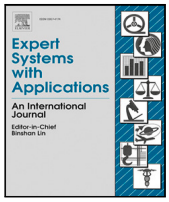
Expert Systems With Applications 251 (2024) 123992

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Fixed-time convergence ZNN model for solving rectangular dynamic full-rank matrices inversion

Bing Zhang [a,b,c,d], Yuhua Zheng [g,*], Shuai Li [e,f], Xinglong Chen [a,b,c,d], Yao Mao [a,b,c,d]

[a] National Key Laboratory of Optical Field Manipulation Science and Technology, Chinese Academy of Sciences, Chengdu, 610209, China
[b] Key Laboratory of Optical Engineering, Chinese Academy of Sciences, Chengdu, 610209, China
[c] Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu, 610209, China
[d] University of Chinese Academy of Science, Beijing, 100049, China
[e] Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu, 90570, Finland
[f] VTT-Technical Research Centre of Finland, Oulu, 90590, Finland
[g] Zhejiang Lab, Hangzhou, 311121, China

ARTICLE INFO

Keywords:
Zeroing neural network
Dynamic matrices Moore–Penrose inversion
Fixed-time convergence
Manipulator-based photoelectric tracking system

ABSTRACT

The Moore–Penrose inverse of dynamic matrices has found widespread application and has garnered significant attention. The zeroing neural network (ZNN) has proven to be an effective solution for computing the Moore–Penrose inverse in dynamic matrices. This paper proposes a novel unified fixed-time ZNN (UFTZNN) model designed to achieve fixed-time convergence and solve both left and right inverse problems using a single model. Theoretical analysis of the convergence and robustness of the UFTZNN model is rigorously presented. Numerical simulations comparing the UFTZNN with existing ZNN models confirm its superiority in addressing left and right inverse problems, convergence time, and robustness. The UFTZNN model is applied to the inverse kinematic tracking problem of a six-degree-of-freedom manipulator-based photoelectric tracking system to demonstrate its potential applications and effectiveness.

1. Introduction

The Moore–Penrose inverse of a dynamic matrix plays a crucial role in solving various problems in the fields of robotics (Kong et al., 2023; Machado & Lopes, 2017), image processing (Chountasis et al., 2010), control engineering (Peter et al., 2012), and numerical optimization (Wei et al., 2019). Despite the development of several numerical methods, such as Newton's iteration (Zhang et al., 2004) and singular value decomposition (Wang et al., 2006), for matrices that remain static over time, these algorithms are not suitable for dynamic matrices. In such cases, the inversion calculation needs to be performed within each sampling period, which may not be feasible for tasks with high sampling rates. To address this issue, various recurrent neural network algorithms have been proposed and studied.

Expanding on the gradient neural network, Zhang et al. (2002) introduced the zeroing neural network (ZNN) to address time-varying problems. By employing a vector cost function instead of a scalar one reduces element-wise errors and tracks time-varying solutions precisely by including the parameter's time derivative. The ZNN model has been extensively studied and applied in various scenarios, such as matrix equality (Xiao, 2015), matrix inequality (Xiao, Song, et al., 2022), Lyapunov equation (Xiao & Liao, 2016), Sylvester equation (Jin et al., 2022), matrix square roots (Sun et al., 2022), and matrix inverses (Dai et al., 2022). In the investigation of matrix inverses, researchers have explored improved ZNN models for computing the Moore–Penrose inverse (Liao & Zhang, 2014; Wu & Zheng, 2020), Drazin inverse (Wang et al., 2017; Zhang et al., 2015), and complex matrices inverse (Xiao et al., 2019, 2020).

The convergence performance and noise tolerance of the ZNN model are crucial factors in evaluating its effectiveness. The original ZNN (OZNN) (Zhang et al., 2011) model uses a linear evolution formula, leading to slow convergence and ineffective noise suppression. Besides, the OZNN model exhibits exponential convergence, approximating the theoretical solution as time approaches infinity. To enhance convergence and noise tolerance, various approaches have been explored. Incorporating the integral function into the ZNN model for the Moore–Penrose inverse has shown strong adaptive capabilities, smoothing out noise (Sun et al., 2020; Xiang et al., 2018). This type of ZNN model is commonly referred to as the integral-enhanced ZNN (IEZNN) model. Jia et al. (2020) and Katsikis et al. (2022) employ a dynamic adjustment

* Corresponding author.
E-mail addresses: zhangbing20@mails.ucas.ac.cn (B. Zhang), zhengyh@zhejianglab.com (Y. Zheng), shuai.li@oulu.fi (S. Li), chenxinglong@ioe.ac.cn (X. Chen), maoyao@ioe.ac.cn (Y. Mao).

https://doi.org/10.1016/j.eswa.2024.123992
Received 30 August 2023; Received in revised form 22 February 2024; Accepted 14 April 2024
Available online 21 April 2024

of the ZNN parameters through a fuzzy function. The ZNN model with variable parameters, adapting the integral coefficient dynamically, is capable of handling noise interference across a broader range (Xiao et al., 2020). However, these methods still exhibit exponential convergence, prompting the use of finite-time theory to achieve faster convergence rates for real-time problems.

To achieve convergence within a finite time frame in the ZNN model for solving the Moore–Penrose inverse, the Li activation function (Guo & Zhang, 2014), sign-power activation function (Chen et al., 2020), and other nonlinear functions play a critical role in its design. Tan et al. (2020) developed a modified ZNN (MZNN) model, which incorporates a noise-resistant and finite-time convergence design through the combination of the Li activation function and integral function. However, a drawback of the finite-time model is its sensitivity to the initial system value, potentially leading to longer convergence times with excessively large initial values. Fixed time and predefined time are specific forms of finite-time convergence that are insensitive to the initial value of the system and are integrated into the design of ZNN models. Jin et al. (2022) and Xiao, He, and Liao (2022) designed ZNN models with fixed time convergence to address Sylvester's equation and linear equations. Predefined time was initially proposed by Li (2018) for designing the ZNN model to solve linear matrix equations. Subsequently, Li et al. (2019) and Dai et al. (2021) introduced several distinct predefined-time ZNN models corresponding to the dynamic matrix square root solution and dynamic Stein matrix equations, respectively. Recently, Li et al. (2024) proposed a strictly predefined time ZNN model for solving linear equations, which features more stringent conditions compared to fixed time. The nonlinearity of the evolution formulas for predefined-time and fixed-time convergence results in their high suppression capability against various forms of noise. However, it is noteworthy that no prior research has examined the utilization of fixed-time stability theory to tackle the dynamic matrix Moore–Penrose inverse problem. Hence, there is considerable significance in investigating a novel fixed-time or predefined-time ZNN model for the matrix Moore–Penrose inverse.

For a full-rank rectangular matrix, it has a left inverse when the number of rows is greater than the number of columns, and a right inverse when the number of columns is greater than the number of rows. Previous research (Liao & Zhang, 2014; Sun et al., 2020; Tan et al., 2020; Xiang et al., 2018; Zhang et al., 2011) has focused on designing models to solve for the left inverse and the right inverse separately, aiming to obtain a unique solution for the matrix inverse. Two distinct analog circuits must be designed to implement the neural network and harness its parallelism. When the left-inverse ZNN model is employed to address the right inverse of the matrix, non-uniqueness arises in the solution. However, in specific scenarios, such as manipulator joint faults or control task switching, this triggers the alternation between the left-inverse solution model and the right-inverse solution model. Currently, there is a lack of research on the ZNN model that can solve the left or right inverses of a matrix using a single model. This has the potential to simplify circuit design and reduce costs.

To overcome the limitations of the existing ZNN models, this paper introduces a novel approach called the unified fixed-time ZNN model (UFTZNN) for dynamic matrix inversion. Unlike the previous ZNN model, the UFTZNN model can solve the left or right inverses of a matrix using a single model. Additionally, it demonstrates the ability to achieve fixed-time convergence, which is a significant contribution of this paper as no such model has been reported before. The primary contributions of this paper are as follows:

1. A new ZNN model is proposed, which is based on fixed-time convergence and is independent of the initial system value. This model is stricter than finite-time convergence, and can solve the Moore–Penrose inverse of a full-rank rectangular matrix without distinguishing between the left and right inverses.

2. The convergence time and convergence boundary of the UFTZNN model are rigorously analyzed and proven in both noise-free and noisy environments.

3. The UFTZNN model is applied to solve for the left and right inverses of a rectangular matrix. Furthermore, an experiment on a manipulator-based photoelectric tracking system is conducted to validate the high accuracy and potential feasibility of the proposed model in real-world environments.

The remainder of the paper is organized as follows: Section 2 provides a review of the necessary background knowledge and several typical ZNN models. It also presents the design process of the UFTZNN model. Section 3 details and proves the fixed-time convergence and robustness of the UFTZNN model. Section 4 presents numerical simulations and experimental verification. Finally, Section 5 concludes the paper.

2. Model description

To facilitate the subsequent discussion and analysis, this section will provide the necessary preliminary knowledge. Four ZNN models for dynamic matrix inversion are presented. The OZNN model, IEZNN model, MZNN model, and UFTZNN model are listed in chronological order.

2.1. Preliminaries

Definition 1 (*Liao & Zhang, 2014; Zhang & Guo, 2015*)**.** Suppose there is a matrix $N(t) \in \mathbb{R}^{m\times n}$. If there exists a matrix $X(t) \in \mathbb{R}^{n\times m}$ that satisfies at least all of the following four Penrose equations:

$$\begin{cases} N(t)X(t)N(t) = N(t) \\ X(t)N(t)X(t) = X(t) \\ (N(t)X(t))^{\mathrm{T}} = N(t)X(t) \\ (X(t)N(t))^{\mathrm{T}} = X(t)N(t) \end{cases}, \tag{1}$$

where superscript T denotes the transpose of a matrix, then the matrix $X(t)$ is called the Moore–Penrose inverse matrix of the $N(t)$, denoted by $N^{+}(t)$.

