RNN for Receding Horizon Control of Redundant Robot Manipulators

Jingkun Yan, Long Jin, *Senior Member, IEEE*, Zhanting Yuan, and Zhiyi Liu

***Abstract*—Redundant manipulators have been studied and applied in many fields. The trajectory tracking of redundant manipulators is an important topic to explore for applications. This article aims to develop a planning scheme for achieving the trajectory tracking of redundant manipulators, from the receding horizon control (RHC) perspective. For the nonlinear model of manipulators, the linearization operation is conducted to obtain predictive outputs through the forward kinematic equation. Subsequently, an RHC scheme, which minimizes tracking error, velocity norm, and acceleration norm, and directly considers joint limits at three levels as well as the terminal equality constraint, is constructed and further simplified as a convex quadratic programming problem. Furthermore, a recurrent neural network (RNN) model is designed for the constructed RHC scheme, with the help of the technique of converting inequality constraints into equality constraints. The proposed RHC scheme solved by the RNN model is compared with other existing planning schemes and solvers through computer simulations and experiments, without and with the sudden external interference. Simulation and experiment results show that the proposed RHC scheme solved by the RNN model is able to make the redundant manipulator track the given trajectory excellently, and is superior to other existing schemes and solvers in terms of high efficiency, quick-response capacity, and strong robustness.**

***Index Terms*—Joint limits, receding horizon control (RHC), recurrent neural network (RNN), redundant manipulator, trajectory tracking.**

I. Introduction

IN RECENT decades, robots have attracted a large amount of attention and have been investigated in various scientific and engineering areas [1]–[4]. Equipped with the high dexterity and efficiency, redundant manipulators that own more degrees of freedom (DOFs) than required are capable of performing complex and knotty tasks with some extra missions (e.g., avoiding obstacles, optimizing various performance indices) fulfilled, thus being employed in extensive scenarios [5]–[7]. For redundant manipulators, the trajectory tracking of the end-effector is an important topic [8]–[11]. Peng *et al.* [8] present a sensorless admittance control scheme for redundant manipulators to interact with unknown environments in the trajectory tracking task, in which the admittance control method is used to modify the desired trajectory to adapt to the environment. Trajectory tracking of the redundant manipulator is achieved by solving the inverse kinematics equation of manipulators in real time to obtain the joint information as the input of the manipulator so that the end-effector of the manipulator tracks the given trajectory. There are some planning schemes in the form of optimization problems designed for redundant manipulators, such as the minimum-velocity-norm (MVN) scheme [12]–[15] and the minimum-acceleration-norm (MAN) scheme [9], [16], [17], which are separately constructed at velocity level and acceleration level. However, most of the existing research schemes, such as those developed in [12]–[17], only consider MVN or MAN separately, and few investigations simultaneously take both into account. In the existing planning schemes, in addition to the optimization of performance indexes and the solution of inverse kinematics, joint limits are considered to be constraints of optimization problems [14]–[17]. However, joint limits at different levels can only take effect indirectly by formulating them together through leveraging conversion techniques [15]–[17], which needs to introduce extra parameters and may reduce the feasible region of redundant manipulators. Specifically, in

Manuscript received December 8, 2020; revised February 1, 2021; accepted February 16, 2021. Date of publication March 3, 2021; date of current version October 27, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0118900, in part by the research project of Huawei Mindspore Academic Award Fund of Chinese Association of Artificial Intelligence under Grant CAAIXSJLJJ-2020-009A, in part by the Team Project of the Natural Science Foundation of Qinghai Province, China, under Grant 2020-ZJ-903, in part by the Key Laboratory of IoT of Qinghai under Grant 2020-ZJ-Y16, in part by the Natural Science Foundation of Gansu Province, China, under Grant 20JR10RA639, in part by the Natural Science Foundation of Chongqing (China) under Grant cstc2020jcyj-zdxmX0028, in part by the Research and Development Foundation of Nanchong (China) under Grant 20YFZJ0018, in part by CAS "Light of West China" Program, in part by the Project Supported by Chongqing Key Laboratory of Mobile Communications Technology under Grant cqupt-mct-202004, and in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2019-89 and Grant lzuxxxy-2019-tm20. *(Jingkun Yan and Long Jin are co-first authors.) (Corresponding author: Long Jin.)*

Jingkun Yan is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China and also with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China (e-mail: jingkunyan@foxmail.com).

Long Jin is with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China and also with the Department of Computer Science, Lanzhou University, Lanzhou 730000, China (e-mail: jinlongsysu@foxmail.com).

Zhanting Yuan is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China (e-mail: ldwm@lzu.edu.cn).

Zhiyi Liu is with the School of Nuclear Science and Technology, Lanzhou University, Lanzhou 730000, China (e-mail: zhiyil@lzu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3062257.

Digital Object Identifier 10.1109/TIE.2021.3062257

an acceleration-level scheme, when angle-level constraints are introduced into the acceleration level via conversion techniques, a margin is supposed to be designed, which may reduce the feasible range of joint angles.

As an advanced control technique, receding horizon control (RHC), also named model predictive control, has been studied and introduced into a variety of fields [18]–[20]. Actually, RHC is a technology realizing the control of a system in an optimization form. To be specific, different from most optimization control schemes considering the whole time horizon, the optimization operation involved in RHC is repeated at each time instant, based on the receding prediction horizon and the predictive output of the system model, with system constraints considered [21]. Afterward, the solved optimal solution is the input sequence, and the first input is taken as the input of the system at this time instant. It is deserved to note that the characteristic of repeated online optimization enables RHC to tackle complex situations that occur during the control process on time. In [22], the model predictive path-following control of robots is investigated, which considers two situations with and without speed assignment, and is further implemented physically. A robust nonlinear RHC scheme is adopted in [23] for the unmanned aircraft control, with adaptive neural networks leveraged to update the predictive model and estimate parameters. By considering the smoothness in the cost function, a smooth and accurate RHC algorithm for vehicle steering is presented, which is superior to the pure-pursuit controller and the standard RHC without smoothness control [24]. Gulbudak and Gokdag present three finite control set-RHC schemes for nine-switch inverter-based AC drive systems and validate the feasibility of presented schemes through field programmable gate arrays in [25]. Equipped with many advantages, such as tackling the multiple-input and multiple-output systems well, explicitly disposing of system constraints, seeking optimal performance, and considering future states of the system, RHC meets the demands of modern industrial development and will receive more attention in the future.

Taking the real-time nature of RHC into consideration, the online solution of optimization problems is pivotal. In this respect, the recurrent neural network (RNN) performs excellently, and some related work has been carried out [26]–[29]. Based on the zeroing neural network (a kind of RNN), a noise-tolerant neural algorithm is presented for the future dynamic nonlinear optimization problem, with a technique of estimating matrix inversion employed [26]. Qi *et al.* [27] investigate and exploit two discrete-time RNN models with global convergence performance to tackle online time-dependent complex-valued quadratic programming (QP) problems with linear constraints. Considering the harmonic noise control, Li *et al.* [28] devise an improved primal-dual neural network (DNN) to solve the perturbed optimization problem constructed for controlling redundant manipulators. In [29], the research about the projection neural network (a kind of RNN) is reviewed, and its vital role in solving a variety of optimization problems is discussed. These outcomes corroborate the excellent performance of RNN in terms of high efficiency, high accuracy, and global convergence for various optimization problems. Thanks to its high efficiency and high effectiveness, RNN has the potential to become an effective online solving method applied in RHC [21], [30].

Different from the existing planning schemes of redundant manipulators formulated as optimization problems taking the whole time horizon into account, this article achieves the trajectory tracking of redundant manipulators from the perspective of RHC. Given that the nonlinear nature of robot kinematics, a certain linearization operation is needed. Directly considering the joint limits at three levels (i.e., angle, velocity, and acceleration), adding terminal equality constraint to guarantee stability, and minimizing tracking error, velocity norm, and acceleration norm, the RHC scheme for the trajectory tracking of redundant manipulators is constructed. For the constructed RHC scheme that can be formulated as a convex QP problem subject to equality and inequality constraints, an RNN model is devised for its online solution.

The remainder of this article is summarized as follows. Section II constructs the RHC scheme and simplifies it to a convex QP problem. The RNN model is proposed for the constructed RHC scheme, and its convergence analysis is provided in Section III. Section IV exhibits simulation and experiment results based on a 7-DOF redundant manipulator, including some comparison results between existing schemes and the proposed RHC scheme. Finally, Section V concludes this article. The contributions of this article are given as follows.

1) This article constructs an RHC scheme for the trajectory tracking of redundant manipulators at the acceleration level, simplifies it to a convex QP problem with equality and inequality constraints, and further proposes an effective RNN model to obtain its online solution.
2) The joint limits of redundant manipulators at different levels are directly considered in the constructed RHC scheme, without the need for complex conversion techniques and the introduction of redundant parameters, nor does it reduce the feasible region of redundant manipulators.
3) The constructed RHC scheme not only minimizes tracking error and acceleration norm but also minimizes the velocity norm, which supplies a reposeful and reasonable control for the redundant manipulator.
4) By introducing a nonlinear complementary problem (NCP) function, the inequality constraint is converted into an equality constraint, which paves the way for devising RNN model for the constructed RHC scheme.

II. Receding Horizon Control

For a p-DOF redundant manipulator, the position of its end-effector is concerned with its joint space according to the forward kinematic equation [31]–[33]

$$\mathbf{r}(t) = f(\boldsymbol{\theta}(t)) \tag{1}$$

where $\mathbf{r}(t) \in \mathbb{R}^m$ denotes the position of end-effector in Cartesian space, $\boldsymbol{\theta}(t) \in \mathbb{R}^p$ stands for the joint angle vector, $f(\cdot) : \mathbb{R}^p \to \mathbb{R}^m$ symbolizes a projection operation, and $p > m$.

Furthermore, the velocity-level relationship between the end-effector and joints is expressed as

$$\dot{\mathbf{r}}(t) = \dot{f}(\boldsymbol{\theta}(t)) = J(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) \tag{2}$$

of which $J(\boldsymbol{\theta}(t)) = \partial f(\boldsymbol{\theta}(t))/\partial\boldsymbol{\theta}(t) \in \mathbb{R}^{m\times p}$ is the Jacobian matrix of the redundant manipulator.

Before applying RHC to the trajectory tracking of the redundant manipulator, the joint velocity is deemed as the input of controller, i.e.,

$$\mathbf{u}(t) = \dot{\boldsymbol{\theta}}(t). \tag{3}$$

The discrete form is formulated as follows:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \tau\,\mathbf{u}(k) \tag{4}$$

where τ is the sampling interval, and k represents the updating index at time instant $t = k\tau$. Besides, for input $\mathbf{u}(k)$, the following formula holds:

$$\mathbf{u}(k) = \Delta\mathbf{u}(k) + \mathbf{u}(k-1) \tag{5}$$

of which $\Delta\mathbf{u}(k)$ signifies the input increment. Furthermore, one can get

$$\begin{aligned}\mathbf{u}(k+i|k) &= \Delta\mathbf{u}(k+i|k) + \Delta\mathbf{u}(k+i-1|k) + \cdots \\ &\quad + \Delta\mathbf{u}(k|k) + \mathbf{u}(k-1), i = 1, 2, \ldots, N_u\end{aligned} \tag{6}$$

where N_u denotes the control horizon; $\mathbf{u}(k+i|k)$ and $\Delta\mathbf{u}(k+i|k)$ are the future input and future input increment corresponding to sampling instant k, respectively. Taking the time derivative of (3) contributes $\dot{\mathbf{u}}(t) = \ddot{\boldsymbol{\theta}}(t)$, and further applying the backward Euler method contributes

$$\frac{\Delta\mathbf{u}(k)}{\tau} \approx \ddot{\boldsymbol{\theta}}(k).$$

Combine (4) and (5) to obtain the following expression of predicted state $\boldsymbol{\theta}(k+i|k)$ at sampling instant k:

$$\begin{aligned}\boldsymbol{\theta}(k+i|k) &= \boldsymbol{\theta}(k) + i\tau\,\Delta\mathbf{u}(k|k) + (i-1)\tau\,\Delta\mathbf{u}(k+1|k) \\ &\quad + \cdots + \tau\,\Delta\mathbf{u}(k+i-1|k) + i\tau\,\mathbf{u}(k-1) \\ &\quad i = 1, 2, \ldots, N\end{aligned} \tag{7}$$

where N denotes the predictive horizon and satisfies $N \geq N_u$. It is assumed that $\Delta\mathbf{u}(k+i|k) = 0$ for $i \geq N_u$, which means that $\mathbf{u}(k+i|k) = \mathbf{u}(k+N_u-1|k)$ for $i \geq N_u$. The control horizon is usually equal to the predictive horizon. In some cases, selecting a control horizon smaller than the predictive horizon is a way to reduce the computational burden. For predicted output $f(\boldsymbol{\theta}(k+i|k))$ at sampling instant k, the linearization operation is needed

$$\begin{aligned}f(\boldsymbol{\theta}(k+i|k)) &= f(\boldsymbol{\theta}(k)) + J(\boldsymbol{\theta}(k))[\boldsymbol{\theta}(k+i|k) - \boldsymbol{\theta}(k)] \\ &\quad i = 1, 2, \ldots, N.\end{aligned} \tag{8}$$

Actually, the aforementioned linearization result is the first-order Taylor expansion with higher order items neglected.

For convenience, the following definitions are given:

$$\begin{aligned}
\widetilde{\theta} &= [\boldsymbol{\theta}(k+1|k), \boldsymbol{\theta}(k+2|k), \ldots, \boldsymbol{\theta}(k+N|k)]^{\mathrm{T}} \in \mathbb{R}^{N\times p} \\
\widetilde{u} &= [\mathbf{u}(k|k), \mathbf{u}(k+1|k), \ldots, \mathbf{u}(k+N_u-1|k)]^{\mathrm{T}} \in \mathbb{R}^{N_u\times p} \\
\widetilde{\Delta u} &= [\Delta\mathbf{u}(k|k), \Delta\mathbf{u}(k+1|k), \ldots, \Delta\mathbf{u}(k+N_u-1|k)]^{\mathrm{T}} \\
&\quad \in \mathbb{R}^{N_u\times p} \\
\widetilde{r_{\mathrm{d}}} &= [\mathbf{r}_{\mathrm{d}}(k+1), \mathbf{r}_{\mathrm{d}}(k+2), \ldots, \mathbf{r}_{\mathrm{d}}(k+N)]^{\mathrm{T}} \in \mathbb{R}^{N\times m} \\
\widetilde{f(\theta)} &= [f(\boldsymbol{\theta}(k+1|k)), f(\boldsymbol{\theta}(k+2|k)), \ldots, f(\boldsymbol{\theta}(k+N|k))]^{\mathrm{T}} \\
&\quad \in \mathbb{R}^{N\times m} \\
\widetilde{\theta^-} &= [\boldsymbol{\theta}^-, \boldsymbol{\theta}^-, \ldots, \boldsymbol{\theta}^-]^{\mathrm{T}} \in \mathbb{R}^{N\times p} \\
\widetilde{\theta^+} &= [\boldsymbol{\theta}^+, \boldsymbol{\theta}^+, \ldots, \boldsymbol{\theta}^+]^{\mathrm{T}} \in \mathbb{R}^{N\times p} \\
\widetilde{\dot{\theta}^-} &= [\dot{\boldsymbol{\theta}}^-, \dot{\boldsymbol{\theta}}^-, \ldots, \dot{\boldsymbol{\theta}}^-]^{\mathrm{T}} \in \mathbb{R}^{N_u\times p} \\
\widetilde{\dot{\theta}^+} &= [\dot{\boldsymbol{\theta}}^+, \dot{\boldsymbol{\theta}}^+, \ldots, \dot{\boldsymbol{\theta}}^+]^{\mathrm{T}} \in \mathbb{R}^{N_u\times p} \\
\widetilde{\ddot{\theta}^-} &= [\ddot{\boldsymbol{\theta}}^-, \ddot{\boldsymbol{\theta}}^-, \ldots, \ddot{\boldsymbol{\theta}}^-]^{\mathrm{T}} \in \mathbb{R}^{N_u\times p} \\
\widetilde{\ddot{\theta}^+} &= [\ddot{\boldsymbol{\theta}}^+, \ddot{\boldsymbol{\theta}}^+, \ldots, \ddot{\boldsymbol{\theta}}^+]^{\mathrm{T}} \in \mathbb{R}^{N_u\times p}
\end{aligned}$$

where $\mathbf{r}_{\mathrm{d}}(k+i) \in \mathbb{R}^m$ for $i = 1, 2, \ldots, N$ denotes the reference trajectory to be tracked by the end-effector of the redundant manipulator; $\boldsymbol{\theta}^-$ and $\boldsymbol{\theta}^+$ symbolize the lower and upper bounds of joint angles of the redundant manipulator, respectively; $\dot{\boldsymbol{\theta}}^-$ and $\dot{\boldsymbol{\theta}}^+$ signify the lower and upper bounds of joint velocities of the redundant manipulator, respectively; $\ddot{\boldsymbol{\theta}}^-$ and $\ddot{\boldsymbol{\theta}}^+$ denote the lower and upper bounds of joint accelerations of the redundant manipulator, respectively; the superscript $^{\mathrm{T}}$ represents the transposition operation. On this basis, (6)–(8) can be reformulated as the following compact forms:

$$\begin{aligned}&\widetilde{u} = I_u + E\widetilde{\Delta u}, \quad \widetilde{\theta} = I_\theta + A\widetilde{\Delta u} + B \\ &\widetilde{f(\theta)} = I_f + A\widetilde{\Delta u}J^{\mathrm{T}} + BJ^{\mathrm{T}}\end{aligned}$$

where

$$I_\theta = \begin{bmatrix}\boldsymbol{\theta}^{\mathrm{T}}(k) \\ \boldsymbol{\theta}^{\mathrm{T}}(k) \\ \vdots \\ \boldsymbol{\theta}^{\mathrm{T}}(k)\end{bmatrix} \in \mathbb{R}^{N\times p}, \quad I_f = \begin{bmatrix}f^{\mathrm{T}}(\boldsymbol{\theta}(k)) \\ f^{\mathrm{T}}(\boldsymbol{\theta}(k)) \\ \vdots \\ f^{\mathrm{T}}(\boldsymbol{\theta}(k))\end{bmatrix} \in \mathbb{R}^{N\times m}$$

$$I_u = \begin{bmatrix}\mathbf{u}^{\mathrm{T}}(k-1) \\ \mathbf{u}^{\mathrm{T}}(k-1) \\ \vdots \\ \mathbf{u}^{\mathrm{T}}(k-1)\end{bmatrix} \in \mathbb{R}^{N_u\times p}, \quad J = J(\boldsymbol{\theta}(k))$$

$$A = \begin{bmatrix}\tau & 0 & \cdots & 0 \\ 2\tau & \tau & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ N_u\tau & (N_u-1)\tau & \cdots & \tau \\ (N_u+1)\tau & N_u\tau & \cdots & 2\tau \\ \vdots & \vdots & \ddots & \vdots \\ N\tau & (N-1)\tau & \cdots & (N-N_u+1)\tau\end{bmatrix} \in \mathbb{R}^{N\times N_u}$$

$$B=\begin{bmatrix}\tau\,\mathbf{u}^{\mathrm{T}}(k-1)\\ 2\tau\,\mathbf{u}^{\mathrm{T}}(k-1)\\ \vdots\\ N_u\tau\,\mathbf{u}^{\mathrm{T}}(k-1)\\ \vdots\\ N\tau\,\mathbf{u}^{\mathrm{T}}(k-1)\end{bmatrix}\in\mathbb{R}^{N\times p},$$

$$E=\begin{bmatrix}1&0&\cdots&0\\ 1&1&\cdots&0\\ \vdots&\vdots&\ddots&\vdots\\ 1&1&\cdots&1\end{bmatrix}\in\mathbb{R}^{N_u\times N_u}.$$

Let $\boldsymbol{\epsilon}=\mathrm{vec}(\widetilde{f(\theta)})-\mathrm{vec}(\widetilde{r}_{\mathrm{d}})=\mathrm{vec}(I_f)+(J\otimes A)\cdot\mathrm{vec}(\widetilde{\Delta u})+\mathrm{vec}(BJ^{\mathrm{T}})-\mathrm{vec}(\widetilde{r}_{\mathrm{d}})$ signify the tracking error and $\mathrm{vec}(\widetilde{u})=\mathrm{vec}(I_u)+(I\otimes E)\cdot\mathrm{vec}(\widetilde{\Delta u})$, with $\otimes$ denoting the Kronecker product, $\mathrm{vec}(\cdot)$ signifying the vectorization operation, and I being an identity matrix. Therefore, the RHC scheme for the trajectory tracking of redundant manipulators is described as

$$\min_{\widetilde{\Delta u}}\ \|\boldsymbol{\epsilon}\|_Q^2+\|\mathrm{vec}(\widetilde{u})\|_{R_1}^2+\|\mathrm{vec}(\widetilde{\Delta u})\|_{R_2}^2 \tag{9a}$$

$$\text{s.t. } \widetilde{\theta^-}\le\widetilde{\theta}\le\widetilde{\theta^+} \tag{9b}$$

$$\widetilde{\dot{\theta}^-}\le\widetilde{u}\le\widetilde{\dot{\theta}^+} \tag{9c}$$

$$\widetilde{\ddot{\theta}^-}\le\frac{\widetilde{\Delta u}}{\tau}\le\widetilde{\ddot{\theta}^+} \tag{9d}$$

$$f(\boldsymbol{\theta}(k+N|k))=\mathbf{r}_{\mathrm{d}}(k+N) \tag{9e}$$

where positive-definite and symmetrical matrices $Q\in\mathbb{R}^{mN\times mN}$, $R_1\in\mathbb{R}^{pN_u\times pN_u}$, and $R_2\in\mathbb{R}^{pN_u\times pN_u}$ are separately the weight matrices for tracking error, input, and input increment; $\|\cdot\|_Q$, $\|\cdot\|_{R_1}$, and $\|\cdot\|_{R_2}$ symbolize Euclidean norms of vectors, i.e., $\|\boldsymbol{\epsilon}\|_Q=\sqrt{\boldsymbol{\epsilon}^{\mathrm{T}}Q\boldsymbol{\epsilon}}$. It is worth pointing out that (9b)–(9d) severally represent the state constraint, input constraint, and input increment constraint, and (9e) signifies the terminal equality constraint for guaranteeing the stability of the aforementioned RHC scheme [34]. Furthermore, the cost function (9a) can be simplified as

$$\Psi=\frac{1}{2}\mathrm{vec}(\widetilde{\Delta u})^{\mathrm{T}}\cdot W\cdot\mathrm{vec}(\widetilde{\Delta u})+\mathbf{q}^{\mathrm{T}}\cdot\mathrm{vec}(\widetilde{\Delta u})$$

with $W=2[(J\otimes A)^{\mathrm{T}}Q(J\otimes A)+(I\otimes E)^{\mathrm{T}}R_1(I\otimes E)+R_2]$ and $\mathbf{q}=2(J\otimes A)^{\mathrm{T}}Q[\mathrm{vec}(I_f)+\mathrm{vec}(BJ^{\mathrm{T}})-\mathrm{vec}(\widetilde{r}_{\mathrm{d}})]+2(I\otimes E)^{\mathrm{T}}R_1\cdot\mathrm{vec}(I_u)$. Note that constant items independent of $\widetilde{\Delta u}$ in the original cost function (9a) are omitted. Besides, constraints (9b)–(9d) can be combined in the following form:

$$\begin{bmatrix}A\\ -A\\ E\\ -E\\ I\\ -I\end{bmatrix}\widetilde{\Delta u}\le\begin{bmatrix}\widetilde{\theta^+}-I_\theta-B\\ -\widetilde{\theta^-}+I_\theta+B\\ \widetilde{\dot{\theta}^+}-I_u\\ -\widetilde{\dot{\theta}^-}+I_u\\ \tau\,\widetilde{\ddot{\theta}^+}\\ -\tau\,\widetilde{\ddot{\theta}^-}\end{bmatrix}.$$

Let

$$M'=\begin{bmatrix}A\\ -A\\ E\\ -E\\ I\\ -I\end{bmatrix},\ h'=\begin{bmatrix}\widetilde{\theta^+}-I_\theta-B\\ -\widetilde{\theta^-}+I_\theta+B\\ \widetilde{\dot{\theta}^+}-I_u\\ -\widetilde{\dot{\theta}^-}+I_u\\ \tau\,\widetilde{\ddot{\theta}^+}\\ -\tau\,\widetilde{\ddot{\theta}^-}\end{bmatrix}$$

and thus constraints (9b)–(9d) can be together converted into $(I\otimes M')\mathrm{vec}(\widetilde{\Delta u})\le\mathrm{vec}(h')$. Defining $\mathbf{c}=[0,0,\ldots,0,1]\in\mathbb{R}^{1\times N}$ and combing (7) and (8), one can get

$$f^{\mathrm{T}}(\boldsymbol{\theta}(k+N|k))=f^{\mathrm{T}}(\boldsymbol{\theta}(k))+[\mathbf{c}A\widetilde{\Delta u}+N\tau\,\mathbf{u}^{\mathrm{T}}(k-1)]J^{\mathrm{T}}.$$

Therefore, constraint (9e) is transformed into

$$(J\otimes(\mathbf{c}A))\mathrm{vec}(\widetilde{\Delta u})=\mathbf{r}_{\mathrm{d}}(k+N)$$
$$-f(\boldsymbol{\theta}(k))-N\tau\,J\mathbf{u}(k-1).$$

Let $\mathbf{x}=\mathrm{vec}(\widetilde{\Delta u})$, $M=I\otimes M'$, $\mathbf{h}=\mathrm{vec}(h')$, $F=(J\otimes(\mathbf{c}A))$, $\mathbf{g}=\mathbf{r}_{\mathrm{d}}(k+N)-f(\boldsymbol{\theta}(k))-N\tau\,J\mathbf{u}(k-1)$, and the constructed RHC scheme (9) can be expressed as the following QP problem:

$$\min_{\mathbf{x}}\ \frac{1}{2}\mathbf{x}^{\mathrm{T}}W\mathbf{x}+\mathbf{q}^{\mathrm{T}}\mathbf{x} \tag{10a}$$

$$\text{s.t. } F\mathbf{x}=\mathbf{g} \tag{10b}$$

$$M\mathbf{x}\le\mathbf{h} \tag{10c}$$

where $\mathbf{x}\in\mathbb{R}^{pN_u}$, $W\in\mathbb{R}^{pN_u\times pN_u}$, $\mathbf{q}\in\mathbb{R}^{pN_u}$, $F\in\mathbb{R}^{m\times pN_u}$, $\mathbf{g}\in\mathbb{R}^{m}$, $M\in\mathbb{R}^{p(2N+4N_u)\times pN_u}$, and $\mathbf{h}\in\mathbb{R}^{p(2N+4N_u)}$.

Through handling QP problem (10) at sampling instant k, the future input increment $\widetilde{\Delta u}$ can be obtained, and its first element $\Delta\mathbf{u}(k)$ is further applied as the input increment during time interval $[k\tau,(k+1)\tau)$. When the next sampling instant $k+1$ arrives, the aforementioned procedure is repeated until the trajectory tracking task ends.

III. RNN Model and Theoretical Analysis

This section aims to propose an RNN model to solve QP problem (10) with high effectiveness and high accuracy. Additionally, the stability and convergence of the proposed RNN model are guaranteed via theoretical analysis.

Evidently, QP problem (10) is a convex QP problem, and thus its optimal solution exists and is unique, satisfying the following

Karush–Kuhn–Tucker (KKT) conditions:

$$\begin{cases} W\mathbf{x}^* + \mathbf{q} + F^{\mathrm{T}}\boldsymbol{\lambda}^* + M^{\mathrm{T}}\boldsymbol{\mu}^* = \mathbf{0} & \text{(11a)}\\ F\mathbf{x}^* - \mathbf{g} = \mathbf{0} & \text{(11b)}\\ \boldsymbol{\mu}^* \geq 0, \mathbf{h} - M\mathbf{x}^* \geq \mathbf{0} & \text{(11c)}\\ \boldsymbol{\mu}^{*\mathrm{T}}(\mathbf{h} - M\mathbf{x}^*) = 0 & \text{(11d)} \end{cases}$$

of which $\mathbf{x}^*$ is the optimal solution to QP (10), and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and $\boldsymbol{\mu}^* \in \mathbb{R}^{p(2N+4N_u)}$ separately symbolize the Lagrangian multipliers corresponding to constraints (10b) and (10c). For (11c) and (11d), one can replace them with an NCP function satisfying $\varphi(\mathbf{a}, \mathbf{b}) = \mathbf{0} \Leftrightarrow \mathbf{a} \geq \mathbf{0}, \mathbf{b} \geq \mathbf{0}, \mathbf{a} \circ \mathbf{b} = \mathbf{0}$, where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^n$, and symbol $\circ$ represents the Hadamard product. The following NCP function is considered [35]:

$$\varphi_\eta(\mathbf{a}, \mathbf{b}) = \eta\left(\mathbf{a} + \mathbf{b} - \sqrt{\mathbf{a} \circ \mathbf{a} + \mathbf{b} \circ \mathbf{b}}\right) + (1-\eta)\mathbf{a}_+ \circ \mathbf{b}_+$$

where $\eta \in (0,1)$ is an arbitrarily fixed parameter, and $z_+ = \max\{0, z\}$ for $z \in \mathbb{R}$. Subsequently, KKT condition (11) is rewritten with the help of the NCP function

$$\begin{cases} W\mathbf{x}^* + \mathbf{q} + F^{\mathrm{T}}\boldsymbol{\lambda}^* + M^{\mathrm{T}}\boldsymbol{\mu}^* = \mathbf{0}\\ F\mathbf{x}^* - \mathbf{g} = \mathbf{0}\\ \varphi_\eta(\mathbf{h} - M\mathbf{x}^*, \boldsymbol{\mu}^*) = \mathbf{0} \end{cases}.$$

On the strength of the aforementioned analyses, solving QP problem (10) is equivalent to solving the following equation:

$$\begin{bmatrix} W & F^{\mathrm{T}} & M^{\mathrm{T}}\\ F & 0 & 0\\ -\eta M & 0 & \eta I \end{bmatrix} \begin{bmatrix} \mathbf{x}\\ \boldsymbol{\lambda}\\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \mathbf{q}\\ -\mathbf{g}\\ \eta\,\mathbf{h} + \mathbf{l} \end{bmatrix} = \mathbf{0} \tag{12}$$

with

$$\mathbf{l} = (1-\eta)(\mathbf{h} - M\mathbf{x})_+ \circ \boldsymbol{\mu}_+ \\ - \eta\sqrt{(\mathbf{h}-M\mathbf{x}) \circ (\mathbf{h}-M\mathbf{x}) + \boldsymbol{\mu} \circ \boldsymbol{\mu}}.$$

Let

$$P = \begin{bmatrix} W & F^{\mathrm{T}} & M^{\mathrm{T}}\\ F & 0 & 0\\ -\eta M & 0 & \eta I \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{x}\\ \boldsymbol{\lambda}\\ \boldsymbol{\mu} \end{bmatrix}, \mathbf{v} = \begin{bmatrix} \mathbf{q}\\ -\mathbf{g}\\ \eta\,\mathbf{h} + \mathbf{l} \end{bmatrix}$$

and (12) can be reformulated as

$$P\mathbf{y} + \mathbf{v} = \mathbf{0}. \tag{13}$$

As a result, solving QP problem (10) is transformed to solve (13).

The next aim is to propose an RNN model to solve (13) with stability and convergence guaranteed. First of all, an error function monitoring the solving process is defined as $\mathbf{e} = P\mathbf{y} + \mathbf{v}$. When $\mathbf{e}$ approaches zero, the obtained solution $\mathbf{y}$ approaches the theoretical solution $\mathbf{y}^*$. With this aim, the design formula $\dot{\mathbf{e}} = -\gamma\,\Upsilon(\mathbf{e})$ is used to facilitate error zeroing [36], of which $\gamma > 0 \in \mathbb{R}$ is a scaling index controlling the convergence rate of $\mathbf{e}$ and $\Upsilon(\cdot)$ signifies an activation function expediting the convergence of $\mathbf{e}$, with odd and monotonically increasing properties. Furthermore, an RNN model solving (13) is generated as follows:

$$S\dot{\mathbf{y}} = -\gamma\,\Upsilon(P\mathbf{y} + \mathbf{v}) \tag{14}$$

where $\Upsilon_i(e_i) = \exp(\kappa e_i) - \exp(-\kappa e_i)$

$$S = \\ \begin{bmatrix} W & F^{\mathrm{T}} & M^{\mathrm{T}}\\ F & 0 & 0\\ -\eta M + \eta D_1 M - (1-\eta) D_3\,\tilde{}\,M & 0 & \eta I - \eta D_2 + (1-\eta) D_4 \end{bmatrix}$$

and

$$\kappa = 3, D_1 = \Lambda((\mathbf{h} - M\mathbf{x}) \oslash \mathbf{w}), D_2 = \Lambda(\boldsymbol{\mu} \oslash \mathbf{w})$$

$$D_3 = \Lambda(\partial(\mathbf{h} - M\mathbf{x})_+ \circ \boldsymbol{\mu}_+), D_4 = \Lambda((\mathbf{h} - M\mathbf{x})_+ \circ \partial\boldsymbol{\mu}_+)$$

$$\mathbf{w} = \sqrt{(\mathbf{h} - M\mathbf{x}) \circ (\mathbf{h} - M\mathbf{x}) + \boldsymbol{\mu} \circ \boldsymbol{\mu}}$$

$$\partial z_+ = \begin{cases} 1, & z > 0\\ [0,1], & z = 0\\ 0, & z < 0 \end{cases}$$

with e_i standing for the ith element of vector $\mathbf{e}$, and $\Lambda(\cdot)$ and $\oslash$ denoting the diagonal matrix and the Hadamard division, respectively.

Ulteriorly, the stability and convergence of RNN model (14) served for (13) are guaranteed by the following theorem.

Theorem 1: The actual solution $\mathbf{y}$ calculated by RNN model (14) converges to the theoretical solution $\mathbf{y}^*$ to (13).

Proof: Aiming at the design formula, one can design a Lyapunov function as $\chi = \mathbf{e}^{\mathrm{T}}\mathbf{e}/2$ that is evidently positive definite. Taking its time derivative generates

$$\dot{\chi} = \mathbf{e}^{\mathrm{T}}\dot{\mathbf{e}} = -\gamma\,\mathbf{e}^{\mathrm{T}}\Upsilon(\mathbf{e}).$$

In view of the odd and monotonically increasing properties of the activation function $\Upsilon(\cdot)$, one can infer that $\dot{\chi} \leq 0$. The Lyapunov stability theory helps us deduce that $\mathbf{e}$ converges to zero with time evolving, which means that $\mathbf{e}^* = 0$ is an equilibrium point of the system. What is noteworthy is that the equilibrium point $\mathbf{e}^*$ corresponds to the theoretical solution $\mathbf{y}^*$ to (13). As a consequence, the conclusion is drawn that the actual solution $\mathbf{y}$ calculated by RNN model (14) converges to the theoretical solution $\mathbf{y}^*$ to (13). ■

IV. Simulation and Experiment Results Based on Manipulator

In this section, the proposed RHC scheme (9) solved by RNN model (14) is applied to the trajectory tracking problem of a redundant manipulator and simulations are conducted on the MindSpore framework, for demonstrating the efficiency and strong robustness of the proposed scheme. For comparison, some existing techniques for handing trajectory tracking problem of redundant manipulators are first introduced.

The MAN scheme considering joint limits and error feedbacks without considering the elimination of joint-angle drift phenomenon is as below [16]

$$\min_{\ddot{\boldsymbol{\theta}}} \frac{1}{2}\ddot{\boldsymbol{\theta}}^{\mathrm{T}}\ddot{\boldsymbol{\theta}} \tag{15a}$$

$$\text{s.t. } J(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} = \ddot{\mathbf{r}}_{\mathrm{d}} - \dot{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \alpha_{\mathrm{v}}\boldsymbol{\epsilon}_{\mathrm{v}} + \alpha_{\mathrm{p}}\boldsymbol{\epsilon}_{\mathrm{p}} \tag{15b}$$

TABLE I
D–H Parameters of Franka Emika Panda

Joint	a (m)	d (m)	α (rad)	θ (rad)
Joint 1	0	0.3330	0	θ_1
Joint 2	0	0	$-\pi/2$	θ_2
Joint 3	0	0.3160	$\pi/2$	θ_3
Joint 4	0.0825	0	$\pi/2$	θ_4
Joint 5	-0.0825	0.3840	$-\pi/2$	θ_5
Joint 6	0	0	$\pi/2$	θ_6
Joint 7	0.0880	0	$\pi/2$	θ_7
Flange	0	0.1070	0	0

$$\boldsymbol{\theta}^{-} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^{+} \tag{15c}$$

$$\dot{\boldsymbol{\theta}}^{-} \leq \dot{\boldsymbol{\theta}} \leq \dot{\boldsymbol{\theta}}^{+} \tag{15d}$$

$$\ddot{\boldsymbol{\theta}}^{-} \leq \ddot{\boldsymbol{\theta}} \leq \ddot{\boldsymbol{\theta}}^{+} \tag{15e}$$

of which $\boldsymbol{\epsilon}_{\mathrm{v}} = \dot{\mathbf{r}}_{\mathrm{d}} - J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\epsilon}_{\mathrm{p}} = \mathbf{r}_{\mathrm{d}} - f(\boldsymbol{\theta})$ separately stand for velocity error and position error of the end-effector of the redundant manipulator with $\mathbf{r}_{\mathrm{d}}$ being the given trajectory, and positive α_{v} and α_{p} denote the corresponding feedback gains. The aforementioned scheme is solved by a piecewise-linear projection equation based neural network (PLPENN) model presented in [16]. Besides, the MVN scheme is also considered as a comparison scheme

$$\min_{\dot{\boldsymbol{\theta}}} \ \frac{1}{2}\dot{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\boldsymbol{\theta}} \tag{16a}$$

$$\text{s.t. } J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\mathbf{r}}_{\mathrm{d}} \tag{16b}$$

$$\dot{\boldsymbol{\theta}}^{-} \leq \dot{\boldsymbol{\theta}} \leq \dot{\boldsymbol{\theta}}^{+} \tag{16c}$$

which does not consider the error feedback and is solved by a discrete perturbation-immunity neural network (DPINN) model [14]. It is worthwhile pointing out that joint limits at different levels of the aforementioned two schemes are not dealt with directly, but indirectly by combining them into the acceleration or velocity levels. For the proposed RHC scheme (9), a simplified DNN model presented in [21] can be utilized to handle it for comparison.

A. Without Sudden External Interference

In the first place, the high efficiency and effectiveness of the proposed RHC scheme (9) solved by RNN model (14) are substantiated based on a 7-DOF redundant manipulator named Franka Emika Panda with $p = 7$ and $m = 3$ under the environment without sudden external interference. The Denavit–Hartenberg (D–H) parameters of Franka Emika Panda are exhibited in Table I. The initial state of the manipulator is set as $\boldsymbol{\theta}(0) = [0, -0.785\,472, 0, -2.35\,425, 0, 1.57\,164, 0.785\,465]^{\mathrm{T}}$ rad, and the joint limits are set as $\boldsymbol{\theta}^{+} = [2.8973, 1.7628, 2.8973, -0.0698, 2.8973, 3.7525, 2.8973]^{\mathrm{T}}$ rad, $\boldsymbol{\theta}^{-} = [-2.8973, -1.7628, -2.8973, -3.0718, -2.8973, -0.0175, -2.8973]^{\mathrm{T}}$ rad, $-\dot{\boldsymbol{\theta}}^{-} = \dot{\boldsymbol{\theta}}^{+} = [2.1750, 2.1750, 2.1750, 2.1750, 2.6100, 2.6100, 2.6100]^{\mathrm{T}}$ rad/s, and $-\ddot{\boldsymbol{\theta}}^{-} = \ddot{\boldsymbol{\theta}}^{+} = [15, 7.5, 10, 12.5, 15, 20, 20]^{\mathrm{T}}$ rad/s^2. The sampling interval τ is set as 0.01 s, and the task duration T is 10 s. Without loss of generality, other parameters are decided through empirical methods: $\eta = 0.95$, $\gamma = 100$, $N = 5$, $N_u = 5$, $Q = 100I$, $R_1 = 0.1I$, and $R_2 = 0.1I$. The manipulator is driven to track a flower-shaped trajectory, and the initial position of the given trajectory to be tracked is the same as that of the end-effector of the manipulator.

The related simulation results synthesized by the proposed RHC scheme (9) solved by RNN model (14) are displayed in Fig. 1. The given trajectory to be tracked and the actual end-effector trajectory performed by the Panda manipulator are depicted in Fig. 1(a), and they coincide evidently. Fig. 1(b) exhibits the trajectory tracking process of the Panda manipulator with reasonable configurations in the three-dimensional space. Fig. 1(c) shows the tracking errors in three dimensions that stabilize at the order 10^{-4} m, which testifies the high effectiveness and high accuracy of the proposed RHC scheme (9) solved by RNN model (14) in the trajectory tracking of the redundant manipulator. Fig. 1(d) through Fig. 1(f) display the joint angle, joint velocity, and joint acceleration of the Panda manipulator during the task of tracking the flower-shaped trajectory, with the related joint limits being also depicted. One can observe that none of these joint parameters exceed the joint limits. To sum up, the given trajectory tracking task is excellently performed by the Panda manipulator driven by RHC scheme (9) solved by RNN model (14), demonstrating their feasibility, effectiveness, and accuracy.

Furthermore, different parameters have different effects on the results. First, Fig. 2 portrays simulation results with $N = 5$, $N_u = 5$, $Q = 100I$, $R_1 = 0$, and $R_2 = 0.1I$. When $R_1 = 0$, the cost function (9a) does not consider minimizing the input norm, i.e., joint velocity norm. As can be seen from Fig. 2, joint angles vary more steeply than those drawn in Fig. 1. There exists hidden trouble that when the joint angle approaches the joint limit and remains at the joint limit, the manipulator may be caught in singularity. Letting the joint angle change gently, that is, minimizing the joint velocity norm, is for acquiring the reposeful control effect, thus being of significance. Second, the influence of the weight matrix Q, predictive horizon N, and control horizon N_u on the tracking error is worth discussing. Fig. 3(a) portrays the tracking error when $N = 5$, $N_u = 5$, $Q = 20000I$, $R_1 = 0.1I$, $R_2 = 0.1I$, and Fig. 3(b) portrays the tracking error when $N = 2$, $N_u = 2$, $Q = 100I$, $R_1 = 0.1I$, and $R_2 = 0.1I$. As shown in Fig. 3, increasing weight matrix Q or decreasing predictive horizon N and control horizon N_u can help improve the tracking accuracy. It is important to note that there is a tradeoff between the high-precision tracking performance and the normal operation of the system.

In addition, a trajectory tracking experiment base on Panda manipulator is carried out to validate the feasibility of RHC scheme (9) solved by RNN model (14). First of all, the experiment platform is displayed in Fig. 4, including a laptop, Panda manipulator, and some necessary components. Second, corresponding parameters are set as $\tau = 0.001$ s, $N = 18$, and $N_u = 12$. Figs. 5 and 6 exhibit snapshots of experiment results

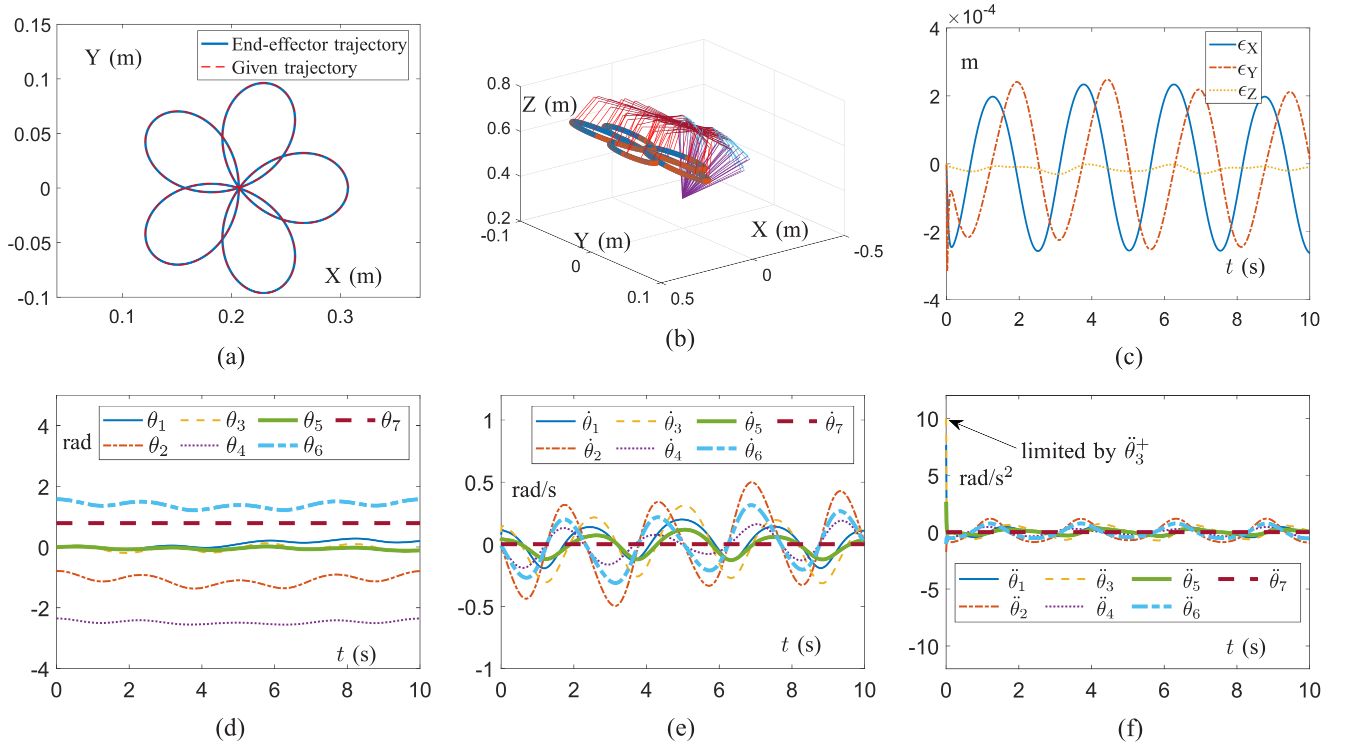


Fig. 1. Simulation results on the Panda manipulator executing the trajectory tracking task, synthesized by RHC scheme (9) solved by RNN model (14). (a) Top view. (b) Tracking process. (c) Tracking error. (d) Joint angle. (e) Joint velocity. (f) Joint acceleration.

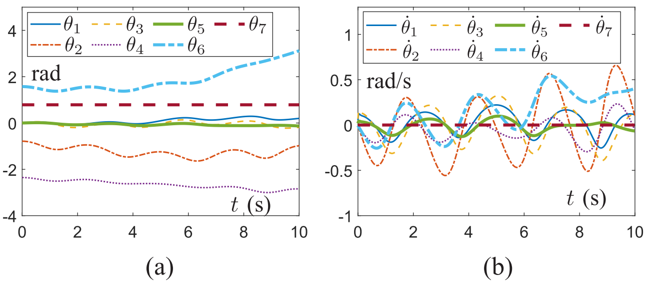


Fig. 2. Simulation results on the Panda manipulator executing the trajectory tracking task, synthesized by RHC scheme (9) solved by RNN model (14), with $N = 5$, $N_u = 5$, $Q = 100I$, $R_1 = 0$, and $R_2 = 0.1I$. (a) Joint angle. (b) Joint velocity.

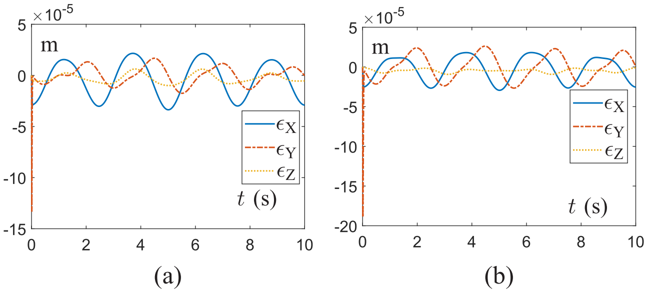


Fig. 3. Tracking errors of the Panda manipulator executing the trajectory tracking task, synthesized by RHC scheme (9) solved by RNN model (14). (a) Error with $N = 5$, $N_u = 5$, $Q = 20000I$, $R_1 = 0.1I$, and $R_2 = 0.1I$. (b) Error with $N = 2$, $N_u = 2$, $Q = 100I$, $R_1 = 0.1I$, and $R_2 = 0.1I$.

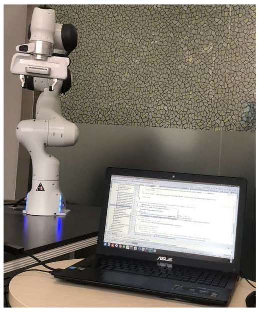


Fig. 4. Experiment platform.

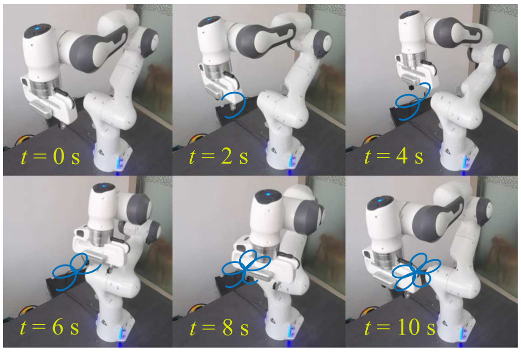


Fig. 5. Snapshots on experiment results, in which Panda manipulator tracks the flower-shaped trajectory with the help of RHC scheme (9) solved by RNN model (14).

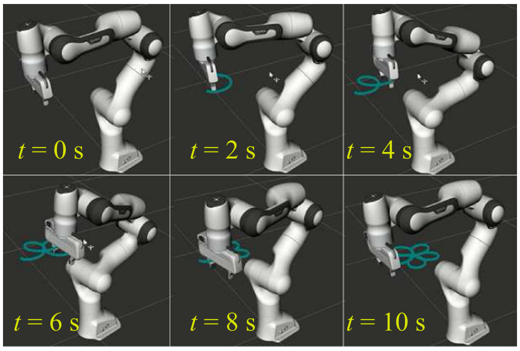
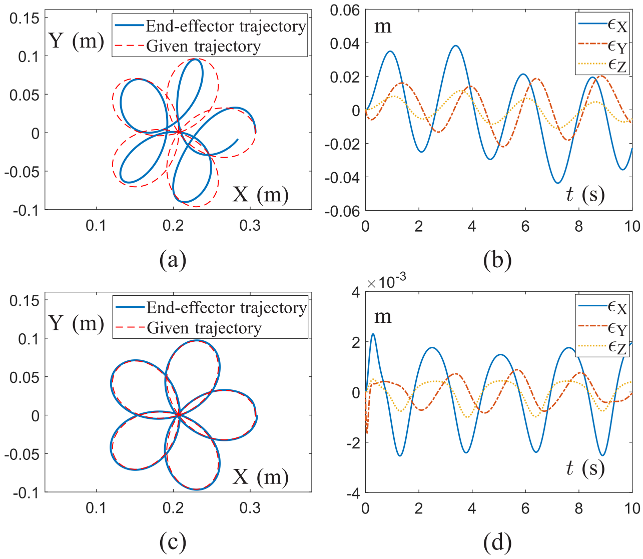


Fig. 6. Snapshots on screen recording of the RVIZ corresponding to the experiment process in Fig. 5, in which Panda manipulator tracks the flower-shaped trajectory with the help of RHC scheme (9) solved by RNN model (14).



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

Fig. 7. Simulation results on the Panda manipulator executing the trajectory tracking task, synthesized by RHC scheme (9) solved by DNN model [21]. (a) Top view, $\gamma = 100$. (b) Tracking error, $\gamma = 100$. (c) Top view, $\gamma = 5 \times 10^4$. (d) Tracking error, $\gamma = 5 \times 10^4$.

and screen recording of the robot visualization tool (RVIZ) corresponding to the experiment process, respectively, in which the end-effector of Panda manipulator moves along a given flower-shaped trajectory. Note that blue flower-shaped trajectories in Fig. 5 are auxiliary lines to help observe the motion of the end-effector of Panda manipulator. By observing snapshots in Figs. 5 and 6, one can know that the trajectory tracking task is commendably fulfilled.

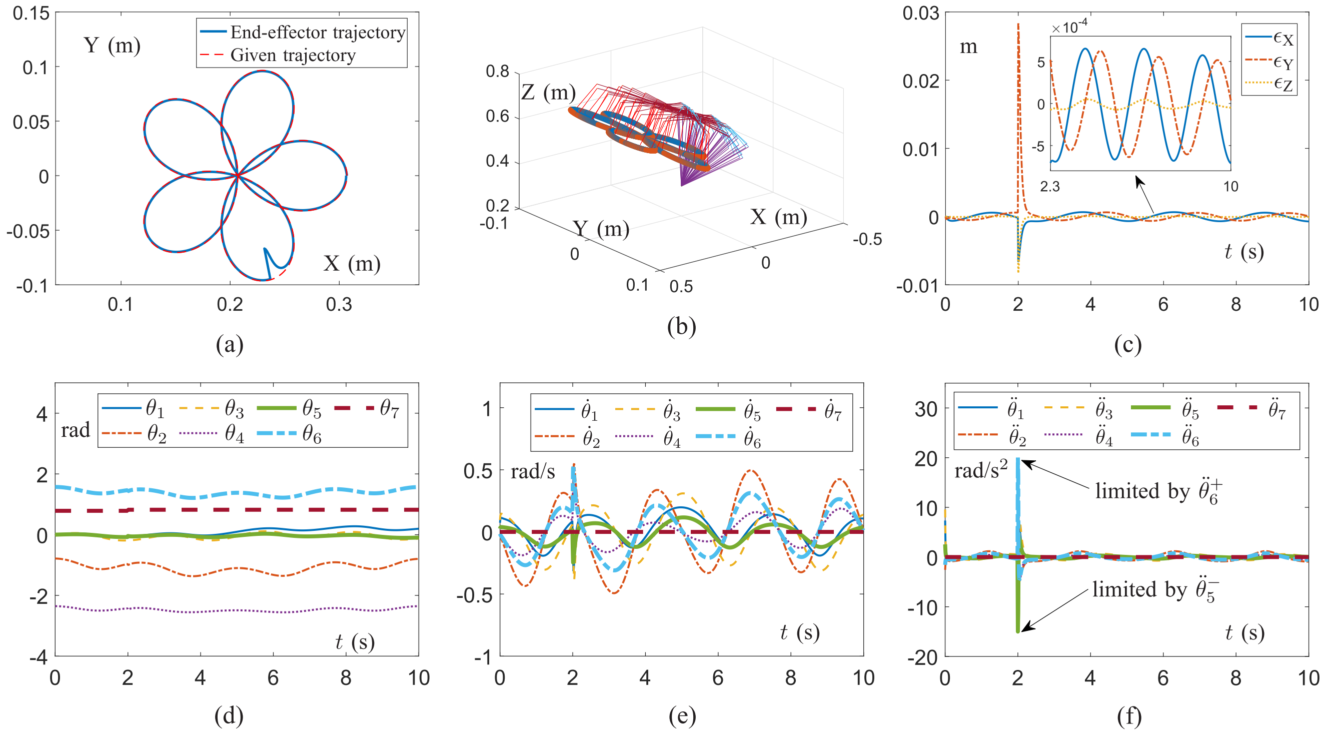
Comparison results synthesized by RHC scheme (9) solved by DNN model [21] are shown in Fig. 7. Note that the scaling index controlling the convergence rate of DNN model [21] is represented by γ. When $\gamma = 100$, the same as that of RNN model (14) in Fig. 1, DNN model [21] cannot solve RHC scheme (9) effectively, which leads to the disordered tracking results displayed in Fig. 7(a) and (b). When $\gamma = 5 \times 10^4$, the end-effector trajectory depicted in Fig. 7(c) barely fits the given trajectory with obvious tracking error. In comparison with RNN model (14) equipped with $\gamma = 100$, DNN model [21] equipped with $\gamma = 5 \times 10^4$ still generates the larger tracking error, which implies the effectiveness and superiority of the proposed RNN model (14) in solving RHC scheme (9).

B. With Sudden External Interference

In this section, the sudden external interference is considered for testing the quick-response capacity and robustness of RHC scheme (9) solved by RNN model (14). To be specific, when $t = 2$ s, the joint angle $\boldsymbol{\theta}$ of the Panda manipulator is increased by $[0.035, -0.035, 0.035, -0.035, 0.035, -0.035, 0.035]^{\mathrm{T}}$ rad. Meanwhile, comparative simulations are carried out using the comparison schemes mentioned previously and simulation results are depicted in Fig. 8 through Fig. 10.

Fig. 8 portrays the related simulation results synthesized by the proposed RHC scheme (9) solved by RNN model (14), under sudden external interference, with $N = 8$ and $N_u = 8$. From Fig. 8(a) and (b), one can see that the end-effector of the Panda manipulator evidently deviates from the given trajectory when the external interference is suddenly injected in the joint angle, although the value of interference is very small. However, thanks to the quick-response capacity and inherent robustness of RHC scheme (9), the end-effector of the Panda manipulator quickly returns to the given trajectory. It is revealed that RHC scheme (9) is capable of tackling the abrupt situations that occur during the control process in a timely manner. Additionally, the tracking errors observed in Fig. 8(c) stabilize again at the order of 10^{-4} m after 0.3 s, indicating the high efficiency and quick-response capability of RHC scheme (9) solved by RNN model (14). Similar to Fig. 1(d) through Fig. 1(f), joint parameters of the Panda manipulator exhibited in Fig. 8(d) through Fig. 8(f) are still restrained in joint limits ensuring the safe running of the manipulator, even if impacted by the sudden external interference.

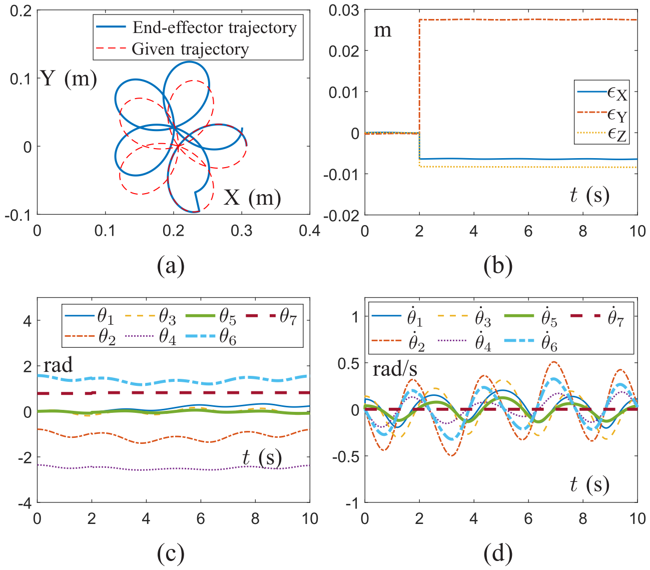
In addition, as shown in Fig. 9, MVN scheme (16) solved by DPINN model [14] performs poorly when the Panda manipulator is subjected to sudden external interference during the execution of the trajectory tracking task. From Fig. 9(a) and (b), it can be observed that MVN scheme (16) does not respond to the considerable tracking errors caused by interference, further causing the subsequent end-effector trajectory to deviate from the given trajectory entirely. This is because error feedback is not considered in MVN scheme (16). Therefore, MVN scheme (16) without error feedback is incapable of coping with sudden external interference. As for the comparative MAN scheme (15) solved by the PLPENN model [16], related simulation results are drawn in Fig. 10. One can see from Fig. 10(a) that, when subjected to sudden external interference, the Panda manipulator driven by MAN scheme (15) returns to the given trajectory slower than that driven by RHC scheme (9). Besides, although MAN scheme (15) takes more response time than RHC scheme (9), its accuracy is worse than that of the latter. It follows that the proposed RHC scheme (9) solved by RNN model (14) has stronger robustness, higher efficiency, and quicker response



(a) (b) (c)

(d) (e) (f)

Fig. 8. Simulation results on the Panda manipulator executing the trajectory tracking task with sudden external interference, synthesized by RHC scheme (9) solved by RNN model (14). (a) Top view. (b) Tracking process. (c) Tracking error. (d) Joint angle. (e) Joint velocity. (f) Joint acceleration.



(a) (b)

(c) (d)

Fig. 9. Simulation results on the Panda manipulator executing the trajectory tracking task with sudden external interference, synthesized by MVN scheme (16) solved by DPINN model [14]. (a) Top view. (b) Tracking error. (c) Joint angle. (d) Joint velocity.

capability to cope with the sudden external interference than MAN scheme (15) solved by PLPENN model [16]. Furthermore, owing to a lack of minimizing the velocity norm in the performance index of MAN scheme (15), joint velocities in Fig. 10(e) are intuitively larger than those in Fig. 8(e), and even touch the velocity limit.

In a nutshell, the simulation results and experiment results displayed in Fig. 1 through Fig. 10 corroborate the high efficiency, quick-response capacity, and strong robustness of RHC (9) solved by RNN model (14) and further imply that it is superior to the comparison schemes and solvers.

Additionally, more comparison schemes are considered, and detailed comparison results are exhibited in Table II. As represented in Table II, in terms of joint limits, only the proposed scheme (9) can directly deal with joint limits at multiple levels. Besides, scheme (9) minimizes both the velocity norm and the acceleration norm, whereas other schemes only minimize one of them. