseudoinverse $J^{\dagger}$ and lessens the computation burden. However, (7) is an implicit neural network model essentially, and it still requires the pseudoinverse when we implement it through programming.

For higher efficiency, we are motivated to devise an explicit neural network model with low computational complexity.

III. LCCZNN Model and Theoretical Analyses

For the sake of a pseudoinverse-free model, the LCCZNN model is constructed step by step in this section. Besides, the novel AF is presented to reinforce the convergence and robustness of the LCCZNN model. Comparisons among the presented AF in this paper and two AFs in [22] are also provided.

A. LCCZNN Model

By drawing lessons from the design scheme of the ZNN in [19], the model with low computational complexity is constructed as follows.

Every neural network needs an error function to monitor its performance and then optimize its inner structure. Therefore, the first step of the construction is determining an error function, that is

$$\nu(t) = \frac{1}{2}\|\boldsymbol{f}\|_2^2 = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}. \tag{10}$$

Second, the ZNN design formula is utilized to force ν to tend towards 0 with the passage of time. It is expressed as [19]:

$$\dot{\nu} = -\mu\phi(\nu), \tag{11}$$

where $\mu > 0$ and ϕ represent a design parameter and an AF, respectively. Since

$$\dot{\nu} = \frac{\partial \nu}{\partial \boldsymbol{x}}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} + \frac{\partial \nu}{\partial t} = \boldsymbol{f}^{\mathrm{T}}J\dot{\boldsymbol{x}} + \boldsymbol{f}^{\mathrm{T}}\frac{\partial \boldsymbol{f}}{\partial t},$$

it follows from (11) that

$$\boldsymbol{f}^{\mathrm{T}} J\dot{\boldsymbol{x}} + \boldsymbol{f}^{\mathrm{T}}\frac{\partial \boldsymbol{f}}{\partial t} = -\mu\phi\left(\frac{\|\boldsymbol{f}\|_2^2}{2}\right).$$

Finally, through a simple transformation, the LCCZNN model is obtained as

$$\dot{\boldsymbol{x}} = -\frac{J^{\mathrm{T}}\boldsymbol{f}}{\|J^{\mathrm{T}}\boldsymbol{f}\|_2^2}\left(\boldsymbol{f}^{\mathrm{T}}\frac{\partial \boldsymbol{f}}{\partial t} + \mu\phi\left(\frac{\|\boldsymbol{f}\|_2^2}{2}\right)\right). \tag{12}$$

Remark 1: Notice that the denominator of model (12) may become zero, resulting in a singularity problem. To address the issue, one approach is to incorporate a regularization term for avoiding the problem, and the corresponding regularized LCCZNN model is formulated as

$$\dot{\boldsymbol{x}} = -\frac{J^{\mathrm{T}}\boldsymbol{f}}{\|J^{\mathrm{T}}\boldsymbol{f}\|_2^2 + \lambda}\left(\boldsymbol{f}^{\mathrm{T}}\frac{\partial \boldsymbol{f}}{\partial t} + \mu\phi\left(\frac{\|\boldsymbol{f}\|_2^2}{2}\right)\right),$$

where $\lambda > 0$ denotes the regularization term. It is evident that the denominator is always greater than zero. Besides, it follows from the subsequent Theorem 4 that the perturbance caused by the regularization term λ can be suppressed by a sufficient large value of parameter μ. Nonetheless, in order to reduce the side effect brought by λ, it is more reasonable to assign a sufficiently small value to the regularization term. We notice that when λ approaches 0, the regularized model has similar performance to model (12) in the experiments. Therefore, we keep focus on the LCCZNN model (12) without the regularization term in the following analyses and experiments.

To enable the LCCZNN model (12) with the capability of finite-time convergence, the novel nonlinear AF is proposed as below:

$$\phi(z) = \operatorname{sig}^r(z)\exp(|z|). \tag{13}$$

In order to explain the computational complexity of the LCCZNN model (12) more clearly, the computing procedure of (12) is illustrated in Algorithm 1. In Algorithm 1, lines 4 and 8 are mainly about computing the square of the Euclidean norm of a vector; line 5 involves the multiplication of a matrix by a vector; line 6 is concerning the inner product of two vectors; line 7 is primarily about mapping a scalar; line 9 is the multiplication of a vector by a scalar; and line 11 comprises the integration of a vector. Evidently, the computational complexities of all these operations are at most $O(mn)$, and thus, each loop in Algorithm 1 takes $O(mn)$ time. That is to say, the computational complexity of model (12) is $O(mn)$.

```
Algorithm 1: LCCZNN model (12) for solving DNES
  Data: Jacobian matrix $J$; partial derivative $\partial\boldsymbol{f}/\partial t$; AF $\phi$;
        design parameter $\mu$; task time $\Gamma$.
  Result: To find $\boldsymbol{x}$ satisfying $\boldsymbol{f} = \boldsymbol{0}$ during time interval
          $[0, \Gamma]$.
1  $t \leftarrow 0$;
2  Assign a random vector to $\boldsymbol{x}$;
3  while $t \leq \Gamma$ do
4      $\nu \leftarrow \|\boldsymbol{f}(t,\boldsymbol{x})\|_2^2/2$;
5      $\boldsymbol{\omega} \leftarrow J^{\mathrm{T}}(t,\boldsymbol{x})\boldsymbol{f}(t,\boldsymbol{x})$;
6      $\kappa_1 \leftarrow \boldsymbol{f}^{\mathrm{T}}(t,\boldsymbol{x})\frac{\partial \boldsymbol{f}}{\partial t}(t,\boldsymbol{x})$;
7      $\kappa_2 \leftarrow \mu\phi(e)$;
8      $\kappa_3 \leftarrow \|\boldsymbol{\omega}\|_2^2$;
9      $\dot{\boldsymbol{x}} \leftarrow -((\kappa_1 + \kappa_2)/\kappa_3)\boldsymbol{\omega}$;
10     Update $t$;
11     Update $\boldsymbol{x}$ by $\dot{\boldsymbol{x}}$;
12 end
```

It is always meaningful to realize the theoretical knowledge physically, instead of staying in theoretical level. According to [34], it is practicable to implement the LCCZNN model (12) as an analog circuit by using analog devices, including resistors, operational amplifiers, and diodes. As an important step of the ultimate hardware implementation, the schematic diagram of model (12) is depicted in Fig. 1.

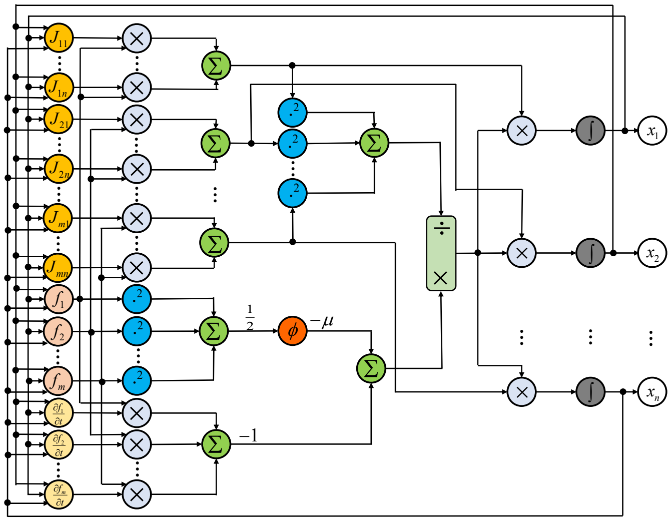


Fig. 1: Schematic diagram of LCCZNN model (12).

Remark 2: From the perspective of computational complexity, model (6) is found to has high computational complexity. Owing to the fact that it needs to compute the pseudoinverse of the Jacobian matrix, the computational complexity of model (6) is $O(mn^2)$. From the perspective of circuit complexity, the structure of model (7) becomes more complicated when the dimension of $\boldsymbol{x}$ gets larger, since (7) has term $(I - J)\dot{\boldsymbol{x}}$. In summary, the proposed LCCZNN model (12) is more prominent in terms of computational complexity and circuit implementation.

B. Convergence and Robustness Analyses

We analyze the LCCZNN model (12) aided by the AF (13), and summarize the research findings as the following theorems.