In general, when considering a rectangular matrix $N(t)$ in the real number field with dimensions $m\times n$, if m is greater than n and the rank of $N(t)$ is equal to n, then the matrix $N(t)$ possesses a left Moore–Penrose inverse. Conversely, if m is less than n and the rank of $N(t)$ is equal to m, then the matrix $N(t)$ possesses a right Moore–Penrose inverse. In this paper, we focus on investigating a full-rank rectangular matrix. Consequently, the following lemma can be formally and unequivocally stated:

Lemma 1 (*Liao & Zhang, 2014; Zhang & Guo, 2015*)**.** *For any full-rank dynamic matrix $N(t) \in \mathbb{R}^{m\times n}$, if $\mathrm{rank}(N(t)) = \min\{m, n\}$, then the unique Moore–Penrose inverse of $N(t)$ can be expressed as:*

$$N^{+}(t) = \begin{cases} N^{\mathrm{T}}(t)(N(t)N^{\mathrm{T}}(t))^{-1} \\ (N^{\mathrm{T}}(t)N(t))^{-1}N^{\mathrm{T}}(t) \end{cases}. \tag{2}$$

Proof. Grouping the first two terms on the left of $N(t)N^{+}(t)N(t) = N(t)$, transposing both sides of the equation, and bringing $(N(t)N^{+}(t))^{\mathrm{T}} = N(t)N^{+}(t)$ into it, we can obtain $N^{\mathrm{T}}(t)N(t)N^{+}(t) = N^{\mathrm{T}}(t)$. Similarly, Grouping the first two terms on the left of $N(t)N^{+}(t)N(t) = N(t)$, transposing both sides of the equation, and bringing $(N^{+}(t)N(t))^{\mathrm{T}} = N^{+}(t)N(t)$ into it, we can obtain $N^{+}(t)N(t)N^{\mathrm{T}}(t) = N^{\mathrm{T}}(t)$. Given that matrix $N(t)$ has full rank, the $N^{\mathrm{T}}(t)N(t)$ or $N(t)N^{\mathrm{T}}(t)$ is nonsingular. Eq. (2) can be derived by solving the linear matrix equation.

Lemma 2 (*Polyakov, 2012*)**.** *Consider the continuous autonomous system $\dot{x}(t) = f(x(t))$ and $f(0) = 0$. Suppose there exists a continuous radially unbounded function $V(x(t))$ that is positive definite and satisfies:*

$$\dot{V}(x(t)) + (v_1 V^{\sigma}(x(t)) + v_2 V^{\varsigma}(x(t)))^{\varrho} \le 0, x(t) \in \mathbb{D} \setminus \{0\},$$

where ν_1, ν_2, σ, ϱ, and ς are all positive real numbers, and $\sigma\varrho < 1$ and $\varsigma\varrho > 1$. The system's origin is globally fixed-time stable. The settling time meets the condition

$$T_r \leq \frac{1}{\nu_1^{\varrho}(1-\sigma\varrho)} + \frac{1}{\nu_2^{\varrho}(\varsigma\varrho-1)}.$$

Proof. The above inequality implies that if $V(x(t)) \leq 1$, then $\dot{V}(x(t)) \leq -\nu_1^{\varrho} V^{\sigma\varrho}(x(t))$, and if $V(x(t)) > 1$, then $\dot{V}(x(t)) \leq -\nu_2^{\varrho} V^{\varsigma\varrho}(x(t))$. Consequently, for any $x(t)$ such that $V(x(0)) > 1$, the latter inequality ensures $V(x(t)) \leq 1$ for $t \geq \frac{1}{\nu_2^{\varrho}(\varsigma\varrho-1)}$. Similarly, for any $x(t)$ such that $V(x(t_0)) \leq 1$, we obtain $V(x(t)) = 0$ for $t \geq t_0 + \frac{1}{\nu_1^{\varrho}(1-\sigma\varrho)}$. Consequently, $V(x(t)) = 0$ holds for all $t \geq \frac{1}{\nu_2^{\varrho}(\varsigma\varrho-1)} + \frac{1}{\nu_1^{\varrho}(1-\sigma\varrho)}$ and for any solution $x(t)$ of $\dot{x}(t) = f(x(t))$.

2.2. OZNN model

With reference to Eq. (2), the following relationship is established:

$$\begin{aligned} N^{+}(t)N(t)N^{\mathrm{T}}(t) &= N^{\mathrm{T}}(t) \\ N^{\mathrm{T}}(t)N(t)N^{+}(t) &= N^{\mathrm{T}}(t) \end{aligned}. \tag{3}$$

To solve the inverse problem of dynamic rectangular matrices, the error matrix $E_r(t)$ and $E_l(t)$ for the right and left inverses are defined as follows:

$$\begin{aligned} E_r(t) &= X(t)N(t)N^{\mathrm{T}}(t) - N^{\mathrm{T}}(t) \in \mathbb{R}^{n\times m} \\ E_l(t) &= N^{\mathrm{T}}(t)N(t)X(t) - N^{\mathrm{T}}(t) \in \mathbb{R}^{n\times m} \end{aligned}. \tag{4}$$

Taking the derivative of Eq. (4) results in:

$$\begin{aligned} \dot{E}_r(t) &= \dot{X}(t)N(t)N^{\mathrm{T}}(t) + X(t)(\dot{N}(t)N^{\mathrm{T}}(t) + N(t)\dot{N}^{\mathrm{T}}(t)) - \dot{N}^{\mathrm{T}} \\ \dot{E}_l(t) &= N^{\mathrm{T}}(t)N(t)\dot{X}(t) + (\dot{N}^{\mathrm{T}}(t)N(t) + N^{\mathrm{T}}(t)\dot{N}(t))X(t) - \dot{N}^{\mathrm{T}} \end{aligned}. \tag{5}$$

During the design of the OZNN model, the evolution formula below is used to ensure that $E_r(t)$ and $E_l(t)$ reach zero (Zhang et al., 2011):

$$\dot{E}_k(t) = -\alpha E_k(t), \tag{6}$$

where α is a positive constant. The subscript k represents r and l, respectively. Therefore, the OZNN model for handling the right (OZNN-R) and left (OZNN-L) Moore–Penrose inverses of dynamic rectangular matrices, respectively, can be described as:

$$\begin{aligned} \dot{X}(t)N(t)N^{\mathrm{T}}(t) &= \dot{N}^{\mathrm{T}}(t) - X(t)(\dot{N}(t)N^{\mathrm{T}}(t) + N(t)\dot{N}^{\mathrm{T}}(t)) \\ &\quad - \alpha(X(t)N(t)N^{\mathrm{T}}(t) - N^{\mathrm{T}}(t)), \end{aligned} \tag{7}$$

$$\begin{aligned} N^{\mathrm{T}}(t)N(t)\dot{X}(t) &= \dot{N}^{\mathrm{T}}(t) - (\dot{N}^{\mathrm{T}}(t)N(t) + N^{\mathrm{T}}(t)\dot{N}(t))X(t) \\ &\quad - \alpha(N^{\mathrm{T}}(t)N(t)X(t) - N^{\mathrm{T}}(t)). \end{aligned} \tag{8}$$

2.3. IEZNN model

To enhance the noise suppression ability of the OZNN model, the evolution formula that incorporates the concept of proportional–integral–derivative control is proposed by Sun et al. (2020):

$$\dot{E}_k(t) = -\alpha E_k(t) - \beta \int_0^t E_k(\tau)\, d\tau, \tag{9}$$

Therefore, the IEZNN model for processing the right (IEZNN-R) and left (IEZNN-L) Moore–Penrose inverses of dynamic rectangular matrices, respectively, can be described as follows:

$$\begin{aligned} \dot{X}(t)N(t)N^{\mathrm{T}}(t) &= \dot{N}^{\mathrm{T}}(t) - X(t)(\dot{N}(t)N^{\mathrm{T}}(t) + N(t)\dot{N}^{\mathrm{T}}(t)) \\ &\quad - \alpha(X(t)N(t)N^{\mathrm{T}}(t) - N^{\mathrm{T}}(t)) \\ &\quad - \beta \int_0^t (X(\tau)N(\tau)N^{\mathrm{T}}(\tau) - N^{\mathrm{T}}(\tau))\, d\tau, \end{aligned} \tag{10}$$

$$\begin{aligned} N^{\mathrm{T}}(t)N(t)\dot{X}(t) &= \dot{N}^{\mathrm{T}}(t) - (\dot{N}^{\mathrm{T}}(t)N(t) + N^{\mathrm{T}}(t)\dot{N}(t))X(t) \\ &\quad - \alpha(N^{\mathrm{T}}(t)N(t)X(t) - N^{\mathrm{T}}(t)) \\ &\quad - \beta \int_0^t (N^{\mathrm{T}}(\tau)N(\tau)X(\tau) - N^{\mathrm{T}}(\tau))\, d\tau. \end{aligned} \tag{11}$$

2.4. MZNN model

Whether it is the OZNN model or the IEZNN model, the error can only converge to zero as time tends to infinity. They all require infinite time to obtain the exact solution. To consider both noise tolerance and convergence rate, the following evolution formula is used (Tan et al., 2020):

$$\begin{aligned} \epsilon_k(t) &= E_i(t) + \beta \int_0^t \psi(E_k(\tau))\, d\tau \\ \dot{\epsilon}_k(t) &= -\alpha\psi(\epsilon_k(t)) \end{aligned}, \tag{12}$$

where $\psi(*) = (\mathrm{sgn}(*)\,|*|)^{\mu} + (\mathrm{sgn}(*)\,|*|)^{1/\mu}$, $\mu \in (0,1)$. Therefore, the MZNN model, which is used to handle the right (MZNN-R) and left (MZNN-L) inverses of the dynamic rectangular matrices, respectively, can be described as (Trim t for space efficiency):

$$\begin{aligned} \dot{X}NN^{\mathrm{T}} &= -\alpha\psi(XNN^{\mathrm{T}} - N^{\mathrm{T}} + \beta \int_0^t \psi(XNN^{\mathrm{T}} - N^{\mathrm{T}})\, d\tau) \\ &\quad - \beta\psi(XNN^{\mathrm{T}} - N^{\mathrm{T}}) + \dot{N}^{\mathrm{T}} - X(\dot{N}N^{\mathrm{T}} + N\dot{N}^{\mathrm{T}}), \end{aligned} \tag{13}$$

$$\begin{aligned} N^{\mathrm{T}}N\dot{X} &= -\alpha\psi(N^{\mathrm{T}}NX - N^{\mathrm{T}} + \beta \int_0^t \psi(N^{\mathrm{T}}NX - N^{\mathrm{T}})\, d\tau) \\ &\quad - \beta\psi(N^{\mathrm{T}}NX - N^{\mathrm{T}}) + \dot{N}^{\mathrm{T}} - (\dot{N}^{\mathrm{T}}N + N^{\mathrm{T}}\dot{N})X. \end{aligned} \tag{14}$$

2.5. UFTZNN model

Despite previous research on noise suppression and convergence time in ZNN models, these models necessitate the utilization of distinct models to solve the left and right inverses of a given matrix. Furthermore, even if finite-time convergence is achieved, the convergence time is affected by the initial value of the system. To address these limitations and solve the left or right inverses of a matrix using a single model while ensuring that the convergence time remains unaffected by the initial value of the system, we propose the UFTZNN model. This model is designed based on the consideration of dynamic quadratic programming problem. It can be stated as follows:

$$\begin{aligned} &\text{minimize } \mathrm{tr}(X^{\mathrm{T}}(t)X(t)/2) \\ &\text{subject to } : X(t)N(t)N^{\mathrm{T}}(t) - N^{\mathrm{T}}(t) = 0 \end{aligned}, \tag{15}$$

where $\mathrm{tr}(\cdot)$ represents the trace of a matrix. By using the related Lagrangian function $L(X(t), \Gamma(t), t) = \mathrm{tr}(X^{\mathrm{T}}(t)X(t)/2) + \mathrm{tr}[\Gamma^{\mathrm{T}}(t)(X(t)N(t)N^{\mathrm{T}}(t) - N^{\mathrm{T}}(t))]$ with $\Gamma(t) \in \mathbb{R}^{n\times m}$ denoting the Lagrange-multiplier matrix. Solving the Moore–Penrose matrix can be achieved by zeroing the following equation:

$$\begin{cases} \frac{\partial L(X(t),\Gamma(t),t)}{\partial X} = X(t) + \Gamma(t)N(t)N^{\mathrm{T}}(t) = 0 \\ \frac{\partial L(X(t),\Gamma(t),t)}{\partial \Gamma} = X(t)N(t)N^{\mathrm{T}}(t) - N^{\mathrm{T}}(t) = 0 \end{cases}, \tag{16}$$

The above equations could be further written as:

$$H(t)Y(t) - P(t) = 0, \tag{17}$$

where

$$H(t) = \begin{bmatrix} I & N(t)N^{\mathrm{T}}(t) \\ N(t)N^{\mathrm{T}}(t) & 0 \end{bmatrix} \in \mathbb{R}^{2m\times 2m},$$

$$Y(t) = \begin{bmatrix} X^{\mathrm{T}}(t) \\ \Gamma^{\mathrm{T}}(t) \end{bmatrix} \in \mathbb{R}^{2m\times n}, P(t) = \begin{bmatrix} 0 \\ N(t) \end{bmatrix} \in \mathbb{R}^{2m\times n}.$$

Furthermore, for enhanced comprehension and comparative analysis, it is acknowledged that the dynamic theoretical solution can be expressed as:

$$Y^{*}(t) = \begin{bmatrix} N^{+}(t) & \Gamma^{*}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2m\times n}. \tag{18}$$

To enhance the explication and surveillance of the evolving process of the Moore–Penrose inverse solution for dynamic matrices, the dynamic error matrix is formally introduced as follows:

$$E(t) = H(t)Y(t) - P(t) \in \mathbb{R}^{2m\times n}. \tag{19}$$

To achieve fixed-time convergence, we utilize the following evolution formula:

$$\dot{E}(t) = -\alpha(E^{h/g} + \beta E^{q/p}), \tag{20}$$

where $\alpha > 0$, $\beta > 0$, $p > q > 0$, $h > g > 0$, and p, q, g and h are all positive odd numbers. By differentiating Eq. (19) and combining Eq. (20) yields the UFTZNN model:

$$\begin{aligned} H(t)\dot{Y}(t) = {} & \dot{P}(t) - \alpha(H(t)Y(t) - P(t))^{h/g} \\ & - \dot{H}(t)Y(t) - \alpha\beta(H(t)Y(t) - P(t))^{q/p}. \end{aligned} \tag{21}$$

When considering the noise in the solution process, the UFTZNN model can be described as follows:

$$\begin{aligned} H(t)\dot{Y}(t) = {} & -\dot{H}(t)Y(t) - \alpha(H(t)Y(t) - P(t))^{h/g} \\ & + \dot{P}(t) - \alpha\beta(H(t)Y(t) - P(t))^{q/p} + \eta(t), \end{aligned} \tag{22}$$

where $\eta(t) \in \mathbb{R}^{2m\times n}$ represents the equivalent measurement noise and disturbance in matrix form. The given model, denoted as (22), is expressed as an implicit equation. To facilitate the modeling process, it is reformulated as:

$$\begin{aligned} \dot{Y}(t) = {} & (I - H(t))\dot{Y}(t) - \dot{H}(t)Y(t) + \dot{P}(t) + \eta(t) \\ & - \alpha(H(t)Y(t) - P(t))^{h/g} - \alpha\beta(H(t)Y(t) - P(t))^{q/p}. \end{aligned} \tag{23}$$

Remark 1. The original problem of inversion is converted into a quadratic optimization problem. This model is applicable to both left and right Moore–Penrose inversion. The solution for the Moore–Penrose inverse using the least squares method can be obtained when the number of rows (m) is less than the number of columns (n). Conversely, the solution for the least norm can be found when m is greater than n. The proposed model is versatile and applicable, as it can be employed irrespective of the dimensions of m and n.

3. Theoretical analysis

In this section, we provide the convergence and robustness analysis of the UFTZNN model.

3.1. Convergence of UFTZNN

Theorem 1. *For a dynamic quadratic program problem (15), under the consideration of such an evolution formula (20), the error between the theoretical solution $Y^*(t)$ and the numerical solution $Y(t)$ used in the UFTZNN model (21) is defined as $\tilde{Y}(t) = Y(t) - Y^*(t)$. Whether the matrix $N(t)$ has full-row rank or full-column rank, the UFTZNN model will ensure that the matrix $Y(t)$ globally converges from any initial value $Y(0)$ to the theoretical solution $Y^*(t)$ in a fixed time:*

$$T_r \le \frac{g}{\alpha(h-g)}2^{\frac{g-h}{2g}} + \frac{p}{\alpha\beta(p-q)}2^{\frac{p-q}{2p}}.$$

In addition, the first $n \times m$ elements of $Y^{\mathrm{T}}(t)$ converge to the theoretical solution $N^{+}(t)$ of the Moore–Penrose inverse of matrix $N(t)$ in a fixed time T_r.

Proof. Bringing $Y(t) = \tilde{Y}(t) + Y^*(t)$ into Eq. (19) yields:

$$E(t) = H(t)\tilde{Y}(t). \tag{24}$$

The time differential on both sides of $H(t)Y^*(t) = P(t)$ can be derived:

$$\dot{H}(t)Y^*(t) + H(t)\dot{Y}^*(t) = \dot{P}(t). \tag{25}$$

The difference between (21) and (25) can be obtained:

$$\dot{H}(t)\tilde{Y}(t) + H(t)\dot{\tilde{Y}}(t) = -\alpha(H(t)Y(t) - P(t))^{h/g} - \alpha\beta(H(t)Y(t) - P(t))^{q/p}. \tag{26}$$

That is, $\dot{E}(t) = -\alpha E^{h/g}(t) - \alpha\beta E^{q/p}(t)$. Let $e_{ij}(t)$ be the ij-th element of matrix $E(t)$. This equation represents a compact form of $2m \times n$ subsystems:

$$\dot{e}_{ij}(t) = -\alpha(e_{ij}^{h/g}(t) + \beta e_{ij}^{q/p}(t)), \ \forall i = 1, 2, \dots, 2m \text{ and } j = 1, 2, \dots, n.$$

Considering a positive definite Lyapunov function $V(e_{ij}(t)) = e_{ij}^2(t)/2$, its differential as follows:

$$\dot{V}(e_{ij}(t)) = e_{ij}(t)\dot{e}_{ij}(t) = e_{ij}(t)(-\alpha e_{ij}^{h/g}(t) - \alpha\beta e_{ij}^{q/p}(t)). \tag{27}$$

Since $\alpha, \beta > 0$, g, h, p, and q are positive odd numbers, it is easy to get that $\dot{V}(e_{ij}(t)) < 0$. Based on the principles of Lyapunov theory, it can be deduced that the ij-th component of the error matrix $E(t)$ exhibits global convergence towards zero. Additionally, for any $i \in [1, 2\ m]$ and $j \in [1, n]$, the individual elements of $E(t)$ also converge to zero. Consequently, it can be inferred that $E(t)$ globally converges to zero. Eq. (27) can be reformulated as follows:

$$\dot{V}(e_{ij}(t)) = -\alpha 2^{\frac{g+h}{2g}}\left(\frac{e_{ij}^2(t)}{2}\right)^{\frac{g+h}{2g}} - \alpha\beta 2^{\frac{p+q}{2p}}\left(\frac{e_{ij}^2(t)}{2}\right)^{\frac{p+q}{2p}}.$$

Let $\sigma = \frac{p+q}{2p}$, $\varsigma = \frac{g+h}{2g}$, $\nu_1 = \alpha\beta 2^{\sigma}$, and $\nu_2 = \alpha 2^{\varsigma}$. It is easy to get $1/2 < \sigma < 1, \varsigma > 1$. According to Lemma 2, the settling time from $e_{ij}(0) \neq 0$ to $e_{ij}(t_{ij}) = 0$ is

$$t_{ij} = \frac{g}{\alpha(h-g)}2^{\frac{g-h}{2g}} + \frac{p}{\alpha\beta(p-q)}2^{\frac{p-q}{2p}}.$$

The time required for all elements in $E(t) \neq 0$ to converge to zero is $T_r \le \max\{t_{ij} | i = 1, 2 \cdots, 2\ m,\ j = 1, 2 \cdots, n\}$.

This completes the proof.

Remark 2. The method proposed achieves fixed-time convergence. The convergence time is only related to the design parameter (p, q, g, h, α, β) and is independent of the distance from the initial state of the system to the equilibrium point. Moreover, the term $E^{h/g}$ primarily determines the convergence rate far from the equilibrium point, and the convergence rate near the equilibrium point mainly depends on the term $E^{q/p}$.

3.2. Robustness of UFTZNN

Theorem 2. *In addition to Theorem 1, suppose there exists a norm-bounded noise $\eta(t)$, and the inequality $\|\eta(t)\| \le \|\eta^*\|$ and $|\eta_{ij}(t)| \le \eta_{ij}^*$ holds, where η_{ij} stands for the ij-th element of $\eta(t)$ and superscript $*$ denotes its upper bound. The neural solution $Y(t)$ of model (22) will converge to the regions of the theoretical solution $Y^*(t)$ as*

$$\|E(t)\| \le \min\left\{\left(\frac{\|\eta^*\|}{(1-\rho)\alpha}\right)^{g/h}, \left(\frac{\|\eta^*\|}{(1-\rho)\alpha\beta}\right)^{p/q}\right\}$$

within fixed time

$$T_r \le \max\left\{\begin{array}{l} \frac{g}{\alpha\rho(h-g)}2^{\frac{g-h}{2g}} + \frac{p}{\alpha\beta(p-q)}2^{\frac{p-q}{2p}}, \\ \frac{g}{\alpha(h-g)}2^{\frac{g-h}{2g}} + \frac{p}{\alpha\beta\rho(p-q)}2^{\frac{p-q}{2p}} \end{array}\right\},$$

where $0 < \rho < 1$ is a constant.

