Discrete Perturbation-Immunity Neural Network for Dynamic Constrained Redundant Robot Control

JUN YANG1, LI CHEN1, YIMENG QI2, MEI LIU2, AND CHEN CUI1,3

1College of Mathematics Physics and Information Engineering, Jiaxing University, Jiaxing 314000, China
2School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
3School of Information Science and Technology, Heilongjiang University, Harbin 150080, China

Corresponding authors: Jun Yang (yangj95@mail2.sysu.edu.cn) and Chen Cui (394874486@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702224, in part by the Natural Science Foundation of Zhejiang Province under Grant LY18F020020 and Grant LY15F020040, in part by Zhejiang Public Welfare Technology Research Project under Grant LGG20F010010, and in part by the Fundamental Research Funds for the Central Universities under Grant Izujbky-2019-89.

ABSTRACT Considering the necessity of merging the physical constraints in joint space for redundant robot motion control in practice by regarding the kinematics of robots, a discrete perturbation-immunity neural network (DPINN) model with high robustness and predominant convergence is proposed in this article for managing the problem. It is worth emphasizing that this proposed neural network is developed to provide a solution algorithm to dispose of the practical applications in robot motion control that is investigated by reforming it into perturbed dynamic quadratic programming (QP) with equality and inequality constraints, which remedies the lack and drawbacks of existing methodologies. Moreover, theoretical verifications and results verify the convergence performance and noise suppression ability of the proposed neural network, which is further verified by simulation examples. In addition, the simulations on robot motion control embody its superiority and powerful versatility compared to the existing methods. In summary, the proposed neural network extends the sphere of application in perturbed dynamic QP with double-bound constraints and offers a feasible scheme capable of controlling robots in a neural-network thought.

INDEX TERMS Discrete perturbation-immunity neural network (DPINN), physical constraints, robots, theoretical verifications.

I. INTRODUCTION

Redundant robots, a type of robot with additional degrees of freedom for more expediently completing some specific or complicated problems, have achieved certain goals, such as the avoidance of joint limits and obstacles [1], [2]. Thus, there is much attention from the scientific and engineering domain, and a series of investigations are being explored and obtained, for example, space exploration, military industries, and so on [3]–[5]. In view of these studies, abundant methods mainly contain numerical algorithms and neural network-based ones have been discussed and applied to robot control, bringing their superiority in saving man-power and improving efficiency in full play in the industry. However, conventional numerical methods, such as the Newton iteration and conjugate gradient method, share a common idea using serial processing solving [6], [7], which is a complicated and long iteration process, brings heavy computational load, and limits the application of such algorithms in large-scale online optimization, especially for the time-critical ones as time-dependent problems or more challenging situations. In contrast, the neural network/dynamic method displays powerful competitiveness with its parallel distribution structure, nonlinear mapping, strong learning ability, and it approximates any function accurately in the control field, including robot motion control [8]–[10] and other aspects [11], [12]. It is worth noting that the realization of robot control is through a quadratic programming (QP) based scheme that can generally be reorganized as a linear equation system with bound constraints neglected or a linear projection one with those involved [13].

A class of Wang neural networks with exponential convergence is developed to settle many of the mathematical problems from the perspective of control [14]–[16] and specializes
in static problems, but those problems in practice are actually dynamical, which is verified in their failure to track the theoretical solution of a dynamic problem due to the lack of velocity compensation of the dynamic parameters process [17]–[19]. In addition, the gradient neural network as a conventional recurrent neural network (RNN) based approach is employed to search the target roots of dynamic problems [20], [21]. As an improvement, the zeroing neural network that can be viewed as a systematic approach in managing dynamic problems is studied to force the error norm approach to zero [22]–[24]. Then, considerable effort is devoted to its finite-time convergence aided with special activation functions with theoretical analyses and simulation results given [25], [26]. All the models/methods introduced above are continuous models that are only feasible at the level of theoretical application and are not suitable for the purpose of digital hardware implementation. Thus, a discrete-time RNN model is further studied in [27] via the Euler-type difference rule, whose residual error is obtained in the accuracy of $O(\delta^2)$ (being the sampling interval). In view of this point, four-instant discretization rules are utilized in [28] to construct a discrete model for managing dynamic QP problems with equality limits in a more favorable and precise manner. Taking robot control as an example, it is inevitable that robots would be interfered with environmental perturbations (such as electromagnetic interference) and internal equipment errors from digital equipment. The models reviewed previously are invalid and unreliable in noisy circumstances, leading to a failure of the solution process or even a collapsed system and thus suppressing perturbations during task completion in real time is inescapable. Furthermore, discrete-time neural dynamics with noise suppression ability is investigated in [29] to promote the robot to execute a determined trajectory subject to equality constraint in the complex domain. In [30], a continuous-time noise-tolerant neural algorithm is designed for robot motion generation in a nonlinear projection form with equality limits with excellent convergence performance.

All the above research results share one thing in common, that is, the dynamic QP problem to be solved is just equipped with equality constraints, limiting the application prospect of the existing models greatly. In terms of robotics concerned in current research, almost all robot manipulators are constrained physically. Particularly, the joint variables should be kept within their physically limited regions [31]–[35] because exceeding the joint physical limits would cause failure in the motion planning of physically constrained redundant robot manipulators. Consequently, it is necessary to study dynamic QP constrained by inequalities. The existing studies may be flawed to some extent. For example, in [36], an RNN for solving the perturbed dynamic linear system with double-bound limits on joint velocity is applied to a physically-limited PUMA 560 robot in tracking trajectory satisfactorily, which is a continuous-time model. In addition, another neural network in the continuous form is developed in [37] for robots to realize repetitive motion generation with equality and inequality constraints but the perturbations are ignored. In [38], [39], a distributed coordination model is developed for multiple robots for competitive target tracking, where sufficient theoretical analysis and simulations are given to demonstrate its feasibility but the continuous-time system is unstable and fragile. As exploited in [40], a dual neural network is designed for manipulator redundancy resolution subject to both joint and joint speed constraints, presenting exponential convergence but disturbance is overlooked.

Considering the defects of existing models in addressing the dynamic redundant robot control with inequality constraints involved, this paper focuses on the design of a discrete and highly robust neural network model, i.e., the discrete perturbation-immunity neural network (DPINN) model, and employs it in robot motion control with double side constraints on joint velocity as a highlight of the study, which extends the applicability of the proposed model. The discussions on the technical drawbacks of previous studies and main contributions of this article as an improvement.

FIGURE 1. The flowchart for setting forth the technical drawbacks of previous studies and main contributions of this article as an improvement.
This paper is centered on a DPINN model with high robustness for managing the dynamic optimization problem with equality and inequality constraints simultaneously, vastly making up for the defects of existing algorithms/models.

This paper starts from reality, taking cutting-edge robotics as an example, and in turn designs a comprehensive neural network to illustrate the practicability and effectiveness of the proposed model in robot motion planning.

To scientifically and authoritatively demonstrate the noise suppression ability and convergence performance of the proposed neural network, the associated theoretical analysis and simulations are conducted and exploited.

II. PROBLEM PRELIMINARY AND MODEL CONSTRUCTION

In this section, to lay a foundation for the design of the robot control scheme, a general form of the robot forward kinematics equation is drawn forth with the design procedures combined with reality. Then, the manipulation process for neural network development is presented in detail.

A. PRELIMINARY ON ROBOT KINEMATICS

Considering a redundant robot manipulator with p-degree of freedom, the corresponding kinematic equation in an affine form is defined as

$$J(\theta(t))\dot{\theta}(t) = \dot{\rho}(t),$$  \tag{1}

of which $J(\theta(t)) \in \mathbb{R}^{m \times p}$ stands for the Jacobian matrix, which is underdetermined (i.e., $m < p$) and nonsingular at any point in time $t \in [0, T]$ to ensure the consistent logic of $\dot{\theta}(t)$ with $T$ the task execution time; $\dot{\theta}(t)$ denoting the joint-velocity vector is the derivative of the joint angle vector $\theta(t) = [\theta_1(t); \theta_2(t); \cdots; \theta_p(t)] \in \mathbb{R}^p$; $\dot{\rho}(t)$ is the time derivative of the Cartesian coordinate $\rho(t) \in \mathbb{R}^m$. All the manipulators are desired to track the expected trajectory $\rho(t)$, thus controlling and limiting the robots with formula (1). Besides, the joint velocity of the robot is restricted in a certain range in reality, to prevent the robot from singularity and damage. Thus, the solution calculated by the above formula is considered to handle the following QP problem constrained by an equality and an inequality:

$$\min \dot{\theta}^T(t)\dot{\theta}(t)/2$$

s.t. $J(\theta(t))\dot{\theta}(t) = \dot{\rho}(t)$

$$\dot{\theta}^-(t) \leq \dot{\theta}(t) \leq \dot{\theta}^+(t),$$ \tag{2}

with the symbol $^T$ the transpose operation of a matrix or a vector; time $t \in [t_0, t_f]$, where $t_0$ and $t_f$ are the initial time and the final time of the computing process; the upper bound $\dot{\theta}^+(t)$ and lower bound $\dot{\theta}^-(t)$. For further constructing the proposed neural network and managing the desired robot joint velocity to control its motion behavior through handling the above QP problem, the following transformation is designed.

The inequality constraint $\dot{\theta}^-(t) \leq \dot{\theta}(t) \leq \dot{\theta}^+(t)$ is able to be generalized as $N(t)\dot{\theta}(t) \leq 0(t)$, where the above two formula are equivalent when $N(t) \in \mathbb{R}^{2p \times p}$ and $c(t) \in \mathbb{R}^{2p}$ are set as $N(t) = [-E_{p \times p}; E_{p \times p}]$ ($E$ being the identity matrix) and $c(t) = [\dot{\theta}^-(t); \dot{\theta}^+(t)]$, respectively. Hence, formula (2) is converted into

$$\min \dot{\theta}^T(t)\dot{\theta}(t)/2$$

s.t. $J(\theta(t))\dot{\theta}(t) = \dot{\rho}(t)$

$$N(t)\dot{\theta}(t) \leq c(t).$$ \tag{3}

Accordingly, introduce the Lagrangian multipliers to the equality and inequality constraints as $\lambda(t) \in \mathbb{R}^m$ and $\mu(t) \in \mathbb{R}^{2p}$, respectively, to calculate the optimal solution to the QP problem (3), which is required to meet the circumstance (Karush-Kuhn-Tucker conditions) below:

$$E_{p \times p}\dot{\theta}(t) + J^T(\theta(t))\lambda(t) + N^T(t)\mu(t) = 0$$

$$J(\theta(t))\dot{\theta}(t) = \dot{\rho}(t)$$

$$N(t)\dot{\theta}(t) - c(t) \leq 0, \mu(t) \geq 0$$

$$\mu^T(t)(N(t)\dot{\theta}(t) - c(t)) = 0.$$ \tag{4}

Then, a nonlinear complementary problem function that is described as $\omega_{bcpl}(\beta_1; \beta_2) = (\beta_1 \circ \beta_1 + \beta_2 \circ \beta_2 + \epsilon)^{1/2} - \beta_1 - \beta_2$ is leveraged here to treat the above nonlinear formula (4) [41]. The symbol $\circ$ signifies the Hadamard product, and $\epsilon \to 0^+ \in \mathbb{R}^p$ is the perturbation item. Consequently, formula (4) is able to be reformulated as

$$E_{p \times p}\dot{\theta}(t) + J^T(\theta(t))\lambda(t) + N^T(t)\mu(t) = 0$$

$$J(\theta(t))\dot{\theta}(t) = \dot{\rho}(t)$$

$$\omega_{bcpl}(\mu(t), N(t)\dot{\theta}(t) - c(t)) = 0.$$ \tag{5}

In line accordance with the definition of the nonlinear complementary problem function $\omega_{bcpl}(\cdot)$, formula (5) is simplified as

$$H(t)x(t) - w(t) = 0.$$ \tag{6}

The parameters involved in (6) are set as $H(t) = [E_{p \times p} J^T(\theta(t)) N^T(t); J(\theta(t)) \ 0 \ 0; -N(t) \ 0 \ E_{2p \times 2p}] \in \mathbb{R}^{(p+m+2p) \times (p+m+2p)}$, $x(t) = [\dot{\theta}(t) \ \lambda(t) \ \mu(t)]^T \in \mathbb{R}^{p+m+2p}$, $w(t) = [0 \ \dot{\rho}_0(t) \ \omega_3(t)]^T \in \mathbb{R}^{p+m+2p}$ with $\omega_3(t) = -c(t) + \sqrt{c(t) \circ c(t) + \mu(t) \circ \mu(t) + \epsilon}$, where $c(t) = N(t)\dot{\theta}(t) - c(t)$. Heretofore, disposing of formula (6) is just finding the optimal solution to formula (2) and thus the QP problem is solved completely.