Theorem 1: Suppose that the DNES (5) has at least one solution. Starting from any initial state $\boldsymbol{x}(0)$, the LCCZNN model (12) activated by the AF (13) converges to one feasible solution $\boldsymbol{x}^*(t)$ of (5) within finite time

$$t_{\mathrm{AF3}} = \frac{1}{\mu}\gamma(1-r,\nu_{\mathrm{ini}}) \leq \frac{\nu_{\mathrm{ini}}^{1-r}\left[1 + (1-r)\exp(-\nu_{\mathrm{ini}})\right]}{\mu(1-r)(2-r)},$$

where $\nu_{\mathrm{ini}} = \nu(0) > 0$ denotes the initial error, and γ denotes the lower incomplete gamma function defined as [35]:

$$\gamma(a,z)=\int_0^z \tau^{a-1}\exp(-\tau)\ \mathrm{d}\tau,$$

where $a\in\mathbb{C}$ with real$(a)>0$ and $z\in\mathbb{R}$ with $z>0$.

Proof: Define a Lyapunov function candidate with respect to ν: $V_1(\nu)=\nu$. It follows from the definition of ν that $V_1\geq 0$, and the equality holds only when $\nu=0$. Then, the time derivative of V_1 along the trajectory of (11) is obtained as

$$\dot{V}_1(\nu)=\frac{\mathrm{d}V_1}{\mathrm{d}t}=-\mu\,\mathrm{sig}^r(\nu)\exp(|\nu|).$$

Evidently, $\dot{V}_1\leq 0$ and the equality holds only when $\nu=0$. Additionally, $V_1\to+\infty$ when $|\nu|\to+\infty$. Therefore, V_1 and $\dot{V}_1$ satisfy the conditions in Lemma 1. As a result, model (12) globally asymptotically converges to one solution $\boldsymbol{x}^*(t)$, which is closest to $\boldsymbol{x}(0)$ among all feasible solutions at time $t=0$.

Next, we prove that model (12) is able to converge to $\boldsymbol{x}^*(t)$ without infinite time. Because $\nu=0$ is globally asymptotically stable, given any initial error $\nu_{\text{ini}}>0$, ν will decrease to 0 and then remain unchanged. Suppose when $t=t_{\text{AF3}}$, ν diminishes to 0. On the basis of previous discussion, it is confirmed that

$$\nu=\dot{\nu}=0,\quad \forall\ t\geq t_{\text{AF3}}.$$

Due to the nonnegativity of ν, (11) is rewritten in differential form as

$$\mathrm{d}t=-\frac{1}{\mu}\nu^{-r}\exp(-\nu)\ \mathrm{d}\nu,\quad \forall\ t\in[0,t_{\text{AF3}}].$$

Then, by integrating the above equation from $t=0$ towards $t=t_{\text{AF3}}$, an integral equation is acquired as below:

$$\begin{aligned}\int_0^{t_{\text{AF3}}}\mathrm{d}t&=-\frac{1}{\mu}\int_{\nu_{\text{ini}}}^{\nu(t_{\text{AF3}})}\nu^{-r}\exp(-\nu)\ \mathrm{d}\nu\\&=\frac{1}{\mu}\int_0^{\nu_{\text{ini}}}\nu^{(1-r)-1}\exp(-\nu)\ \mathrm{d}\nu\\&=\frac{1}{\mu}\gamma(1-r,\nu_{\text{ini}}).\end{aligned}\tag{14}$$

In light of Theorem 4.1 in [36], the following inequality

$$\frac{a}{z^a}\gamma(a,z)\leq\frac{1}{a+1}\left(1+a\exp(-z)\right)$$

holds true. It follows from (14) that

$$t_{\text{AF3}}=\frac{1}{\mu}\gamma(1-r,\nu_{\text{ini}})\leq\frac{\nu_{\text{ini}}^{1-r}\left(1+(1-r)\exp(-\nu_{\text{ini}})\right)}{\mu(1-r)(2-r)}.\tag{15}$$

Thus, model (12) converges to one feasible solution of (5) within finite time t_{AF3}, and the proof is completed. ■

Theorem 2: Suppose that the DNES (5) has at least one solution. The time required for the LCCZNN model (12) activated by the AF (13) with any initial state to converge to one solution of (5), can be shortened by 1) increasing parameter μ; 2) decreasing parameter r.

Proof: Please see Appendix for the detailed proof. ■

Remark 3: As mentioned before, not all nonlinear AFs are capable to enable the ZNN to converge within finite time. Thus, it is of great significance to find an AF possessing such a merit. Besides, the convergence time of the ZNN varies by the adopted AF. It is necessary to compare the convergence performance of the ZNN aided by different AFs. By following the reasoning in [22], the convergence times of the LCCZNN model (12) aided by (8) and (9) are obtained as below:

$$t_{\text{AF1}}=\frac{\nu_{\text{ini}}^{1-r}}{\mu(1-r)}$$

and

$$t_{\text{AF2}}=\frac{\ln(1+\nu_{\text{ini}}^{1-r})}{\mu(1-r)}.$$

It follows from Theorem 1 that for any $\nu_{\text{ini}}>0$,

$$\frac{t_{\text{AF3}}}{t_{\text{AF1}}}=\frac{\gamma(1-r,\nu_{\text{ini}})}{\nu_{\text{ini}}^{1-r}/(1-r)}\leq\frac{1+(1-r)\exp(-\nu_{\text{ini}})}{2-r}<1.$$

Thereby, the AF (13) endows the LCCZNN model (12) with faster convergence speed than the AF (8). As for the magnitude relationship between t_{AF3} and t_{AF2}, Theorem 3 discusses it subsequently.

Theorem 3: Suppose that the DNES (5) has at least one solution. Suppose that t_{AF2} and t_{AF3} respectively correspond to the convergence times of the LCCZNN model (12) activated by the AFs (9) and (13) with any initial state. Then, there exists a sufficiently small $\bar{r}\in(0,1)$ such that

$$t_{\text{AF3}}<t_{\text{AF2}},\quad\forall\ r\in(0,\bar{r}),\quad\forall\ \nu_{\text{ini}}\in(0,+\infty).$$

Proof: In view of Theorem 1 and Remark 3, the difference between t_{AF3} and t_{AF2} is

$$t_{\text{AF3}}-t_{\text{AF2}}=\frac{1}{\mu}\int_0^{\nu_{\text{ini}}}\nu^{-r}\exp(-\nu)\mathrm{d}\nu-\frac{\ln(1+\nu_{\text{ini}}^{1-r})}{\mu(1-r)}.$$

The partial derivative of the difference with respect to ν_{ini} is

$$\frac{\partial(t_{\text{AF3}}-t_{\text{AF2}})}{\partial\nu_{\text{ini}}}=\frac{\nu_{\text{ini}}^{-r}}{\mu}\left(\exp(-\nu_{\text{ini}})-\frac{1}{1+\nu_{\text{ini}}^{1-r}}\right).$$

According to Lemma 3, equation $\exp(-\nu_{\text{ini}})=1/(1+\nu_{\text{ini}}^{1-r})$ only has one root $\zeta\in(0,1)$, and when r tends to 0, ζ tends to 0. Besides, $\partial(t_{\text{AF3}}-t_{\text{AF2}})/\partial\nu_{\text{ini}}$ is positive on $(0,\zeta)$ and negative on $(\zeta,+\infty)$. As a consequence, the following inequality is obtained:

$$\begin{aligned}t_{\text{AF3}}-t_{\text{AF2}}\leq&\frac{1}{\mu}\int_0^{\zeta}\nu^{-r}\exp(-\nu)\mathrm{d}\nu-\frac{\ln\left(1+\zeta^{1-r}\right)}{\mu(1-r)}\\\leq&\frac{1}{\mu(1-r)}\Bigg(\frac{\zeta^{1-r}\left(1+(1-r)\exp(-\zeta)\right)}{2-r}-\\&\ln\left(1+\zeta^{1-r}\right)\Bigg)\\<&\frac{1}{\mu(1-r)}\left(\zeta^{1-r}-\ln\left(1+\zeta^{1-r}\right)\right).\end{aligned}$$

Moreover, the value of $\lim_{r\to0^+}\left(\zeta^{1-r}-\ln(1+\zeta^{1-r})\right)$ is evaluated as below:

$$\lim_{r\to0^+}\left(\zeta^{1-r}-\ln(1+\zeta^{1-r})\right)=\lim_{\zeta\to0^+}\left(\zeta-\ln(1+\zeta)\right)=0.$$

That is to say, when $r\to0^+$, $t_{\text{AF3}}-t_{\text{AF2}}$ is less than 0 for any $\nu_{\text{ini}}\in(0,+\infty)$. Therefore, there exists a sufficiently small $\bar{r}\in(0,1)$ such that for any $r\in(0,\bar{r})\subset(0,1)$ and any

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2023.3319132

$\nu_{\text{ini}} \in (0, +\infty)$, $t_{\text{AF3}} - t_{\text{AF2}} < 0$ holds. The proof is thus completed. ■

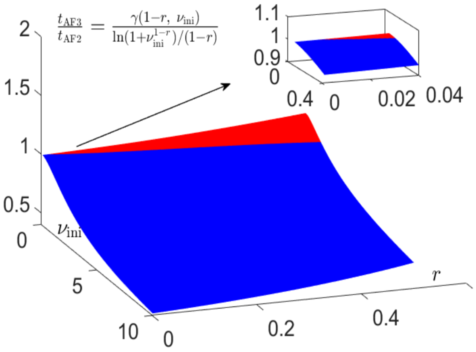


Fig. 2: Graph of ratio $t_{\text{AF3}}/t_{\text{AF2}}$ with $\nu_{\text{ini}} \in (0, 10]$ and $r \in (0, 0.5]$, where blue area represents $t_{\text{AF3}} < t_{\text{AF2}}$ and red one represents $t_{\text{AF3}} \geq t_{\text{AF2}}$.

Remark 4: The relationship between t_{AF3} and t_{AF2} is shown more apparently in Fig. 2. It is found in the local enlarged drawing that $\bar{r}$ mentioned in Theorem 3 exists on interval $(0,\ 0.02)$. Besides, for any $r \in (0, 1)$, there exists a $\hat{\zeta} > 0$ such that ratio $t_{\text{AF3}}/t_{\text{AF2}} < 1$ when $\nu_{\text{ini}} \in (\hat{\zeta}, +\infty)$ and $t_{\text{AF3}}/t_{\text{AF2}} \geq 1$ when $\nu_{\text{ini}} \in (0, \hat{\zeta}]$. Therefore, it is better to choose the AF (13) to assist the LCCZNN model (12) when the initial error ν_{ini} is relatively large. Since $\hat{\zeta} \to 0^+$ as $r \to 0^+$, the AF (13) is superior when r is relatively small.

Note that we have not taken noises into account so far. In fact, it is unreasonable not to consider noise disturbance in practical implementations due to noise ubiquity. Hence, it is of necessity to analyze the robustness of the LCCZNN model (12) under different kinds of noises. In this work, all sorts of noises are deemed additive for convenience [37]. Then, the disturbed LCCZNN model and its error dynamics are respectively formulated as

$$\dot{\boldsymbol{x}} = -\frac{J^{\mathrm{T}}\boldsymbol{f}}{\|J^{\mathrm{T}}\boldsymbol{f}\|_2^2}\left(\boldsymbol{f}^{\mathrm{T}}\frac{\partial \boldsymbol{f}}{\partial t} + \mu\phi\left(\frac{\|\boldsymbol{f}\|_2^2}{2}\right)\right) + \boldsymbol{\delta}(t), \tag{16}$$

and

$$\dot{\nu} = -\mu\phi(\nu) + d(t),$$

where $d(t) = \boldsymbol{f}^{\mathrm{T}}J\boldsymbol{\delta}(t)$, and $\boldsymbol{\delta}(t) \in \mathbb{R}^n$ represents the aggregation of all dynamic noises. The forthcoming Theorem 4 investigates the robustness of the disturbed LCCZNN model (16) in detail.

Theorem 4: Consider the DNES (5) with at least one solution. Let $\tilde{\nu} = \|\boldsymbol{f}\|_2 = (2\nu)^{1/2}$ be the residual error synthesized by the interfered LCCZNN model (16). Suppose that d is bounded and satisfies $|d| \leq \bar{d}$. Then, the steady-state residual error $\lim_{t\to+\infty}\tilde{\nu}$ synthesized by the interfered LCCZNN model (16) activated by the AF (13) with any initial state is bounded by $(2\ln(1+2\bar{d}/\mu))^{1/2}$. Moreover, the steady-state residual error $\lim_{t\to+\infty}\tilde{\nu}$ can be arbitrarily small as long as μ is sufficiently large.

Proof: Define a Lyapunov function candidate with respect to ν: $V_2(\nu) = (\text{sig}^r(\nu)\exp(|\nu|))^p = (\nu^r\exp(\nu))^p$, where $p = 1+1/r$. Correspondingly, the time derivative of V_2 is expressed as

$$\dot{V}_2(t,\nu) = \frac{\mathrm{d}V_2}{\mathrm{d}t} = -p(r\nu^r + \nu^{1+r})\exp(p\nu)(\mu\nu^r\exp(\nu) + d) \leq -p(r\nu^r + \nu^{1+r})\exp(p\nu)(\mu\nu^r\exp(\nu) - \bar{d}).$$

Let η satisfy $\mu\eta^r\exp(\eta) = 2\bar{d}$. Let $\alpha(\cdot)$ and $\beta(\cdot)$ be class $\mathcal{K}_\infty$ functions defined as $\alpha(|\nu|) = (\exp(\nu) - 1)^p$ and $\beta(|\nu|) = (\nu^r\exp(\nu))^p$. Inequality $\alpha \leq V_2$ holds for any $\nu \in [1, +\infty)$ evidently, and when $\nu \in [0, 1)$, inequality $\alpha \leq V_2$ is proved by the deduction below:

$$\begin{aligned} &1 - \nu^r \leq 1 - \nu \leq \exp(-\nu) \\ &\Rightarrow (1-\nu^r)\exp(\nu) \leq 1 \\ &\Rightarrow \exp(\nu) - 1 \leq \nu^r\exp(\nu) \\ &\Rightarrow (\exp(\nu) - 1)^p \leq (\nu^r\exp(\nu))^p. \end{aligned}$$

Then, the following results are obtained:

1) $\alpha \leq V_2 \leq \beta$, $\forall\ t \in [0, +\infty)$ and $\forall\ \nu \in [0, +\infty)$,
2) $\dot{V}_2 \leq -\bar{d}p(r\nu^r + \nu^{1+r})\exp(p\nu)$, $\forall \nu \geq \eta$.

In light of Lemma 2, for any initial state $\boldsymbol{x}(0)$, there exists a $T = T(\eta, \nu_{\text{ini}})$ such that

$$|\nu| = \nu \leq \alpha^{-1}(\beta(\eta)) = \ln\left(1 + \frac{2\bar{d}}{\mu}\right)$$

holds for any $t \geq T$. As a consequence,

$$\tilde{\nu} = \|\boldsymbol{f}\|_2 = \sqrt{2\nu} \leq \sqrt{2\ln\left(1 + \frac{2\bar{d}}{\mu}\right)}, \quad \forall\ t \geq T.$$

The upper bound of the steady-state residual error is negatively correlated with μ, and thus, the steady-state residual error can be arbitrarily small as long as μ is sufficiently large. The proof is thus completed. ■

We have provided a lot of proofs on the properties of the LCCZNN model and the comparisons among the AFs in this section. Next, experimental results are displayed to verify the properties.

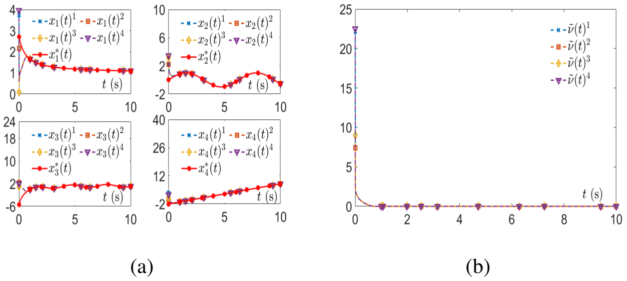


Fig. 3: Solutions and residual errors synthesized by LCCZNN model (12) starting from four different initial states with $\mu = 5$ and $r = 0.6$, where $\boldsymbol{x}^*(t)$ represents one feasible solution of system (17), and $\boldsymbol{x}(t)^i$ and $\tilde{\nu}(t)^i$ respectively denote solution and residual error generated by (12) starting from initial state $\boldsymbol{x}(0)^i$ with $i \in \{1, 2, 3, 4\}$. (a) Neural states $\boldsymbol{x}(t)^i$. (b) Residual errors $\tilde{\nu}(t)^i$.

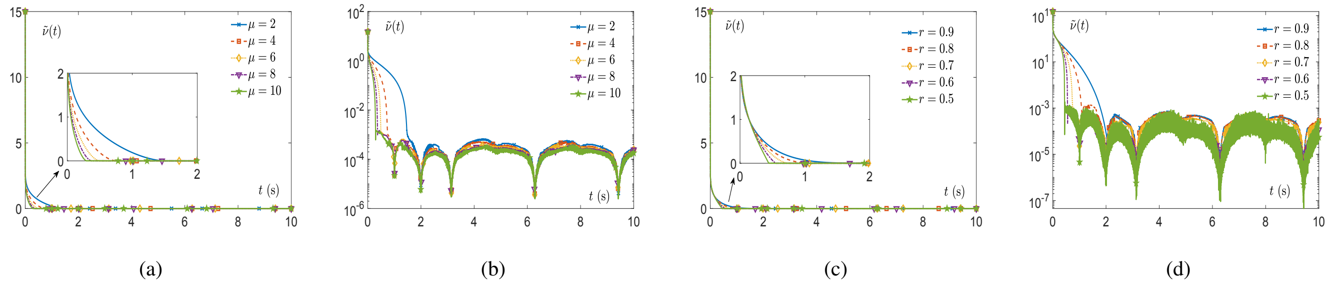


Fig. 4: Residual errors synthesized by LCCZNN model (12) with different values of parameters. (a) Residual errors $\tilde{\nu}(t)$ with $r = 0.7$ and different values of μ. (b) Semilog plot of residual errors $\tilde{\nu}(t)$ with $r = 0.7$ and different values of μ. (c) Residual errors $\tilde{\nu}(t)$ with $\mu = 4$ and different values of r. (d) Semilog plot of residual errors $\tilde{\nu}(t)$ with $\mu = 4$ and different values of r.

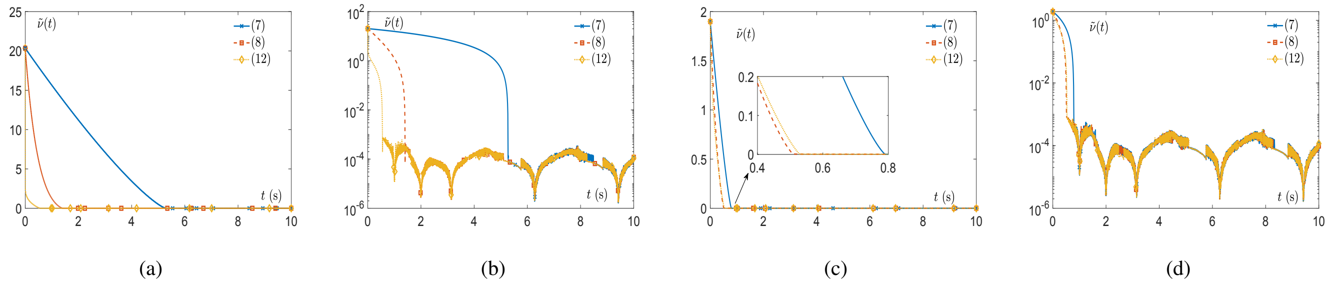


Fig. 5: Residual errors synthesized by LCCZNN model (12) activated by three distinct functions (i.e., (8), (9), and (13)) with $\mu = 4$ and $r = 0.6$. (a) Residual errors $\tilde{\nu}(t)$ with relatively large initial error $\nu(0)$. (b) Semilog plot of residual errors $\tilde{\nu}(t)$ with relatively large initial error $\nu(0)$. (c) Residual errors $\tilde{\nu}(t)$ with relatively small initial error $\nu(0)$. (d) Semilog plot of residual errors $\tilde{\nu}(t)$ with relatively small initial error $\nu(0)$.

IV. Numerical and Simulative Experiments

In this section, numerical experiments are first conducted to verify the theories corroborated in Section III, including the effectiveness and robustness of the proposed LCCZNN model (12), and the effects of various AFs and parameter values on the model. Then, model (12) is applied to the trajectory-following control of a UR5 manipulator with six degrees of freedom. The successful simulation outcome substantiates the practical value of model (12). It is worth mentioning that (13) is the default AF for the LCCZNN model (12) in the experiments.

A. Case Study without Noises

Consider the following DNES:

$$\boldsymbol{f}(t,\boldsymbol{x}(t)) = \begin{cases} \ln(x_1(t)) - 1/(t+1) = 0, \\ x_1(t)x_2(t) - \sin(t)\exp(1/(t+1)) = 0, \\ x_1^2(t) - \sin(t)x_2(t) + x_3(t) - 2 = 0, \\ x_1^2(t) - x_2^2(t) + x_3(t) + x_4(t) - t = 0, \end{cases} \tag{17}$$

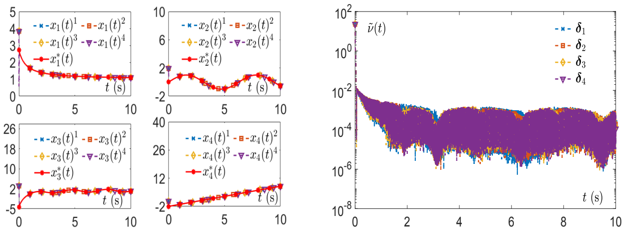
where $x_1(t) \neq 0$. One feasible solution of system (17) is formulated as

$$\boldsymbol{x}^*(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} \exp(1/(t+1)) \\ \sin(t) \\ 2 - \exp(2/(t+1)) + \sin^2(t) \\ t-2 \end{bmatrix}.$$

First, in order to validate whether the LCCZNN model (12) is competent to deal with DNESs, model (12) is utilized to solve system (17). In the experiments, parameters μ and r are respectively set as 4 and 0.7, and four initial states $\boldsymbol{x}(0) \in [0,4]^{4\times 1}$ are randomly generated to observe the influence of the initial state on the convergence. The corresponding results are displayed in Fig. 3. In Fig. 3(a), the solid curves with red color and markers $*$ represent the feasible solution, and the dashed ones with other colors and markers are solutions generated by model (12) with different initial states. Evidently, the dashed curves overlap with the solid ones in a short time. The residual errors $\tilde{\nu}$ synthesized by model (12) with different initial states are depicted in Fig. 3(b), where each $\tilde{\nu}$ dwindles to 0 within 1 second. Therefore, the LCCZNN model (12) is capable to handle the DNES and insensitive to the initial state. To put it in another way, it is verified that $\nu = 0$ is globally asymptotically stable.

Second, we pay attention to the effects of parameters μ and r on the convergence performance of model (12), by fixing the initial state $\boldsymbol{x}(0)$ and one of the parameters, and adjusting the other one. As observed in Fig. 4, it is verified that the convergence time of model (12) is cut down by raising μ or diminishing r, which is completely in coincidence with Theorem 2 in Section III. Besides, it is seen that the variation of the parameters has nothing to do with the precision of model (12), which is between 10^{-3} and 10^{-4}.

Third, we focus on the impact of various AFs on the convergence performance of model (12). In the experiments, parameters μ and r are respectively set as 4 and 0.6, and the corresponding results are shown in Fig. 5. As seen in Fig. 5(a), starting from a relatively large initial error $\nu(0)$, model (12) takes the least time to converge when aided by the AF (13). It is also observed that the convergence times of model (12) assisted by (9) and (13) are almost the same, when it starts from a relatively small initial error $\nu(0)$. Besides, model (12) aided by (8) takes the most time to converge no matter what the initial error $\nu(0)$ is. The above observations are consistent with Remark 3 and Theorem 3 in Section III. In addition, different AFs also have no effect on the precision of model (12), as spotted in Fig. 5(b) and (d).



(a) (b)

Fig. 6: Solutions and residual errors synthesized by LCCZNN model (16) disturbed by different noises with $\mu = 50$ and $r = 0.5$, where $\boldsymbol{x}^*(t)$ represents one feasible solution of system (17), and $\boldsymbol{x}(t)^i$ and $\tilde{\nu}(t)^i$ respectively denote solution and residual error generated by (16) disturbed by $\boldsymbol{\delta}_i(t)$ with $i \in \{1, 2, 3, 4\}$. (a) Neural states $\boldsymbol{x}(t)^i$. (b) Semilog plot of residual errors $\tilde{\nu}(t)^i$.

B. Case Study with Noises

Let us reconsider system (17) of interest under noisy circumstances. In this study, a constant noise, two dynamic noises, and a random noise are incorporated, and the disturbed LCCZNN model (16) is exploited to cope with system (17). Specifically, each element of the constant noise $\boldsymbol{\delta}_1$, the dynamic linear noise $\boldsymbol{\delta}_2$, and the dynamic quadratic noise $\boldsymbol{\delta}_3$ is 5, $0.6t$, and $0.2(t-5)^2$. The random noise $\boldsymbol{\delta}_4$ is composed of four randomly generated values ranging from 0 to 5. Then, We adopt these noises to interfere model (16), and check whether its residual errors $\tilde{\nu}$ are able to converge to a certain bound or not.

Under the four kinds of noises, the performance of model (16) assisted by (13) with parameters $\mu = 50$ and $r = 0.5$ is presented in Fig. 6. In Fig. 6(a), the solid curves with red color and markers $*$ represent the feasible solution, and the dashed ones with other colors and markers are solutions generated by model (16) interfered with different noises. It is evident that the solutions synthesized by model (16) still converge to the feasible solution swiftly, which means that model (12) assisted by (13) is of robustness to an extent. Additionally, the residual errors synthesized by model (16) remain below approximately 10^{-3} after 3 s, as seen in Fig. 6(b).

C. Application to UR5 Manipulator

In this subsection, the LCCZNN model (12) is applied to a practical problem. Robots are becoming more common in industry and real life [38]. As a sort of robots, manipulators have replaced human labor in lots of industrial activities and improved labor productivity. Trajectory-following control is one important problem for manipulators, and it has many applications, such as writing and assembly [28].

On the basis of model (12), a controller that controls the UR5 manipulator is designed to online follow a desired trajectory in three-dimensional space. In light of [39], the forward kinematics of the redundant manipulator is generalized as

$$\boldsymbol{\psi}(\boldsymbol{\theta}(t)) = \boldsymbol{p}_{\mathrm{a}}(t),$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^{\mathrm{T}} \in \mathbb{R}^6$ denotes its joint angles, $\boldsymbol{p}_{\mathrm{a}} = [p_{\mathrm{aX}}, p_{\mathrm{aY}}, p_{\mathrm{aZ}}]^{\mathrm{T}} \in \mathbb{R}^3$ denotes the actual position of the end-effector in the Cartesian space, and $\boldsymbol{\psi}\colon \mathbb{R}^6 \mapsto \mathbb{R}^3$ maps $\boldsymbol{\theta}$ onto $\boldsymbol{p}_{\mathrm{a}}$ nonlinearly. In the trajectory-following task, $\boldsymbol{p}_{\mathrm{a}}(t)$ is expected to overlap a desired position denoted as $\boldsymbol{p}_{\mathrm{d}}(t)$ at any time t. That is to say, $\boldsymbol{p}_{\mathrm{a}}(t) - \boldsymbol{p}_{\mathrm{d}}(t) = \boldsymbol{\psi}(\boldsymbol{\theta}(t)) - \boldsymbol{p}_{\mathrm{d}}(t) = \mathbf{0}$ is wanted. Let $\boldsymbol{\Psi}(t, \boldsymbol{\theta}(t)) = \boldsymbol{\psi}(\boldsymbol{\theta}(t)) - \boldsymbol{p}_{\mathrm{d}}(t)$. Then, the essence of the trajectory-following task is to solve $\boldsymbol{\Psi}(t, \boldsymbol{\theta}(t)) = \mathbf{0}$, which is a DNES. Derived from model (12), a pseudoinverse-free controller is formulated as

$$\dot{\boldsymbol{\theta}} = \frac{J_{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{\Psi}}{\|J_{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{\Psi}\|_2^2} \left(\boldsymbol{\Psi}^{\mathrm{T}} \dot{\boldsymbol{p}}_{\mathrm{d}} - \mu \phi \left(\frac{\|\boldsymbol{\Psi}\|_2^2}{2} \right) \right), \tag{18}$$

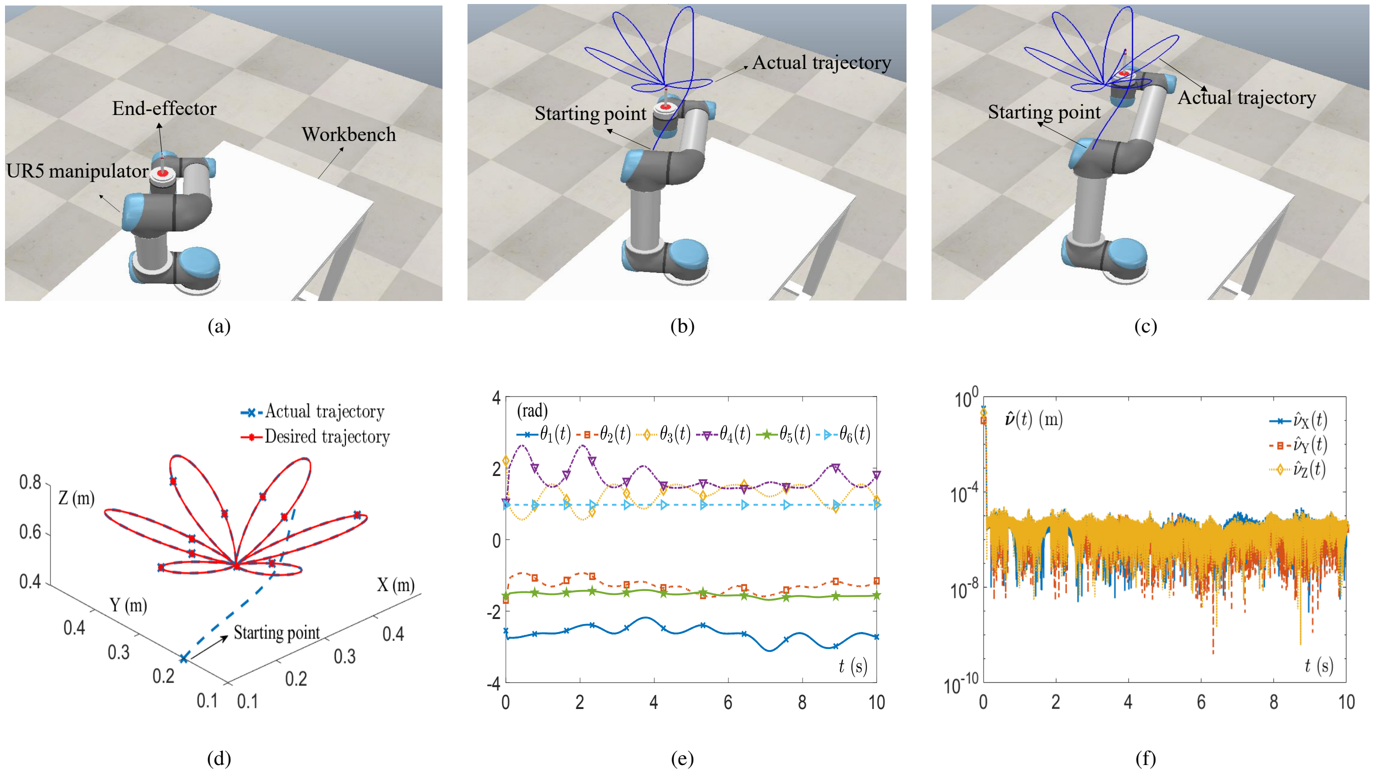
where $\dot{\boldsymbol{\theta}}$ represents the joint velocities, $J_{\boldsymbol{\psi}} = \partial \boldsymbol{\psi} / \partial \boldsymbol{\theta} \in \mathbb{R}^{3\times 6}$ represents the Jacobian matrix of the manipulator, and $\dot{\boldsymbol{p}}_{\mathrm{d}} = \mathrm{d}\boldsymbol{p}_{\mathrm{d}}/\mathrm{d}t \in \mathbb{R}^3$ represents the desired velocity of the end-effector. In the same way, the controller based on model (6) with a linear function array $\boldsymbol{\phi}(\boldsymbol{\Psi}) = \boldsymbol{\Psi}$ is presented as

$$\dot{\boldsymbol{\theta}} = J_{\boldsymbol{\psi}}^{\dagger} \left(\dot{\boldsymbol{p}}_{\mathrm{d}} - \mu \boldsymbol{\Psi} \right). \tag{19}$$

In the simulation, parameters μ and r are respectively set to be 6 and 0.5, simulation time T is set as 10 s, and a flower-shaped trajectory desired to be tracked is described as

$$\boldsymbol{p}_{\mathrm{d}}(t) = \frac{1}{10} \begin{bmatrix} 3 + (1 + 1\sin(6\pi t/5)) \cos(\pi t/5)/2 \\ 3 + (1 + 1\sin(6\pi t/6)) \sin(\pi t/5)/2 \\ 6 + \sin(6\pi t/5)/2 \end{bmatrix} \mathrm{m}.$$

The simulation environment is exhibited in Fig. 7(a), where the UR5 manipulator with a pen installed as the end-effector is put on the workbench, and its joint angles are initialized as $\boldsymbol{\theta}(0) = [-4\pi/5, -\pi/2, 7\pi/10, 3\pi/10, -\pi/2, 3\pi/10]^{\mathrm{T}}$ rad. The motion process of the manipulator controlled by (18) is displayed in Fig. 7(b) through (f). Specifically, snapshots of the manipulator controlled by (18) during the motion process are provided in Fig. 7(b) and (c). As seen from Fig. 7(d), the actual trajectory overlaps the desired one after a period. The corresponding resolved joint angles are presented in Fig. 7(e). The tracking errors along X-, Y-, and Z-axes displayed in Fig. 7(f) are defined as $\hat{\nu}_{\mathrm{X}}(t) = |p_{\mathrm{aX}}(t) - p_{\mathrm{dX}}(t)|$, $\hat{\nu}_{\mathrm{Y}}(t) = |p_{\mathrm{aY}}(t) - p_{\mathrm{dY}}(t)|$, and $\hat{\nu}_{\mathrm{Z}}(t) = |p_{\mathrm{aZ}}(t) - p_{\mathrm{dZ}}(t)|$. It is seen in Fig. 7(f) that errors $\hat{\nu}_{\mathrm{X}}$, $\hat{\nu}_{\mathrm{Y}}$, and $\hat{\nu}_{\mathrm{Z}}$ converge within 0.5 s, and the magnitudes of the steady-state errors $\hat{\nu}_{\mathrm{X}}$, $\hat{\nu}_{\mathrm{Y}}$, and $\hat{\nu}_{\mathrm{Z}}$ are all approximately 10^{-5} m, which illustrates that the trajectory-



(a) (b) (c)

(d) (e) (f)

Fig. 7: Motion process of UR5 manipulator controlled by controller (18) with $\mu = 6$ and $r = 0.5$. (a) Initial state. (b) Intermediate state. (c) Final state. (d) Actual and desired trajectories. (e) Joint angles. (f) Tracking errors along X-, Y-, and Z-axes.

following task is completed successfully, and controller (18) is competent and powerful. The simulation result synthesized by controller (19) resembles the one synthesized by controller (18) and thus omitted here, except that controller (19) spends more time (about 5.5 s) to converge.

TABLE I: Running times of controllers (18) and (19) with $\mu = 6$ and $r = 0.5$.

Index	Controller (18)	Controller (19)
T_u ($\times 10^{-5}$ s)	0.74	3.79
T_o ($\times 10^{-5}$ s)	2.86	5.61
N	132672	80000
NT_o (s)	3.79	4.49

In the aspect of the running time, two criteria in [40] are used in the simulation for comparison. Concretely, the average time for the controller to update $\boldsymbol{\theta}$ and the average time for it to finish one control operation are denoted as T_u and T_o, respectively. Furthermore, the number of operations N and the total running time NT_o are taken into consideration. Table I displays the running times of controllers (18) and (19). On the one hand, T_u of controller (18) is shorter than that of controller (19). On the other hand, in spite of more operations, the total running time NT_o of controller (18) is less than that of controller (19). Both observations substantiate that the LCCZNN model (12) has lower computational complexity than model (6). Besides, on the basis of the fact that the total running times NT_o of two controllers are less than the simulation time $T = 10$ s, it follows that both controllers have the capability to complete the trajectory tracking online.

TABLE II: Running times of perturbed controllers (20) and (21) with $\mu = 6$ and $r = 0.5$.

Index	Controller (20)	Controller (21)
T_u ($\times 10^{-5}$ s)	0.95	4.33
T_o ($\times 10^{-5}$ s)	3.07	6.15
N	195088	115672
NT_o (s)	5.99	7.11

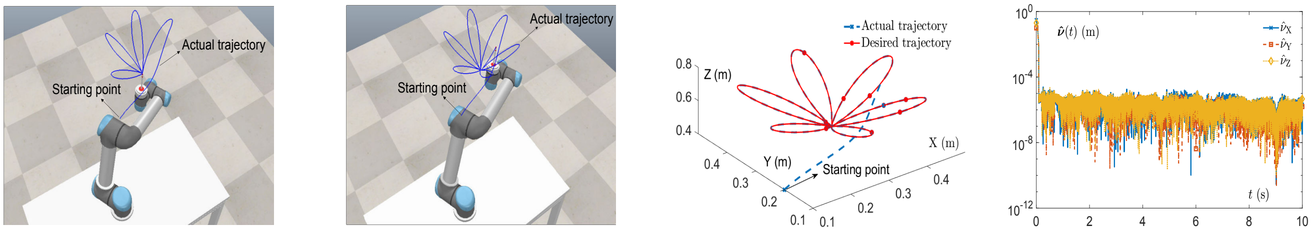
Additionally, noise disturbance is also considered in the simulation. Suppose that the above controllers with perturbance are formulated as

$$\dot{\boldsymbol{\theta}} = \frac{J_{\boldsymbol{\psi}}^{\mathrm{T}}\boldsymbol{\Psi}}{\|J_{\boldsymbol{\psi}}^{\mathrm{T}}\boldsymbol{\Psi}\|_2^2}\left(\boldsymbol{\Psi}^{\mathrm{T}}\dot{\boldsymbol{p}}_{\mathrm{d}} - \mu\phi\left(\frac{\|\boldsymbol{\Psi}\|_2^2}{2}\right)\right) + \boldsymbol{\delta}_{\boldsymbol{\theta}} \qquad (20)$$

and

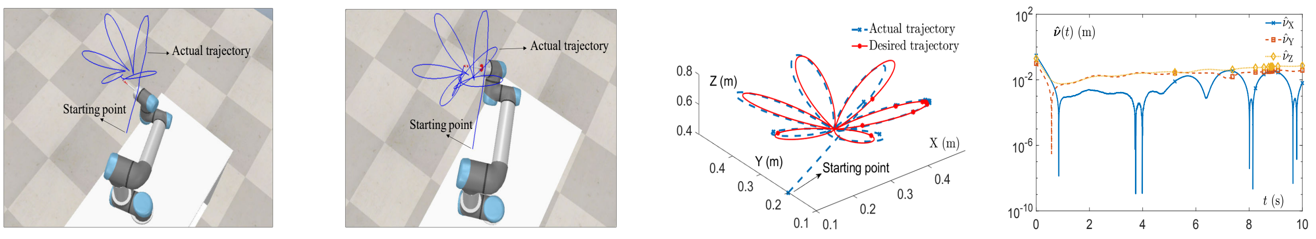
$$\dot{\boldsymbol{\theta}} = J_{\boldsymbol{\psi}}^{\dagger}\left(\dot{\boldsymbol{p}}_{\mathrm{d}} - \mu\boldsymbol{\Psi}\right) + \boldsymbol{\delta}_{\boldsymbol{\theta}}, \qquad (21)$$

where $\boldsymbol{\delta}_{\boldsymbol{\theta}} \in \mathbb{R}^6$ is a random noise whose elements range from 0 to 3. With the same parameters and settings, the motion processes of the manipulator and the running times of the controllers are illustrated in Figs. 8 and 9 along with Table II. As seen in Fig. 8 and Table II, under the noise interference,



(a) (b) (c) (d)

Fig. 8: Motion process of UR5 manipulator controlled by perturbed controller (20) with $\mu = 6$ and $r = 0.5$. (a) Intermediate state. (b) Final state. (c) Actual and desired trajectories. (d) Tracking errors along X-, Y-, and Z-axes.



(a) (b) (c) (d)

Fig. 9: Motion process of UR5 manipulator controlled by perturbed controller (21) with $\mu = 6$. (a) Intermediate state. (b) Final state. (c) Actual and desired trajectories. (d) Tracking errors along X-, Y-, and Z-axes.

the interfered controller (20) still accomplishes the tracking task successfully and keeps the same high precision as in the absence of perturbance. The steady-state tracking error $\|\hat{\boldsymbol{\nu}}\|_2$ synthesized by (20) is about 4.95×10^{-5} m. On the contrary, as observed in Fig. 9 and Table II, although the interfered controller (21) maintains the real-time processing ability, it fails to control the manipulator to follow the desired trajectory accurately in the presence of perturbance and the tracking error $\|\hat{\boldsymbol{\nu}}\|_2$ diverges as time elapses. In a word, the LCCZNN model (12) possesses a better real-time processing capability and is of certain robustness, meaning that it is more proper for handling the practical problem.

V. Conclusion

In this paper, the LCCZNN model has been proposed for solving problems that can be mathematically described as DNESs. The LCCZNN model gets rid of any pseudoinverse computation and thus has $O(mn)$ computational complexity, which is a great progress compared with the $O(mn^2)$ model. It is also convenient to implement the model via an analog circuit, owing to its core of explicit dynamics. Besides, the novel AF modified from the sign-bi-power function has been acquired for strengthening the model in terms of convergence and robustness. Through Lyapunov theory, we have determined the upper bounds of the model's convergence time and the disturbed one's steady-state residual error. We have as well compared the new AF with two other AFs in the aspect of the model's convergent rate, and found out the advantages of the newly proposed AF. Finally, the numerical experiments coupled with the simulation on the trajectory-following control of the UR5 manipulator have been conducted to validate the effectiveness and relevant analysis results of the proposed model.

Appendix

The detailed proofs of Lemma 3 and Theorem 2 are provided here.

Proof of Lemma 3

An auxiliary function $h_1(z) = \exp(z) - z^a - 1$ with $z \in (0, +\infty)$ is introduced, and its first- and second-order derivatives are

$$h_1' = \frac{\mathrm{d}h_1}{\mathrm{d}z} = \exp(z) - az^{-(1-a)}$$

and

$$h_1'' = \frac{\mathrm{d}^2 h_1}{\mathrm{d}z^2} = \exp(z) + a(1-a)z^{-(2-a)}.$$

Since $a \in (0,1)$ and $z \in (0, +\infty)$, h_1'' is always larger than 0 on the domain. As a consequence, h_1' is strictly increasing with respect to z. It is observed that

$$\lim_{z \to 0^+} h_1'(z) = -\infty$$

and

$$h_1'(1) = \exp(1) - a > 0.$$

Therefore, there only exists one root $\rho \in (0,1)$ satisfying $h_1'(\rho) = 0$. As a result, h is strictly decreasing on $(0, \rho)$ and strictly increasing on $(\rho, +\infty)$. It follows from $\lim_{z \to 0^+} h_1(z) = 0$ that $h_1(\rho) < 0$. On account of the fact that $h_1(1) > 0$, we deduce that there only exists one root $\zeta \in (\rho, 1) \subset (0, 1)$ satisfying $h_1(\zeta) = 0$.

Suppose that ζ and $\bar{\zeta}$ respectively correspond to the roots of equations $\exp(z) - z^a - 1 = 0$ and $\exp(z) - z^{\bar{a}} - 1 = 0$, with $0 < a < \bar{a} < 1$. Through simple transformations, a and $\bar{a}$ are expressed as

$$a = \frac{\ln(\exp(\zeta) - 1)}{\ln \zeta}$$

and

$$\bar{a} = \frac{\ln(\exp(\bar{\zeta}) - 1)}{\ln \bar{\zeta}}.$$

Another auxiliary function is introduced as

$$h_2(z) = \frac{\ln(\exp(z) - 1)}{\ln z}, \quad z \in (0, 1),$$

and correspondingly, its first-order derivative is

$$h_2' = \frac{\mathrm{d}h_2}{\mathrm{d}z} = \frac{1}{\ln^2 z}\left(\frac{\exp(z)\ln z}{\exp(z) - 1} - \frac{\ln(\exp(z) - 1)}{z}\right).$$

Since inequality $\exp(z)z\ln z < (\exp(z) - 1)\ln(\exp(z) - 1)$ holds for any $z \in (0, 1)$, h_2' is negative on the domain. Hence, h_2 is strictly decreasing with respect to z.

Finally, according to the relationship between a and $\bar{a}$ (i.e., $a < \bar{a}$), and the monotonicity of h_2, it is inferred that $\zeta > \bar{\zeta}$. That is to say, the root of equation (3) decreases with the increase of a. In view of the fact that

$$\lim_{z \to 0^+} h_2 = \lim_{z \to 0^+} \frac{\ln(\exp(z) - 1)}{\ln z} = 1,$$

it is evident that

$$\lim_{a \to 1^-} \zeta = 0.$$

Hence, the proof is completed.

Proof of Theorem 2

On the basis of Theorem 1, model (12) converges to one solution of (5) in finite time $t_{\mathrm{AF3}} = \gamma(1 - r, \nu_{\mathrm{ini}})/\mu$. It is easy to verify that the first manner (i.e., increasing parameter μ) is useful to shorten the convergence time, since t_{AF3} is inversely proportional to μ.

In order to prove the second manner, the partial derivative of t_{AF3} with respect to r is written out:

$$\frac{\partial t_{\mathrm{AF3}}}{\partial r} = \frac{1}{\mu}\int_0^{\nu_{\mathrm{ini}}} -\nu^{-r}\exp(-\nu)\ln\nu \ \mathrm{d}\nu.$$

$\partial t_{\mathrm{AF3}}/\partial r > 0$ holds for any $\nu_{\mathrm{ini}} \in (0, 1]$. When $\nu_{\mathrm{ini}} \in (1, +\infty)$, it follows that

$$\begin{aligned}
&\int_0^{\nu_{\mathrm{ini}}} \quad \nu^{-r}\exp(\quad \nu)\ln\nu \ \mathrm{d}e \\
&= \int_0^1 \quad \nu^{-r}\exp(\quad \nu)\ln\nu \ \mathrm{d}\nu + \int_1^{\nu_{\mathrm{ini}}} \quad \nu^{-r}\exp(\quad \nu)\ln\nu \ \mathrm{d}\nu \\
&= \int_0^1 \quad \nu^{-r}\exp(\quad \nu)\ln\nu \ \mathrm{d}\nu + \int_{1/\nu_{\mathrm{ini}}}^1 \nu^{-(2-r)}\exp\left(\quad \frac{1}{\nu}\right)\ln\nu \ \mathrm{d}\nu \\
&> \quad \int_{1/\nu_{\mathrm{ini}}}^1 \left(\nu^{-r}\exp(\quad \nu) \quad \nu^{-(2-r)}\exp\left(\quad \frac{1}{\nu}\right)\right)\ln\nu \ \mathrm{d}\nu.
\end{aligned}$$

Since inequality $1/\nu - \nu + 2(1 - r)\ln\nu > 0$ holds for any $\nu \in (1/\nu_{\mathrm{ini}}, 1) \subset (0, 1)$, the following deduction is obtained:

$$\begin{aligned}
&\frac{1}{\nu} - \nu + 2(1 - r)\ln\nu > 0 \Leftrightarrow \frac{1}{\nu} - \nu > -2(1 - r)\ln\nu \\
&\Leftrightarrow \exp\left(\frac{1}{\nu} - \nu\right) > \nu^{-2(1-r)} \\
&\Leftrightarrow \nu^{-r}\exp(-\nu) - \nu^{-(2-r)}\exp\left(-\frac{1}{\nu}\right) > 0.
\end{aligned}$$

Consequently, $\partial t_{\mathrm{AF3}}/\partial r > 0$ holds when $\nu_{\mathrm{ini}} \in (1, +\infty)$.

To sum up, decreasing parameter r is also functional to shorten the convergence time. The proof is thus completed.

References

[1] L. Jin, L. Wei, and S. Li, "Gradient-based differential neural-solution to time-dependent nonlinear optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 620–627, Jan. 2023.

[2] L. Xiao, J. Dai, L. Jin, W. Li, S. Li, and J. Hou, "A noise-enduring and finite-time zeroing neural network for equality-constrained time-varying nonlinear optimization," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 8, pp. 4729–4740, Aug. 2021.

[3] N. Tan, X. Gu, and H. Ren, "Pose characterization and analysis of soft continuum robots with modeling uncertainties based on interval arithmetic," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 570–584, Apr. 2019.

[4] D. Chen, S. Li, W. Li, and Q. Wu, "A multi-level simultaneous minimization scheme applied to jerk-bounded redundant robot manipulators," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 463–474, Jan. 2020.

[5] W. Guo, G. Li, Q. Zhang, Y. Luo, and Z. Wang, "Solving nonlinear equation systems by a two-phase evolutionary algorithm," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 9, pp. 5652–5663, Sep. 2021.

[6] W. Li, L. Xiao, and B. Liao, "A finite-time convergent and noise-rejection recurrent neural network and its discretization for dynamic nonlinear equations solving," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3195–3207, Jul. 2020.

[7] T. Liu, Z. Qin, Y. Hong, and Z.-P. Jiang, "Distributed optimization of nonlinear multiagent systems: A small-gain approach," *IEEE Trans. Autom. Control*, vol. 67, no. 2, pp. 676–691, Feb. 2022.

[8] J. Santiaguillo-Salinas and E. Aranda-Bricaire, "Time-varying containment problem for multi-agent systems," in *Proc. Int. Conf. Electr. Eng. Comput. Sci. Automat. Contr.*, Mexico City, Mexico, 2013, pp. 358–363.

[9] F.B. Hilderbrand, *Introduction to Numerical Analysis*. Mineola, NY, USA: Dover Publications, 1987.

[10] M.S. Petkovic, B. Neta, L.D. Petkovic, and J. Dznic, "Multipoint methods for solving nonlinear equations: A survey," *Appl. Math. Comput.*, vol. 226, pp. 635–660, Jan. 2014.

[11] H. Ramos and M.T.T Monteiro, "A new approach based on the Newton's method to solve systems of nonlinear equations," *J. Comput. Appl. Math.*, vol. 318, pp. 3–13, Jul. 2017.

[12] Y. Zhang, Y. Shi, L. Xiao, and B. Mu, "Convergence and stability results of Zhang neural network solving systems of time-varying nonlinear equations," in *Proc. Int. Conf. Nat. Comput.*, Chongqing, China, 2012, pp. 143–147.

[13] Y. Xia, G. Feng, and J. Wang, "A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints," *IEEE Trans. Neural Netw.*, vol. 19, no. 8, pp. 1340–1353, Aug. 2008.

[14] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3310–3315, Dec. 2015.

[15] L. Xiao, S. Li, L. Jin, K. Li, and B. Liao, "Co-design of finite-time convergence and noise suppression: A unified neural model for time varying linear equations with robotic applications," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 12, pp. 5233–5243, Dec. 2020.

[16] W. Li, L. Han, X. Xiao, B. Liao, and C. Peng, "A gradient-based neural network accelerated for vision-based control of an RCM-constrained surgical endoscope robot," *Neural. Comput. Appl.*, vol. 34, pp. 1329–1343, Jan. 2022.

[17] Y. Zhang, P. Xu, and N. Tan, "Further studies on Zhang neural-dynamics and gradient dynamics for online nonlinear equations solving," in *Proc. IEEE Int. Conf. Autom. Logist.*, Shenyang, China, 2009, pp. 566–571.

[18] Y. Zhang, C. Yi, and D. Guo, "Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation," *Neural Comput. Appl.*, vol. 20, no. 1, pp. 1–7, Feb. 2011.

[19] Y. Zhang, D. Jiang, and J. Wang, "A recurrent neural network for solving Sylvester equation with time-varying coefficients," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1053–1063, Sep. 2002.

[20] L. Xiao and Y. He, "A noise-suppression ZNN model with new variable parameter for dynamic Sylvester equation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7513–7522, Nov. 2021.

[21] J. Li, Y. Zhang, S. Li, and M. Mao, "New discretization-formula-based zeroing dynamics for real-time tracking control of serial and parallel manipulators," *IEEE Trans Ind. Informat.*, vol. 14, no. 8, pp. 3416–3425, Aug. 2018.

[22] L. Xiao, Z. Zhang, and S. Li, "Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 11, pp. 2210–2220, Nov. 2019.

[23] S. Li, S. Chen, and B. Liu, "Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function," *Neural Process. Lett.*, vol. 37, no. 2, pp. 189–205, Apr. 2013.

[24] L. Xiao, J. Tao, and W. Li, "An arctan-type varying-parameter ZNN for solving time-varying complex Sylvester equations in finite time," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 3651–3660, Jun. 2022.

[25] Z. Zhang, L. Kong, L. Zheng, P. Zhang, X. Qu, B. Liao, and Z. Yu, "Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying QM and QP problems and applications," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 12, pp. 5106–5118, Dec. 2020.

[26] L. Xiao, Y. Zhang, J. Dai, J. Li, and W. Li, "New noise-tolerant ZNN models with predefined-time convergence for time-variant Sylvester equation solving," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 6, pp. 3629–3640, Jun. 2021.

[27] Z. Tan, W. Li, L. Xiao, and Y. Hu, "New varying-parameter ZNN models with finite-time convergence and noise suppression for time-varying matrix Moore-Penrose inversion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2980–2992, Aug. 2020.

[28] N. Tan and P. Yu, "Robust model-free control for redundant robotic manipulators based on zeroing neural networks activated by nonlinear functions," *Neurocomputing*, vol. 438, pp. 44–54, May. 2021.

[29] H.K. Khalil, *Nonlinear Systems Third Edition*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[30] S.P. Bhat and D.S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM J. Control Optim.*, vol. 38, no. 3, pp. 751–766, Feb. 2000.

[31] L. Jin, Y. Zhang, S. Li, and Y. Zhang, "Noise-tolerant ZNN models for solving time-varying zero-finding problems: A control-theoretic approach," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 992–997, Feb. 2017.

[32] Z. Zhang, Y. Lu, L. Zheng, S. Li, Z. Yu, and Y. Li, "A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4110–4125, Dec. 2018.

[33] Z. Zhang, L. Zheng, T. Qiu, and F. Deng, "Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 874–881, Feb. 2020.

[34] S. Li and Y. Li, "Nonlinearly activated neural network for solving time-varying complex Sylvester equation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1397–1407, Oct. 2013.

[35] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Washington, DC, USA: Government Publishing Office, 1964.

[36] E. Neuman, "Inequalities and bounds for the incomplete gamma function," *Results Math.*, vol. 63, no. 3, pp. 1209–1214, Jun. 2013.

[37] V.G. Rao and D.S. Bernstein, "Naive control of the double integrator: A comparison of a dozen diverse controllers under off-nominal conditions," in *Proc. Am. Control Conf.*, San Diego, USA, 1999, pp. 1477–1481.

[38] M. Yang, Y. Zhang, X. Zhou, and H. Hu, "Pose control of constrained redundant arm using recurrent neural networks and one-iteration computing algorithm," *Appl. Soft Comput.*, vol. 113, p. 108007, Dec. 2021.

[39] J.J. Craig, *Introduction to Robotics: Mechanics and Control*. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2005.

[40] L. Zheng and Z. Zhang, "Time-varying quadratic-programming-based error redefinition neural network control and its application to mobile redundant manipulators," *IEEE Trans. Autom. Control*, vol. 67, no. 11, pp. 6151–6158, Nov. 2022.

Kangze Zheng received the B.S. degree in computer science and technology from Sun Yat-sen University, Guangzhou, China, in 2021. He is currently pursuing the M.S. degree at the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.

His current research interests include robotic control, neural networks, multiagent systems, and intelligent optimization.

Shuai Li (Senior Member, IEEE) received the B.E. degree in precision mechanical engineering from the Hefei University of Technology, Hefei, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, Hefei, in 2008, and the Ph.D. degree in electrical and computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2014.

He is currently a full professor with Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland, and also with VTT-Technology Research Center of Finland. His current research interests include dynamic neural networks, robotics, machine learning, and autonomous systems.

Dr. Li is the Founding Editor-in-Chief of the International Journal of Robotics and Control, an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMSII: EXPRESS BRIEFS, and the General Co-Chair of the 2018 International Conference on Advanced Robotics and Intelligent Control.

Yunong Zhang (Member, IEEE) received the B.S. degree in industrial electrical automation from Huazhong University of Science and Technology, Wuhan, China, in 1996, the M.S. degree in control theory and control engineering from South China University of Technology, Guangzhou, China, in 1999, and the Ph.D. degree in mechanical and automation engineering from Chinese University of Hong Kong, Shatin, Hong Kong, China, in 2003.

He is currently a professor in the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. Before joining Sun Yat-sen University in 2006, he had been with National University of Singapore, University of Strathclyde, and National University of Ireland at Maynooth, since 2003. His main research interests include robotics, neural networks, computation and optimization. His web-page is now available at http://cse.sysu.edu.cn/content/2477