Proof. In the presence of the noise, $\eta_{ij} \neq 0$. Similarly, we can obtain:

$$\begin{aligned} \dot{V}(e_{ij}(t)) = {} & e_{ij}(t)(-\alpha e_{ij}^{h/g}(t) - \alpha\beta e_{ij}^{q/p}(t)) + e_{ij}(t)\eta_{ij}(t) \\ \le {} & e_{ij}(t)(-\alpha e_{ij}^{h/g}(t) - \alpha\beta e_{ij}^{q/p}(t)) + e_{ij}(t)\eta_{ij}^* \\ = {} & -\alpha 2^{\frac{g+h}{2g}}\left(\frac{e_{ij}^2(t)}{2}\right)^{\frac{g+h}{2g}} - \alpha\beta 2^{\frac{p+q}{2p}}\left(\frac{e_{ij}^2(t)}{2}\right)^{\frac{p+q}{2p}} + e_{ij}(t)\eta_{ij}^* \end{aligned}. \tag{28}$$

Let $\Delta = \alpha e_{ij}^{h/g}(t) + \alpha\beta e_{ij}^{q/p}(t)$. Eq. (28) can be restated as:

$$\dot{V}(e_{ij}(t)) \le -e_{ij}(t)(\Delta + \eta_{ij}^*).$$

Given that p, q, g, and h are odd numbers, it can be inferred that both Δ and $e_{ij}(t)$ share the same sign. The stability of the system in the sense of Lyapunov is guaranteed as long as $\Delta > \eta_{ij}^*$ ensures $\dot{V}(e_{ij}(t)) < 0$. However, when $\Delta < \eta_{ij}^*$, $\dot{V}(e_{ij}(t)) > 0$ and the system cannot converge to zero. As Δ has a large value when it is far from zero and decreases as it approaches zero, the system first satisfies $\Delta > \eta_{ij}^*$ when the error is far from zero, but then $\Delta < \eta_{ij}^*$ occurs as the error gradually approaches

zero. Consequently, the system can only converge to one boundary in the presence of noise, and the size of this boundary is influenced by the magnitudes of Δ and η_{ij}^*.

There exists a scalar $0 < \rho < 1$ such that above inequality (28) can be expressed as:

$$\dot{V}(e_{ij}(t)) \leq -\rho\nu_2 V^{\varsigma}(e_{ij}(t)) - \nu_1 V^{\sigma}(e_{ij}(t)) - (1-\rho)\nu_2 V^{\varsigma}(e_{ij}(t)) + e_{ij}(t)\eta_{ij}^*, \quad (29)$$

$$\dot{V}(e_{ij}(t)) \leq -\nu_2 V^{\varsigma}(e_{ij}(t)) - \rho\nu_1 V^{\sigma}(e_{ij}(t)) - (1-\rho)\nu_1 V^{\sigma}(e_{ij}(t)) + e_{ij}(t)\eta_{ij}^*. \quad (30)$$

According to Lemma 2, for Eq. (29) , $e_{ij}(t)$ will be driven into the region

$$\lim_{t\to T_r} e_{ij}(t) \in \left\{ V^{\varsigma}(e_{ij}(t)) \leq \frac{e_{ij}(t)\eta_{ij}^*}{(1-\rho)\nu_2} \right\} \quad (31)$$

in fixed time

$$T_r \leq \frac{1}{\rho\nu_2(\varsigma-1)} + \frac{1}{\nu_1(1-\sigma)}. \quad (32)$$

For Eq. (30), $e_{ij}(t)$ will be forced to converge within the specified region

$$\lim_{t\to T_r} e_{ij}(t) \in \left\{ V^{\sigma}(e_{ij}(t)) \leq \frac{e_{ij}(t)\eta_{ij}^*}{(1-\rho)\nu_1} \right\} \quad (33)$$

in fixed time

$$T_r \leq \frac{1}{\nu_2(\varsigma-1)} + \frac{1}{\rho\nu_1(1-\sigma)}. \quad (34)$$

Thus, the system's origin is practically fixed-time stable, with the residual set for the system's solution defined as follows:

$$\lim_{t\to T_r} e_{ij}(t) \in \left\{ V(x) \leq \min\left\{ \left(\frac{e_{ij}(t)\eta_{ij}^*}{(1-\rho)\nu_2}\right)^{1/\varsigma}, \left(\frac{e_{ij}(t)\eta_{ij}^*}{(1-\rho)\nu_1}\right)^{1/\sigma} \right\} \right\}. \quad (35)$$

The setting time is bounded as:

$$T_r \leq \max\left\{ \frac{1}{\rho\nu_2(\varsigma-1)} + \frac{1}{\nu_1(1-\sigma)}, \frac{1}{\nu_2(\varsigma-1)} + \frac{1}{\rho\nu_1(1-\sigma)} \right\}. \quad (36)$$

Furthermore, when $e_{ij}(0) \neq 0$ reach the region

$$|e_{ij}(t)| \leq \min\left\{ \left(\frac{\eta_{ij}^*}{(1-\rho)\alpha}\right)^{g/h}, \left(\frac{\eta_{ij}^*}{(1-\rho)\alpha\beta}\right)^{p/q} \right\}, \quad (37)$$

the time required for all elements in $E(t) \neq 0$ is

$$T_r \leq \max\left\{ \begin{matrix} \frac{g}{\alpha\rho(h-g)} 2^{\frac{g-h}{2g}} + \frac{p}{\alpha\beta(p-q)} 2^{\frac{p-q}{2p}}, \\ \frac{g}{\alpha(h-g)} 2^{\frac{g-h}{2g}} + \frac{p}{\alpha\beta\rho(p-q)} 2^{\frac{p-q}{2p}} \end{matrix} \right\}. \quad (38)$$

This completes the proof.

Remark 3. The settling time of a system is independent of the initial error value, but it is influenced by the presence of noise. In the absence of noise, the selection of appropriate parameters for the UFTZNN model is simplified, and it is easier to guarantee that the derivative of the Lyapunov function is negative. However, in the presence of noise, it becomes necessary to choose larger values for parameters such as α and β to ensure stability and convergence. It is typically required that the value of the evolutionary formula exceeds the magnitude of the noise. Additionally, the presence of noise affects the boundary of the region. A lower level of noise leads to a smaller final convergence boundary with the same set of parameters. Consequently, larger parameter values are needed when aiming for a smaller expected final region boundary.

4. Performance verification

By addressing the issue of determining the Moore–Penrose inverse of a matrix, this study conducts a comparative analysis of the OZNN model, IEZNN model, MZNN model, and UFTZNN model. The effectiveness of the UFTZNN model has been empirically demonstrated. The effectiveness of the UFTZNN model is further substantiated through the utilization of illustrative examples and practical experiments conducted within the context of a manipulator-based photoelectric tracking system.

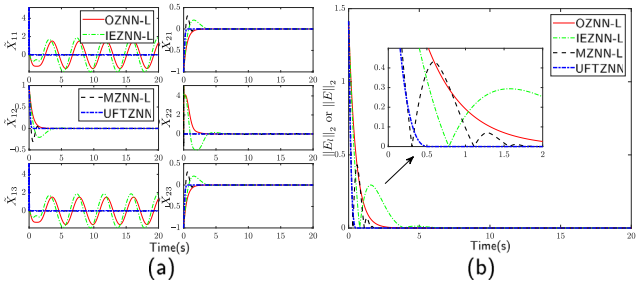


Fig. 1. The simulation results of solving the left Moore–Penrose inverse using the OZNN-L, IEZNN-L, MZNN-L, and UFTZNN models when $X(0) = 0$. (a) The dynamic trajectory of the residual element errors $\tilde{X}_{ij}(t)$. (b) The dynamic trajectory of the residual error matrix norm $\|E_l(t)\|$ or $\|E(t)\|$.

Table 1
Convergence time of the error matrix norm less than 10^{-3} under different initial values.

ZNN models	OZNN-L	IEZNN-L	MZNN-L	UFTZNN
$\Delta N = 0.2$	3.396	6.652	1.397	0.356
$\Delta N = 0.8$	4.011	8.643	2.075	0.576
$\Delta N = 10$	5.215	9.856	2.772	1.127
$\Delta N = 40$	5.987	12.161	2.922	1.457
$\Delta N = 100$	6.510	12.632	2.991	1.666
$\Delta N = 1000$	7.649	15.282	3.028	2.123

Table 2
The root mean square value of the error matrix norm for solving the left inverse under various noise conditions.

ZNN models	OZNN-L	IEZNN-L	MZNN-L	UFTZNN
Fig. 3(a)	0.3142	0.3154	0.2052	0.1854
Fig. 3(b)	0.3242	0.3259	0.2093	0.1874
Fig. 3(c)	0.4993	0.4711	0.3042	0.2199
Fig. 3(d)	0.3371	0.3091	0.2102	0.1882

4.1. Illustrative example

4.1.1. Left Moore–Penrose inverse

To assess the model's capacity to compute the left inverse of a matrix, we examine a time-varying coefficient column full-rank matrix, denoted as (Sun et al., 2020):

$$N(t) = \begin{bmatrix} \sin 1.5t & \cos 1.5t \\ -\cos 1.5t & \sin 1.5t \\ \sin 1.5t & \cos 1.5t \end{bmatrix},$$

and its theory resolution of the left Moore–Penrose inverse $N^+(t)$ can be generalized as:

$$N^+(t) = \begin{bmatrix} 0.5\sin 1.5t & -\cos 1.5t & 0.5\sin 1.5t \\ 0.5\cos 1.5t & \sin 1.5t & 0.5\cos 1.5t \end{bmatrix}.$$

In the simulation, the model parameters are taken as $\alpha = \beta = 2$, $\mu = q/p = 3/5$, and $h/g = 17/15$. Without loss of generality, let us assume the initial value $X(0) = 0$ and $Y(0) = 0$. The left Moore–Penrose inverse problem is solved by using the OZNN-L (8), IEZNN-L (11), MZNN-L (14), and UFTZNN (21) model.

The results are presented in Fig. 1. Fig. 1(a) illustrates the discrepancy between the state solution $X(t)$ and the theoretical solution $N^+(t)$ for each element, denoted as $\tilde{X}_{ij}(t) = X_{ij}(t) - N_{ij}^+(t)$. Fig. 1(b) displays the dynamic trajectory of the model error matrix norm $\|E_l(t)\|$ or $\|E(t)\|$. The errors of various elements in the alternative methods tend to converge to zero, with the exception of $\tilde{X}_{11}$ and $\tilde{X}_{31}$ in the OZNN-L and IEZNN-L models. This discrepancy is primarily influenced by the initial distance between $X(0)$ and $N^+(0)$, which will be further discussed. Notably, the error matrix norm of the model eventually converges to zero and remains stable. The IEZNN-L and MZNN-L models exhibit an overshoot phenomenon due to the impact of the integral

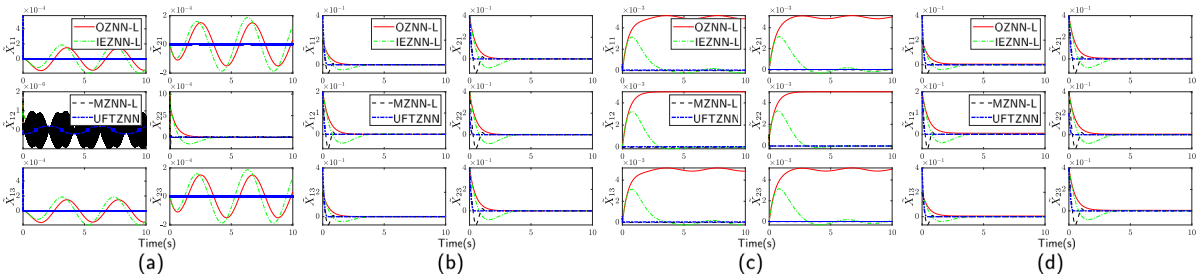


Fig. 2. The trajectory of the residual element errors in solving the left Moore–Penrose inverse using the OZNN-L, IEZNN-L, MZNN-L, and UFTZNN models with different initial values. (a) $X(0) = N^{+}(0)$, $\eta_{ij}(t) = 0$. (b) $X(0) = N^{+}(0) + 0.2$, $\eta_{ij}(t) = 0$. (c) $X(0) = N^{+}(0)$, $\eta_{ij}(t) = 0.01$. (d) $X(0) = N^{+}(0) + 0.2$, $\eta_{ij}(t) = 0.01$.

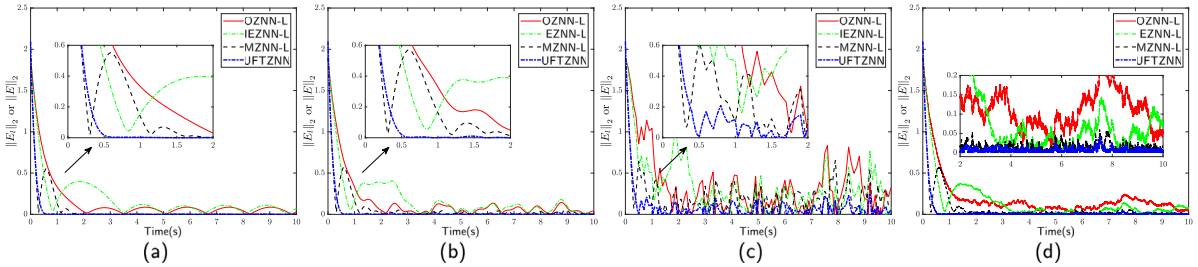


Fig. 3. The trajectory of the error matrix norm in solving the left Moore–Penrose inverse using the OZNN-L, IEZNN-L, MZNN-L, and UFTZNN models under different noise levels. (a) With single-frequency sinusoidal noise $\eta_{ij}(t) = 0.1 \sin 2t$. (b) With composite-frequency sinusoidal noise $\eta_{ij}(t) = 0.1 \sin 2t + 0.1 \sin 5t + 0.1 \sin 10t$. (c) With 10 Hz sampled random noise with a mean of 0.1 and a variance of 1. (d) With 1000 Hz sampled random noise with a mean of 0.1 and a variance of 1.

Table 3
The percentage improvement of the root mean square error of the UFTZNN model compared to other models for solving the left inverse.

Other models	OZNN-L(%)	IEZNN-L(%)	MZNN-L(%)
UFTZNN	40.98	41.21	**9.63**
UFTZNN	42.18	42.49	10.46
UFTZNN	**55.96**	53.34	27.72
UFTZNN	44.17	39.11	10.46

term. The convergence time for the error matrix norms of the OZNN-L, IEZNN-L, MZNN-L, and UFTZNN models to reach 10^{-3} and remain stable were recorded as 3.664 s, 6.603 s, 1.781 s, and 0.49 s, respectively. In comparison to the other three models, the UFTZNN model demonstrated an improvement in convergence rate of 86.63%, 92.58%, and 72.49%, respectively.

To examine the impact of the initial value $X(0)$ on the solution process, we select $X(0) = N^{+}(0) + \Delta N$, where ΔN takes on values of 0 and 0.2 for comparison. Fig. 2 illustrates the dynamic trajectory of the solution error $\tilde{X}_{ij}(t)$ for each element in the absence of noise and constant noise. From Fig. 2(a), when the initial value is close to the actual initial value, the model exhibits varying degrees of oscillation during the solution process. This is because, in this example, the real solution of the matrix has zero elements at $t = 0$, resulting in an oscillation of certain elements in Fig. 1(a). Fig. 2(b) demonstrates that when the initial value deviates slightly from the actual value, the convergence of the solution improves. Fig. 2(c) reveals that when the initial value is in close proximity to the actual value, both the OZNN-L and IEZNN-L models fail to complete the solution task. Noise can cause the OZNN-L model to fail and induce oscillation in the IEZNN-L model. However, when the distance between the initial value and the actual value increases, as shown in Fig. 2(d), the result still converges. Therefore, it can be concluded that the UFTZNN model is only minimally affected by the initial value, indicating excellent performance of the UFTZNN model. It is important to note that in practical applications, noise is inevitable, and obtaining the true initial value for the inverse solution of a matrix can be challenging. Using a zero initial value can pose risks in specific applications.

Based on Theorem 1, the UFTZNN model has a theoretical convergence time of $T_r \leq 4.2986$ s. To demonstrate the benefits of fixed-time convergence, the convergence time of several models (OZNN-L, IEZNN-L, MZNN-L, and UFTZNN) is calculated under different initial values. The convergence time is determined by the residual error matrix norm being less than 10^{-3} and the holding time. The results are presented in Table 1. The UFTZNN model exhibits a small convergence time across various initial values, indicating its superior convergence rate. Additionally, as the initial error increases, the convergence time increases, albeit remaining below the theoretical calculation time. This suggests that the theoretical calculation of convergence time is overly conservative, a finding supported by Polyakov (2012). Furthermore, in comparison to the OZNN-L and IEZNN-L models, the UFTZNN model demonstrates a smaller variation in convergence time. It should be noted that MZNN exhibits a degree of fixed-time convergence.

To further investigate the anti-interference capabilities of the UFTZNN model, this study conducted additional anti-interference experiments on four ZNN models using time-varying and random noise. To maintain consistency, all initial values $X(0)$ in the subsequent experiments were set to 0.5. Fig. 3 illustrates the error matrix norm trajectories of the four models when subjected to single-frequency sinusoidal noise, composite-frequency sinusoidal noise, and random noise with varying sampling frequencies. The error matrix norm of the OZNN and IEZNN models deviates significantly from the equilibrium point. Conversely, the UFTZNN model consistently reaches a minimum error matrix norm value regardless of the noise condition. Additionally, the root mean square value of the error matrix norm depicted in Fig. 3 is recorded in Table 2. Moreover, Table 3 presents the percentage improvement of the root mean square value achieved by the UFTZNN model in comparison to the other three models. The UFTZNN model outperforms the other models, exhibiting a minimum improvement of 9.63% and a maximum improvement of 55.96%. Thus, the superiority of the UFTZNN model in handling noisy conditions is substantiated.

4.1.2. Right Moore–Penrose inverse

To assess the model's capacity to compute the right inverse of a matrix, we examine a row full-rank matrix with a time-dependent

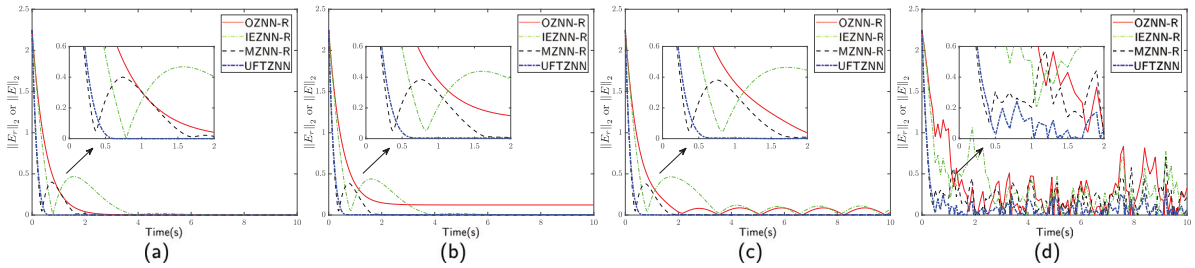


Fig. 4. The trajectory of the error matrix norm in solving the right Moore–Penrose inverse using the OZNN-R, IEZNN-R, MZNN-R, and UFTZNN models under different noise levels. (a) With no noise. (b) With constant noise $\eta_{ij}(t) = 0.1$. (c) With sinusoidal noise $\eta_{ij}(t) = 0.1 \sin 2t$. (d) With 10 Hz sampled random noise with a mean of 0.1 and a variance of 1.

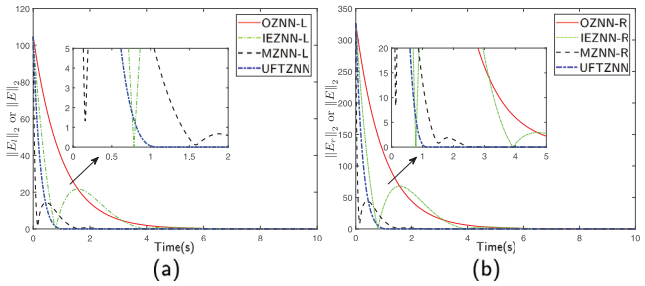


Fig. 5. The trajectory of the error matrix norm in solving the high-dimensional matrix Moore–Penrose inverse. (a) The left inverse is obtained using the OZNN-L, IEZNN-L, MZNN-L, and UFTZNN models. (b) The right inverse is obtained using the OZNN-R, IEZNN-R, MZNN-R, and UFTZNN models.

coefficient. This matrix can be defined as follows (Sun et al., 2020; Tan et al., 2020):

$$N(t) = \begin{bmatrix} \sin 1.5t & \cos 1.5t & -\sin 1.5t \\ -\cos 1.5t & \sin 1.5t & \cos 1.5t \end{bmatrix},$$

and its theory resolution of the right Moore–Penrose inverse $N^{+}(t)$ can be generalized as:

$$N^{+}(t) = \begin{bmatrix} 0.5\sin 1.5t & -0.5\cos 1.5t \\ \cos 1.5t & \sin 1.5t \\ -0.5\sin 1.5t & 0.5\cos 1.5t \end{bmatrix}.$$

The solution to the right Moore–Penrose inverse is obtained using the OZNN-R, IEZNN-R, MZNN-R, and UFTZNN models, as described in Eqs. (7), (10), (13), and (21), respectively, at different levels of noise. The trajectories of the error matrix norm are depicted in Fig. 4. The UFTZNN model exhibits the fastest rate of convergence regardless of the level of noise. Furthermore, it yields results that are closest to the equilibrium point, even in the presence of random noise. The convergent times are 3.907 s, 6.764 s, 2.342 s, and 0.609 s for the error matrix norms of the OZNN-R, IEZNN-R, MZNN-R, and UFTZNN models to attain and remain stable at a value of 10^{-3}, respectively. The UFTZNN model has a noticeably greater convergence rate when compared to the other three models, with improvements of 84.41%, 91.00%, and 74.00%, respectively. The error matrix norm's root mean square value, as depicted in Fig. 4, is listed in Table 4. The percentage improvement in the root mean square value attained by the UFTZNN model in comparison to the other three models is shown in Table 5. The data unambiguously shows that the UFTZNN model performs better than the other models, with gains varying from 4.51% to 54.26%.

Drawing insights from the simulation examples discussed above, it can be concluded that the UFTZNN model is suitable for the left inverse problem or the right inverse problem, irrespective of the dimensions of the rectangular matrix. Additionally, the UFTZNN model demonstrates superior convergence and robustness against noise.

Table 4
The root mean square value of the error matrix norm for solving the right inverse under various noise conditions.

ZNN models	OZNN-R	IEZNN-R	MZNN-R	UFTZNN
Fig. 4(a)	0.3559	0.3559	0.2186	0.2066
Fig. 4(b)	0.3859	0.3569	0.2183	0.2085
Fig. 4(c)	0.3668	0.3679	0.2170	0.2070
Fig. 4(d)	0.5312	0.5047	0.3197	0.2430

Table 5
The percentage improvement of the root means square error of the UFTZNN model compared to other models for solving the right inverse.

Other models	OZNN-R(%)	IEZNN-R(%)	MZNN-R(%)
UFTZNN	41.95	41.95	5.49
UFTZNN	45.98	41.58	**4.51**
UFTZNN	43.56	43.74	4.61
UFTZNN	**54.26**	51.86	24.00

4.1.3. High-dimensional matrix Moore–Penrose inverse

To assess the ability to solve high-dimensional matrices, we examine a time-dependent Toepliz matrix. By exploiting the properties of the Toepliz matrix, we designate the first column as follows:

$$N_{i1}(t) = \begin{bmatrix} 2+\sin(t) & 2\sin(t) & \cdots & 2\sin(t)/(m-1) \end{bmatrix}^{\mathrm{T}}.$$

The first row is as follows:

$$N_{1j}(t) = \begin{bmatrix} 2+\sin(t) & 2\cos(t) & \cdots & 2\cos(t)/(n-1) \end{bmatrix}.$$

When the dimensions m and n are unequal, an asymmetric Toeplitz matrix is obtained. In this simulation, we evaluate the performance of solving the left inverse by selecting m and n as 10 and 5, respectively. The results are depicted in Fig. 5(a). Similarly, to assess the performance of solving the right inverse, we choose m and n as 10 and 30, respectively, and the results are presented in Fig. 5(b). The ZNN models are capable of solving high-dimensional matrices. Analogous to the low-dimensional case, the UFTZNN model demonstrates the shortest final convergence time for solving both the left inverse and the right inverse.

4.2. Application to photoelectric tracking system based on manipulator

In this section, the proposed UFTZNN model is applied to the photoelectric tracking servo system that is based on the manipulator.

4.2.1. Problem description

The photoelectric tracking system is a servo system commonly employed for tracking targets. Traditionally, this system consists of two axes: the azimuth axis and the pitch axis. The pitch axis is equipped with a television imaging system. The fundamental principle of operation involves maintaining a specific distance between the target and the device, while simultaneously driving the azimuth and pitch axes to ensure that the target remains at the center of the image. Fig. 8(a)

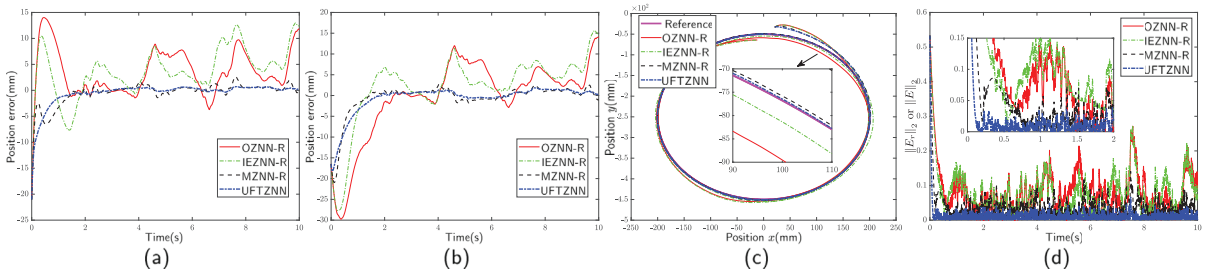


Fig. 6. The simulation tracking result of the reference path $\dot{x}_d(t) = [200\sin 0.4\pi t, 200\cos 0.4\pi t - 250]^{\mathrm{T}}$ mm. (a) The position tracking error of the end effector moving in the x-direction. (b) The position tracking error of the end effector moving in the y-direction. (c) The reference path and actual motion trajectory of the end effector. (d) The dynamic trajectory of the residual error matrix norm.

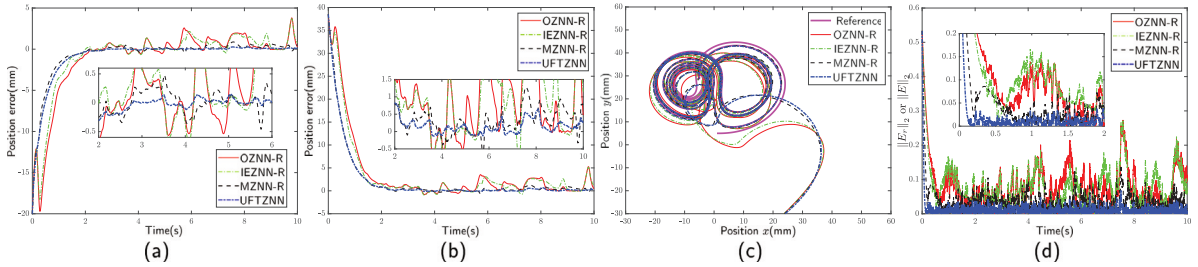


Fig. 7. The simulation tracking result of the reference path $\dot{x}_d(t) = [L_y, L_z]^{\mathrm{T}}$ mm, where L_y and L_z are the solution set of Lorenz system (1) in Jin et al. (2016) when $t < 20$. (a) The position tracking error of the end effector moving in the x-direction. (b) The position tracking error of the end effector moving in the y-direction. (c) The reference path and actual motion trajectory of the end effector. (d) The dynamic trajectory of the residual error matrix norm.

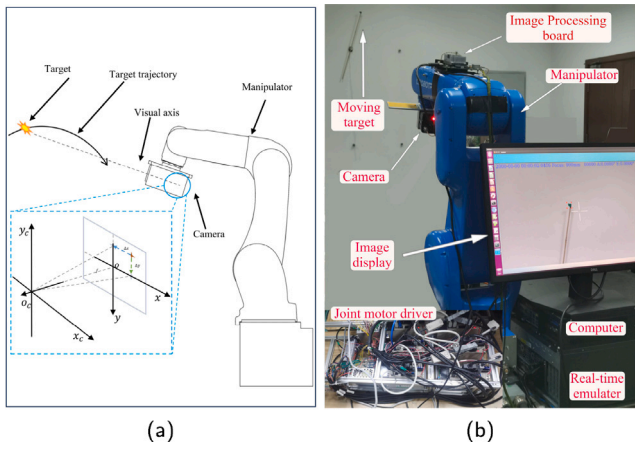


Fig. 8. The photoelectric tracking system based on the manipulator. (a) Schematic diagram. (b) Experimental installation.

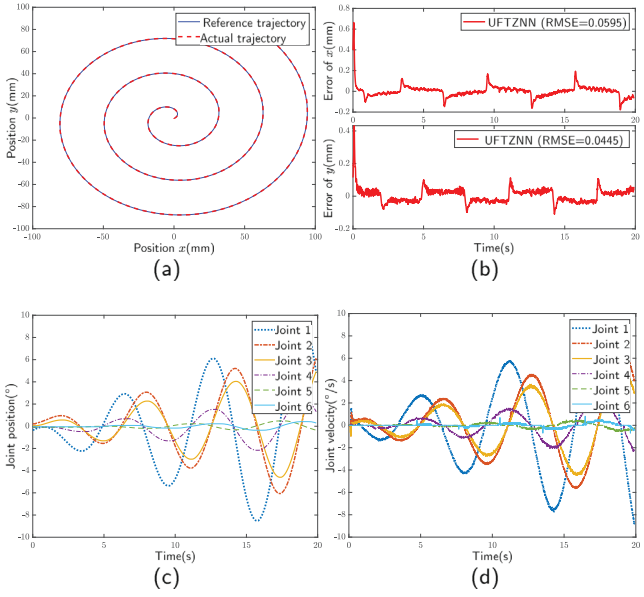


Fig. 9. The guide tracking result of the reference path $\dot{x}_d(t) = [(5t+1)\cos t, (5t+1)\sin t]^{\mathrm{T}}$ mm by using the UFTZNN model. (a) The expected path and actual motion trajectory of the end effector. (b) The guide tracking error of the end effector moving in the x-direction and y-direction. (c) The encoder position of each joint (remove the average value). (d) The velocity of each joint.

illustrates a schematic diagram of a photoelectric tracking system that utilizes a manipulator. The manipulator's standard Denavit–Hartenberg parameters are listed in Table 6.

In this system, a camera is positioned at the end of a six-degree-of-freedom manipulator to capture images of the target. The system typically operates in two modes: the guidance mode and the television closed-loop mode. In the guidance mode, the target's position in the world coordinate system is determined, and the target is displayed within the camera's field of view. The deviation of the target's off-target coordinates in the image coordinate system oxy is denoted as $(\Delta x, \Delta y)$. Subsequently, a television closed-loop control is implemented to ensure that the target remains centered within the image coordinate system. Both the guidance mode and the television closed-loop mode are typically based on the velocity mode of the servo driver. Consequently, the manipulator faces a typical inverse kinematics problem at the velocity level.

In the case of a manipulator with six degrees of freedom, the joint angles are denoted as $\theta = \left[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\right]^{\mathrm{T}}$. The relationship between the joint angles and the resulting spatial motion can be represented as $x_d(t) = f(\theta(t))$, where $x_d(t)$ represents the desired spatial motion at time t and $f(\theta(t))$ represents the mapping function. By taking the time derivative of both sides of the equation, the rate of change of the spatial motion can be determined:

$$\dot{x}_d(t) = J(\theta(t))\dot{\theta}(t), \tag{39}$$

where $J(\theta(t)) = \partial f(\theta(t))/\partial\theta(t)$ is the Jacobi matrix, and $\dot{x}_d(t)$ represents the workspace velocity, and $\dot{\theta}(t)$ represents the joint velocity. Therefore, the inverse kinematics solution can be expressed as:

$$\dot{\theta}(t) = J^{+}(\theta(t))\dot{x}_d(t), \tag{40}$$

where $J^{+}(\theta(t))$ is the right Moore–Penrose inverse matrix of $J(\theta(t))$ since $J(\theta(t)) \in \mathbb{R}^{2\times 6}$. Let $N(t) = J(\theta(t))$, so the UFTZNN model (21) can be used to solve the $J^{+}(\theta(t))$.

The symbol $\Lambda(t)$ is used to represent the tracking error. In the guidance mode, $\Lambda(t) = x_d(t) - f(\theta)$. The control law is designed in the form of minimum velocity norm (Wu & Zheng, 2020):

$$\begin{aligned} U =& \dot{\theta}(t) \\ =& J^{+}(\theta(t))\left(\dot{x}_d(t) + \kappa(x_d(t) - f(\theta))\right), \\ =& X(t)\left(\dot{x}_d(t) + \kappa\Lambda(t)\right) \end{aligned} \tag{41}$$

where κ is the feedback gain. In the television closed-loop mode, $\Lambda(t) = [\Delta x, \Delta y]^{\mathrm{T}}$ represents the miss distance and the control law is designed as:

$$U = X(t)(\dot{x}_d(t) + \kappa\Lambda(t) + \lambda\int \Lambda(\tau)d\tau), \tag{42}$$

where λ is the closed-loop integral coefficient, and $\dot{x}_d(t)$ is synthesized by using all encoders and $\Lambda(t)$, that is:

$$\dot{x}_d(t) = \left(f(\theta)\exp(-\tau_c s) + \Lambda(t)\right)s/(\tau_f s + 1), \tag{43}$$

where τ_c represents the system delay time. τ_f represents the coefficient of low-pass filter. s denotes the Laplace transform.

4.2.2. Simulation validation

To establish the groundwork for the experiment, the initial step involves conducting a simulation to validate the guidance tracking mode. In this simulation, the OZNN-R, IEZNN-R, and MZNN-R models are utilized for comparison purposes. The value of κ is set to 2, while the remaining parameters remain consistent with the aforementioned simulation. Additionally, random noise with a variance of 2 is introduced to this simulation. The reference path is defined as

$$\dot{x}_d(t) = [200\sin 0.4\pi t, 200\cos 0.4\pi t - 250]^{\mathrm{T}}\ \mathrm{mm},$$

and

$$\dot{x}_d(t) = [L_y, L_z]^{\mathrm{T}}\ \mathrm{mm},$$

where L_y and L_z represent the solution set of the Lorenz system (1) in Jin et al. (2016) when $t < 20$.

The simulation results are depicted in Figs. 6 and 7. Subfigure (a) illustrates the position tracking error of the end effector along the x-direction, while subfigure (b) showcases the position tracking error along the y-direction. Subfigure (c) displays the anticipated path alongside the actual motion trajectory of the end effector. Both the MZNN-R model and the UFTZNN model outperform the OZNN-R model and the IEZNN model in terms of task completion. Furthermore, the UFTZNN model exhibits a lower tracking error compared to the MZNN-R model in finer detail. Subfigure (d) displays the error matrix norm of the matrix pseudoinverse calculation in trajectory tracking. The UFTZNN model demonstrates a faster convergence rate and a smaller static error in the resolution process.

4.2.3. Experimental validation

The experimental configuration is depicted in Fig. 8(b). The 6-axis vertical multi-joint manipulator is equipped with a 17-bit absolute encoder at the motor end of each joint. This manipulator demonstrates a repetitive positioning accuracy of ±0.03 mm, can handle an end load of 6 kg, and has a maximum coverage of 916 mm. The real-time emulator is facilitated by a software called Links-RT, which enables real-time simulation. The system operates at a frequency of 1000 Hz, while the television's sampling frequency is 50 Hz. The system delay, denoted as τ_c, is measured to be 40 ms. The camera's focal length (f) is

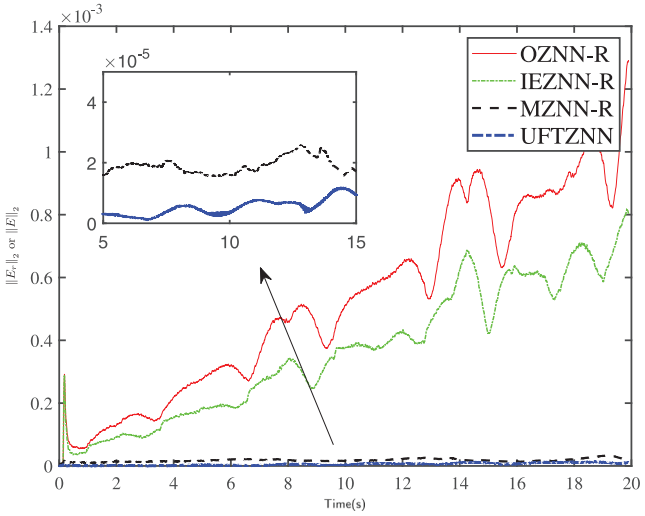


Fig. 10. The error matrix norm of the right inverse during trajectory tracking in an experimental environment.

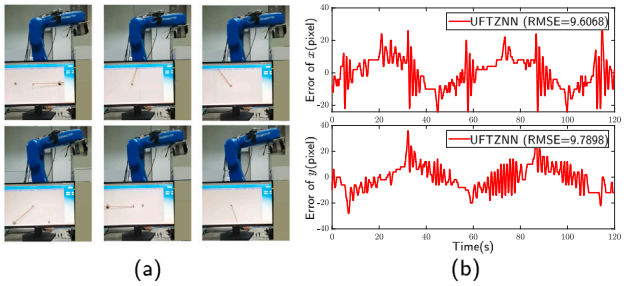


Fig. 11. The closed-loop of the television mode by using the UFTZNN model. (a) Snapshots of the closed-loop process. (b) The closed-loop tracking error.

Table 6
The standard Denavit–Hartenberg parameters of the manipulator.

Joint number	1	2	3	4	5	6
Link length a_i (mm)	40	435	40	0	0	0
Link offset d_i (mm)	389	0	0	440	0	92
Link twist α_i (rad)	$\pi/2$	0	$\pi/2$	$\pi/2$	$\pi/2$	0
Joint angle θ_i (rad)	0	$\pi/2$	0	π	π	0

adjustable from 5.2 mm to 98 mm. The relevant parameters are set as follows: $\kappa = 20$, $\lambda = 4$, and $\tau_f = 0.01$. The UFTZNN model is implemented in both the system's guidance mode and the closed-loop mode of the television. In the experimental setting, it is assumed that the motor speed can be achieved based on the speed mode of the servo driver, following the provided instructions.

The reference path provided in this experiment is described by the spiral equation $\dot{x}_d(t) = [(5t+1)\cos t, (5t+1)\sin t]^{\mathrm{T}}$ in millimeters. The tracking results are presented in Fig. 9. From Fig. 9(a), the expected trajectory closely matches the actual end trajectory. The maximum steady-state error is found to be as low as 0.19 mm, while the root mean square value of the tracking error is calculated to be as low as 0.06 mm from Fig. 9(b) . Furthermore, Fig. 9(c) and (d) demonstrate that the joint angle and velocity changes of the actual equipment exhibit smooth transitions without sudden changes. This characteristic makes it highly suitable for engineering applications. In the context of trajectory tracking, the OZNN-R, IEZNN-R, MZNN-L, and UFTZNN models are employed to compute the pseudoinverse $J^{+}(\theta(t))$ of the Jacobian matrix. The outcomes of this computation are depicted in Fig. 10. The

OZNN-R and IEZNN-R models exhibit the least favorable performance, while the UFTZNN model achieves the lowest error matrix norm.

To assess the closed-loop capability of the television mode, a dynamic tracking experiment was conducted on a target performing circular motion. The snapshot of the tracking process is depicted in Fig. 11(a), and the closed-loop tracking error is shown in Fig. 11(b). It can be observed that the moving target consistently remains positioned in the center of the image, and the television closed-loop root mean square error is less than 10 pixels. These experimental results provide evidence for the feasibility and effectiveness of utilizing the UFTZNN model to control real-world systems. Consequently, this experiment serves as a valuable reference for engineering applications.

5. Conclusion

The paper proposes a UFTZNN model designed to solve the Moore–Penrose inverse problem for dynamic matrices. The UFTZNN model achieves fixed-time convergence and is independent of initial values. Compared to other models whose convergence time depends on initial values, the UFTZNN model is more stable and effective. The UFTZNN model can uniformly solve for both the left and right inverses of a matrix without the need to design separate models for each. The UFTZNN model has a stronger anti-interference ability and can quickly converge to the theoretical solution even under noisy conditions. Through simulation validation and practical application, the UFTZNN model outperforms other typical ZNN models in terms of convergence speed and anti-interference capability. The UFTZNN model has been successfully applied to solve the inverse kinematics of robots and has achieved good real-time tracking effects. This demonstrates the effectiveness and practicality of the UFTZNN model in solving real engineering problems. Overall, the UFTZNN model proposed in this paper has significant innovations in fixed-time convergence, unifying left and right inverses, anti-interference ability, and practical application effects, providing an effective solution to the Moore–Penrose inverse problem for dynamic matrices. One future direction involves utilizing this framework to address other time-varying problems. Another development direction entails creating new types of ZNN models tailored to handle different noise types within the framework.

Abbreviations

ZNN	Zeroing neural network
OZNN	Original zeroing neural network
IEZNN	Integrated-enhanced zeroing neural network
MZNN	Modified zeroing neural network
UFTZNN	Unified fixed-time zeroing neural network
-R	As suffixes for the OZNN, IEZNN, and MZNN, they represent models designed for solving right Moore–Penrose inverses.
-L	As suffixes for the OZNN, IEZNN, and MZNN, they represent models designed for solving left Moore–Penrose inverses.

Funding

This work was supported by the National Natural Science Foundation of China [grant number 62271109].

CRediT authorship contribution statement

Bing Zhang: Investigation, original draft, Visualization, Writing – review & editing. **Yuhua Zheng:** Formal analysis, Visualization, Validation, Writing – review & editing. **Shuai Li:** Conceptualization, Methodology, Writing – review & editing. **Xinglong Chen:** Software, Data curation, Supervision. **Yao Mao:** Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Chen, D., Li, S., Lin, F.-J., & Wu, Q. (2020). New super-twisting zeroing neural-dynamics model for tracking control of parallel robots: A finite-time and robust solution. *IEEE Transactions on Cybernetics*, *50*(6), 2651–2660. http://dx.doi.org/10.1109/TCYB.2019.2930662.

Chountasis, S., Katsikis, V. N., & Pappas, D. (2010). Digital image reconstruction in the spectral domain utilizing the Moore-Penrose inverse. *Mathematical Problems in Engineering*, *45*, 1–14. http://dx.doi.org/10.1155/2010/750352.

Dai, J., Chen, Y., Xiao, L., Jia, L., & He, Y. (2022). Design and analysis of a hybrid GNN-ZNN model with a fuzzy adaptive factor for matrix inversion. *IEEE Transactions on Industrial Informatics*, *18*(4), 2434–2442. http://dx.doi.org/10.1109/TII.2021.3093115.

Dai, J., Jia, L., & Xiao, L. (2021). Design and analysis of two prescribed-time and robust ZNN models with application to time-variant Stein matrix equation. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(4), 1668–1677. http://dx.doi.org/10.1109/TNNLS.2020.2986275.

Guo, D., & Zhang, Y. (2014). Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time-varying Jacobian matrix pseudoinversion. *Applied Soft Computing*, *24*, 158–168. http://dx.doi.org/10.1016/j.asoc.2014.06.045.

Jia, L., Xiao, L., Dai, J., & Cao, Y. (2020). A novel fuzzy-power zeroing neural network model for time-variant matrix Moore-Penrose inversion with guaranteed performance. *IEEE Transactions on Fuzzy Systems*, *29*(9), 2603–2611. http://dx.doi.org/10.1109/TFUZZ.2020.3005272.

Jin, L., Zhang, Y., Qiao, T., Tan, M., & Zhang, Y. (2016). Tracking control of modified lorenz nonlinear system using ZG neural dynamics with additive input or mixed inputs. *Neurocomputing*, *196*, 82–94. http://dx.doi.org/10.1016/j.neucom.2015.12.115.

Jin, J., Zhu, J., Gong, J., & Chen, W. (2022). Novel activation functions-based ZNN models for fixed-time solving dynamirc Sylvester equation. *Neural Computing and Applications*, *34*(17), 14297–14315. http://dx.doi.org/10.1007/s00521-022-06905-2.

Katsikis, V. N., Stanimirović, P. S., Mourtas, S. D., Xiao, L., Karabašević, D., & Stanujkić, D. (2022). Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix. *IEEE Transactions on Fuzzy Systems*, *30*(9), 3426–3435. http://dx.doi.org/10.1109/TFUZZ.2021.3115969.

Kong, Y., Chen, S., Jiang, Y., Wang, H., & Chen, H. (2023). Zeroing neural network with fuzzy parameter for cooperative manner of multiple redundant manipulators. *Expert Systems with Applications*, *212*, 118735.1–118735.9. http://dx.doi.org/10.1016/j.eswa.2022.118735.

Li, W. (2018). A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations. *IEEE Transactions on Industrial Informatics*, *14*(12), 5289–5298. http://dx.doi.org/10.1109/TII.2018.2817203.

Li, W., Guo, C., Ma, X., & Pan, Y. (2024). A strictly predefined-time convergent and noise-tolerant neural model for solving linear equations with robotic applications. *IEEE Transactions on Industrial Electronics*, *71*(1), 798–809. http://dx.doi.org/10.1109/TIE.2023.3241393.

Li, W., Liao, B., Xiao, L., & Lu, R. (2019). A recurrent neural network with predefined-time convergence and improved noise tolerance for dynamic matrix square root finding. *Neurocomputing*, *337*(APR.14), 262–273. http://dx.doi.org/10.1016/j.neucom.2019.01.072.

Liao, B., & Zhang, Y. (2014). From different ZFs to different ZNN models accelerated via Li activation functions to finite-time convergence for time-varying matrix pseudoinversion. *Neurocomputing*, *133*, 512–522. http://dx.doi.org/10.1016/j.neucom.2013.12.001.

Machado, J. T., & Lopes, A. M. (2017). A fractional perspective on the trajectory control of redundant and hyper-redundant robot manipulators. *Applied Mathematical Modelling*, *46*, 716–726. http://dx.doi.org/10.1016/j.apm.2016.11.005.

Peter, Benner, Pablo, Ezzatti, EnriqueS, Quintana-Ortí, Alfredo, & Remón (2012). Matrix inversion on CPU-GPU platforms with applications in control theory. *Concurrency Computations: Practice and Experience*, *25*(8), 1170–1182. http://dx.doi.org/10.1002/cpe.2933.

Polyakov, A. (2012). Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Transactions on Automatic Control*, *57*(8), 2106–2110. http://dx.doi.org/10.1109/CDC.2011.6160190.

Sun, Z., Li, F., Jin, L., Shi, T., & Liu, K. (2020). Noise-tolerant neural algorithm for online solving time-varying full-rank matrix Moore-Penrose inverse problems: A control-theoretic approach. *Neurocomputing*, *413*, 158–172. http://dx.doi.org/10.1016/j.neucom.2020.06.050.

Sun, Z., Wang, G., Jin, L., Cheng, C., Zhang, B., & Yu, J. (2022). Noise-suppressing zeroing neural network for online solving time-varying matrix square roots problems: A control-theoretic approach. *Expert Systems with Applications*, *192*, 116272.1–116272.15. http://dx.doi.org/10.1016/j.eswa.2021.116272.

Tan, Z., Xiao, L., Chen, S., & Lv, X. (2020). Noise-tolerant and finite-time convergent ZNN models for dynamic matrix Moore-Penrose inversion. *IEEE Transactions on Industrial Informatics*, *16*(3), 1591–1601. http://dx.doi.org/10.1109/TII.2019.2929055.

Wang, H., Li, J., & Liu, H. (2006). Practical limitations of an algorithm for the singular value decomposition as applied to redundant manipulators. In *2006 IEEE conference on robotics, automation and mechatronics* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/RAMECH.2006.252609.

Wang, X., Ma, H., & Stanimirović, P. S. (2017). Recurrent neural network for computing the W-weighted Drazin inverse. *Applied Mathematics and Computation*, *300*, 1–20. http://dx.doi.org/10.1016/j.amc.2016.11.030.

Wei, L., Jin, L., Yang, C., Chen, K., & Li, W. (2019). New noise-tolerant neural algorithms for future dynamic nonlinear optimization with estimation on Hessian matrix inversion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *PP*(99), 1–13. http://dx.doi.org/10.1109/TSMC.2019.2916892.

Wu, W., & Zheng, B. (2020). Improved recurrent neural networks for solving Moore-Penrose inverse of real-time full-rank matrix. *Neurocomputing*, *418*, 221–231. http://dx.doi.org/10.1016/j.neucom.2020.08.026.

Xiang, Q., Liao, B., Xiao, L., & Jin, L. (2018). A noise-tolerant Z-type neural network for time-dependent pseudoinverse matrices. *Optik*, *165*, 16–28. http://dx.doi.org/10.1016/j.ijleo.2018.03.078.

Xiao, L. (2015). A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation. *Neurocomputing*, *167*, 254–259. http://dx.doi.org/10.1016/j.neucom.2015.04.070.

Xiao, L., He, Y., Dai, J., Liu, X., Liao, B., & Tan, H. (2020). A variable-parameter noise-tolerant zeroing neural network for time-variant matrix inversion with guaranteed robustness. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(4), 1535–1545. http://dx.doi.org/10.1109/TNNLS.2020.3042761.

Xiao, L., He, Y., & Liao, B. (2022). A parameter-changing zeroing neural network for solving linear equations with superior fixed-time convergence. *Expert Systems with Applications*, *208*, 118086.1–118086.9. http://dx.doi.org/10.1016/j.eswa.2022.118086.

Xiao, L., & Liao, B. (2016). A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation. *Neurocomputing*, *193*, 213–218. http://dx.doi.org/10.1016/j.neucom.2016.02.021.

Xiao, L., Song, W., Li, X., Jia, L., Sun, J., & Wang, Y. (2022). Design and analysis of a noise-resistant ZNN model for settling time-variant linear matrix inequality in predefined-time. *IEEE Transactions on Industrial Informatics*, *18*(10), 6840–6847. http://dx.doi.org/10.1109/TII.2021.3135383.

Xiao, L., Zhang, Y., Li, K., Liao, B., & Tan, Z. (2019). A novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion. *Neurocomputing*, *331*, 483–492. http://dx.doi.org/10.1016/j.neucom.2018.11.071.

Xiao, L., Zhang, Y., Zuo, Q., Dai, J., Li, J., & Tang, W. (2020). A noise-tolerant zeroing neural network for time-dependent complex matrix inversion under various kinds of noises. *IEEE Transactions on Industrial Informatics*, *16*(6), 3757–3766. http://dx.doi.org/10.1109/TII.2019.2936877.

Zhang, Y., & Guo, D. (2015). Time-varying matrix left pseudoinverse. In *Zhang functions and various models* (pp. 91–104). Berlin, Heidelberg: Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-662-47334-4_7.

Zhang, Y., Jiang, D., & Wang, J. (2002). A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Transactions on Neural Networks*, *13*(5), 1053–1063. http://dx.doi.org/10.1109/TNN.2002.1031938.

Zhang, Y., Qiu, B., Jin, L., Guo, D., & Yang, Z. (2015). Infinitely many Zhang functions resulting in various ZNN models for time-varying matrix inversion with link to Drazin inverse. *Information Processing Letters*, *115*(9), 703–706. http://dx.doi.org/10.1016/j.ipl.2015.03.007.

Zhang, Y., Yang, Y., Tan, N., & Cai, B. (2011). Zhang neural network solving for time-varying full-rank matrix Moore-Penrose inverse. *Computing*, *92*, 97–121. http://dx.doi.org/10.1007/s00607-010-0133-9.

Zhang, B., Zhang, H., & Ge, S. S. (2004). Face recognition by applying wavelet subband representation and kernel associative memory. *IEEE Transactions on Neural Networks*, *15*(1), 166–177. http://dx.doi.org/10.1109/tnn.2003.820673.