A Simultaneous Learning and Control Scheme for Redundant Manipulators With Physical Constraints on Decision Variable and Its Derivative

Mei Liu, Jialiang Fan, Yu Zheng, *Senior Member, IEEE*, Shuai Li, *Senior Member, IEEE*, and Long Jin, *Senior Member, IEEE*

***Abstract*—In this article, a simultaneous learning and control scheme built on the joint velocity level with physical constraints on the decision variable and its derivative, i.e., joint angle, joint velocity, and joint acceleration constraints, is proposed for the redundant manipulator control. The scheme works when the structure parameters involved in the forward kinematics are unknown or implicit. The learning and control parts are incorporated simultaneously in the scheme, which is finally formulated as a quadratic programming problem solved by a devised recurrent neural network (RNN). The convergences of learning and control abilities of the RNN are proved theoretically. Simulations and physical experiments on a 7-degrees of freedom (DOFs) redundant manipulator show that, aided with the proposed scheme and the related RNN solver, a redundant manipulator with unknown structure parameters can perform a given inverse kinematics task with high accuracy while satisfying physical constraints on the decision variable and its derivative.**

***Index Terms*—Learning and control, physical constraints on decision variable and its derivative, recurrent neural network (RNN), redundant manipulators.**

I. Introduction

AS AN important type of robot, manipulators play a pivotal role in all walks of life, on which the research is booming [1]–[3]. Redundant manipulators are a special type of manipulators that have more degrees of freedom (DOFs) than the task required. These extra DOFs provide infinite solutions to the primary task and can be utilized to achieve secondary tasks, such as obstacle avoidance [4], joint limit avoidance [5], torque constraints [6], etc. Incorporating a performance index into redundant manipulator control has been a research hotspot in recent years because different performance indices reflect the requirements of a manipulator in different scenarios. A scheme is built on acceleration level for redundant manipulators in [7], which takes repetitive motion as the performance index, solving the joint drift problem existing in redundant manipulator control successfully. In [8], the manipulability optimization is chosen as a performance index in the cooperative control of distributed robotic systems, which can effectively handle the singularity phenomenon in the cooperative control of redundant manipulators.

Physical constraints consideration is an inevitable topic in robot control that has great practical significance. Although many commercial manipulators have protective measures on joints to avoid causing damage to motors or sensors, these measures sometimes are too broad for those operating in narrow spaces, which need stricter constraints. Therefore, it is necessary to make constraints on joint angles considering the limitations of operating space. In addition, excessive joint angle velocities may cause damage to the motors of manipulators. Physical constraints on acceleration level are also crucial because excessive joint accelerations may lead to discontinuous joint velocities, thus interrupting the task execution. There are many related works concerning about physical constraints of redundant manipulators. For instance, a control method is devised for redundant robot [9], which achieves joint limit avoidance and smooth

Manuscript received December 6, 2021; revised March 11, 2022; accepted March 27, 2022. Date of publication April 12, 2022; date of current version May 2, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62176109, in part by the CIE-Tencent Robotics X Rhino-Bird Focused Research Program under Grant 2021-01, in part by the Natural Science Foundation of Gansu Province under Grant 21JR7RA531, in part by the National Key Research and Development Program of China under Grant 2017YFE0118900, in part by the Special Projects of the Central Government in Guidance of Local Science and Technology Development under Grant YDZX20216200001297, in part by the Science and Technology Project of Chengguan Discrict of Lanzhou under Grant 2021JSCX0014, in part by the Tibetan Information Processing and Machine Translation Key Laboratory of Qinghai Province under Grant 2021-Z-003, and in part by the Supercomputing Center of Lanzhou University. *(Corresponding author: Long Jin.)*

Mei Liu, Jialiang Fan, Shuai Li, and Long Jin are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China, with the State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining 810016, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: mliu@lzu.edu.cn; fanjl20@lzu.edu.cn; lishuai@lzu.edu.cn; jinlong@lzu.edu.cn).

Yu Zheng is with Robotics X, Tencent, Shenzhen 518057, China (e-mail: petezheng@tencent.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2022.3165279.

Digital Object Identifier 10.1109/TIE.2022.3165279

trajectories. It can be applied to online sensor-based trajectory tracking tasks or act as a null-space velocity with planned task trajectories. In [10], an acceleration-level scheme is presented to solve the joint drift problem existing in manipulator control. In addition, the joint acceleration constraint is incorporated as an inequality constraint. Some related works take into account the joint acceleration constraint [11], [12] but are all built on joint acceleration level. However, the acceleration-level redundancy resolution requires more computations than the velocity-level resolution, which is unsuitable for many real-time redundant manipulator control scenarios. Therefore, it is a practical and challenging problem to control a manipulator from the velocity level while considering the joint acceleration constraint. To this end, Zhang *et al.* [13] presented a velocity-level redundancy resolution for manipulators subject to joint acceleration limits An extra parameter is computed and applied to the neural network controller with only one extra computational step.

The work discussed earlier mainly focuses on the kinematics of a manipulator, which requires structure parameters, i.e., Denavit–Hartenberg (D-H) parameters, to deduce the corresponding Jacobian matrix if a scheme is built on velocity or acceleration level. In many realistic scenarios, the structure parameters of a manipulator may be inaccurate due to abrasion of the mechanical structure by long-term high-intensity use, which may cause failures in achieving the desired accuracy. The actual structure information of a redundant manipulator possibly deviates from factory settings with D-H parameters deviating. Besides, the structure information is perhaps indeterminate and alterable for a homemade manipulator utilized for hardware development. There are some methods to deal with this issue, such as the kinematics calibration, which is of great significance for improving the manipulator's accuracy and can be divided into two types, i.e., 1) model-based; and 2) nonparametric kinematics calibration. Chen *et al.* [14] introduced several kinematics calibration methods and discuss their advantages and disadvantages. In addition, exploiting a data-driven technology based on a gradient descent method to address this problem is also an effective way that has made outstanding achievements in recent years. In [15], a model-free dual neural network is devised, which unifies the learning and control parts simultaneously to find a redundancy resolution of the manipulator. Xie *et al.* [16] extended data-driven approaches to redundant manipulators with unknown structure parameters to perform a cyclic-motion generation task, and the task is successfully fulfilled with tiny tracking errors.

Inspired by the discussions earlier, this article proposes a new simultaneous learning and control scheme for redundant manipulators. The scheme is built on the velocity level considering not only the joint angle and velocity constraints, but also the joint acceleration constraint and is suitable for manipulators with unknown structure parameters. In addition, the repetitive motion planning (RMP) performance index is constructed as the objective function. The physical constraints at the joint angle and joint velocity level are unified as one inequality constraint. Besides, a parameter is designed to guarantee that the joint acceleration will not exceed the predefined threshold. Finally, the scheme is formulated as a quadratic programming (QP) problem solved by a recurrent neural network (RNN). The research route map of this article is briefly presented in Fig. 1. The main contributions in this article can be summarized as follows:

1) For the first time, a simultaneous learning and control scheme is built on the velocity level but has acceleration-level constraints, without structure parameters of the manipulator required.
2) The scheme is reformulated as a QP problem and an RNN is devised to solve the problem, which simultaneously unifies the learning and control parts.
3) Theoretical analysis substantiates the stability and convergence of the proposed RNN. In addition, computer simulations and physical experiment are presented to verify the correctness and efficiency of the proposed scheme.

The rest of this article is organized as follows. In Section II, the preliminary knowledge of the scheme is introduced, and the scheme is presented as a QP problem. In Section III, an RNN is devised for the scheme. Section IV presents the theoretical analysis of the RNN. Then, computer simulations, the physical experiment, and comparisons are shown in Section V. Finally, Section VI concludes this article.

II. Preliminaries and Scheme Formulation

In this section, the preliminary knowledge of the scheme is presented. Then, the scheme is formulated as a QP problem.

A. Preliminaries

The forward kinematics plays a basic role in the manipulator control, which is determined by a manipulator's structure parameters, i.e., D-H parameters, describing a nonlinear mapping from joint space to Cartesian space [17]. The basic mapping from base coordinate to Cartesian coordinate is usually described as a nonlinear function $f(\cdot)$ from joint angle $\boldsymbol{\theta}(t) \in \mathbb{R}^a$ to end-effector position $\boldsymbol{p}(t) \in \mathbb{R}^b$. In the following part, the argument t is omitted somewhere for description convenience. A more common way to handle the forward kinematics is to turn the mapping into a velocity level as

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{p}} \tag{1}$$

where $\boldsymbol{J} = \partial f/\partial \boldsymbol{\theta} \in \mathbb{R}^{b\times a}$ is the so-called Jacobian matrix, representing the structure parameters of the manipulator on the velocity level; $\dot{\boldsymbol{\theta}} \in \mathbb{R}^a$ represents the velocity in joint space; $\dot{\boldsymbol{p}} \in \mathbb{R}^b$ represents the end-effector velocity in Cartesian space. Generally, for a trajectory tracking task of the redundant manipulator, the manipulator's end-effector position and velocity should conform to the desired trajectory and velocity, that is, $\boldsymbol{p}_{\mathrm{r}} \to \boldsymbol{p}_{\mathrm{d}}$ and $\dot{\boldsymbol{p}}_{\mathrm{r}} \to \dot{\boldsymbol{p}}_{\mathrm{d}}$, where the subscripts "r" and "d" represent the actual and desired values, respectively. To fulfill the requirements, it can be designed as

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} + \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}}) = \dot{\boldsymbol{p}}_{\mathrm{d}} \tag{2}$$

where $\kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}})$ is a position error feedback item [18]; $\kappa > 0$ is the corresponding coefficient. When the manipulator executing a closed trajectory tracking task, the joints may not return to their initial states, which is called the joint angle drift phenomenon [10]. To minimize the position error between the initial

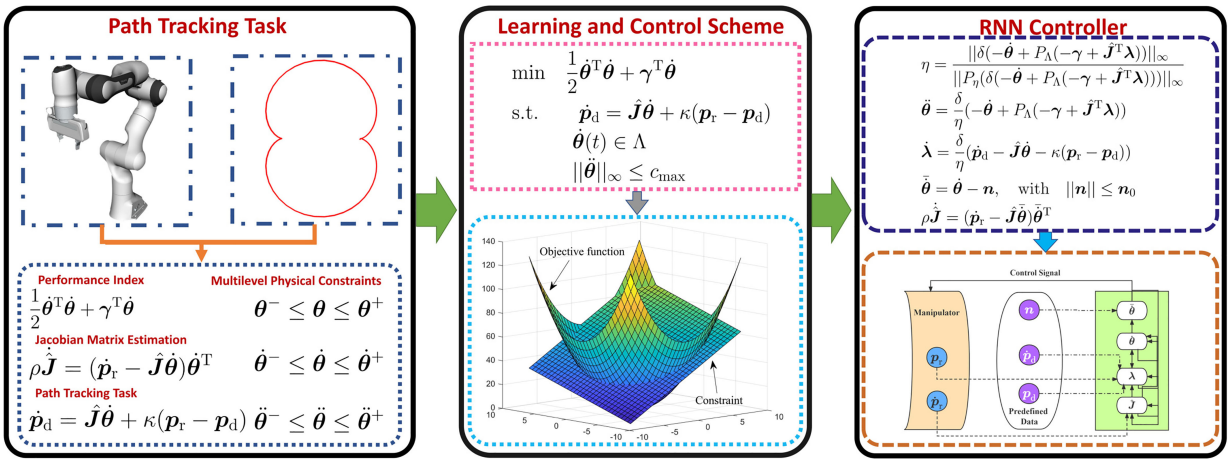


Fig. 1. Research route map of this article.

and final states, the RMP performance index can be devised as $\dot{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\boldsymbol{\theta}}/2+\boldsymbol{\gamma}^{\mathrm{T}}\dot{\boldsymbol{\theta}}$ where $\boldsymbol{\gamma}=d(\boldsymbol{\theta}-\boldsymbol{\theta}_0)$; $\boldsymbol{\theta}_0$ represents the initial state of the manipulator, and $d>0$ is used to scale the magnitude of the manipulator response to the joint displacement [19].

Recall that the manipulator control requires structure parameters. However, there always exists inevitable abrasion due to high-intensity usage in many industrial scenarios. Acquiring the precise value of robot structure parameters for high-accuracy control is becoming a significant problem in the robotic field. Hence, this article takes the problem into account to control manipulators with an unknown structure, and the forward kinematics on the velocity level is designed as

$$\hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}}=\hat{\dot{\boldsymbol{p}}}_{\mathrm{r}} \tag{3}$$

where $\hat{\boldsymbol{J}}$ represents the estimated value of the real-time Jacobian matrix and $\hat{\dot{\boldsymbol{p}}}_{\mathrm{r}}$ represents the calculated end-effector velocity. If $\hat{\boldsymbol{J}}$ is infinitely close to $\boldsymbol{J}$, $\hat{\dot{\boldsymbol{p}}}_{\mathrm{r}}$ is approximately equivalent to actual end-effector velocity $\dot{\boldsymbol{p}}_{\mathrm{r}}$. For a highly accurate Jacobian matrix estimation, an error function is designed as $\boldsymbol{\psi}=||\hat{\dot{\boldsymbol{p}}}_{\mathrm{r}}-\dot{\boldsymbol{p}}_{\mathrm{r}}||_2^2/2$. Then, by utilizing the gradient descent method, the following equation is derived [15], [20], [21]:

$$\hat{\boldsymbol{J}}^{k+1}=\hat{\boldsymbol{J}}^{k}-\xi\nabla\boldsymbol{\psi}(\hat{\boldsymbol{J}}^{k}) \tag{4}$$

where ξ represents the step size of the gradient descent method and k represents the iteration index. Furthermore, a continuous-time adaptation law is conducted from (4) as

$$\dot{\hat{\boldsymbol{J}}}=\frac{\hat{\boldsymbol{J}}^{k+1}-\hat{\boldsymbol{J}}^{k}}{\tau}=\frac{-\xi\nabla\boldsymbol{\psi}(\hat{\boldsymbol{J}}^{k})}{\tau}=-\frac{\xi}{\tau}(\hat{\dot{\boldsymbol{p}}}_{\mathrm{r}}-\dot{\boldsymbol{p}}_{\mathrm{r}})\dot{\boldsymbol{\theta}}^{\mathrm{T}} \tag{5}$$

where $\dot{\hat{\boldsymbol{J}}}$ represents the derivative of $\hat{\boldsymbol{J}}$ with respect to time; $\tau>0$ is tiny enough. The continuous-time learning law is finally derived as

$$\rho\dot{\hat{\boldsymbol{J}}}=(\dot{\boldsymbol{p}}_{\mathrm{r}}-\hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}^{\mathrm{T}} \tag{6}$$

where $\rho=\tau/\xi\ll 1$ represents the step size of each iteration. It can be observed from (5) that the current $\hat{\boldsymbol{J}}$ is calculated from the measured end-effector velocity $\dot{\boldsymbol{p}}_{\mathrm{r}}$, $\hat{\boldsymbol{J}}$ in last time instant, and joint velocity $\dot{\boldsymbol{\theta}}$. It is worth noting that a smaller step size ρ leads to a more accurate $\hat{\boldsymbol{J}}$.

Remark 1: It is worth pointing out that we focus on the position control of the manipulator with the learning law on the Jacobian matrix that can be divided into position and orientation parts. Besides, the problem of the learning part can be regarded as a convex problem where the local minimum is identical to the global minimum. Therefore, the local minimum issue will not be encounter in this method.