B. NEURAL NETWORK CONSTRUCTION

For promoting the construction of the proposed neural network, an error function is firstly designed as

$$\zeta(t) = H(t)x(t) - w(t) \in \mathbb{R}^{p+m+2p}.$$ \tag{7}

Notably, the error function $\zeta(t)$ is desired to approach zero for obtaining the solution to formula (6). In addition, the evolution law is applied to $\zeta(t)$ as

$$\dot{\zeta}(t) = -\alpha\zeta(t) - \eta \int_0^t \zeta(\sigma)d\sigma.$$
of which the parameters $\alpha > 0$ and $\eta > 0$ are constants corresponding to the convergence rate. Put another way, by projecting the neural network model from the control viewpoint, a nonlinear controller with $x(t)$ as the state variable, $u^*(t)$ as the input and $\varsigma(t)$ as the output can be obtained as below

$$
\begin{align*}
    u^*(t) &= \Delta x(t)/dt \\
    \varsigma(t) &= H(t)x(t) - w(t).
\end{align*}
$$

Consequently, promoting the error function $\varsigma(t)$ to approach zero in control terms is able to be regarded as exploring a controller $u^*(t)$ to propel $\varsigma(t)$ to zero and then the state variable $x(t)$ to its expected one $x^*(t)$. Moreover, combining formula (7) and the above formula leads to

$$
\dot{x}(t) = H^{-1}(t)(-\dot{H}(t)x(t) + \dot{w}(t) - \alpha(H(t)x(t) - w(t)) - \eta \int_0^{t_1} (H(\sigma)x(\sigma) - w(\sigma))d\sigma), \quad (8)
$$

where the symbol $^{-1}$ denotes the inverse operation on a matrix. When taking perturbations $\psi(t) \in \mathbb{R}^{p+m+2p}$ into consideration, the above formula is evolved into

$$
\dot{x}(t) = H^{-1}(t)(-\dot{H}(t)x(t) + \dot{w}(t) - \alpha(H(t)x(t) - w(t)) - \eta \int_0^{t_1} (H(\sigma)x(\sigma) - w(\sigma))d\sigma + \psi(t)). \quad (9)
$$

To successfully employ the proposed neural network in the digital computer platform for managing dynamic problems, a numerical differentiation method is investigated to generate a discrete-time neural network model to dispose of the dynamic problem in a predictive way. One of the feasible numerical differentiation methods is the Euler forward difference formula $\dot{x}(t) = (x_{k+1} - x_k)/\delta$ ($\delta$ being the sampling interval), which is utilized into the discretization of formula (8). Then, it is got

$$
x_{k+1} = x_k + H_k^{-1}(-(H_k - H_{k-1})x_k + w_k - w_{k-1} - \gamma(H_kx_k - w_k) - \kappa \sum_{i=0}^{k} (H_i x_i - w_i)), \quad (10)
$$

with $\gamma = \alpha \delta$ and $\kappa = \eta \delta^2$. Formula (10) is the DPINN model proposed in this paper with the corresponding state equation in the discrete form expressed as

$$
\begin{align*}
    u_k &= H_k^{-1}(-(H_k - H_{k-1})x_k + w_k - w_{k-1}) \\
    \gamma(H_kx_k - w_k) - \kappa \sum_{i=0}^{k} (H_i x_i - w_i))/\delta \\
    x_{k+1} &= x_k + \delta u_k \\
    \varsigma_{k+1} &= H_{k+1}x_{k+1} - w_{k+1}.
\end{align*}
$$

Remark 1: The proposed DPINN model (10) is able to be equalized into $1 + \gamma \varsigma_k - \varsigma_{k-1} + \kappa \sum_{i=0}^{k} \varsigma_i + O(\delta^2) = 0$ with $O(\delta^2)$ a vector made up of $O(\delta^2)$. Concretely, the proposed DPINN model (10) is able to be simplified into

$$
H_kx_{k+1} = H_{k-1}x_k + w_k - w_{k-1} - \gamma(H_kx_k - w_k)
$$

$$
-\kappa \sum_{i=0}^{k} (H_i x_i - w_i),
$$

where the Taylor expansion is utilized for further facilitating that $x_{k+1} = x_k + \dot{x}_k \delta + O(\delta^2)$ and $x_k = x_{k-1} + \dot{x}_k \delta + O(\delta^2)$, thus deriving

$$
H_k(x_k + \dot{x}_k \delta + O(\delta^2))
$$

$$
= H_{k-1}(x_{k-1} + \dot{x}_k \delta + O(\delta^2))
$$

$$
+ w_k - w_{k-1} - \gamma(H_kx_k - w_k) - \kappa \sum_{i=0}^{k} (H_i x_i - w_i).
$$

III. THEORETICAL VERIFICATIONS AND RESULTS

In this section, the potential performances of the proposed DPINN model (10) are explored and the corresponding theoretical verifications are established. Firstly, for arranging the proposed DPINN model (10) and simplifying the study process, a remark is introduced as below.

The realization block diagram of the proposed DPINN model (10) is portrayed as Figure 2. Additionally, the above formula is improved as the following one with the aid of formula (9) and the Euler forward difference formula injected with perturbations.

$$
x_{k+1} = x_k + H_k^{-1}(-(H_k - H_{k-1})x_k + w_k - w_{k-1} - \gamma(H_kx_k - w_k) - \kappa \sum_{i=0}^{k} (H_i x_i - w_i) + \psi_k).
$$

FIGURE 2. Realizations of DPINN model (10) employed to the robot for tracking the desired trajectory of (2) regarded as a feedback control system, of which the plant denotes the problem to be solved.
Disposing of the above formula obtains $H_kx_k - w_k + H_k\hat{x}_k\delta = H_kx_{k-1} - w_{k-1} + H_{k-1}\hat{x}_{k-1}\delta + O(\delta^2) - \gamma (H_kx_k - w_k) - \kappa \sum_{i=0}^{k-1} (H_kx_i - w_i)$. Given the definition of the residual error $\xi_k$, the above formula is reformulated as $\xi_k + H_k\hat{x}_k\delta = \xi_{k-1} + H_{k-1}\hat{x}_{k-1}\delta + O(\delta^2) - \gamma \xi_k - \kappa \sum_{i=0}^{k-1} \xi_i$. Similarly, applying the Taylor expansion to matrix $H_k$ as $H_k = H_{k-1} + \dot{H}_k\delta + \Lambda(\delta^2)$ with the matrix $\Lambda(\delta^2)$ made up of $O(\delta^2)$ obtains $\xi_k + H_k\dot{\xi}_k\delta$

$$= \xi_{k-1} + (H_k - \dot{H}_k\delta + \Lambda(\delta^2))\hat{x}_k\delta + O(\delta^2) - \gamma \xi_k - \kappa \sum_{i=0}^{k-1} \xi_i,$$

Then, adjusting the above formula generates

$$\xi_k - \xi_{k-1} = -\gamma \xi_k - \kappa \sum_{i=0}^{k-1} \xi_i + O(\delta^2),$$

which is able to be transformed into

$$(1 + \gamma)\xi_k - \xi_{k-1} + \kappa \sum_{i=0}^{k-1} \xi_i + O(\delta^2) = 0.$$  

Theorem 1: For the proposed DPINN model (10) employed to the robot for tracking the desired trajectory of (2) disturbed by linear time-varying perturbation $\psi(t) = \phi t + \zeta$ (the discrete form $\psi_k = \phi \delta + \zeta$), the generated residual error $\|\xi_k\|_2$ approaches to $O(\delta^2)$ globally.

Proof: Based on the equivalent transformation (11), its $r$th ($r \in [1, 3p + m]$) subsystem at time interval $k\delta$ is expressed as

$$(1 + \gamma)\xi'_{k+1} - \xi'_k + \kappa \sum_{i=0}^{k} \xi'_i + O(\delta^2) = 0,$$

which is utilized to subtract the $r$th subsystem at time interval $(k - 1)\delta$ producing

$$(1 + \gamma + \kappa)\xi'_{k+1} = -\xi'_{k-1} + (2 + \gamma)\xi'_k + O(\delta^2).$$

The above formula is able to be reformulated in a matrix form as

$$\varphi'_{k+1} = P\varphi_k + O(\delta^2),$$

where $\varphi'_{k+1} = [\xi'_{k+1}; \xi'_{k+1}; \ldots; \xi'_{k+1}]$ and $P = [0; -1/(1 + \gamma + \kappa); (2 + \gamma)/(1 + \gamma + \kappa)]$ with its eigenvalues being $\text{Eig}_{1,2} = 0.5 + (\sqrt{\gamma^2 - 4}\gamma) + 2)/(1 + \gamma + \kappa)$. Managing formula (13) to explore the convergence performance of the proposed DPINN model (10) gets

$$\lim_{k \to \infty} \|\varphi'_{k+1}\|_2 = \|P_{k}\varphi_{k}\|_2 + O(\delta^2) \leq \|P\|_2 \cdot \|\varphi_k\|_2 + O(\delta^2) \leq \|P\|_2^2 \cdot \|\varphi_{k-1}\|_2 + O(\delta^2) \leq \ldots \leq \|P\|_2^k \cdot \|\varphi_1\|_2 + O(\delta^2),$$

with $\lim_{k \to \infty} \|P\|_2^k = 0$ in view of the eigenvalues $\text{Eig}_{1,2}$ of matrix $P$ both located in the unit circle. Thus, it is able to be got that $\lim_{k \to \infty} \|\varphi_{k+1}\|_2 = \lim_{k \to \infty} \|P\|_2 \|\varphi_k\|_2 + O(\delta^2) = O(\delta^2)$. In this sense, considering the definition of $\varphi_{k+1}$, the conclusion that $\lim_{k \to \infty} \|\xi_k\|_2 = O(\delta^2)$ is demonstrated. □

In the next part of the abovementioned series, the theoretical verifications on the convergence performance of the proposed DPINN model (10) are discussed with various injected perturbations.

Theorem 2: For the proposed DPINN model (10) employed to the robot for tracking the desired trajectory of (2) disturbed by linear time-varying perturbation $\psi(t) = \phi t + \zeta$ (the discrete form $\psi_k = \phi \delta + \zeta$), the generated residual error $\|\xi_k\|_2$ approaches to $O(\delta^2)$ globally.

Proof: It is able to be easily concluded from (11) that the proposed DPINN model (10) is a linear system with the superposition nature, and thus $\psi_k$ and $O(\delta^2)$ are able to be analyzed in several. Then, taking the similar effect as $O(\delta^2)$ in formula (12) with $\psi_k$, the $r$th subsystem of the proposed DPINN model (10) generates

$$(1 + \gamma)\psi'_{k+1} - \psi'_k + \kappa \sum_{i=0}^{k} \psi'_i + \phi \delta + \zeta = 0.$$  

Arranging the above formula with $Z$-transform obtains

$$(1 + \gamma + \kappa)z(\psi'(z) - \psi'(0)) = \psi'(z) + \kappa \psi'(z) \frac{1}{1 - z^{-1}} + \phi \delta + \zeta \frac{z^{-1}}{1 - z^{-1}} = 0,$$

with the initial value $\psi'(0)$ of $\psi'_k$. Handling the above formula evolves

$$\psi'(z) = -\phi \delta \frac{z\zeta + (z - 1)\zeta' + z(1 + \gamma + \kappa)\psi'(0)}{(z - 1)^2 + \kappa z(z - 1) + z(1 - 1 + \gamma + \kappa)},$$

whose poles are $\text{Pol}_{1,2} = 0.5 \sqrt{\sqrt{\gamma^2 - 4}} + 2/(1 + \gamma + \kappa)$ and $\text{Pol}_{1,3} = 1$. All the three poles are located in/on the unit circle embodying the stability of formula (14). Specifically, to prove the convergence performance of the residual error $\xi_k$, it is derived as follows:

$$\lim_{z \to 1} \|\varphi'_{k+1}\|_2 = \lim_{z \to 1} \|\psi'(0)\|_2 \frac{1}{\kappa} = \frac{\phi \delta \zeta}{\kappa}.$$  

Therefore, $\lim_{k \to \infty} \|\xi_k\|_2 = \|\phi \delta / \kappa\|_2$ is gotten with linear time-varying perturbations considered solely. Combining with the intrinsic residual error $O(\delta^2)$ of the proposed DPINN model (10), $\lim_{k \to \infty} \|\xi_k\|_2 = \|\phi \delta / \kappa\|_2 + O(\delta^2)$ is obtained doubtlessly in this circumstance. □

On the strength of the theoretical result in Theorem 2, the convergence performance of the proposed neural network disturbed by randomly-generated constant perturbations is discussed in the following remark.
Remark 2: For the proposed DPINN model (10) employed to the robot for tracking the desired trajectory of (2) disturbed by the randomly-generated constant perturbations, the generated residual error $\| \mathbf{s}_k \|_2$ approaches to $O(\delta^2)$ globally. In particular, given that the linear dynamic perturbation $\Psi(t) = \phi(t) + \zeta$ would degrade into the randomly-generated constant one with $\phi = 0$, the conclusion $\lim_{k \to \infty} \| \mathbf{s}_k \|_2 = \| \phi \delta / \kappa \|_2 + O(\delta^2)$ verified in Theorem 2 is converted into $O(\delta^2)$, which indicates that the generated residual error $\| \mathbf{s}_k \|_2$ globally approaches to $O(\delta^2)$ with constant perturbations injected.

Theorem 3: For the proposed DPINN model (10) employed to the robot for tracking the desired trajectory of (2) disturbed by the bounded random perturbations $\Psi(t)$, the generated residual error $\| \mathbf{s}_k \|_2$ is convergent globally with an upper bound of $2(m + 3p) \cdot \sup_{1 \leq i \leq k, 1 \leq \nu \leq (m + 3p)} \| \mathbf{w} \|_1 / (1 - \| P \|_2) + O(\delta^2)$.

Proof: The analysis is able to be discussed from two perspectives similar to that in Theorem 2, i.e., the bounded random perturbations $\Psi(t)$ and $O(\delta^2)$. Then, taking the similar effect as $O(\delta^2)$ in formula (12) with $\Psi_k$, the $b$th subsystem of the proposed DPINN model (10) generates

$$(1 + \gamma) \mathbf{s}_{k+1}^b - \mathbf{s}_k^b + \mathbf{s}_k^b + \Psi_k = 0.$$  

Subtracting the subsystem at time interval $(k - 1)\delta$ from that at time interval $k\delta$ gets

$$(1 + \gamma + \kappa) \mathbf{s}_{k+1}^b - (2 + \gamma) \mathbf{s}_k^b + \mathbf{s}_k^b + \Psi_k - \Psi_{k-1} = 0,$$

which is able to be further rewritten as

$$(1 + \gamma + \kappa) \mathbf{s}_{k+1}^b = (2 + \gamma) \mathbf{s}_k^b + \mathbf{s}_k^b + \Psi_{k-1} - \Psi_k.$$  

To arrange the above formula into a matrix form of state space, define $\mathbf{s}_k^b = [0; \Psi_{k-1} - \Psi_k]$. With the assistant of the definition of $\mathbf{s}_k$ and $\mathbf{P}$ in Theorem 1, it is obtained

$$\mathbf{s}_k^b = P \mathbf{s}_k + \mathbf{e}_k.$$

Afterwards, it is deduced that

$$\| \mathbf{e}_k \|_2 \leq \| \mathbf{e}_k \|_2 + \| \mathbf{e}_k \|_2,$$

$$\| \mathbf{e}_k \|_2 \leq \| \mathbf{P} \|_2 \cdot \| \mathbf{e}_k \|_2 + \| \mathbf{e}_k \|_2,$$

$$\| \mathbf{e}_k \|_2 \leq \| \mathbf{P} \|_2 \cdot \| \mathbf{e}_{k-1} \|_2 + \| \mathbf{P} \|_2 \cdot \| \mathbf{e}_{k-1} \|_2 + \| \mathbf{e}_k \|_2,$$

$$\| \mathbf{e}_k \|_2 \leq \| \mathbf{P} \|_2 \cdot \max_{0 \leq l \leq k} \| \mathbf{e}_l \|_2 / (1 - \| \mathbf{P} \|_2).$$

Review $\lim_{k \to \infty} \| \mathbf{P} \|_2 = 0$ and then $\lim_{k \to \infty} \| \mathbf{s}_{k+1} \|_2 < 2 \max_{0 \leq l \leq k} \| \mathbf{e}_l \|_2 / (1 - \| \mathbf{P} \|_2)$ is achieved. In consequence, $\lim_{k \to \infty} \| \mathbf{s}_k \|_2 < 2(m + 3p) \cdot \sup_{1 \leq i \leq k, 1 \leq \nu \leq (m + 3p)} \| \mathbf{w} \|_1 / (1 - \| \mathbf{P} \|_2) + O(\delta^2)$ is finally demonstrated.

IV. NUMERICAL SIMULATIONS

Hereinafter, the explanatory simulations are devised and organized to embody the practicability and availability of the proposed neural network. Additionally, some of the existing typical methods are employed here for comparison purposes. First, these methods are introduced below.

Hereby, two classic algorithms for managing constrained QP with inequality and bound constraint named 94LVI [42] and E47 [43] are introduced, which are similar in managing this type of problem by converting it equally into a piecewise-linear projection equation as below:

$$h(s(t)) = s(t) - f(s(t) - (M(t)s(t) + \mathbf{f}(t))) = 0,$$

of which the auxiliary variables $M(t) = [E_{p \times p} - F^T(\theta(t)); J(\theta(t))0] \in \mathbb{R}^{(p+m) \times (p+m)}$; $\mathbf{f}(t) = (0; -\hat{\rho}_d(t)) \in \mathbb{R}^{p+m}; s(t) = [\hat{\theta}(t); \mathbf{v}(t)] \in \mathbb{R}^{p+m}$ with the dual decision vector $\mathbf{v}(t)$ related to the equality constraint in (2); the mapping function $F(\cdot)$ defined as

$$F(\kappa) = \begin{cases} b_1^+, & \kappa_1 < b_1^- \\ b_1^-, & b_1^- \leq \kappa_1 \leq b_1^+ \\ b_1^+, & \kappa_1 > b_1^+. \end{cases}$$

It has been verified in [43] that managing (2) is equivalent to compute the solution $s(t)$ to the above formula. Besides, some other definitions are set as $s^+(t) = [\hat{\theta}(t); -\beta 1] \in \mathbb{R}^{p+m}$; $s^-(t) = [\hat{\theta}^-(t); \beta 1] \in \mathbb{R}^{p+m}; \beta \in \mathbb{R}$ tending to infinity; $\mathbf{1}$ a vector composed by 1 being each element. Then, the 94LVI algorithm and the E47 one are described in detail below.

Algorithm 1 94LVI

1. Preface coefficient matrices or vectors $J(\theta(t))$, $\hat{\rho}_d(t)$, $\hat{\theta}^-(t)$, $\hat{\theta}^+(t)$ and the randomly initial variable $\hat{\theta}(0)$.
2. Obtain $M(t)$, $\mathbf{f}(t)$, $s^-(t)$, $s^+(t)$ and $s(0)$.
3. Compute $h(s(t))$ and judge whether $\| h(s(t)) \|_2 < 10^{-3}$. If yes, output $s_k$ where $\hat{\theta}_k$ to be solved constitues the first $p$ terms of $s_k$. If no, go to step 4.
4. Calculate

$$\sigma(s_k) = (M^T + E_{(p+m) \times (p+m)})h(s_k)$$

$$\mathbf{f}(s_k) = \| h(s_k) \|_2^2 / \| \sigma(s_k) \|_2^2$$

and then one is able to be obtained

$$s_k = s_k - \| \mathbf{f}(s_k) \|_2 \sigma(s_k).$$

5. Update the index $k$ to the next time instant $k + 1$, and return to step 3 until the last moment comes.
Algorithm 2 E47

1. Preset coefficient matrices or vectors $J(\theta(t))$, $\rho_d(t)$, $\dot{\theta}^ -(t)$, $\dot{\theta}^ + (t)$ and the randomly initial variable $\bar{\theta}(0)$.
2. Obtain $M(t)$, $f(t)$, $s^ -(t)$, $s^ + (t)$ and $s(0)$.
3. Compute $h(s_k)$ and judge whether $\|h(s_k)\|_2 < 10^{-3}$. If yes, output $s_k$ where $\hat{\theta}_k$ to be solved constitutes the first $p$ terms of $s_k$. If no, go to step 4.
4. Calculate

$$
\sigma(s_k) = M_k^T h(s_k) + M_k s_k + f_k
$$

$$
\xi(s_k) = \left\| h(s_k) \right\|^2 \left( M_k^T + E_{(p+m)X(p+m)} h(s_k) \right) \frac{2}{2}
$$

and then one is able to obtain

$$
s_k = F(s_k - \xi(s_k) \sigma(s_k)). \quad (20)
$$

5. Update the index $k$ to the next time instant $k + 1$, and return to step 3 until the last moment comes.

Secondly, for promoting the simulation process, the adopted parameters are defined as the dimensions $p = 6$ and $m = 3$; the desired path being the plum blossom curve; the whole execution time $T = 7$ s; the sampling gap $\delta = 0.001$ s; the selected robot being UR5 [44]; the lower bound $\dot{\theta}^ -(t)$ and the upper bound $\dot{\theta}^ + (t)$ being set as $[-0.35]_{6 \times 1}$ rad/s and $[0.35]_{6 \times 1}$ rad/s, respectively; the injected perturbation $\psi$ is supposed to random one that $\psi \in [3, 7]_{6 \times 1}$; the index reflecting the motion effect of the robot $\Delta = \rho - \rho_d$ embodying the distance between the generated path and the expected one. Besides, the design parameters related to the models are set as $\gamma = \kappa = 1000$. In accordance with these definitions, the corresponding simulations are conducted on a personal computer with an Intel processor (Core i5-8500, 3.0 GHz), a Windows 10 (64-bit) operating system and MATLAB R2016a software and the results are concluded as portrayed in Figure 3 through Figure 5.

The experimental results are analyzed as follows. First, it should be clear that in practical applications, all kinds of environmental perturbations and equipment errors that may exist (such as impulse noise and harmonic noise) are regarded as random ones in the simulation. How the proposed DPINN model (10) drives the redundant robot to track a determined trajectory within the preset time is portrayed in Figure 3. Notably, although the random perturbation is considered in the tracking process, the destination path is completed successfully with the generated trajectory coinciding perfectly with the desired one as portrayed in Figure 3(a). As a continuation, the UR5 robot driven by the proposed DPINN model (10) performs the given plum curve during the whole time, and the joint angles move smoothly without singularity and malformation in Figure 3(b). It is explicit that the constraint imposed on the joint velocity $\dot{\theta}^ -(t) \leq \dot{\theta}(t) \leq \dot{\theta}^ + (t)$ takes effect slickly with the joint velocities portrayed in Figure 3(c) all controlled in the restrictions $[-0.35, 0.35]$ rad/s with success. It is worth noting that the constraints in the joint velocity are indispensable in practice for preventing the robots from breaking down and ensuring the safety of the system. Ulteriorly, the explicit position errors at X-axis, Y-axis and Z-axis are recorded over time as portrayed in Figure 3(d), which are all of the order $10^{-4}$ m.
FIGURE 5. Simulations on the UR5 robot for dynamically tracking the plum curve driven by the introduced E47 method (20) with random perturbation $\psi(t) \in [3. 7]_{6 \times 1}$. (a) Top view of the desired trajectory and the generated one. (b) Trajectory tracking process. (c) Joint velocity. (d) Position error.

FIGURE 6. Joint angles of the UR5 robot for dynamically tracking the plum curve driven by the introduced 94LVI method (19) and the introduced 94LVI method (20) with random perturbation $\psi(t) \in [3. 7]_{6 \times 1}$. (a) Changing of joint angle by (19). (b) Changing of joint angle by (20).

TABLE 1. Performance comparisons among different networks for redundant robot dynamic motion control.

| Network          | Continuous vs Discrete (hardware implementation) | Time-dependent problems (lagging error) | Disturbance resistance | Robot physical constraint | Computing complexity ($\hat{m} = m + p$) |
|------------------|-------------------------------------------------|----------------------------------------|------------------------|--------------------------|-----------------------------------------|
| DPINN (10)       | Discrete (✓)                                    | Yes (✓)                                | Yes                    | Yes                      | $\hat{m}^3 + 9\hat{m}^2 + 6\hat{m}$     |
| 94LVI (19)       | Discrete (✓)                                    | Yes (✓)                                | No                     | Yes                      | $2\hat{m}^3 + 2\hat{m}^2 + 4\hat{m} - 1$ |
| E47 (20)         | Discrete (✓)                                    | Yes (✓)                                | No                     | Yes                      | $2\hat{m}^3 + 4\hat{m}^2 + 7\hat{m} - 1$ |
| NI [17]          | Discrete (✓)                                    | No (✓)                                 | No                     | No                       | $\hat{m}^3 + 4\hat{m}^2$                |
| GNN [18]         | Continuous (✓)                                 | No (✓)                                 | No                     | No                       | $5\hat{m}^2$                            |
| Network [28]     | Discrete (✓)                                    | Yes (✓)                                | No                     | No                       | $2\hat{m}^3 + 10\hat{m}^2 + 9\hat{m}$   |
| Network [14] [29]| Discrete (✓)                                    | Yes (✓)                                | No                     | No                       | $\hat{m}^3 + 9\hat{m}^2 + 6\hat{m}$     |
| Network [40]     | Continuous (✓)                                 | Yes (✓)                                | No                     | Yes                      | $4\hat{m}^3 + 4\hat{m}^2 + 6\hat{m} + 2\hat{m}$ |
| Network [16] [29]| Discrete (✓)                                    | Yes (✓)                                | No                     | No                       | $\hat{m}^3 + 7\hat{m}^2 + 2\hat{m}$     |

accounting for the accurate implementation and superiority of the proposed DPINN model (10).

The contrast simulations synthesized by the introduced 94LVI method (19) and E47 method (20), as portrayed in Figure 4 and Figure 5 share similar consequences and poor performance in perturbation immunity. Specifically, Figure 4(a) and Figure 5(a) illustrate that the actual trajectories generated by the two methods significantly deviate from the theoretical ones, indicating that the expected effects cannot be obtained by the two methods in practice with random perturbations. In addition, Figure 4(b) and Figure 5(b) show that the movement of each robot manipulator is chaotic during path planning. To understand the changes of the joint angles more clearly, Figure 6 is given corresponding to the introduced 94LVI method (19) and E47 method (20), where the sudden change existing in joint angles is potentially dangerous for the robot because, in practice, it is likely to exceed the allowable angle of robot motion and cause damage to the robot. Figure 4(c) and Figure 5(c) present that the joint velocities hit a large range belonging to $[-5, 10]$ rad/s, which is against reality, and that the constraint imposed on the angular velocity is null and void. Moreover, the position errors synthesized by the introduced 94LVI method (19) and E47 method (20) are both of the order $10^{-2}$ m, which are higher than that of the proposed DPINN model (10). All the results reveal the unpromising performance of the introduced 94LVI method (19) and E47 method (20) in managing the situations with perturbations, and the vigorous and predominant perturbation immunity ability and practicability of the proposed DPINN model (10).

For the purpose of enhancing the effectiveness of the proposed model (10), the introduced 94LVI method (19), E47 method (20), Newton iteration method (NI) [17], gradient neural network (GNN) [18] and networks in [28], [29], [40] are brought in as a contrast, mainly concerning the structural characteristics of these networks and the industrial application effect. The results are shown in Table 1. Specifically, on behalf of better implementing the resolution algorithm in hardware and even extending it to practical areas, the model is usually required to be discrete with satisfactory performance. Second, most problems in practice being dynamic or time-dependent appeal for various methods capable of handling such problems. It can be found that even though the dual neural network in [40] is capable of settling dynamic problems, there is a significant lag error, greatly reducing its accuracy and feasibility. Moreover, only the proposed DPINN model (10) and model (14) [29] are
equipped with the content anti-disturbance performance, nevertheless, the latter does not consider the potential robot constraints in robot control, such as the joint velocity constraint, which is of the essence to protect the robot from singularities or even damage when performing tasks and ensure the safe and smooth operation of the whole control system. Finally, a comparative analysis of the computational complexity [45] shows that the complexities of the networks are roughly of the order \( (p + m)^3 \), which is related to the dimension of the selected robot’s Jacobian matrix. It is worth noting that the time cost by this level of computational complexity is small given the general hardware capabilities, and that the computational complexity is mainly attributed to the matrix inversion, which is able to be eliminated by adopting matrix inverse estimation algorithms. For instance, the Davidon-Fletcher-Powell algorithm and the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno method. Overall, the proposed DPINN model (10) manifests exceptional problem-solving ability and robust stability in redundant robot control.

In particular, the NI method [17] and Network (16) [29] are introduced here for comparison with the proposed DPINN model (10) for dynamically tracking the plum curve disturbed by the random perturbation \( \psi(t) \in [3.7]_{6 \times 1} \) with the results portrayed in Figure 7 and Figure 8. It is clearly seen from Figure 7 that in front of the random perturbations, the motion control system of the robot constructed by the NI method [17] completely breaks down, thus failing to complete the assigned task. By the way, the joint speed reaches the value of the order \( 10^7 \) rad/s, which is not allowed in practical application and the equipment is vulnerable to damage. Besides, Figure 8 presents the tracking process synthesized by Network (16) [29], which illustrates the fragility and instability of the network in the noisy environment as well. The corresponding analysis is similar to that in Figure 7 and thus omitted. A conclusion is able to be drawn from the above simulations that the proposed DPINN model (10) is endowed with robustness stability and control efficiency, which should be noticeable for wide application. Additionally, the proposed DPINN model (10) and the introduced 94LVI method (19) are selected to complete the task in the

![Figure 7](image-url)

**Figure 7.** Simulations on the UR5 robot for dynamically tracking the plum curve driven by NI method [17] with random perturbation \( \psi(t) \in [3.7]_{6 \times 1} \).

(a) Top view of the desired trajectory and the generated one. (b) Trajectory tracking process. (c) Joint velocity. (d) Position error.

![Figure 8](image-url)

**Figure 8.** Simulations on the UR5 robot for dynamically tracking the plum curve driven by Network (16) [29] with random perturbation \( \psi(t) \in [3.7]_{6 \times 1} \).

(a) Top view of the desired trajectory and the generated one. (b) Trajectory tracking process. (c) Joint velocity. (d) Position error.

![Figure 9](image-url)

**Figure 9.** Snapshots on the UR5 robot for dynamically tracking the plum curve driven by the proposed DPINN model (10) with random perturbation \( \psi(t) \in [3.7]_{6 \times 1} \) in the V-rep.
virtual robot experimentation platform (V-rep) perturbed by random noises. The snapshots captured in the movement process are described in Figures 9 and 10. It is evident that the proposed DPINN model (10) is still valid when the latter shows poor performance. Overall, the above-mentioned experimental results adequately substantiate the high accuracy and robustness of the proposed DPINN model (10) for dynamic robot control.

V. CONCLUSION

This paper has proposed a DPINN model for managing dynamic constrained redundant robot control, providing neural-dynamic thought in controlling robots to complete motion generation, task realization and other missions ining dynamic constrained redundant robot control, provid-eracy and robustness of the proposed DPINN model (10) for dynamic robot control.

REFERENCES

[1] J. Yan, X. Xiao, H. Li, J. Zhang, Y. Yan, and M. Liu, “Noise-tolerant zeroing neural network for solving non-stationary Lyapunov equation,” IEEE Access, vol. 7, pp. 41517–41524, 2019.
[2] N. Najmaei and M. R. Kermani, “Applications of artificial intelligence in safe Human–Robot interactions,” IEEE Trans. Syst., Man, Cybern., B (Cybern.), vol. 41, no. 2, pp. 448–459, Apr. 2011.
[3] W. He, W. Ge, Y. Li, Y.-J. Liu, C. Yang, and C. Sun, “Model identification and control design for a humanoid robot,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 47, no. 1, pp. 45–57, Jan. 2017.
[4] H. M. La, W. Sheng, and J. Chen, “Cooperative and active sensing in mobile sensor networks for scalar field mapping,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 45, no. 1, pp. 1–12, Jan. 2015.
[5] L. Jin, S. Li, H. M. La, and X. Luo, “Manipulability optimization of redundant manipulators using dynamic neural networks,” IEEE Trans. Ind. Electron., vol. 64, no. 6, pp. 4710–4720, Jun. 2017.
[6] Y. Yang and J. Cao, “Solving quadratic programming problems by delayed projection neural network,” IEEE Trans. Neural Netw., vol. 17, no. 6, pp. 1630–1634, Nov. 2006.
[7] X. Gao and L.-Z. Liao, “A new one-layer neural network for linear and quadratic programming,” IEEE Trans. Neural Netw., vol. 21, no. 6, pp. 918–929, Jun. 2010.
[8] A. Zou, Z. Hou, S. Fu, and M. Tan, “Neural networks for mobile robot navigation: A survey,” in Adv. Neural Networks: 3rd Int. Symp. Neural Netw. (ISNN), vol. 2006, pp. 1218–1226.
[9] L. Chen, Z. Huang, Y. Li, N. Zeng, M. Liu, A. Peng, and L. Jin, “Weight and structure determination neural network aided with double pseudoinversion for diagnosis of flat foot,” IEEE Access, vol. 7, pp. 33001–33008, 2019.
[10] L. Jin, Z. Huang, Y. Li, Z. Sun, H. Li, and J. Zhang, “On modified multi-output chebyshev-polynomial feed-forward neural network for pattern classification of wine regions,” IEEE Access, vol. 7, pp. 1973–1980, 2019.
[11] X. Luo, P.-F. Zhang, Z. Huang, L. Nie, and X.-S. Xu, “Discrete hashing with multiple supervision,” IEEE Trans. Image Process., vol. 28, no. 6, pp. 2962–2975, Jun. 2019.
[12] X. Luo, Y. Wang, Y. Hao, X.-J. Li, C.-M. Liu, X.-X. Fei, C.-H. Ya, and P. Cao, “Research of single-event burnout and hardening of AlGaN/GaIn-based MIFSET,” IEEE Trans. Electron Devices, vol. 66, no. 2, pp. 1118–1122, Feb. 2019.
[13] S. Li, M. Zhou, and X. Luo, “Modified dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 10, pp. 1–11, Dec. 2017.
[14] S. Hu and J. Wang, “Global asymptotic stability and global exponential stability of continuous-time recurrent neural networks,” IEEE Trans. Autom. Control, vol. 47, no. 5, pp. 802–807, May 2002.
[15] K. Chen, “Robustness analysis of wang neural network for online linear equation solving,” Electron. Lett., vol. 48, no. 22, p. 1391, 2012.
[16] Z. Li and Z. Yin, “Extended wang neural network for online solving a set of linear equations,” Electron. Lett., vol. 52, no. 12, pp. 1001–1003, Jun. 2016.
[17] Y. Zhang, Z. Zhang, D. Chen, Z. Xiao, and X. Yan, “From davidenko method to zhang dynamics for nonlinear equation systems solving,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 47, no. 11, pp. 2817–2830, Nov. 2017.
[18] Y. Zhang, J. Wang, T. Qiao, Z. Xiao, and M. Mao, “The second-order ZD, GD and hybrid systems solving nonlinear equations compared with other dynamics,” in Proc. Chin. Control Decis. Conf. (CCDC), May 2016, pp. 933–938.
[19] Y. Zhang and C. Yi, Neural Networks and Neural-Dynamic Method, New York, NY, USA: Nova, 2011.
[20] L. Jin, J. Yan, X. Du, X. Xiao, and D. Fu, “RNN for solving time-variant generalized Sylvester equation with applications to robots and acoustic source localization,” IEEE Trans. Ind. Inform., early access, Jan. 8, 2020, doi: 10.1109/TII.2020.2964817.
[21] Y. Zhang, S. Li, and X. Liu, “Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 6227–6241, Dec. 2018.
[22] L. Jin, Y. Zhang, and S. Li, “Integration-enhanced zhang neural network for Real-Time-Varying matrix inversion in the presence of various kinds of noises,” IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 12, pp. 2615–2627, Dec. 2016.
[23] Y. Qi, L. Jin, H. Li, Y. Li, and M. Liu, “Discrete computational neural dynamics models for solving time-dependent Sylvester equations with applications to robotics and MIMO systems,” IEEE Trans. Ind. Inform., early access, Jan. 14, 2020, doi: 10.1109/TII.2020.2966544.
[24] L. Jin, S. Li, and B. Hu, “RNN models for dynamic matrix inversion: A control-theoretical perspective,” IEEE Trans. Ind. Inform., vol. 14, no. 1, pp. 189–199, Jan. 2018.
[25] L. Xiao, “Accelerating a recurrent neural network to finite-time convergence using a new design formula and its application to time-varying matrix square root,” J. Franklin Inst., vol. 354, no. 13, pp. 5667–5677, Sep. 2017.
[26] L. Xiao, B. Liao, S. Li, Z. Zhang, L. Ding, and L. Jin, “Design and analysis of FTZINN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator,” IEEE Trans. Ind. Inform., vol. 14, no. 1, pp. 98–105, Jan. 2018.
JUN YANG received the M.S. degree from the School of Computer, Guangdong University of Technology, China, in 2010, and the Ph.D. degree from the School of Information Science and Technology, Sun Yat-sen University, China, in 2016. He joined the College of Mathematics and Information Engineering, Jiaxing University, where he has been a Lecturer, since July 2016. His current research interests include digital watermark, multimedia information analysis and processing, pattern recogntion.

LI CHEN received the Ph.D. degree in computer software and theory from the Harbin Institute of Technology.

She is currently a Lecturer with the College of Mathematics and Information Engineering, Jiaxing University, China. Her research interests include machine learning and mobile communications technology.

MEI LIU received the B.E. degree in communication engineering from Yantai University, Yantai, China, in 2011, and the M.E. degree in pattern recognition and intelligent system from Sun Yat-sen University, Guangzhou, China, in 2014. She is currently a Teacher with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Before joining Lanzhou University, in 2017, she was a Lecturer with the College of Physics, Mechanical and Electrical Engineering, Jishou University, Jishou, China. Her main research interests include neural networks and robotics.

CHEN CUI received the B.S. and M.E. degrees in software engineering and the Ph.D. degree in computer science from the Harbin Institute of Technology, China, in 2006, 2009, and 2016, respectively. He is currently an Associate Professor with the School of Information Science and Technology, Heilongjiang University, China. His research interests include digital watermarking, digital forensics and image processing, and so on.

* * *

[27] Y. Zhang, H. Qiu, C. Peng, Y. Shi, and H. Tan, “Simply and effectively proved square characteristics of discrete-time zd solving systems of time-varying nonlinear equations,” in Proc. IEEE Int. Conf. Inf. Autom., Aug. 2015, pp. 1457–1462.

[28] Z. Yin, K. Liu, Y. Chen, R. Xue, J. Li, and Y. Shi, “Performance analyses of four-instant discretization formulas with application to Generalized-Sylvester-Type future matrix equation,” IEEE Access, vol. 7, pp. 152258–152266, 2019, doi: 10.1109/ACCESS.2019.2948082.

[29] Y. Qi, L. Jin, Y. Wang, L. Xiao, and J. Zhang, “Complex-valued discrete-time neural dynamics for perturbed time-dependent complex quadratic programming with applications,” IEEE Trans. Neural Netw. Learn. Syst., early access, Nov. 8, 2019, doi: 10.1109/TNNLS.2019.2944992.

[30] L. Wei, L. Jin, C. Yang, K. Chen, and W. Li, “New noise-tolerant neural algorithms for future dynamic nonlinear optimization with estimation on hessian matrix inversion,” IEEE Trans. Syst., Man, Cybern. Syst., early access, May 27, 2019, doi: 10.1109/TSMC.2019.2916892.

[31] D. Guo, K. Li, and B. Liao, “Bi-criteria minimization with MWVN-INAM type for motion planning and control of redundant robot manipulators,” Robotica, vol. 36, no. 5, pp. 655–675, May 2018.

[32] L. Jin, S. Li, J. Yu, and J. He, “Robot manipulator control using neural networks: A survey,” Neurocomputing, vol. 285, pp. 23–34, Apr. 2018.

[33] Y. Zhang and S. Li, “A neural controller for image-based visual servoing of manipulators with physical constraints,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 11, pp. 5419–5429, Nov. 2018.

[34] L. Jin and S. Li, “Distributed task allocation of multiple robots: A control perspective,” IEEE Trans. Control Syst. Technol., vol. 100, pp. 75–81, Feb. 2019.

[35] Y. Zhang, S. Li, J. Gui, and X. Luo, “Velocity-level control with compliancy to acceleration-level constraints: A novel scheme for manipulator redundancy resolution,” IEEE Trans. Ind. Informat., vol. 14, no. 3, pp. 921–930, Mar. 2018.

[36] H. Lu, L. Jin, X. Luo, B. Guo, and L. Xiao, “RNN for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables,” IEEE Trans. Ind. Informat., vol. 15, no. 11, pp. 5931–5942, Nov. 2019.

[37] Z. Xie, L. Jin, X. Luo, S. Li, and X. Xiao, “A data-driven cyclic-motion generation scheme for kinematic control of redundant manipulators,” IEEE Trans. Control Syst. Technol., early access, Jan. 15, 2020, doi: 10.1109/TCST.2019.2963017.

[38] Y. Zhang, S. Li, J. Gui, and X. Luo, “Velocity-level control with compliance to acceleration-level constraints: A novel scheme for manipulator redundancy resolution,” IEEE Trans. Ind. Informat., vol. 14, no. 3, pp. 921–930, Mar. 2018.

[39] L. Jin, S. Li, J. Yu, and J. He, “Robot manipulator control using neural networks: A survey,” Neurocomputing, vol. 285, pp. 23–34, Apr. 2018.

[40] Y. Zhang and S. Li, “A neural controller for image-based visual servoing of manipulators with physical constraints,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 11, pp. 5419–5429, Nov. 2018.