

Pose control of constrained redundant arm using recurrent neural networks and one-iteration computing algorithm[☆]

Min Yang^a, Yunong Zhang^b, Xuefeng Zhou^c, Haifeng Hu^{a,*}

^a School of Electronics and Information Technology, Sun Yat-sen University (SYSU), Guangzhou 510006, China

^b School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou 510006, China

^c Guangdong Key Laboratory of Modern Control Technology, Guangzhou 510070, China

ARTICLE INFO

Article history:

Received 5 June 2020

Received in revised form 2 February 2021

Accepted 12 October 2021

Available online 30 October 2021

Keywords:

Pose control

Redundant arm

Physical constraints

Zeroing neural network

Projection neural network

One-iteration computing algorithm

ABSTRACT

Accuracy and safety are two important factors in redundant arm control. Generally, physical constraints should be taken into account for safe operation of redundant arm. By investigating the kinematics of redundant arm, the formulation of end-effector's position and orientation is presented in detail. By utilizing zeroing neural network method, a pose control scheme considering physical constraints is proposed. For solving the pose control scheme, it is reformulated as a standard quadratic programming at joint angular velocity level. Then, a projection neural network solver is developed to handle the quadratic programming. Further, for better computer operation and real-time requirements, the discrete algorithm is desired. As a result, a one-iteration computing algorithm is developed for pose control of redundant arm by discretizing the projection neural network solver. Theoretical results are presented to guarantee the properties of the proposed scheme and the one-iteration computing algorithm. In addition, plenty of comparative experiments on the basis of UR5 arm verify the efficacy and superiority of the one-iteration computing algorithm, and substantiate theoretical results.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

With the rapid development of science and technology, robots will be an important part of the production and life in the future. As a significant component of a robot, or seen as a simple robot, redundant arm (or termed serial redundant manipulator) has attracted much attention [1–3]. The redundant arm has been applied to taking the place of human in many industrial processes. The mechanical structure of redundant arm consists of a sequence of links interconnected by means of joints [1]. Besides, the redundant arm is equipped with an end-effector that performs the task. Redundant arms are divided into two categories: planar redundant arm [4] and three-dimensional (3-D) redundant arm [5]. The 3-D redundant arm is more popular and effective,

which is investigated in this paper. Besides, the redundant arm has more degrees of freedom (DOF) than the workspace, which is meaningful [1].

Because of the important role of redundant arm, many control algorithms have been proposed, developed, and investigated [6–12]. For example, recurrent neural network (RNN) method [6, 13, 14] and fuzzy control method [15] can be applied to redundant arm control. In [7], an adaptive scheme with parameters estimation for redundant arm control was proposed, which has finite-time convergence property. Jin et al. proposed a dynamic neural network model, which optimizes the manipulability of redundant arm [8]. In [9], a distributed control algorithm was developed for cooperative control of redundant arms. Li et al. designed a modified dual neural network model to eliminate harmonic noises for redundant arm control [10]. An adaptive controller combined with the radial basis function neural network was developed by Yang et al. for redundant arms [3]. He et al. proposed a reinforcement learning control strategy for a two-link arm [16].

Considering physical constraints [1] of redundant arm, the tracking control becomes more practical and also complicated. When controlling the redundant arm, it may be damaged if physical constraints are not satisfied. Generally speaking, physical constraints are formulated in inequality form, which is difficult to be handled. In this case, many researchers have devoted themselves into this topic [17–19]. For example, Chen et al. developed

[☆] The work is aided by the National Natural Science Foundation of China (with numbers 61976230 and 62076262), the Key-Area Research and Development Program of Guangzhou, China (with number 202007030004), the Research Fund Program of Guangdong Key Laboratory of Modern Control Technology (with number 2017B030314165), the Natural Science Foundation of Guangdong Province, China (with number 2017A030311029), and also the Project Supported by Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (with number 2018).

* Corresponding author.

E-mail addresses: yangm46@mail2.sysu.edu.cn (M. Yang), zhynong@mail.sysu.edu.cn (Y. Zhang), xf.zhou@gia.ac.cn (X. Zhou), huhai@fmail.sysu.edu.cn (H. Hu).

a multi-layer simultaneous optimal scheme for jerk-bounded redundant arm in [17]. In [18], considering joint angle constraints, joint angular velocity constraints, and joint angular acceleration constraints, a minimum-acceleration-norm scheme was presented for obstacle avoidance of redundant arm. A minimum-velocity-norm coordinating scheme with joint angle constraints and joint angular velocity constraints considered was proposed and investigated in [19]. In [20], a novel RNN was applied to optimal motion control of mobile arms with joint angular velocity constraints.

Note that the above control methods/algorithms are mainly developed for position control of redundant arm. In addition to the position control, orientation control is also significant for redundant arm. Note that pose control includes position control and orientation control. In many situations, although the position is correct, the task cannot be completed if the orientation is not appropriate. In [19], the orientation control of redundant arm was investigated briefly with physical constraints. Lai et al. proposed a position-orientation control method utilizing genetic algorithm for planar four-link redundant arm [21]. In [22], another position-orientation control method using intelligent optimization was developed for planar three-link redundant arm. Actually, the above algorithms are not very effective for time-varying tracking tasks.

Different from existing results, the pose control of redundant arm with physical constraints is investigated in depth in this paper. By investigating the kinematics of redundant arm, the position formulation and orientation formulation are presented, which are straightforward and effective. Then, by applying zeroing neural network (ZNN, also termed Zhang neural network) method [23–28] and considering joint angle constraints as well as joint angular velocity constraints, a pose control scheme is proposed for redundant arm, which is presented as an optimization problem [29,30]. By using some mathematical knowledge, the pose control scheme is reformulated as a standard quadratic programming (QP) [17] at joint angular velocity level. Further, a projection neural network (PNN) [1,31] model is developed as the QP solver for pose control of redundant arm. Note that ZNN and PNN both belong to RNN. In practical, the continuous control algorithm should be transformed into discrete control algorithm for better computer operation and real-time control requirements [24,32]. For example, a discrete ZNN model was developed for redundant arm control with guaranteed motion precision in [32]. Therefore, a one-iteration computing algorithm is further developed to realize the pose control of redundant arm by referring to the previous work [1,33].

The remainder of the paper is divided into five sections. In the next section, the kinematics modeling including the position and orientation of end-effector is investigated. In Section 3, the pose control scheme considering physical constraints is designed and proposed. The QP formulation and corresponding solvers are presented in Section 4. Section 5 shows experimental results and comparisons to substantiate the efficacy and superiority of the method developed for pose control of redundant arm. A conclusion is given in the last section. At the end of this section, the main contributions are summarized as below.

1. By investigating the kinematics formulation of redundant arm and utilizing ZNN method, a pose control scheme is first proposed and investigated.
2. The pose control scheme is reformulated as a standard QP at joint angular velocity level, and a PNN solver is developed with theoretical analyses.
3. For better computer operation as well as real-time requirements, a one-iteration computing algorithm is further developed to realize pose control of redundant arm.
4. Sufficient experiments are conducted, which verify the efficacy and superiority of the proposed scheme and solvers for pose control of redundant arm.

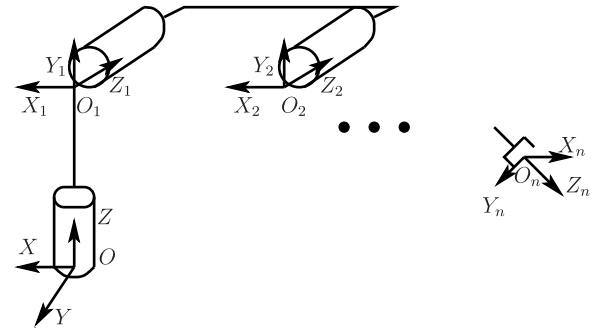


Fig. 1. 3-D model of redundant arm.

2. Kinematics modeling of redundant arm

Let us take an n -DOF redundant arm as an example, which works in 3-D space. The joint angle vector is termed $\theta \in \mathbb{R}^n$. The 3-D model of redundant arm is presented in Fig. 1. According to the knowledge of redundant arm [1], the following matrix is computed:

$$A_i^{i-1} = \begin{bmatrix} \cos \theta^i & -\sin \theta^i \cos \alpha_i & \sin \theta^i \sin \alpha_i & a_i \cos \theta^i \\ \sin \theta^i & \cos \theta^i \cos \alpha_i & -\cos \theta^i \sin \alpha_i & a_i \sin \theta^i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where d_i , a_i , and α_i are Denavit–Hartenberg (D–H) parameters, and $i = 1, 2, \dots, n$. Besides, θ^i denotes the i th element of joint angle vector θ . Then, one computes the coordinate transformation matrix relative to the base coordinate system O -XYZ [1]:

$$T_j^0 = A_1^0 \cdots A_j^{j-1},$$

where $j = 1, 2, \dots, n$. Concretely, T_n^0 has the following form of entries [1]:

$$T_n^0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Generally, considering the position and orientation in real-time t , one obtains the corresponding kinematics equations. Actually, defining nonlinear function $f(\theta(t)) = [p_x, p_y, p_z]^T$, one has the widely-known kinematics equation [1]:

$$f(\theta(t)) = \mathbf{r}_a(t), \quad (1)$$

where $\mathbf{r}_a(t) \in \mathbb{R}^3$ denotes the actual position of the end-effector at time t , and $\theta(t) \in \mathbb{R}^n$ denotes the joint angle vector. Computing the time derivative on both sides of (1) yields

$$J_1(\theta(t))\dot{\theta}(t) = \dot{\mathbf{r}}_a(t), \quad (2)$$

where $J_1(\theta(t)) = \partial f(\theta(t))/\partial \theta(t) \in \mathbb{R}^{3 \times n}$ is termed position Jacobian matrix, $\dot{\theta}(t) \in \mathbb{R}^n$ denotes the joint angular velocity vector, and $\dot{\mathbf{r}}_a(t)$ denotes the actual position-velocity vector of the end-effector.

The orientation of the end-effector can be described by vector $[a_x, a_y, a_z]^T$, which is straightforward, effective, and convenient. Hence, defining nonlinear function $g(\theta(t)) = [a_x, a_y, a_z]^T$, one obtains

$$g(\theta(t)) = \mathbf{o}_a(t), \quad (3)$$

where $\mathbf{o}_a(t) \in \mathbb{R}^3$ denotes the actual orientation of the end-effector at time t . Besides, note that $\mathbf{o}_a(t)$ is dimensionless and

2-norm $\|\mathbf{o}_a(t)\|_2 = 1$ is satisfied. In like manner, computing the time derivative on both sides of (3) yields

$$J_2(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) = \dot{\mathbf{o}}_a(t), \quad (4)$$

where $J_2(\boldsymbol{\theta}(t)) = \partial g(\boldsymbol{\theta}(t))/\partial \boldsymbol{\theta}(t) \in \mathbb{R}^{3 \times n}$ is termed orientation Jacobian matrix, and $\dot{\mathbf{o}}_a(t)$ denotes the actual orientation-velocity vector of the end-effector. On the ensuing pages, time t is usually omitted for brevity.

3. Design of pose control scheme

Given a desired (time-varying) path $\mathbf{r}_d \in \mathbb{R}^3$ and according to (2), the conventional pseudo-inverse-based (PIB) model is formulated as follows [32]:

$$\dot{\boldsymbol{\theta}} = J_1^\dagger(\boldsymbol{\theta})\dot{\mathbf{r}}_d, \quad (5)$$

where superscript † denotes the pseudo-inverse operator, and $\dot{\mathbf{r}}_d$ denotes the time derivative of the desired path. However, the above model involves no feedback, which is not effective enough. To construct strict feedback, the following error function is defined:

$$\mathbf{e}_1 = \mathbf{r}_a - \mathbf{r}_d \in \mathbb{R}^3, \quad (6)$$

and ZNN design formula is utilized as $\dot{\mathbf{e}}_1 = -\gamma \mathbf{e}_1$ with design parameter $\gamma > 0$ [23,24]. Therefore, one obtains

$$J_1(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\mathbf{r}}_d - \gamma(\mathbf{r}_a - \mathbf{r}_d).$$

Similarly, (4) has no feedback, which is also not effective enough. To realize the orientation control of the end-effector, one defines another error function as

$$\mathbf{e}_2 = \mathbf{o}_a - \mathbf{o}_d \in \mathbb{R}^3. \quad (7)$$

By utilizing ZNN design formula $\dot{\mathbf{e}}_2 = -\lambda \mathbf{e}_2$ with $\lambda > 0$, the following result is obtained:

$$J_2(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\mathbf{o}}_d - \lambda(\mathbf{o}_a - \mathbf{o}_d).$$

Note that the primary task is to track the desired path. Hence the orientation control is formulated as an optimization index $\|J_2(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \lambda(\mathbf{o}_a - \mathbf{o}_d) - \dot{\mathbf{o}}_d\|_2^2/2$. Evidently, the optimization index is rewritten as

$$\frac{1}{2}\dot{\boldsymbol{\theta}}^T J_2^T(\boldsymbol{\theta})J_2(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + (\lambda(\mathbf{o}_a - \mathbf{o}_d) - \dot{\mathbf{o}}_d)^T J_2(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}.$$

Moreover, considering joint angle constraints and joint angular velocity constraints, the pose control scheme is formulated as follows:

$$\text{minimize } \frac{1}{2}\dot{\boldsymbol{\theta}}^T J_2^T(\boldsymbol{\theta})J_2(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + (\lambda(\mathbf{o}_a - \mathbf{o}_d) - \dot{\mathbf{o}}_d)^T J_2(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \quad (8)$$

$$\text{subject to } J_1(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\mathbf{r}}_d - \gamma(\mathbf{r}_a - \mathbf{r}_d), \quad (9)$$

$$\boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^+, \quad (10)$$

$$\dot{\boldsymbol{\theta}}^- \leq \dot{\boldsymbol{\theta}} \leq \dot{\boldsymbol{\theta}}^+. \quad (11)$$

Remark 1. The selection of the pose control scheme is meaningful and efficient. The corresponding explanation is divided into three parts.

- (1) Considering the equal importance of the position control and orientation control of the end-effector, the simultaneous control of the position and orientation in the pose control scheme has better applicability.

- (2) In addition, the presence of physical constraints (mainly joint angle constraints and joint angular velocity constraints) is an inherent property of redundant arms. Handling physical constraints not only improves the control efficiency but also ensures the operation safety.
- (3) Furthermore, because of the handling of joint angle constraints and joint angular velocity constraints, the pose control scheme should be formulated at the joint angular velocity level for the development of the one-iteration computing algorithm. Therefore, utilizing ZNN method not only introduces strict feedbacks but also helps the formulation of the pose control scheme.

4. QP formulation and solvers

In this section, pose control scheme (8)–(11) is reformulated as a standard QP. Then, QP solvers are developed to handle the standard QP.

4.1. QP formulation

Note that pose control scheme (8)–(11) is resolved at joint angular velocity level, and it is difficult to be solved directly. In this case, joint angle constraints need to be converted into the velocity level. Particularly, $\boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^+$ can be converted into

$$\eta(\boldsymbol{\theta}^- - \boldsymbol{\theta}) \leq \dot{\boldsymbol{\theta}} \leq \eta(\boldsymbol{\theta}^+ - \boldsymbol{\theta}),$$

where $\eta > 0$ is used to scale the feasible region of $\dot{\boldsymbol{\theta}}$ [1]. Furthermore, considering the original constraints $\boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^+$, the combinative constraints are obtained as

$$\boldsymbol{\omega}^- \leq \dot{\boldsymbol{\theta}} \leq \boldsymbol{\omega}^+,$$

where lower bound $\boldsymbol{\omega}^- = \max\{\eta(\boldsymbol{\theta}^- - \boldsymbol{\theta}), \dot{\boldsymbol{\theta}}^-\}$ and upper bound $\boldsymbol{\omega}^+ = \min\{\eta(\boldsymbol{\theta}^+ - \boldsymbol{\theta}), \dot{\boldsymbol{\theta}}^+\}$.

For brevity, one defines

$$Q = J_2^T(\boldsymbol{\theta})J_2(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n},$$

$$\mathbf{p} = J_2^T(\boldsymbol{\theta})(\lambda(\mathbf{o}_a - \mathbf{o}_d) - \dot{\mathbf{o}}_d) \in \mathbb{R}^n,$$

$$A = J_1(\boldsymbol{\theta}) \in \mathbb{R}^{3 \times n},$$

and

$$\mathbf{b} = \dot{\mathbf{r}}_d - \gamma(\mathbf{r}_a - \mathbf{r}_d) \in \mathbb{R}^3.$$

Then, the following standard QP is formulated:

$$\text{minimize } \frac{1}{2}\dot{\boldsymbol{\theta}}^T Q \dot{\boldsymbol{\theta}} + \mathbf{p}^T \dot{\boldsymbol{\theta}}, \quad (12)$$

$$\text{subject to } A\dot{\boldsymbol{\theta}} = \mathbf{b}, \quad (13)$$

$$\boldsymbol{\omega}^- \leq \dot{\boldsymbol{\theta}} \leq \boldsymbol{\omega}^+. \quad (14)$$

The decision variable is $\dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ in the above standard QP.

4.2. Projection neural network

For solving the above standard QP (12)–(14). The following lemma is introduced for further investigation.

Lemma 1. Assume that the optimal solution $\dot{\boldsymbol{\theta}}^*$ of QP (12)–(14) exists. Solving QP (12)–(14) is equivalent to solving the following piecewise-linear projection function [1]:

$$P_{\Omega}(\mathbf{x} - (M\mathbf{x} + \mathbf{c})) - \mathbf{x} = \mathbf{0}, \quad (15)$$

Table 1

Description about one-iteration computing algorithm (17)–(21) for pose control of redundant arm.

1. Input: $\theta_0, \mathbf{x}_0, T, \lambda, \gamma, \tau, \eta, \varsigma, \mathbf{r}_d$, and \mathbf{o}_d
2. For $k = 0 : T/\tau - 1$
3. Compute $\mathbf{e}(\mathbf{x}_k)$ via (17)
4. Compute $\mathbf{d}(\mathbf{x}_k)$ via (18)
5. Compute $\rho(\mathbf{x}_k)$ via (19)
6. Compute \mathbf{x}_{k+1} via (20)
7. Compute θ_{k+1} via (21)
8. Output: θ_{k+1} and θ_{k+1}
9. End

where $\mathbf{x} = [\dot{\theta}; \mathbf{y}] \in \mathbb{R}^{n+3}$ in MATLAB manner [34], $\mathbf{c} = [\mathbf{p}; -\mathbf{b}] \in \mathbb{R}^{n+3}$, and

$$M = \begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \in \mathbb{R}^{(n+3) \times (n+3)}.$$

Besides, $\mathbf{y} \in \mathbb{R}^3$ is the dual decision vector defined corresponding to (13). The upper and lower bounds of \mathbf{x} are defined as follows:

$$\mathbf{x}^+ = \begin{bmatrix} \omega^+ \\ \varsigma \mathbf{1}_v \end{bmatrix} \in \mathbb{R}^{n+3},$$

$$\mathbf{x}^- = \begin{bmatrix} \omega^- \\ -\varsigma \mathbf{1}_v \end{bmatrix} \in \mathbb{R}^{n+3},$$

where $\mathbf{1}_v = [1, 1, 1]^T \in \mathbb{R}^3$, and constant ς should be defined large enough (e.g., 10^6). In addition, set $\Omega = \{\mathbf{x} | \mathbf{x}^- \leq \mathbf{x} \leq \mathbf{x}^+\}$. Moreover, the i th element of projection operation $P_\Omega(\cdot) : \mathbb{R}^{n+3} \rightarrow \Omega$ is defined as

$$[P_\Omega(\mathbf{z})]_i = \begin{cases} x_i^-, & \text{if } z_i \leq x_i^-, \\ z_i, & \text{if } x_i^- < z_i < x_i^+, \\ x_i^+, & \text{if } z_i \geq x_i^+, \end{cases}$$

where $i = 1, 2, \dots, n+3$.

Proof. It can be generalized from [1]. ■

On the basis of Lemma 1, the PNN solver [1] is designed as below.

Lemma 2. To solve QP (12)–(14), with $\delta \in \mathbb{R}^+$ adjusting the convergence rate and I being the identity matrix, the PNN solver is developed as

$$\dot{\mathbf{x}} = \delta(I + M^T)(P_\Omega(\mathbf{x} - (M\mathbf{x} + \mathbf{c})) - \mathbf{x}), \quad (16)$$

Assume that the optimal solution $\dot{\theta}^*$ of QP (12)–(14) exists. Starting from any initial state \mathbf{x}_0 , the variable vector \mathbf{x} of PNN solver (16) converges to the PNN theoretical solution \mathbf{x}^* . The first n elements of \mathbf{x} constitute the optimal solution $\dot{\theta}^*$.

Proof. It can be generalized from [1]. ■

4.3. Numerical algorithm

In practical applications, the QP solver designed in continuous form is not effective enough because the real-time control requirements are usually desired [1,24]. Besides, the discrete algorithm is also convenient for computer operation [24].

Hence, a one-iteration computing algorithm (i.e., QP solver in discrete form) are developed in this subsection by discretizing PNN solver (16). Meanwhile, $(\cdot)_k$ is defined as the value of the vector/matrix at time instant $t_k = k\tau$ with τ being the sampling gap, e.g., $\mathbf{x}_k = \mathbf{x}(t_k)$. Meanwhile, T/τ is assumed to be an integer, in which T is the task duration. Note that the first n elements of \mathbf{x} constitute the joint angular velocity vector $\dot{\theta}$.

Algorithm. With initial value \mathbf{x}_0 and I being the identity matrix, the one-iteration computing algorithm for solving QP (12)–(14) is developed as

$$\mathbf{e}(\mathbf{x}_k) = \mathbf{x}_k - P_\Omega(\mathbf{x}_k - (M_k\mathbf{x}_k + \mathbf{c}_k)), \quad (17)$$

$$\mathbf{d}(\mathbf{x}_k) = M_k^T \mathbf{e}(\mathbf{x}_k) + M_k\mathbf{x}_k + \mathbf{c}_k, \quad (18)$$

$$\rho(\mathbf{x}_k) = \frac{\|\mathbf{e}(\mathbf{x}_k)\|_2^2}{\|(M_k^T + I)\mathbf{e}(\mathbf{x}_k)\|_2^2}, \quad (19)$$

$$\mathbf{x}_{k+1} = P_\Omega(\mathbf{x}_k - \rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k)), \quad (20)$$

$$\theta_{k+1} = \theta_k + \tau \dot{\theta}_{k+1}. \quad (21)$$

Note that the derivation of (17)–(20) can be generalized from [1,33]. Differently, the algorithm developed in this paper just needs one iteration during each sampling gap, while the algorithm in [1,33] needs several iterations. Besides, (21) is obtained by using Euler backward difference method [34]. The algorithm description is presented in Table 1.

Theorem 1. Assume that the theoretical solution \mathbf{x}_k^* of projection function (15) exists. Sequence $\{\mathbf{x}_k\}$ (with update index $k = 0, 1, 2, \dots$) generated by one-iteration computing algorithm (17)–(21) satisfies

$$\|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^*\|_2^2 \leq \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \rho(\mathbf{x}_k)\|\mathbf{e}(\mathbf{x}_k)\|_2^2 + \mathbf{O}(\tau). \quad (22)$$

Proof. See Appendix A. ■

Theorem 2. Assume that the optimal solution $\dot{\theta}_k^*$ of QP (12)–(14) exists. With $\tau \rightarrow 0$ and $k \rightarrow +\infty$, \mathbf{x}_k computed by one-iteration computing algorithm (17)–(21) converges to the theoretical solution \mathbf{x}_k^* . The first n elements of \mathbf{x}_k constitute the optimal solution $\dot{\theta}_k^*$. In addition, θ_k converges to the optimal solution θ_k^* .

Proof. See Appendix B. ■

Remark 2. The computational complexity of the one-iteration computing algorithm is analyzed as follows. First, let $N = n+3$.

- (1) Compute $\mathbf{e}(\mathbf{x}_k) = \mathbf{x}_k - P_\Omega(\mathbf{x}_k - (M_k\mathbf{x}_k + \mathbf{c}_k))$
For $M_k\mathbf{x}_k$: N^2 multiplications and $N(N-1)$ additions.
For $M_k\mathbf{x}_k + \mathbf{c}_k$: N additions.
For $\mathbf{x}_k - P_\Omega(\mathbf{x}_k - (M_k\mathbf{x}_k + \mathbf{c}_k))$: $2N$ additions.
- (2) Compute $\mathbf{d}(\mathbf{x}_k) = M_k^T \mathbf{e}(\mathbf{x}_k) + M_k\mathbf{x}_k + \mathbf{c}_k$
For $M_k^T \mathbf{e}(\mathbf{x}_k)$: N^2 multiplications and $N(N-1)$ additions.
For $M_k^T \mathbf{e}(\mathbf{x}_k) + M_k\mathbf{x}_k + \mathbf{c}_k$: N additions [with $(M_k\mathbf{x}_k + \mathbf{c}_k)$ computed already].
- (3) Compute $\rho(\mathbf{x}_k) = \|\mathbf{e}(\mathbf{x}_k)\|_2^2 / \|(M_k^T + I)\mathbf{e}(\mathbf{x}_k)\|_2^2$
For $\|\mathbf{e}(\mathbf{x}_k)\|_2^2$: N multiplications and $(N-1)$ additions.
For $\|(M_k^T + I)\mathbf{e}(\mathbf{x}_k)\|_2^2$: N multiplications and $(2N-1)$ additions [($M_k^T + I)\mathbf{e}(\mathbf{x}_k) = M_k^T \mathbf{e}(\mathbf{x}_k) + \mathbf{e}(\mathbf{x}_k)$ with $M_k^T \mathbf{e}(\mathbf{x}_k)$ computed already].
For $\|\mathbf{e}(\mathbf{x}_k)\|_2^2 / \|(M_k^T + I)\mathbf{e}(\mathbf{x}_k)\|_2^2$: 1 multiplications.
- (4) Compute $\mathbf{x}_{k+1} = P_\Omega(\mathbf{x}_k - \rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k))$
For $\rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k)$: N multiplications.
For $P_\Omega(\mathbf{x}_k - \rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k))$: N additions.
- (5) Compute $\theta_{k+1} = \theta_k + \tau \dot{\theta}_{k+1}$
For $\tau \dot{\theta}_{k+1}$: $(N-3)$ multiplications.
For $\theta_k + \tau \dot{\theta}_{k+1}$: $(N-3)$ additions.

By combining (1)–(5), the one-iteration computing algorithm contains $(2N^2 + 4N - 2)$ multiplications and $(2N^2 + 7N - 5)$ additions. Therefore, the computational complexity of the one-iteration computing algorithm is $O(N^2)$.

Table 2
D-H parameters of UR5 arm.

#	θ^i (rad)	d_i (m)	a_i (m)	α_i (rad)
1	θ^1	0.0892	0	$\pi/2$
2	θ^2	0	-0.4250	0
3	θ^3	0	-0.3923	0
4	θ^4	0.1092	0	$\pi/2$
5	θ^5	0.0947	0	$-\pi/2$
6	θ^6	0.0823	0	0

Table 3
Physical limits of UR5 arm.

#	θ^- (rad)	θ^+ (rad)	$\dot{\theta}^-$ (rad/s)	$\dot{\theta}^+$ (rad/s)
1	$-\pi/2$	$\pi/2$	-0.5	0.5
2	$-\pi$	0	-0.5	0.5
3	$-\pi$	0	-0.5	0.5
4	$-\pi/2$	$\pi/2$	-0.5	0.5
5	0	π	-0.5	0.5
6	$-\pi/2$	$\pi/2$	-0.5	0.5

Remark 3. When QP (12)–(14) is disturbed by noises and the theoretical solution becomes $\hat{\mathbf{x}}^*(t)$, the solution computed by the one-iteration computing algorithm converges to $\hat{\mathbf{x}}^*(t)$. The one-iteration computing algorithm itself is not anti-disturbance. The robustness is embodied in the control scheme design. Because ZNN design formula is utilized in (8) and (9), the pose of the end-effector continues to converge to the desired pose after the disturbance.

5. Computer experiments and verifications

In this section, experiments are conducted on the basis of UR5 arm [35], which is a classic type of redundant arms. It has six DOF and works in 3-D space. The D-H parameters [35] are presented in Table 2. Note that joint angle vector $\boldsymbol{\theta} = [\theta^1, \theta^2, \theta^3, \theta^4, \theta^5, \theta^6]^T \in \mathbb{R}^6$ and joint angular velocity vector $\dot{\boldsymbol{\theta}} = [\dot{\theta}^1, \dot{\theta}^2, \dot{\theta}^3, \dot{\theta}^4, \dot{\theta}^5, \dot{\theta}^6]^T \in \mathbb{R}^6$. Besides, actual orientation vector $\mathbf{o}_a = [o_a^1, o_a^2, o_a^3]^T \in \mathbb{R}^3$ and desired orientation vector $\mathbf{o}_d = [o_d^1, o_d^2, o_d^3]^T \in \mathbb{R}^3$.

Remark 4. The discussion on the selection and determination of parameters is divided into three parts.

- (1) Specifically, there are three design parameters (λ , γ , and η) in standard QP (12)–(14), which are all ZNN design parameters scaling the convergence rate of the pose control. The ZNN design parameter generally corresponds to the reciprocal of the product of a resistance parameter and a capacitance parameter in potential electronic circuit [24]. Hence, these three parameters should be set as large as the hardware would permit, or appropriately large for practical application purposes. As a suggestion, these three parameters can be set in the interval [1, 100].
- (2) There are two design parameters (δ and ς) in PNN solver (16). Parameter δ is used to scale the convergence rate of the PNN solution, which also corresponds to the reciprocal of the product of a resistance parameter and a capacitance parameter. Hence, δ should also be set as large as the hardware would permit, or appropriately large for practical application purposes. As a suggestion, δ can be set in the interval $[10^4, 10^{12}]$. Furthermore, ς is used to numerically represent positive infinity, which should be set as large as possible. As a suggestion, ς can be set in the interval $[10^4, +\infty)$.

- (3) There are two parameters (ζ and τ) in one-iteration computing algorithm (17)–(21). Parameter ζ is set the same as the PNN solver. Sampling gap τ should be set relatively small to guarantee high control precision. As a suggestion, τ can be set less than 0.1 s.

One-iteration computing algorithm (17)–(21) is applied to completing the circle path tracking and five-ring path tracking for UR5 arm.

5.1. Pose control of circle path

Based on the previous work [32], the conventional PIB model utilizing ZNN method is developed as

$$\dot{\boldsymbol{\theta}} = J_1^+(\boldsymbol{\theta})(\dot{\mathbf{r}}_d - \gamma(\mathbf{r}_a - \mathbf{r}_d)), \quad (23)$$

where $\gamma > 0$ is a design parameter. Furthermore, the corresponding discrete model (termed DZNN-PIB model) is developed as below [32]:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + J_{1,k}^+(\tau \dot{\mathbf{r}}_{d,k} - h(\mathbf{r}_{a,k} - \mathbf{r}_{d,k})), \quad (24)$$

where $J_{1,k} = J_1(\boldsymbol{\theta}_k)$ and $h = \tau \gamma$.

In this subsection, a circle path in horizontal plane is desired to be tracked by the end-effector of UR5 arm. The radius of the circle path is 0.15 m. The end-effector orientation is desired to be downward when the end-effector tracks the circle path. That is, one has desired orientation vector $\mathbf{o}_d = [0, 0, -1]^T$. Besides, initial joint angles are set as $\boldsymbol{\theta}_0 = [0, -2\pi/3, -2\pi/3, -\pi/6, 2\pi/3, 0]^T$ rad. The tracking-task duration is set as $T = 20$ s.

First, DZNN-PIB model (24) is applied to performing the tracking task. Parameters in DZNN-PIB model (24) are set as $h = 0.1$ and $\tau = 1$ ms. The corresponding results are displayed in Fig. 2. Specifically, Fig. 2(a) shows the motion trajectories of UR5 arm and actual trajectory of end-effector. As expected, the circle path is tracked by the end-effector successfully. However, because the end-effector orientation is not controlled by DZNN-PIB model (24), one can find that the end-effector orientation is not downward. Fig. 2(b) and Fig. 2(c) present profiles of joint angles and joint angular velocities, respectively. One can find that joint angles satisfy physical constraints. Nevertheless, joint angular velocities go beyond physical constraints as shown in the locally enlarged figure. Further, to illustrate the end-effector orientation during the tracking process in detail, profiles of end-effector orientation are displayed in Fig. 2(d). Evidently, the end-effector orientation is relatively haphazard, which is not convergent to the desired orientation. Errors between actual orientation and desired orientation are presented in Fig. 2(e), which are not equal to zero. These results illustrate the fact that DZNN-PIB model (24) is not effective for pose control with physical constraints. Nevertheless, DZNN-PIB model (24) can realize high-precision position tracking, as shown in Fig. 2(f).

By setting parameters as $\lambda = \gamma = 10$, $\eta = 2$, $\varsigma = 10^6$, and $\tau = 1$ ms, computing algorithm (17)–(21) is utilized to perform the same tracking task. The corresponding results are exhibited in Fig. 3. From Fig. 3(a), one can see the motion trajectories of UR5 arm, and one can preliminarily find that the end-effector orientation is downward. In Fig. 3(b), values of joint angles in tracking process are presented in detail. Besides, one can see the profiles of joint angular velocities from Fig. 3(c). In this case, joint angles and joint angular velocities are controlled within physical constraints, which illustrates the superiority of computing algorithm (17)–(21) for pose control of redundant arm with physical constraints. Further, profiles of end-effector orientation are shown in Fig. 3(d) to illustrate the efficacy of computing algorithm (17)–(21). About 1 s later, o_a^1 converges to 0, o_a^2 converges to 0, and o_a^3 converges to -1. That is, end-effector orientation \mathbf{o}_a is consistent with desired

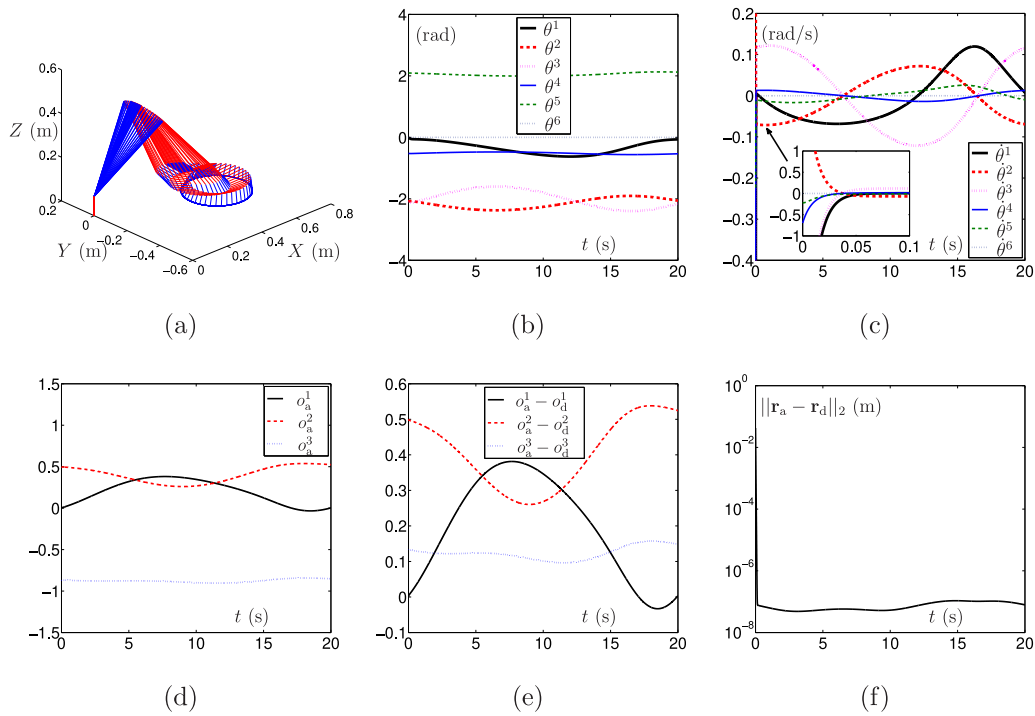


Fig. 2. Numerical and graphical results synthesized by conventional DZNN-PIB model (24) when UR5 arm tracks desired circle path in horizontal plane. (a) Motion trajectories of UR5 arm and actual trajectory of end-effector. (b) Profiles of joint angles. (c) Profiles of joint angular velocities. (d) Profiles of end-effector orientation. (e) Profiles of orientation errors. (f) Profile of 2-norm of position errors.

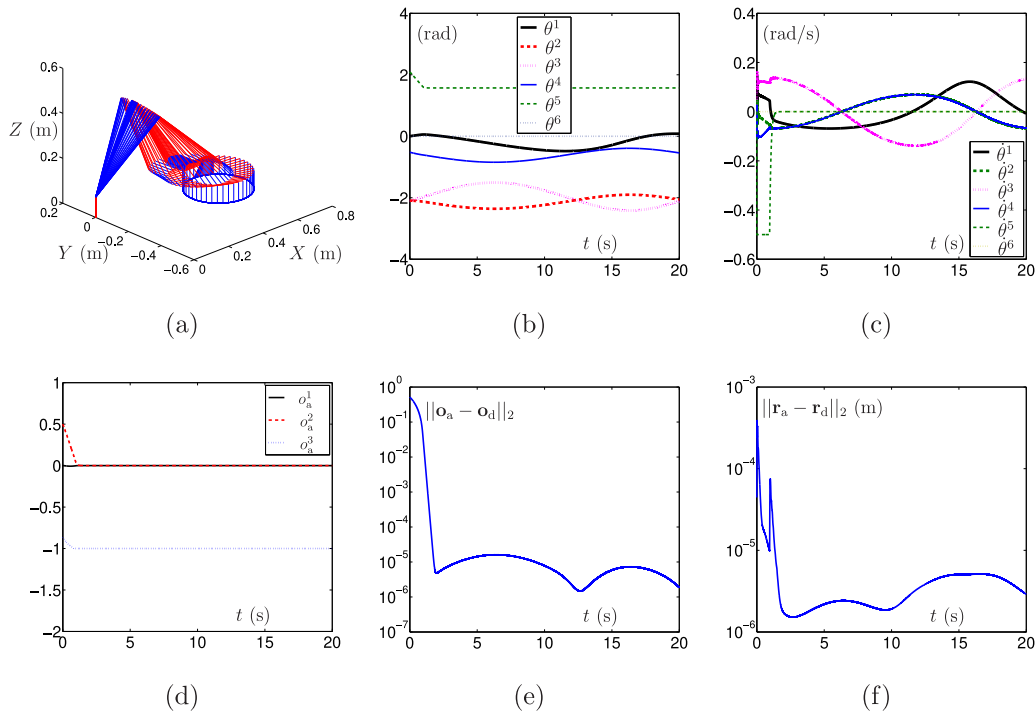


Fig. 3. Numerical and graphical results by computing algorithm (17)–(21) when UR5 arm tracks desired circle path in horizontal plane with desired downward orientation. (a) Motion trajectories of UR5 arm and actual trajectory of end-effector. (b) Profiles of joint angles. (c) Profiles of joint angular velocities. (d) Profiles of end-effector orientation. (e) Profile of 2-norm of orientation errors. (f) Profile of 2-norm of position errors.

orientation \mathbf{o}_d rapidly. Not only that, the control precision about end-effector orientation is high (about 10^{-6} to 10^{-5}), which is displayed in Fig. 3(e). In addition, the control precision about end-effector position is also high, as shown in Fig. 3(f). Hence, compared with DZNN-PIB model (24) only for position control of

redundant arm, the efficacy and superiority of computing algorithm (17)–(21) for pose control of redundant arm with physical constraints are preliminarily substantiated. Naturally, the efficacy of pose control scheme (8)–(11) as well as standard QP (12)–(14) is also verified.

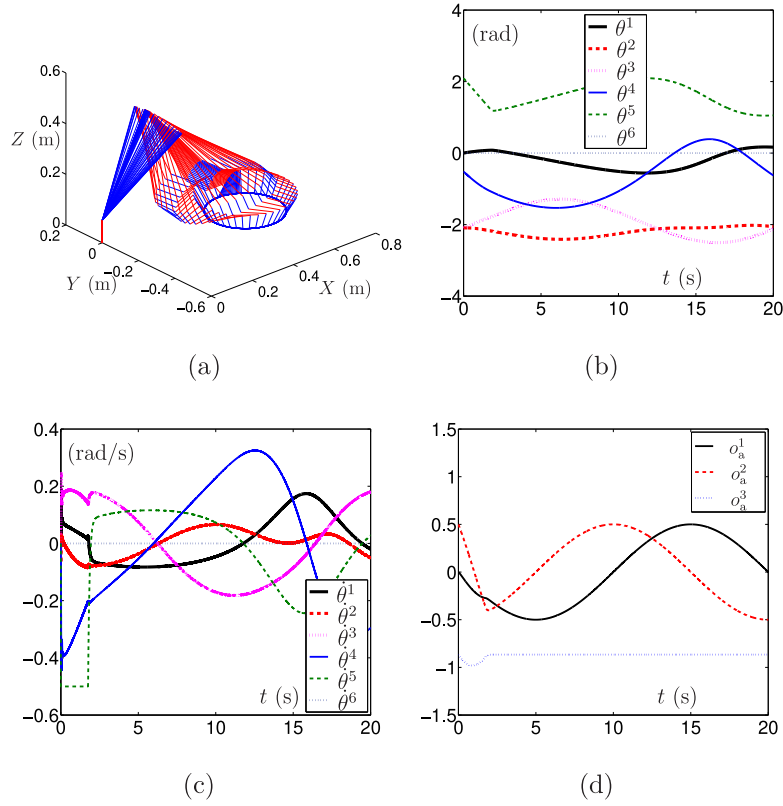


Fig. 4. Numerical and graphical results synthesized by computing algorithm (17)–(21) when UR5 arm tracks desired circle path in horizontal plane with desired 60°-slope orientation. (a) Motion trajectories of UR5 arm and actual trajectory of end-effector. (b) Profiles of joint angles. (c) Profiles of joint angular velocities. (d) Profiles of end-effector orientation.

Further, to illustrate the efficacy of computing algorithm (17)–(21) for pose control, the desired orientation of end-effector is downward with 60° slope. Concretely, the end-effector is desired to point to the node $\sqrt{3}r$ below the origin of desired circle in real time t . In this case, the desired orientation vector is time-varying and complicated. The related parameters setting of computing algorithm (17)–(21) is the same as before. The corresponding results are illustrated in Fig. 4. Different with the above motion trajectories as shown in Fig. 3(a), motion trajectories of UR5 arm are presented in Fig. 4(a), in which one can find the slope of the end-effector. Profiles of joint angles and joint angular velocities are displayed in Fig. 4(b) and Fig. 4(c), respectively. Evidently, physical constraints are satisfied as expected. Besides, the end-effector orientation is consistent with the desired orientation rapidly (after about 2 s), which can be found in Fig. 4(d). Note that the precision of the orientation and position control is also high, but it is not presented hereafter because of space limitation.

5.2. Pose control of five-ring path

By combining $J_1(\theta)\dot{\theta} = \dot{\mathbf{r}}_d - \gamma(\mathbf{r}_a - \mathbf{r}_d)$, and $J_2(\theta)\dot{\theta} = \dot{\mathbf{o}}_d - \lambda(\mathbf{o}_a - \mathbf{o}_d)$, the developed PIB model is obtained for pose control of redundant arm:

$$\dot{\theta} = J^\dagger(\theta)\dot{\mathbf{h}}, \quad (25)$$

in which

$$J = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}, \quad \dot{\mathbf{h}} = \begin{bmatrix} \dot{\mathbf{r}}_d - \gamma(\mathbf{r}_a - \mathbf{r}_d) \\ \dot{\mathbf{o}}_d - \lambda(\mathbf{o}_a - \mathbf{o}_d) \end{bmatrix}.$$

Furthermore, the corresponding discrete model (termed DZNN-PIB2 model) is developed as below [32]:

$$\theta_{k+1} = \theta_k + \tau J_k^\dagger \dot{\mathbf{h}}_k. \quad (26)$$

In this subsection, a complicated path named as five-ring path is desired to be tracked, to further substantiate the efficacy and superiority of pose control scheme (8)–(11), standard QP (12)–(14), and one-iteration computing algorithm (17)–(21).

Likewise, the desired five-ring path is to be tracked on a horizontal plane first. The tracking-task duration is set to be $T = 20$ s as well. Besides, $\theta_0 = [0, -2\pi/3, -2\pi/3, -\pi/6, 2\pi/3, 0]^T$ rad. By setting parameters $h = 0.1$ and $\tau = 1$ ms, DZNN-PIB2 model (24) is applied to performing this task, and the results are briefly shown in Fig. 5. The desired path is tracked successfully, and the orientation of end-effector is also consistent with the desired orientation, as illustrated in Fig. 5(a). Meanwhile, the profiles of joint angles are presented in Fig. 5(b). Besides, Fig. 5(c) presents the profiles of joint angular velocities during the tracking process. Although the pose control task is completed, physical constraints cannot be satisfied by using DZNN-PIB2 model (26) (as illustrated in Fig. 5(c)). Hence, the safe operation of redundant arm cannot be guaranteed by the PIB method.

Differently, with desired orientation (downward) considered, the detailed results synthesized by computing algorithm (17)–(21) for pose control of redundant arm are illustrated in Fig. 6. From Fig. 6(a), one can find the desired five-ring path is tracked successfully as well. Not any that, the end-effector orientation during the tracking process is tidy and downward, which reflects the efficacy of computing algorithm (17)–(21). Values of joint angles and joint angular velocities are illustrated in Fig. 6(b) and Fig. 6(c), respectively. Joint angles and joint angular velocities are controlled successfully within physical constraints shown in Table 3. Meanwhile, the end-effector orientation during the tracking process is shown in Fig. 6(d), which converges to the desired orientation after a short time. These results substantiate the efficacy of pose control scheme (8)–(11), standard QP (12)–(14), and computing algorithm (17)–(21) in complicated task situation.

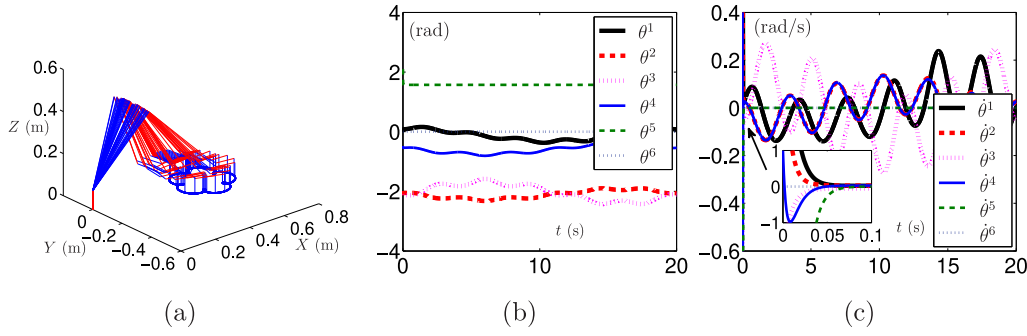


Fig. 5. Numerical and graphical results synthesized by developed DZNN-PIB2 model (26) when UR5 arm tracks desired five-ring path in horizontal plane. (a) Motion trajectories of UR5 arm and actual trajectory of end-effector. (b) Profiles of joint angles. (c) Profiles of joint angular velocities.

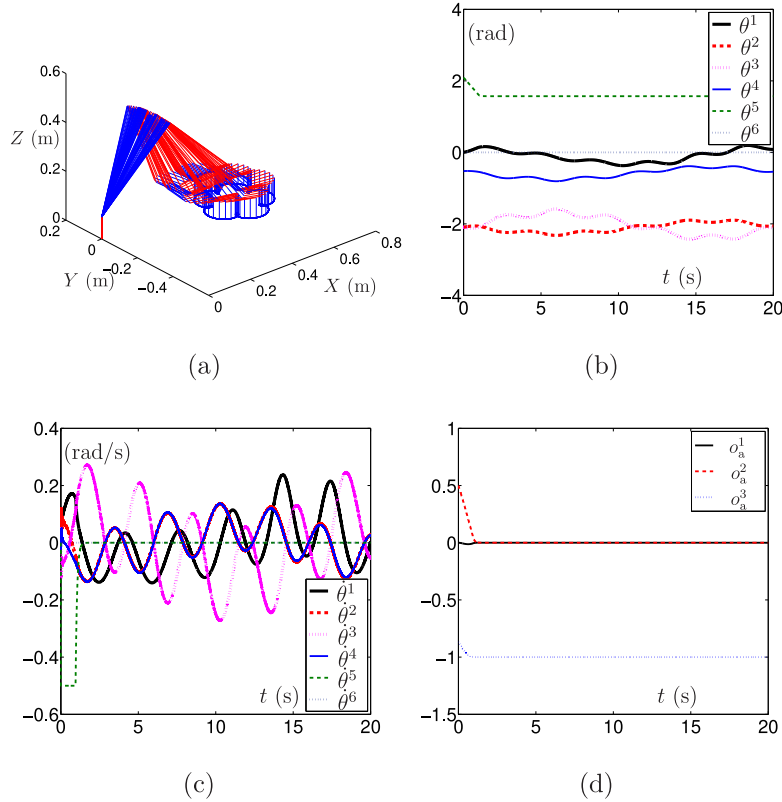


Fig. 6. Numerical and graphical results synthesized by computing algorithm (17)–(21) when UR5 arm tracks desired five-ring path in horizontal plane. (a) Motion trajectories of UR5 arm and actual trajectory of end-effector. (b) Profiles of joint angles. (c) Profiles of joint angular velocities. (d) Profiles of end-effector orientation.

Finally, the task is to track the five-ring path in a vertical plane with end-effector pointing to the path vertically. Concretely, one has $\mathbf{o}_d = [1, 0, 0]^T$. Initial joint angles of UR5 arm are $\theta_0 = [0, -\pi/2, -\pi/2, 0, 2\pi/3, 0]^T$ rad. Detailed results synthesized by computing algorithm (17)–(21) are presented in Fig. 7. Evidently, in this situation, computing algorithm (17)–(21) is also effective. The position control and orientation control are both realized for redundant arm with physical constraints.

Remark 5. In summary, through sufficient experiments, pose control scheme (8)–(11) and standard QP (12)–(14) are substantiated to be effective. The efficiency of the one-iteration computing algorithm is also substantiated by experimental results. Furthermore, the experimental illustration about the efficiency of the one-iteration computing algorithm indirectly substantiates the efficiency of the PNN solver. Meanwhile, the PNN solver is not convenient in practical digital control, and the one-iteration computing algorithm is readily for computer operation and real-time

requirements. Therefore, the validation for the PNN solver is not presented in this paper due to the page limitation.

6. Conclusion

To realize pose control (including position control and orientation control) of redundant arm with physical constraints, the pose control scheme has been proposed and analyzed by utilizing ZNN method. To solve the pose control scheme, standard QP has been presented and PNN solver has been developed. For better computer operation and real-time control requirements, a one-iteration computing algorithm has been further developed. Plenty of experiments about pose control for redundant arm have been conducted, which further verify the effectiveness of the method developed in this paper. Compared with conventional PIB method, the method developed in this paper can deal with physical constraints. Besides, compared with the other numerical

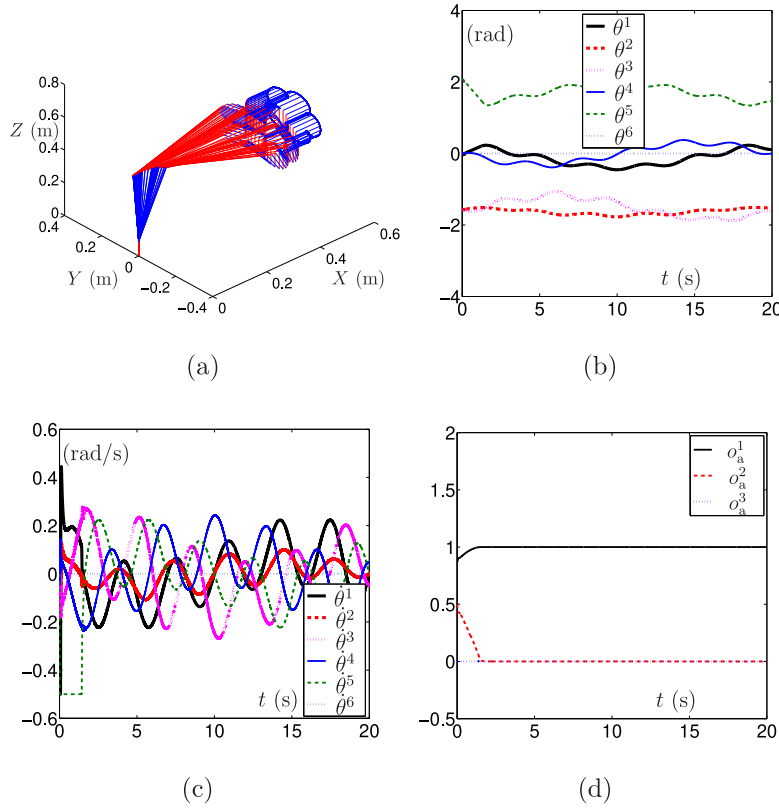


Fig. 7. Numerical and graphical results synthesized by computing algorithm (17)–(21) when UR5 arm tracks desired five-ring path in vertical plane. (a) Motion trajectories of UR5 arm and actual trajectory of end-effector. (b) Profiles of joint angles. (c) Profiles of joint angular velocities. (d) Profiles of end-effector orientation.

algorithms for QP solving, the one-iteration computing algorithm has lower computational complexity because just one iteration is needed during the sampling gap.

The pose control of various redundant robots designed by ZNN method is one of future research directions, such as mobile arms and parallel manipulators. Meanwhile, developing effective computing algorithms for time-varying QP solving is a future research direction, such as active set algorithm improvement. Utilizing the kinematics and dynamics information, such as Jacobian matrix and time-derivative information, to control the configuration of redundant arms could be a future research direction. Moreover, conducting more comparative experiments and physical experiments on the basis of effective computing algorithms is also one of future research directions.

CRedit authorship contribution statement

Min Yang: Methodology, Conceptualization, Software, Validation, Investigation, Writing – original draft, Writing – review & editing. **Yunong Zhang:** Methodology, Writing – review & editing, Supervision. **Xuefeng Zhou:** Writing – review & editing, Supervision. **Haifeng Hu:** Writing – review & editing, Supervision.

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.

Appendix A. Proof of Theorem 1

According to the projection property [33,36], for any $\psi \in \mathbb{R}^{n+3}$ and any $\phi \in \Omega$, the following inequality holds true:

$$(\psi - P_{\Omega}(\psi))^T(\phi - P_{\Omega}(\psi)) \leq 0. \quad (27)$$

Setting $\psi = \mathbf{x}_k - \rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k)$ and $\phi = \mathbf{x}_k^*$, one gets

$$(\mathbf{x}_k - \rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k) - \mathbf{x}_{k+1})^T(\mathbf{x}_k^* - \mathbf{x}_{k+1}) \leq 0,$$

i.e.,

$$(\tilde{\mathbf{x}}_k - \mathbf{x}_{k+1})^T(\mathbf{x}_k^* - \mathbf{x}_{k+1}) \leq 0,$$

where $\tilde{\mathbf{x}}_k = \mathbf{x}_k - \rho(\mathbf{x}_k)\mathbf{d}(\mathbf{x}_k)$ for convenient expression. Note that

$$(\tilde{\mathbf{x}}_k - \mathbf{x}_{k+1})^T(\mathbf{x}_k^* - \mathbf{x}_{k+1}) = \frac{1}{2}(\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|_2^2 + \|\tilde{\mathbf{x}}_k - \mathbf{x}_{k+1}\|_2^2 - \|\tilde{\mathbf{x}}_k - \mathbf{x}_k^*\|_2^2).$$

Thus, one obtains

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|_2^2 \leq -\|\tilde{\mathbf{x}}_k - \mathbf{x}_{k+1}\|_2^2 + \|\tilde{\mathbf{x}}_k - \mathbf{x}_k^*\|_2^2.$$

Based on the above result, one has

$$\begin{aligned} & \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \rho(\mathbf{x}_k)\|\mathbf{e}(\mathbf{x}_k)\|_2^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|_2^2 \\ & \geq \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \rho(\mathbf{x}_k)\|\mathbf{e}(\mathbf{x}_k)\|_2^2 + \|\tilde{\mathbf{x}}_k - \mathbf{x}_{k+1}\|_2^2 - \|\tilde{\mathbf{x}}_k - \mathbf{x}_k^*\|_2^2 \\ & = \|\mathbf{x}_k\|_2^2 + \|\mathbf{x}_{k+1}\|_2^2 - 2\mathbf{x}_k^T\mathbf{x}_{k+1} + 2\mathbf{x}_k^T\mathbf{x}_{k+1} - 2\mathbf{x}_k^T\mathbf{x}_k^* \\ & \quad - 2\tilde{\mathbf{x}}_k^T\mathbf{x}_{k+1} + 2\tilde{\mathbf{x}}_k^T\mathbf{x}_k^* - \rho(\mathbf{x}_k)\|\mathbf{e}(\mathbf{x}_k)\|_2^2 \\ & = \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2 + 2(\mathbf{x}_k - \tilde{\mathbf{x}}_k)^T(\mathbf{x}_{k+1} - \mathbf{x}_k^*) - \rho(\mathbf{x}_k)\|\mathbf{e}(\mathbf{x}_k)\|_2^2 \\ & = \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2 + 2\rho(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k^*)^T\mathbf{d}(\mathbf{x}_k) - \rho(\mathbf{x}_k)\|\mathbf{e}(\mathbf{x}_k)\|_2^2. \end{aligned} \quad (28)$$

Furthermore, setting $\psi = \mathbf{x}_k^* - (M_k\mathbf{x}_k^* + \mathbf{c}_k)$ and $\phi = P_{\Omega}(\mathbf{x}_k - (M_k\mathbf{x}_k + \mathbf{c}_k))$, in view of $\mathbf{x}_k^* = P_{\Omega}(\mathbf{x}_k^* - (M_k\mathbf{x}_k^* + \mathbf{c}_k))$, one has

$$(\mathbf{x}_k^* - (M_k\mathbf{x}_k^* + \mathbf{c}_k) - \mathbf{x}_k^*)^T(P_{\Omega}(\mathbf{x}_k - (M_k\mathbf{x}_k + \mathbf{c}_k)) - \mathbf{x}_k^*) \leq 0,$$

i.e.,

$$(P_{\Omega}(\mathbf{x}_k - (M_k\mathbf{x}_k + \mathbf{c}_k)) - \mathbf{x}_k^*)^T(M_k\mathbf{x}_k^* + \mathbf{c}_k) \geq 0. \quad (29)$$

Meanwhile, one has

$$\begin{aligned}
 (\mathbf{x}_k - \mathbf{x}_k^*)^T \mathbf{d}(\mathbf{x}_k) &= (\mathbf{x}_k - \mathbf{x}_k^*)^T (M_k^T \mathbf{e}(\mathbf{x}_k) + M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= (\mathbf{x}_k - \mathbf{x}_k^*)^T M_k^T \mathbf{e}(\mathbf{x}_k) + \mathbf{e}^T(\mathbf{x}_k) \mathbf{c}_k - \mathbf{e}^T(\mathbf{x}_k) \mathbf{c}_k \\
 &\quad + (\mathbf{x}_k - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k + \mathbf{c}_k) - \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k^* + \mathbf{c}_k) \\
 &\quad + (\mathbf{x}_k - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k + \mathbf{c}_k) - (\mathbf{x}_k - \mathbf{x}_k^* + \mathbf{x}_k^*)^T \\
 &\quad - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)) (M_k \mathbf{x}_k^* + \mathbf{c}_k) + (\mathbf{x}_k - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k + \mathbf{c}_k) - (\mathbf{x}_k - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k^* + \mathbf{c}_k) \\
 &\quad + (P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)) - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k^* + \mathbf{c}_k) \\
 &\quad + (\mathbf{x}_k - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k + \mathbf{c}_k) + (\mathbf{x}_k - \mathbf{x}_k^*)^T M_k (\mathbf{x}_k - \mathbf{x}_k^*) \\
 &\quad + (P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)) - \mathbf{x}_k^*)^T (M_k \mathbf{x}_k^* + \mathbf{c}_k).
 \end{aligned} \tag{30}$$

In view of (29) and $(\mathbf{x}_k - \mathbf{x}_k^*)^T M_k (\mathbf{x}_k - \mathbf{x}_k^*) \geq 0$ (with M_k being positive semi-definite), one gets the following result from (30):

$$(\mathbf{x}_k - \mathbf{x}_k^*)^T \mathbf{d}(\mathbf{x}_k) \geq \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k + \mathbf{c}_k). \tag{31}$$

Based on (31), one has

$$\begin{aligned}
 (\mathbf{x}_{k+1} - \mathbf{x}_k^*)^T \mathbf{d}(\mathbf{x}_k) &= (\mathbf{x}_{k+1} - \mathbf{x}_k)^T \mathbf{d}(\mathbf{x}_k) + (\mathbf{x}_k - \mathbf{x}_k^*)^T \mathbf{d}(\mathbf{x}_k) \\
 &\geq (\mathbf{x}_{k+1} - \mathbf{x}_k)^T \mathbf{d}(\mathbf{x}_k) + \mathbf{e}^T(\mathbf{x}_k) (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= (\mathbf{x}_{k+1} - \mathbf{x}_k)^T (M_k^T \mathbf{e}(\mathbf{x}_k) + M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &\quad + (\mathbf{x}_k - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)))^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &= (\mathbf{x}_{k+1} - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)))^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 &\quad + (\mathbf{x}_{k+1} - \mathbf{x}_k)^T M_k^T \mathbf{e}(\mathbf{x}_k).
 \end{aligned} \tag{32}$$

Using projection property (27) again with $\psi = \mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)$ and $\phi = \mathbf{x}_{k+1}$, one gets

$$\begin{aligned}
 (\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k) - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)))^T (\mathbf{x}_{k+1} \\
 - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k))) \leq 0,
 \end{aligned}$$

which leads to the following inequality:

$$\begin{aligned}
 (\mathbf{x}_{k+1} - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)))^T (M_k \mathbf{x}_k + \mathbf{c}_k) \\
 \geq (\mathbf{x}_k - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k)))^T (\mathbf{x}_{k+1} \\
 - \mathbf{x}_k + \mathbf{x}_k - P_\Omega(\mathbf{x}_k - (M_k \mathbf{x}_k + \mathbf{c}_k))) \\
 = (\mathbf{x}_{k+1} - \mathbf{x}_k)^T \mathbf{e}(\mathbf{x}_k) + \|\mathbf{e}(\mathbf{x}_k)\|_2^2.
 \end{aligned}$$

Hence, according to (32), one has

$$(\mathbf{x}_{k+1} - \mathbf{x}_k^*)^T \mathbf{d}(\mathbf{x}_k) \geq (\mathbf{x}_{k+1} - \mathbf{x}_k)^T (M_k^T + I) \mathbf{e}(\mathbf{x}_k) + \|\mathbf{e}(\mathbf{x}_k)\|_2^2. \tag{33}$$

Meanwhile, because $\|(\mathbf{x}_{k+1} - \mathbf{x}_k) + \rho(\mathbf{x}_k)(M_k^T + I) \mathbf{e}(\mathbf{x}_k)\|_2^2 \geq 0$, one has

$$\begin{aligned}
 2\rho(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)^T (M_k^T + I) \mathbf{e}(\mathbf{x}_k) \\
 \geq -\rho^2(\mathbf{x}_k) \|(M_k^T + I) \mathbf{e}(\mathbf{x}_k)\|_2^2 - \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2.
 \end{aligned} \tag{34}$$

Based on (28), (33), and (34), one obtains

$$\begin{aligned}
 \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|_2^2 \\
 \geq \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2 + 2\rho(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k^*)^T \mathbf{d}(\mathbf{x}_k) - \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2 \\
 \geq \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2 + 2\rho(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)^T (M_k^T + I) \mathbf{e}(\mathbf{x}_k) \\
 + \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2 \\
 \geq -\rho^2(\mathbf{x}_k) \|(M_k^T + I) \mathbf{e}(\mathbf{x}_k)\|_2^2 + \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2 \\
 \geq \rho(\mathbf{x}_k) (\|\mathbf{e}(\mathbf{x}_k)\|_2^2 - \rho(\mathbf{x}_k) \|(M_k^T + I) \mathbf{e}(\mathbf{x}_k)\|_2^2) \\
 = 0.
 \end{aligned}$$

That is,

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|_2^2 \leq \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2.$$

Furthermore, referring to Taylor expansion, one has $\mathbf{x}_{k+1}^* = \mathbf{x}_k^* + \mathbf{O}(\tau)$. Then, one has $\|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^*\|_2^2 = \|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|_2^2 + \mathbf{O}(\tau)$. Therefore, the following result is obtained:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^*\|_2^2 \leq \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2 + \mathbf{O}(\tau).$$

The proof is thus completed. ■

Appendix B. Proof of Theorem 2

Because $\rho(\mathbf{x}_k) = \|\mathbf{e}(\mathbf{x}_k)\|_2^2 / \|(M_k^T + I) \mathbf{e}(\mathbf{x}_k)\|_2^2$, one gets $\rho(\mathbf{x}_k) \geq 1 / \|(M_k^T + I)\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. Hence, there exists a positive constant ϵ satisfying $\rho(\mathbf{x}_k) \geq \epsilon$. Then, one has

$$\epsilon \|\mathbf{e}(\mathbf{x}_k)\|_2^2 \leq \rho(\mathbf{x}_k) \|\mathbf{e}(\mathbf{x}_k)\|_2^2 \leq \|\mathbf{x}_k - \mathbf{x}_k^*\|_2^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^*\|_2^2 + \mathbf{O}(\tau).$$

With $\kappa = T/\tau - 1$, one further obtains

$$\begin{aligned}
 \sum_{k=0}^{\kappa} \epsilon \|\mathbf{e}(\mathbf{x}_k)\|_2^2 &\leq \|\mathbf{x}_0 - \mathbf{x}_0^*\|_2^2 - \|\mathbf{x}_{\kappa+1} - \mathbf{x}_{\kappa+1}^*\|_2^2 + (\kappa + 1) \mathbf{O}(\tau) \\
 &\leq \|\mathbf{x}_0 - \mathbf{x}_0^*\|_2^2 + (\kappa + 1) \mathbf{O}(\tau) \\
 &= \|\mathbf{x}_0 - \mathbf{x}_0^*\|_2^2 + \mathbf{O}(T).
 \end{aligned}$$

With $\tau \rightarrow 0$ (i.e., correspondingly, $\kappa \rightarrow +\infty$), one has $\sum_{k=0}^{\kappa} \|\mathbf{e}(\mathbf{x}_k)\|_2^2 < +\infty$. There exists a positive constant ϵ satisfying

$$\lim_{\kappa \rightarrow +\infty} \sum_{k=0}^{\kappa} \|\mathbf{e}(\mathbf{x}_k)\|_2^2 = \epsilon.$$

According to the definition of series convergence, $\lim_{k \rightarrow +\infty} \mathbf{e}(\mathbf{x}_k) = \mathbf{0}$. One has $\lim_{k \rightarrow +\infty} \mathbf{x}_k = \mathbf{x}_k^*$ and also $\lim_{k \rightarrow +\infty} \dot{\theta}_k = \dot{\theta}_k^*$. Furthermore, θ_k computed by (21) converges to the optimal solution θ_k^* . The proof is completed. ■

References

- [1] Y. Zhang, Z. Zhang, Repetitive Motion Planning and Control of Redundant Robot Manipulators, Springer-Verlag, Berlin, Germany, 2013.
- [2] X. Zhang, J. Liu, J. Feng, Y. Liu, Z. Ju, Effective capture of nongrasable objects for space robots using geometric cage pairs, IEEE/ASME Trans. Mechatronics 25 (1) (2020) 95–107.
- [3] C. Yang, G. Peng, L. Cheng, J. Na, Z. Li, Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks, IEEE Trans. Syst. Man Cybern. Syst 51 (5) (2021) 3282–3292.
- [4] J. Na, B. Jing, Y. Huang, G. Gao, C. Zhang, Unknown system dynamics estimator for motion control of nonlinear robotic systems, IEEE Trans. Ind. Electron. 67 (5) (2020) 3850–3859.
- [5] U.E. Ogenyi, J. Liu, C. Yang, Z. Ju, H. Liu, Physical human-robot collaboration: Robotic systems, learning methods, collaborative strategies, sensors, and actuators, IEEE Trans. Cybern. 51 (4) (2021) 1888–1901.
- [6] Y. Zhang, S. Li, S. Kady, B. Liao, Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints, IEEE Trans. Cybern. 49 (12) (2019) 4194–4205.
- [7] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive parameter estimation and control design for robot manipulators with finite-time convergence, IEEE Trans. Ind. Electron. 65 (10) (2018) 8112–8123.
- [8] L. Jin, S. Li, H.M. La, X. Luo, Manipulability optimization of redundant manipulators using dynamic neural networks, IEEE Trans. Ind. Electron. 64 (6) (2017) 4710–4720.
- [9] L. Jin, S. Li, X. Luo, Y. Li, B. Qin, Neural dynamics for cooperative control of redundant robot manipulators, IEEE Trans. Ind. Inf. 14 (9) (2018) 3812–3821.
- [10] S. Li, M. Zhou, X. Luo, Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises, IEEE Trans. Neural Netw. Learn. Syst. 29 (10) (2018) 4791–4801.
- [11] A. Belkadi, H. Oulhadj, Y. Touati, S.A. Khan, B. Daachi, On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach, Appl. Soft Comput. 60 (2017) 87–100.

- [12] Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties, *Neurocomputing* 329 (2019) 255–266.
- [13] S. Qin, X. Yang, X. Xue, J. Song, A one-layer recurrent neural network for pseudoconvex optimization problems with equality and inequality constraints, *IEEE Trans. Cybern.* 47 (10) (2017) 3063–3074.
- [14] S. Qin, X. Xue, A two-layer recurrent neural network for nonsmooth convex optimization problems, *IEEE Trans. Neural Netw. Learn. Syst.* 26 (6) (2015) 1149–1160.
- [15] Y. Liu, S. Tong, Adaptive fuzzy control for a class of nonlinear discrete-time systems with backlash, *IEEE Trans. Fuzzy Syst.* 22 (5) (2014) 1359–1365.
- [16] W. He, H. Gao, C. Zhou, C. Yang, Z. Li, Reinforcement learning control of a flexible two-link manipulator: An experimental investigation, *IEEE Trans. Syst. Man Cybern. Syst* (2021) <http://dx.doi.org/10.1109/TSMC.2020.2975232> in press.
- [17] D. Chen, S. Li, W. Li, Q. Wu, A multi-level simultaneous minimization scheme applied to jerk-bounded redundant robot manipulators, *IEEE Trans. Autom. Sci. Eng.* 17 (1) (2020) 463–474.
- [18] D. Guo, Y. Zhang, Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators, *IEEE Trans. Ind. Electron.* 61 (12) (2014) 6903–6914.
- [19] Y. Zhang, W. Li, Z. Zhang, Physical-limits-constrained minimum velocity norm coordinating scheme for wheeled mobile redundant manipulators, *Robotica* 33 (6) (2015) 1325–1350.
- [20] D. Chen, S. Li, L. Liao, A recurrent neural network applied to optimal motion control of mobile robots with physical constraints, *Appl. Soft Comput.* 85 (2019) 105880.
- [21] X. Lai, P. Zhang, Y. Wang, M. Wu, Position-posture control of a planar four-link underactuated manipulator based on genetic algorithm, *IEEE Trans. Ind. Electron.* 64 (6) (2017) 4781–4791.
- [22] P. Zhang, X. Lai, Y. Wang, C. Su, W. Ye, M. Wu, A novel position-posture control method using intelligent optimization for planar underactuated mechanical systems, *Mech. Mach. Theory* 140 (2019) 258–273.
- [23] M.D. Petkovic, P.S. Stanimirovic, V.N. Katsikis, Modified discrete iterations for computing the inverse and pseudoinverse of the time-varying matrix, *Neurocomputing* 289 (2018) 155–165.
- [24] M. Yang, Y. Zhang, H. Hu, B. Qiu, General 7-instant DCZNN model solving future different-level system of nonlinear inequality and linear equation, *IEEE Trans. Neural Netw. Learn. Syst.* 31 (9) (2020) 3204–3214.
- [25] D. Guo, Y. Zhang, Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time-varying Jacobian matrix pseudoinversion, *Appl. Soft Comput.* 24 (2014) 158–168.
- [26] L. Xiao, A nonlinearly-activated neurodynamic model and its finite-time solution to equality-constrained quadratic optimization with nonstationary coefficients, *Appl. Soft Comput.* 40 (2016) 252–259.
- [27] L. Jin, S. Li, H. Wang, Z. Zhang, Nonconvex projection activated zeroing neurodynamic models for time-varying matrix pseudoinversion with accelerated finite-time convergence, *Appl. Soft Comput.* 62 (2018) 840–850.
- [28] Z. Hu, L. Xiao, K. Li, K. Li, J. Li, Performance analysis of nonlinear activated zeroing neural networks for time-varying matrix pseudoinversion with application, *Appl. Soft Comput.* 98 (2021) 106735.
- [29] N. Liu, S. Qin, A novel neurodynamic approach to constrained complex-variable pseudoconvex optimization, *IEEE Trans. Cybern.* 49 (11) (2019) 3946–3956.
- [30] N. Liu, S. Qin, A neurodynamic approach to nonlinear optimization problems with affine equality and convex inequality constraints, *Neural Netw.* 109 (2019) 147–158.
- [31] L. Jin, S. Li, B. Hu, M. Liu, A survey on projection neural networks and their applications, *Appl. Soft Comput.* 76 (2019) 533–544.
- [32] D. Guo, F. Xu, L. Yan, The application of ZFD formula to kinematic control of redundant robot manipulators with guaranteed motion precision, *IEEE Access* 6 (2018) 64777–64783.
- [33] Y. Zhang, J. Li, M. Mao, W. Li, S. Fu, Complete theory for E47 and 94LVI algorithms solving inequality-and-bound constrained quadratic program efficiently, in: *Proc. of Chinese Automation Congress*, Wuhan, 2015, pp. 183–189.
- [34] J.H. Mathews, K.D. Fink, *Numerical Methods Using MATLAB*, fourth ed., Prentice Hall, Englewood Cliffs, NJ, USA, 2004.
- [35] F. Zhang, Y. Zhao, Kinematics and singularity analysis of UR5 robot, *J. Hebei Univ. Sci. Technol.* 40 (1) (2019) 51–59.
- [36] B. He, Solving a class of linear projection equations, *Numer. Math.* 68 (1) (1994) 71–80.