Due to the inherent properties of the manipulator and the consideration of the practical application, the physical constraints issue of the manipulator is inevitable during the control. The joint physical constraints in terms of the angle, velocity, and acceleration are presented as

$$\begin{aligned}\boldsymbol{\theta}^{-}&\le\boldsymbol{\theta}\le\boldsymbol{\theta}^{+}\\ \dot{\boldsymbol{\theta}}^{-}&\le\dot{\boldsymbol{\theta}}\le\dot{\boldsymbol{\theta}}^{+}\\ \ddot{\boldsymbol{\theta}}^{-}&\le\ddot{\boldsymbol{\theta}}\le\ddot{\boldsymbol{\theta}}^{+}\end{aligned} \tag{7}$$

where the superscripts $^{+}$ and $^{-}$ denote the upper and lower bounds of the manipulator's feasible regions, respectively; $\ddot{\boldsymbol{\theta}}$ denotes the joint acceleration of the manipulator.

B. Scheme Formulation

In this section, a QP-based scheme is devised and formulated to address the two existing issues (unknown structure parameters and physical constraints on the decision variable and its derivative) in redundant manipulator control. In the scheme, the RMP performance index is chosen as the objective function for repetitive trajectory tracking tasks, while the manipulator's structure parameters are assumed as unknown. Besides, physical constraints on the joint angle, joint velocity, and joint acceleration level are considered simultaneously. The scheme is presented as

$$\min\frac{1}{2}\dot{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\boldsymbol{\theta}}+\boldsymbol{\gamma}^{\mathrm{T}}\dot{\boldsymbol{\theta}} \tag{8a}$$

$$\text{s.t. }\hat{\dot{\boldsymbol{p}}}_{\mathrm{d}}=\hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}}+\kappa(\boldsymbol{p}_{\mathrm{r}}-\boldsymbol{p}_{\mathrm{d}}) \tag{8b}$$

$$\boldsymbol{\theta}^{-}\le\boldsymbol{\theta}\le\boldsymbol{\theta}^{+} \tag{8c}$$

$$\dot{\boldsymbol{\theta}}^{-} \leq \dot{\boldsymbol{\theta}} \leq \dot{\boldsymbol{\theta}}^{+} \tag{8d}$$

$$\ddot{\boldsymbol{\theta}}^{-} \leq \ddot{\boldsymbol{\theta}} \leq \ddot{\boldsymbol{\theta}}^{+}. \tag{8e}$$

It is a challenging problem to find a redundancy resolution of (8) due to the different level physical constraints (8c) and (8e), which are not directly relevant to the control variable $\dot{\boldsymbol{\theta}}$. According to [22], the joint constraints (8c) and (8d) can be transformed into a velocity-level constraint as

$$\boldsymbol{\zeta}^{-}(t) \leq \dot{\boldsymbol{\theta}}(t) \leq \boldsymbol{\zeta}^{+}(t) \tag{9}$$

where $\boldsymbol{\zeta}^{-}(t) = \max\{\dot{\boldsymbol{\theta}}^{-}, \alpha(\boldsymbol{\theta}^{-} - \boldsymbol{\theta}(t))\}$ and $\boldsymbol{\zeta}^{+}(t) = \min\{\dot{\boldsymbol{\theta}}^{+}, \alpha(\boldsymbol{\theta}^{+} - \boldsymbol{\theta}(t))\}$, with $\alpha > 0$ to scale the feasible region of $\dot{\boldsymbol{\theta}}$. For the notation presentation convenience, let $\Lambda = \{\dot{\boldsymbol{\theta}} \in \mathbb{R}^{a} | \boldsymbol{\zeta}^{-}(t) \leq \dot{\boldsymbol{\theta}} \leq \boldsymbol{\zeta}^{+}(t)\}$. Then, (9) is rewritten as

$$\dot{\boldsymbol{\theta}}(t) \in \Lambda. \tag{10}$$

As for the joint acceleration constraint (8e), we exploit the infinite-norm to guarantee that the joint accelerations of manipulators are within the range, which is presented as

$$||\ddot{\boldsymbol{\theta}}||_{\infty} \leq c_{\max} \tag{11}$$

where $c_{\max}$ denotes the minimum absolute value of all elements in $\ddot{\boldsymbol{\theta}}^{+}$ and $\ddot{\boldsymbol{\theta}}^{-}$; $||\cdot||_{\infty}$ denotes the infinity norm of a vector. Based on the derivation of the abovementioned formulas (8)–(11), the scheme is finally formulated as

$$\min \frac{1}{2}\dot{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\boldsymbol{\theta}} + \boldsymbol{\gamma}^{\mathrm{T}}\dot{\boldsymbol{\theta}} \tag{12a}$$

$$\text{s.t.}\ \dot{\boldsymbol{p}}_{\mathrm{d}} = \hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}} + \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}}) \tag{12b}$$

$$\dot{\boldsymbol{\theta}}(t) \in \Lambda \tag{12c}$$

$$||\ddot{\boldsymbol{\theta}}||_{\infty} \leq c_{\max}. \tag{12d}$$

Hereto, the formulation of the scheme is finished.

Remark 2: In scheme (12), the RMP performance index is chosen as the objective function for repetitive motion. Equation (12b) guarantees that the manipulator moves along the predefined trajectory. The joint angle and joint velocity constraints are unified into one level. Besides, the joint acceleration constraint is transformed into an inequality constraint. It should be noted that scheme (12) is not a regular QP problem for (12d) built on acceleration level, whereas the decision variable on the velocity level.

III. RNN Solver

In this section, an RNN based on the Lagrange methods and the Karush–Kuhn–Tucker (KKT) condition is presented for finding the redundancy resolution of the scheme (12). The design of the RNN is divided into two parts. First, we assume that the Jacobian matrix $\boldsymbol{J}$ of the redundant manipulator is known. Then, a neural network controller is devised for the scheme. In the second part, the derivation formula for the Jacobian matrix is incorporated into the devised scheme. Besides, an intentional noise signal is added to the neural network to activate the control system.

The Lagrange function according to (12) is defined as $L = \dot{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\boldsymbol{\theta}}/2 + \boldsymbol{\gamma}^{\mathrm{T}}\dot{\boldsymbol{\theta}} + \boldsymbol{\lambda}^{\mathrm{T}}(\dot{\boldsymbol{p}}_{\mathrm{d}} - \boldsymbol{J}\dot{\boldsymbol{\theta}} - \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}}))$, where $\boldsymbol{\lambda} \in \mathbb{R}^{b}$ is the Lagrange multiplier of (12b). To obtain the optimal solution of (12), first, we take the partial derivatives to $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\lambda}$, respectively, and the following equation set should be satisfied:

$$\frac{\partial L}{\partial \dot{\boldsymbol{\theta}}} = \dot{\boldsymbol{\theta}} + \boldsymbol{\gamma} - \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda} = 0 \tag{13a}$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = \dot{\boldsymbol{p}}_{\mathrm{d}} - \boldsymbol{J}\dot{\boldsymbol{\theta}} - \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}}) = 0. \tag{13b}$$

Besides, in the light of the KKT conditions [23] and the definition of projection operator [24], constraint (12c) is applied on $\dot{\boldsymbol{\theta}}$ as the following format based on (13a):

$$\dot{\boldsymbol{\theta}} = P_{\Lambda}\left(\dot{\boldsymbol{\theta}} - \frac{\partial L}{\partial \dot{\boldsymbol{\theta}}}\right) = P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}) \tag{14}$$

where $P_{\Lambda}(\mathbf{x})$ represents the projection of $\mathbf{y}$ in set Λ satisfying the following equation:

$$P_{\Lambda}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y}\in\Lambda}||\mathbf{x} - \mathbf{y}||_{2}. \tag{15}$$

Then, according to the existing RNN scheme [25], the controller for the scheme is devised as

$$\ddot{\boldsymbol{\theta}} = \delta(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})) \tag{16a}$$

$$\dot{\boldsymbol{\lambda}} = \delta(\dot{\boldsymbol{p}}_{\mathrm{d}} - \boldsymbol{J}\dot{\boldsymbol{\theta}} - \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}})) \tag{16b}$$

where $\delta > 0 \in \mathbb{R}$ is a parameter that takes control of the neural network's convergence rate. In the control process, the control signal, i.e., $\dot{\boldsymbol{\theta}}$ is yielded by the neural dynamic $\ddot{\boldsymbol{\theta}}$ from each iteration of the RNN.

In scheme (12), we define a bound $c_{\max}$ for the infinite norm of the joint acceleration of the manipulator. The constraint (12e) can be transformed according to the devised neural network controller

$$||\delta(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))||_{\infty} \leq c_{\max}. \tag{17}$$

In order to incorporate (17) into the controller, a parameter $\eta > 0 \in \mathbb{R}$ is defined to scale the joint acceleration, which is presented as

$$\eta = \frac{||\delta(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))||_{\infty}}{||P_{\eta}(\delta(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})))||_{\infty}} \tag{18}$$

where $\Omega = \{\mathbf{x} \mid ||\mathbf{x}||_{\infty} < c_{\max}\}$; $P_{\eta}(\cdot)$ denotes a projection operation into Ω and makes the max absolute value of the element in $\mathbf{x}$ less than $c_{\max}$. Finally, the neural network controller with joint acceleration constraint is presented as follows:

$$\eta = \frac{||\delta(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))||_{\infty}}{||P_{\eta}(\delta(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})))||_{\infty}} \tag{19a}$$

$$\ddot{\boldsymbol{\theta}} = \frac{\delta}{\eta}(-\dot{\boldsymbol{\theta}} + P_{\Lambda}(-\boldsymbol{\gamma} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})) \tag{19b}$$

$$\dot{\boldsymbol{\lambda}} = \frac{\delta}{\eta}(\dot{\boldsymbol{p}}_{\mathrm{d}} - \boldsymbol{J}\dot{\boldsymbol{\theta}} - \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}})). \tag{19c}$$

Compared with RNN (16), only one extra computational procedure is added to the neural network controller (19), which means that the computational complexity is not much increased.

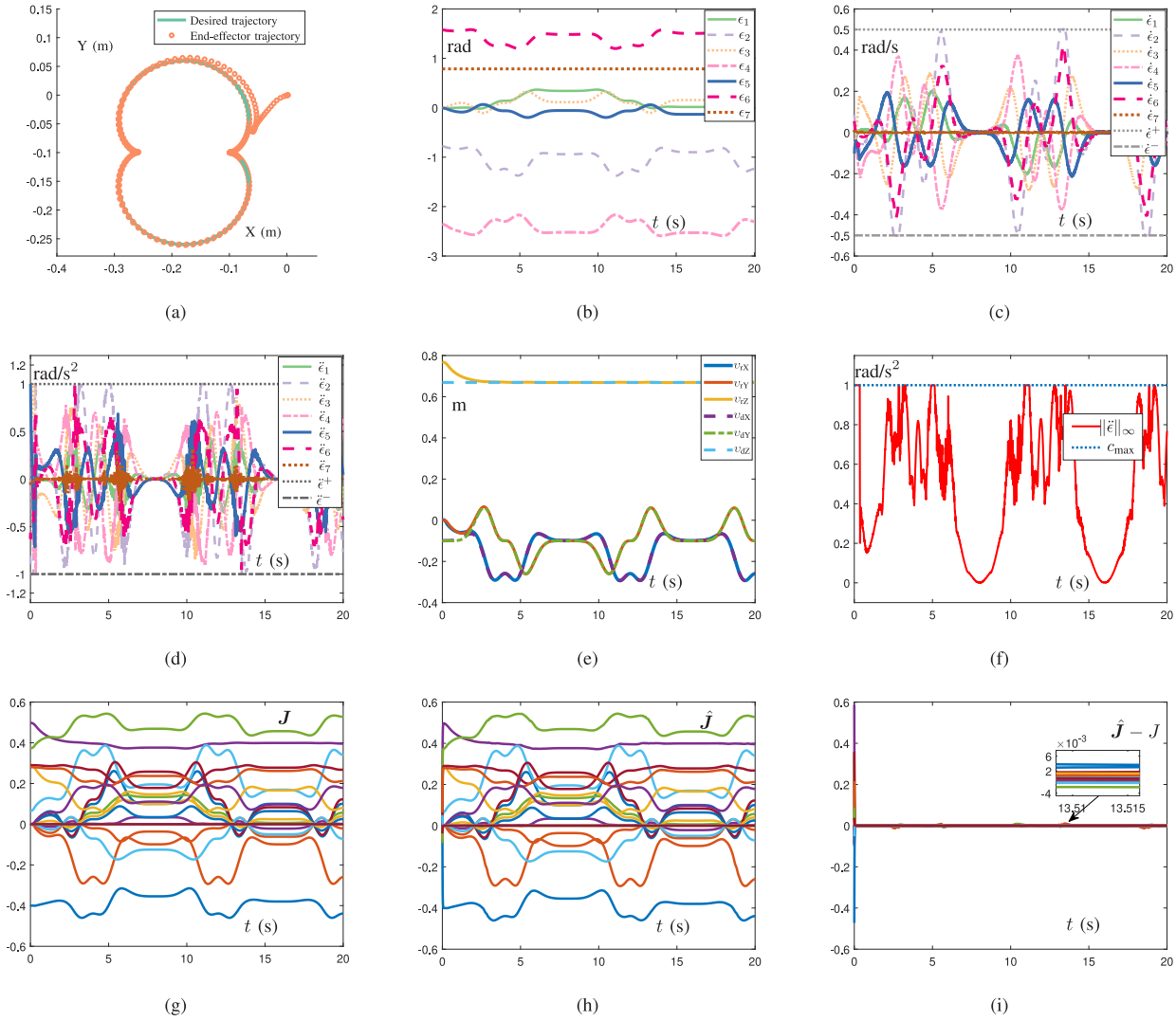


Fig. 2. Computer simulations on employing scheme (12) to track a nephroid trajectory. (a) Desired trajectory and the end-effector's trajectory. (b) Time history of joint angles. (c) Time history of joint velocities. (d) Time history of joint accelerations. (e) Time history of position errors. (f) Time history of the infinity norm of joint accelerations with bound $c_{\max} = 1\text{rad/s}^2$. (g) Time history of the real Jacobian matrix. (h) Time history of the estimated Jacobian matrix. (i) Time history of the Jacobian matrix error.

When the structure parameters of the redundant manipulator are unknown, revisit that the aforementioned derivation (1) of the Jacobian matrix requires the manipulator's structure parameters. To extend the scheme to the manipulator whose structure parameters are unknown, the Jacobian matrix in the controller (19) is replaced by the estimated one in (3). Besides, an intentionally devised noise is applied to drive the control system and increase the diversity of signals [15]

$$\eta = \frac{||\delta(-\dot{\boldsymbol{\theta}} + P_\Lambda(-\boldsymbol{\gamma} + \hat{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{\lambda}))||_\infty}{||P_\eta(\delta(-\dot{\boldsymbol{\theta}} + P_\Lambda(-\boldsymbol{\gamma} + \hat{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{\lambda})))||_\infty} \tag{20a}$$

$$\ddot{\boldsymbol{\theta}} = \frac{\delta}{\eta}(-\dot{\boldsymbol{\theta}} + P_\Lambda(-\boldsymbol{\gamma} + \hat{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{\lambda})) \tag{20b}$$

$$\dot{\boldsymbol{\lambda}} = \frac{\delta}{\eta}(\dot{\boldsymbol{p}}_{\mathrm{d}} - \hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}} - \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}})) \tag{20c}$$

$$\bar{\dot{\boldsymbol{\theta}}} = \dot{\boldsymbol{\theta}} - \boldsymbol{n}, \quad \text{with} \quad ||\boldsymbol{n}|| \leq \boldsymbol{n}_0 \tag{20d}$$

$$\rho\dot{\hat{\boldsymbol{J}}} = (\dot{\boldsymbol{p}}_{\mathrm{r}} - \hat{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}})\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}} \tag{20e}$$

where $\boldsymbol{n}_0 > 0$ and σ are the bound and derivation of the independent identically distributed noise $\boldsymbol{n}$, respectively.

Remark 3: The devised neural network (20) has two special features compared with most of the existing controllers of redundant manipulators: velocity-level resolution with acceleration-level physical constraint and no structure parameters of the manipulator. An extra computational procedure (20a) is introduced to keep the value of acceleration within a Certain range. In addition, there are no structure parameters involved in the control. The learning and control steps are unified simultaneously in the control loop. It is worth mentioning that $\boldsymbol{n}$ increases the diversity of input signals, and it allows the algorithm to explore neighborhoods around the saddle point [26]. Besides,

convergence errors due to $\boldsymbol{n}$ can be ignored by setting a proper value of $\boldsymbol{n}_0$ according to the system's sensitivity.

IV. Theoretical Analysis

In this section, the theoretical analysis is presented to verify the feasibility of RNN (20). First, the learning convergence of RNN (20) is presented. Then, the control convergence is proved when the acceleration constraint is maintained.

Theorem 1: The estimation error $\boldsymbol{J}-\hat{\boldsymbol{J}}$ is convergent to zero when adopting RNN (20) to control a redundant manipulator.

Proof: First, we define the estimation error $\bar{\boldsymbol{J}}=\boldsymbol{J}-\hat{\boldsymbol{J}}$, and then, a Lyapunov function is devised as $V_1=||\bar{\boldsymbol{J}}||_{\mathrm{F}}^2=\mathrm{tr}(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}})/2$ where $||\cdot||_{\mathrm{F}}$ represents the Frobenius norm of a matrix, and $\mathrm{tr}(\cdot)$ represents the trace of a matrix. Taking the derivative of V_1 yields the following equation:

$$\begin{aligned}\dot{V}_1&=\mathrm{tr}(\dot{\bar{\boldsymbol{J}}}\bar{\boldsymbol{J}}^{\mathrm{T}})\\&=-\mathrm{tr}\left(1/\rho\bar{\boldsymbol{J}}^{\mathrm{T}}(\dot{\boldsymbol{p}}-\hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}^{\mathrm{T}}\right)\\&=-\mathrm{tr}\left(1/\rho\bar{\boldsymbol{J}}^{\mathrm{T}}(\boldsymbol{J}-\hat{\boldsymbol{J}})\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}^{\mathrm{T}}\right)\\&=-\mathrm{tr}\left(1/\rho\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})^{\mathrm{T}}\right)\\&=-\mathrm{tr}\left(1/\rho(\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n}))^{\mathrm{T}}\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})\right)\\&=-1/\rho\|\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})\|_{\mathrm{F}}^2\end{aligned}\tag{21}$$

where one can naturally derive that $\dot{V}_1\le 0$ and $V_1>0$. Therefore, the largest invariant set is found through the LaSalle's invariant set principle [27]. When $\dot{V}_1=0$ holds, $\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})=0$ is derived as $t\to\infty$. We multiply $(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})^{\mathrm{T}}\bar{\boldsymbol{J}}^{\mathrm{T}}$ on the above-mentioned equation and calculate the expected value as

$$\begin{aligned}&E\left((\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})^{\mathrm{T}}\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})\right)\\&=E\left(\mathrm{tr}\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})(\bar{\dot{\boldsymbol{\theta}}}+\boldsymbol{n})^{\mathrm{T}}\right)\right)\\&=\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}}\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}}\right)\right)+\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\right)\right)\\&\quad+\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}}\boldsymbol{n}^{\mathrm{T}}\right)\right)+\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\boldsymbol{n}\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}}\right)\right)\\&=\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}}\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}}\right)\right)+\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\right)E\left(\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\right)\right)\\&\quad+\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}}\right)E^{\mathrm{T}}(\boldsymbol{n})\right)+\mathrm{tr}\left(E\left(\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}}\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\right)E(\boldsymbol{n})\right)\\&=\mathrm{tr}\left(E\left(\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}}\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}}\right)\right)+\sigma^2\,\mathrm{tr}\left(E\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\right)\right)\\&=0,\quad\text{as } t\to\infty,\end{aligned}\tag{22}$$

where $E(\cdot)$ represents the expected value of a matrix. Considering that $\mathrm{tr}(E(\bar{\dot{\boldsymbol{\theta}}}^{\mathrm{T}}\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}\bar{\dot{\boldsymbol{\theta}}}))\ge 0$, we can draw the conclusion that $\mathrm{tr}(E(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}))=E(\mathrm{tr}(\bar{\boldsymbol{J}}^{\mathrm{T}}\bar{\boldsymbol{J}}))=E(||\bar{\boldsymbol{J}}||_{\mathrm{F}}^2)=0$, which further implies $\bar{\boldsymbol{J}}=\boldsymbol{J}-\hat{\boldsymbol{J}}=0$ as $t\to\infty$. Here to, the estimation error of the Jacobian matrix converging to zero during the control process is successfully proved, which further indicates the learning convergence of RNN (20). ■

Lemma 1: For any convex set $\Lambda\in\mathbb{R}^p$, according to [28], there has

$$(\mathbf{m}-P_\Lambda(\mathbf{m}))^{\mathrm{T}}(P_\Lambda(\mathbf{m})-\boldsymbol{\omega})\ge 0\quad\forall\mathbf{m}\in\mathbb{R}^p\ \forall\boldsymbol{\omega}\in\Lambda.\tag{23}$$

The equality holds only if $\mathbf{m}\in\Lambda$. Furthermore, there has

$$\begin{aligned}&(\mathbf{m}-P_\Lambda(\mathbf{m}))^{\mathrm{T}}(\mathbf{m}-\boldsymbol{\omega})\\&=(\mathbf{m}-P_\Lambda(\mathbf{m}))^{\mathrm{T}}(\mathbf{m}-P_\Lambda(\mathbf{m})+P_\Lambda(\mathbf{m})-\boldsymbol{\omega})\\&=\|(\mathbf{m}-P_\Lambda(\mathbf{m}))\|_2^2+(\mathbf{m}-P_\Lambda(\mathbf{m}))^{\mathrm{T}}(P_\Lambda(\mathbf{m})-\boldsymbol{\omega})\ge 0.\end{aligned}\tag{24}$$

Theorem 2: The neural dynamics $\mathbf{u}=[\dot{\boldsymbol{\theta}}^{\mathrm{T}},\boldsymbol{\lambda}^{\mathrm{T}}]^{\mathrm{T}}$ converges globally to a steady state with joint acceleration constraint satisfaction when applying RNN (20) on scheme (12).

Proof: We first prove that the joint acceleration is constrained within the predefined bounds. Given that the convergence of estimation error is proved in the previous part, $\boldsymbol{J}$ is used to represent the estimated Jacobian matrix in the following theoretical analysis. Revisit that the definition of scaling parameter η is presented as

$$\eta=\begin{cases}1,\ \text{if } \delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))\in\Omega\\ \frac{\|\delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))\|_\infty}{c_{\max}}>0,\ \text{otherwise.}\end{cases}\tag{25}$$

When the joint acceleration exceeds the predefined constraint, based on (20) and (25), we have

$$\begin{aligned}\left\|\ddot{\boldsymbol{\theta}}\right\|_\infty&=\left\|\frac{\delta}{\eta}(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))\right\|_\infty\\&=\left\|\frac{c_{\max}\delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})}{\delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})}\right\|_\infty\\&=c_{\max}\left\|\frac{\delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})}{\delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})}\right\|_\infty\\&=c_{\max}.\end{aligned}$$

If the joint acceleration is within the predefined constraint, we have $\eta=1$, which further implies

$$\begin{aligned}\left\|\ddot{\boldsymbol{\theta}}\right\|_\infty&=\left\|\frac{\delta}{\eta}(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda}))\right\|_\infty\\&=\left\|\delta(-\dot{\boldsymbol{\theta}}+P_\Lambda(-\boldsymbol{\gamma}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda})\right\|_\infty\\&<c_{\max}.\end{aligned}$$

According to the discussion of the abovementioned two cases, it is concluded that $\ddot{\boldsymbol{\theta}}\le c_{\max}$ always holds, which means that the joint acceleration constraint (17) is satisfied all the time. ■

Then, the global convergence of RNN (20) is proved as follows. For the convenience of analyzing, a new variable is defined as $\mathbf{u}=[\dot{\boldsymbol{\theta}}^{\mathrm{T}},\boldsymbol{\lambda}^{\mathrm{T}}]^{\mathrm{T}}$, and the corresponding neural control

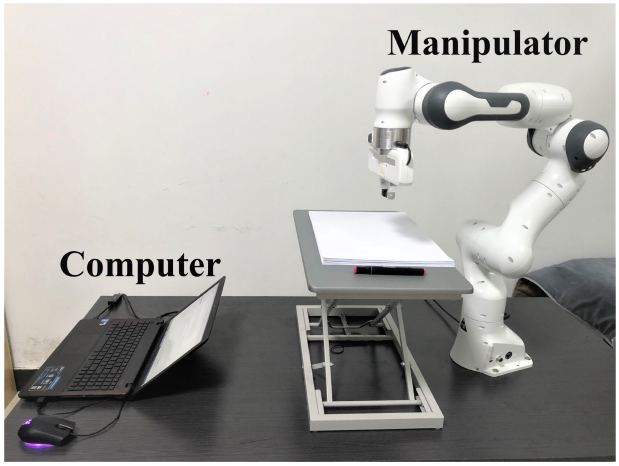


Fig. 3. Physical experiment platform.

law is rewritten as

$$\frac{\eta}{\delta}\dot{\mathbf{u}} = -\mathbf{u} + P_{\bar{\Lambda}}(\mathbf{u} - g(\mathbf{u})) \tag{26}$$

where $P_{\bar{\Lambda}} = \{(\dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}), \dot{\boldsymbol{\theta}} \in \Lambda \subset \mathbb{R}^a, \boldsymbol{\lambda} \in \mathbb{R}^b\}$, and $g(\cdot)$ is defined as

$$g(\mathbf{u}) = g(\dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \begin{bmatrix} \dot{\boldsymbol{\theta}} + \boldsymbol{\gamma} - \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\lambda} \\ \boldsymbol{J}\dot{\boldsymbol{\theta}} + \kappa(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}}) - \dot{\boldsymbol{p}}_{\mathrm{d}} \end{bmatrix}. \tag{27}$$

Calculate the gradient of $g(\mathbf{u})$ as

$$\nabla g = \frac{\partial g(\mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} I & -\boldsymbol{J} \\ \boldsymbol{J}^{\mathrm{T}} & 0 \end{bmatrix} \tag{28}$$

and

$$\nabla g + \nabla^{\mathrm{T}} g = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \tag{29}$$

Equation (29) is a semipositive definite matrix. In addition, according to the mean-value theorem: for any x, y, and z, when $z = \nu x + (1-\nu)y$, $(0 \le \nu \le 1)$, there exists $g(x) - g(y) = \nabla g(z)(x - y)$. The following formula is derived from the abovementioned analysis:

$$(x-y)^{\mathrm{T}}(g(x) - g(y)) = (x-y)^{\mathrm{T}}\nabla g(z)(x-y) \ge 0. \tag{30}$$

It is natural to conduct from formula (30) that $g(\cdot)$ is a monotone function, which is used in the following analysis. Define a Lyapunov function $V_2 = \|\mathbf{u} - P_{\bar{\Lambda}}(\mathbf{u})\|_2^2/2$. It can be easily conducted that $V_2 \ge 0$ and the equality holds when $\mathbf{u} \in \bar{\Lambda}$. The time derivative of V_2 is calculated as

$$\begin{aligned} \dot{V}_2 &= (\mathbf{u} - P_{\bar{\Lambda}}(\mathbf{u}))^{\mathrm{T}}\dot{\mathbf{u}} \\ &= -(\mathbf{u} - P_{\bar{\Lambda}}(\mathbf{u}))^{\mathrm{T}}(\mathbf{u} - P_{\bar{\Lambda}}(\mathbf{u} - g(\mathbf{u})))\,\delta/\eta. \end{aligned} \tag{31}$$

According to Lemma 1 and the monotony of $g(\cdot)$, we can derive that $(\mathbf{u} - P_{\bar{\Lambda}}(\mathbf{u}))^{\mathrm{T}}(\mathbf{u} - P_{\bar{\Lambda}}(\mathbf{u} - g(\mathbf{u}))) \ge 0$, and the equality holds only if $\mathbf{u} \in \bar{\Lambda}$. Combined with the abovementioned analysis, we can draw the conclusion that $\dot{V}_2 \le 0$ and the equality holds when $\mathbf{u} \in \bar{\Lambda}$. Thus, $\mathbf{u}$ converges in $\bar{\Lambda}$ with time according to the LaSalle's invariance principle. The proof is complete.

V. Simulations, Physical Experiment, and Comparisons

In this section, simulations, physical experiments, and comparisons are given to validate scheme (12). The simulations and the physical experiment are conducted on a 7-DOF redundant manipulator named Franka Emika Panda. The manipulator is given trajectory tracking tasks under the premise that the structure of the manipulator is unknown. The D-H parameters of the manipulator can be found in [30].

A. Simulations

In the simulation, the initial angular state of the manipulator is set to $\boldsymbol{\theta}_0 = [0, -\pi/4, 0, -3\pi/4, 0, \pi/2, \pi/4]^{\mathrm{T}}$ rad. The upper bound of the joint angle is set to $\boldsymbol{\theta}^+ = [2.8973, 1.7628, 2.8973, 0.0698, 2.8973, 3.7525, 2.8973]^{\mathrm{T}}$ rad according to Franka control interface documentation [30], and the lower bound of the joint angle is defined as $\boldsymbol{\theta}^- = [-2.8973, -1.7628, -2.8973, -3.0718, -2.8973, -0.0175, -2.8973]^{\mathrm{T}}$ rad. The scaling coefficient of RNN (20) is set to $\delta = 1000$. In order to verify the learning ability of the proposed scheme (12), the D-H parameters of the manipulator are assumed as unknown in the simulation, and the initial value of the estimated Jacobian matrix $\hat{\boldsymbol{J}}_{\boldsymbol{0}}$ is generated by the uniformly distributed random numbers in the range of [0,1].To validate the performance of the scheme (12), the manipulator is given a nephroid trajectory to track, and the radius of the trajectory is set to 0.04 m. The execution time of the trajectory is set to 8 s for one cycle, with the task execution time being 20 s. To better validate the convergence of the scheme (12), an initial position error $\boldsymbol{p}_{\mathrm{r0}} - \boldsymbol{p}_{\mathrm{d0}} = 0.1$ m is defined between the end-effector's initial position and the desired trajectory. Assume that the joint velocity constraint is set to $\dot{\boldsymbol{\theta}}^+ = -\dot{\boldsymbol{\theta}}^- = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]^{\mathrm{T}}$ rad/s, and the joint acceleration constraint of the manipulator is set to $\ddot{\boldsymbol{\theta}}^+ = -\ddot{\boldsymbol{\theta}}^- = [1.2, 1.6, 1.4, 1.4, 1.5, 1.8, 1]^{\mathrm{T}}\mathrm{rad/s}^2$; thus $c_{\max} = 1\mathrm{rad/s}^2$. The values of ξ and τ are set to 10^{-7} and 0.1, respectively, which means ρ is set as 10^{-6}, which obtains a superior performance in calculating the estimated Jacobian matrix $\hat{\boldsymbol{J}}$.

The simulation results are shown in Fig. 2. Fig. 2(a) depicts the comparison of the desired trajectory and the end-effector's trajectory, from which it can be observed that the end-effector quickly moves to the desired trajectory from the initial position, after which the end-effector's trajectory fits the desired trajectory pretty well. Fig. 2(b) shows the time history of the joint angles during the task execution, and it can be seen that the joint angles fluctuate smoothly within predefined physical constraints. Fig. 2(c) depicts the time history of the joint velocity, from which we can find that, when the joint velocity, e.g., $\dot{\boldsymbol{\theta}}_2$ reaches the bound of $\dot{\boldsymbol{\theta}}^+ = 0.5$ rad/s, its value does not increase anymore. Fig. 2(d) displays the time history of the joint accelerations. As shown in the figure, the joint accelerations do not violate the physical constraints when reaching the upper bound or the lower bound. Fig. 2(e) depicts the coordinate values of the

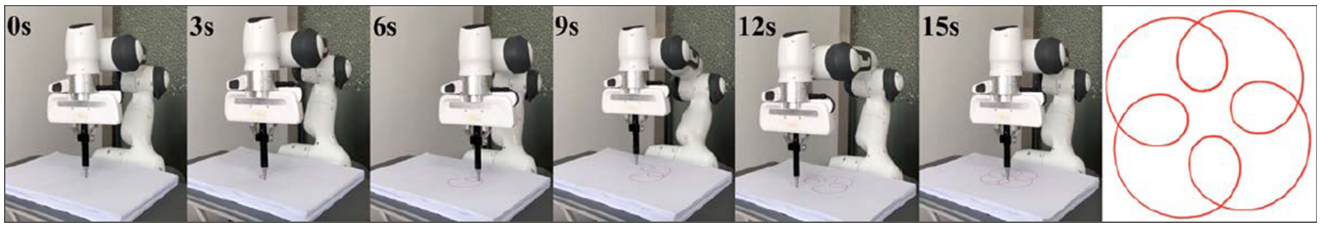
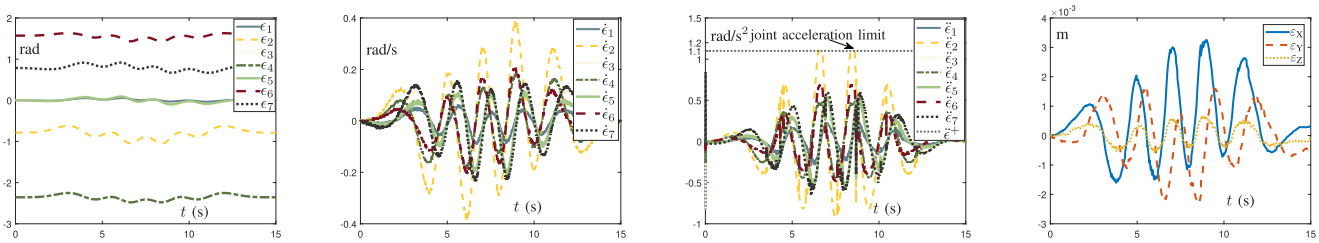


Fig. 4. Snapshots of the manipulator and the experiment results when performing the given task via scheme (12) aided with RNN (20). The experiment video is available on https://youtu.be/dSnp8O2A_W4/.



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

Fig. 5. Physical experiment results on employing scheme (12) to track a four-ring like trajectory. (a) Time history of joint angles. (b) Time history of joint velocities. (c) Time history of joint accelerations. (d) Time history of end-effector errors $(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_{\mathrm{d}})$.

end-effector trajectory $\boldsymbol{p}_{\mathrm{r}}$ and the desired trajectory $\boldsymbol{p}_{\mathrm{d}}$, which is similar as Fig. 2(a), and the trajectory of the end-effector quickly converges to the desired trajectory. Fig. 2(f) shows the infinity norm of the joint accelerations, and it is bounded during the task execution. It can be found that the infinity norm of the joint accelerations reaches the bound many times but is well constrained. Fig. 2(g) and (h) depict the time history of the real Jacobian matrix and the estimated one, respectively, from which we can find that the real Jacobian matrix and the desired one have the same fluctuations, reflecting the superior learning ability of the scheme (12). Fig. 2(i) depicts the time history of the Jacobian matrix estimation error, and we can find that there exists a huge error in the initial state because the initial values are random, while it immediately converges to a small range (10^{-3}), which has a limited influence on the control performance and can be ignored.

Fig. 2 exhibits a superior performance of the scheme (12), which has good abilities on learning and control parts, simultaneously. In the following section, we will depict physical experiment results to further validate the scheme (12).

B. Physical Experiment

To further validate the realistic application of the scheme (12), the physical experiment is carried out on the Franka Emika Panda manipulator. The manipulator is given a four-ring like trajectory to track within 15 s. The values of ξ and τ are set to 10^{-7} and 0.1, respectively; $c_{\max} = 1.1\mathrm{rad/s}^2$; the parameter of RNN (20) is set to $\delta = 1000$. The experiment platform is shown in Fig. 3.

Fig. 4 presents the states of the manipulator when tracking the desired trajectory in different time instants. We can observe that the end-effector's trajectory is closed and smooth, which means that the task is well performed. Fig. 5 depicts the time history of the joint angles, joint velocities, and joint accelerations during the experiment. Fig. 5(a) and (b) depict the time history of joint angles and velocities of the manipulator, which fluctuate smoothly and return to their initial states at the end of the task, indicating that the RMP performance index is successfully fulfilled. Fig. 5(c) depicts the time history of the joint accelerations during the task. As shown in Fig. 5(c), the joint accelerations are well constrained when reaching the bound. Fig. 5(d) depicts the end-effector error between the actual position and the desired position, and it can be seen that the error fluctuates within a small range (10^{-3} m).

C. Comparisons

To further prove the novelty of the proposed scheme, in this section, the proposed scheme (12) is compared with robot schemes in [9], [11]–[13], [15], [16], [29], [31], [32] from five perspectives, i.e., the driven signal of the controller, joint constraints, decision variable and derivative level constraints (DVDLC), physical experiment verification, and unknown parameters. The comparison results are shown in Table I. As shown in the table, schemes in [13], [31], and scheme (12) are all built on the velocity level with multilevel physical constraints considered, but only (12) can work when the structure parameters of the manipulator are unknown, and only scheme (12) is verified by the physical experiment. Schemes in [15], [32] can be applied when the model parameters of the manipulator are unknown, but the acceleration constraints are not fulfilled. The comparison results verify the novelty and superiority of the scheme (12).

TABLE I
COMPARISONS AMONG DIFFERENT CONTROLLERS FOR REDUNDANCY RESOLUTION OF MANIPULATORS

	Driven signal	Joint constraints	DVDLC*	Physical experiment verification	Unknown parameters
Scheme (12)	Velocity	Angle & Velocity & Acceleration	Yes	Yes	Yes
Scheme in [9]	Velocity	Angle	No	No	No
Scheme in [11]	Acceleration	Angle & Velocity & Acceleration	No	Yes	No
Scheme in [12]	Acceleration	Angle & Velocity & Acceleration	No	No	No
Scheme in [13]	Velocity	Angle & Velocity & Acceleration	Yes	No	No
Scheme in [15]	Velocity	Angle & Velocity	No	No	Yes
Scheme in [16]	Velocity	Angle & Velocity	No	No	Yes
Scheme in [29]	Acceleration	Velocity	No	No	No
Scheme in [31]	Velocity	Angle & Velocity & Acceleration	Yes	No	No
Scheme in [32]	Velocity	Angle & Velocity	No	Yes	Yes

*"DVDLC" means whether the scheme has decision variable and its derivative level constraints, e.g., the scheme built on the velocity level has velocity and acceleration constraints.

VI. CONCLUSION

This work has achieved a simultaneous learning and control scheme for redundant manipulators with physical constraints on the decision variable and its derivative considered. The scheme has fulfilled the trajectory tracking task of redundant manipulators with RMP performance index as the objective function. Aided with data-driven methods, the scheme can be applied to redundant manipulators whose structure parameters involved in the kinematics are implicit or unknown. Besides, different from most manipulator control schemes, the proposed velocity-level scheme has considered the joint acceleration constraint with only one extra computational step. A RNN has been devised to address the learning and control problems. Theoretical analysis has proved the correctness of the scheme. Simulations and the physical experiment on the Franka Emika Panda manipulator have validated the effectiveness and feasibility of the proposed scheme. To the best of our knowledge, this is the first time to combine the learning and control simultaneously with physical constraints on the decision variable and its derivative considered. Future works can be extended to integrate the data-driven method to orientation control of the manipulator or investigate the data-driven method on dynamics control of the manipulator.

REFERENCES

[1] G. Hwang, J. Park, D. S. D. Cortes, K. Hyeon, and K.-U. Kyung, "Electroadhesion-based high-payload soft gripper with mechanically strengthened structure," *IEEE Trans. Ind. Electron.*, vol. 69, no. 1, pp. 642–651, Jan. 2022.

[2] B. Hu, Z.-H. Guan, F. L. Lewis, and C. L. P. Chen, "Adaptive tracking control of cooperative robot manipulators with Markovian switched couplings," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2427–2436, Mar. 2021.

[3] S. Baek, H. Lee, and S. Han, "Communication-efficient event-triggered time-delay control and its application to robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9288–9297, Sep. 2022.

[4] Z. Xu, X. Zhou, H. Wu, X. Li, and S. Li, "Motion planning of manipulators for simultaneous obstacle avoidance and target tracking: An RNN approach with guaranteed performance," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3887–3897, Apr. 2022.

[5] J. Obregón-Flores, G. Arechavaleta, H. M. Becerra, and A. Morales-Díaz, "Predefined-time robust hierarchical inverse dynamics on torque-controlled redundant manipulators," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 962–978, Jun. 2021.

[6] Z. Xu, S. Li, X. Zhou, S. Zhou, T. Cheng, and Y. Guan, "Dynamic neural networks for motion-force control of redundant manipulators: An optimization perspective," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1525–1536, Feb. 2021.

[7] L. Jin, Z. Xie, M. Liu, K. Chen, C. Li, and C. Yang, "Novel joint-drift-free scheme at acceleration level for robotic redundancy resolution with tracking error theoretically eliminated," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 1, pp. 90–101, Feb. 2021.

[8] L. Jin *et al.*, "Perturbed manipulability optimization in a distributed network of redundant robots," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7209–7220, Aug. 2021.

[9] A. Atawnih, D. Papageorgiou, and Z. Doulgeri, "Kinematic control of redundant robots with guaranteed joint limit avoidance," *Robot. Auton. Syst.*, vol. 79, pp. 122–131, May 2016.

[10] Z. Xie, L. Jin, X. Du, X. Xiao, H. Li, and S. Li, "On generalized RMP scheme for redundant robot manipulators aided with dynamic neural networks and nonconvex bound constraints," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5172–5181, Sep. 2019.

[11] D. Guo and Y. Zhang, "Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6903–6914, Dec. 2014.

[12] A. D. Prete, "Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 281–288, Jan. 2018.

[13] Y. Zhang, S. Li, and X. Zhou, "Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3573–3582, May 2019.

[14] G. Chen, T. Li, M. Chu, Q. X. Jia, and H. X. Sun, "Review on kinematics calibration technology of serial robots," *Int. J. Precis. Eng. Manuf.*, vol. 15, no. 8, pp. 1759–1774, Aug. 2014.

[15] S. Li, Z. Shao, and Y. Guan, "A dynamic neural network approach for efficient control of manipulators," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 49, no. 5, pp. 932–941, May 2019.

[16] 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.*, vol. 29, no. 1, pp. 53–63, Jan. 2021.

[17] B. M. Yilmaz, E. Tatlicioglu, A. Savran, and M. Alci, "Self-adjusting fuzzy logic based control of robot manipulators in task space," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1620–1629, Feb. 2022.

[18] Z. Zhang, S. Chen, and J. Liang, "Discrete-time circadian rhythms neural network for perturbed redundant robot manipulators tracking problem with periodic noises," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 242–251, Jan. 2022.

[19] Z. Li, B. Liao, F. Xu, and D. Guo, "A new repetitive motion planning scheme with noise suppression capability for redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 12, pp. 5244–5254, Dec. 2020.

[20] M. Liu, L. Chen, X. Du, and M. Shang, "Activated gradients for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 1, 2021, doi: 10.1109/TNNLS.2021.3106044.

[21] L. Jin, L. Wei, and S. Li, "Gradient-based differential neural-solution to time-dependent nonlinear optimization," *IEEE Trans. Autom. Control*, early access, Jan. 20, 2022, doi: 10.1109/TAC.2022.3144135.

[22] W. Li, C. Song, and Z. Li, "An accelerated recurrent neural network for visual servo control of a robotic flexible endoscope with joint limit constraint," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10787–10797, Dec. 2020.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

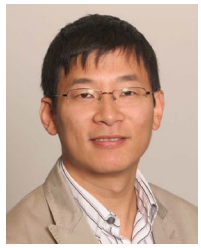
[24] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 3, pp. 1423–1434, Jun. 2017.
[25] Y. Zhang, S. Li, S. Kadry, and B. Liao, "Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4194–4205, Dec. 2019.
[26] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—Online stochastic gradient for tensor decomposition," in *Proc. Conf. Learn. Theory*, 2015, pp. 797–842.
[27] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.
[28] X.-B. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 426–431, Mar. 2003.
[29] K. A. Khudir and A. De Luca, "Faster motion on Cartesian paths exploiting robot redundancy at the acceleration level," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3553–3560, Oct. 2018.
[30] Franka Control Interface Documentation, 2021. [Online]. Available: https://frankaemika.github.io/docs/index.html
[31] 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.
[32] Y. Zhang, S. Chen, S. Li, and Z. Zhang, "Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4909–4920, Jun. 2018.

Mei Liu received the B.E. degree in communication engineering from the Yantai University, Yantai, China, in 2011, and the M.E. degree in pattern recognition and intelligent system from the Sun Yat-sen University, Guangzhou, China, in 2014.

She is currently with Lanzhou University, Lanzhou, China, and the University of Chinese Academy of Sciences, Beijing, China. She joined the College of Physics, Mechanical and Electrical Engineering, Jishou University, Jishou, China, as a Lecturer. From 2016 to 2017, she was with the Department of Computing, The Hong Kong Polytechnic University as a Research Assistant. Her research interests include neural networks, robotics, and optimization.

Jialiang Fan received the B.E. degree in software engineering from Shandong University, Jinan, China, in 2019. He is currently working toward the M.E. degree in computer technology with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China.

His research interests include robotics, neural networks, intelligent information processing, artificial intelligence, and optimization theory.



Yu Zheng (Senior Member, IEEE) received the dual B.E. degrees in mechanical engineering and computer science and the Ph.D. degree in mechatronics from Shanghai Jiao Tong University, Shanghai, China, in 2001 and 2007, respectively. He has also received the M.S. and Ph.D. degrees in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2011 and 2014, respectively.

Between 2007 and 2009, he was a Postdoctoral Research Fellow with the Department of Mechanical Engineering, National University of Singapore, Singapore. Between 2010 and 2014, he worked as a Lab Associate, a Research Associate, and finally a Postdoctoral Researcher with Disney Research Pittsburgh, Pittsburgh, PA, USA. From 2014 to 2018, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn, Dearborn, MI, USA. In September 2018, he joined Tencent Robotics X, Singapore, where he is currently a Principal Research Scientist and the Team Lead of Control Center. His research interests include multicontact/multibody robotic systems, robotic grasping and manipulation, legged locomotion, and various algorithms for robotics.

Dr. Zheng is an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS.

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

His research interests include dynamic neural networks, wireless sensor networks, robotic networks, machine learning, and other dynamic problems defined on a graph.

Dr. Li is on the Editorial Board of the *International Journal of Distributed Sensor Networks*.

Long Jin (Senior Member, IEEE) received the B.E. degree in automation and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2011 and 2016, respectively.

From 2016 to 2017, he received a Postdoctoral training with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. In 2017, he joined the School of Information Science and Engineering, Lanzhou University, Lanzhou, China, as a Professor of computer science and engineering. His research interests include neural networks, robotics, optimization, and intelligent computing.

Dr. Jin is currently serving as an Associate Editor for Neural Processing Letters.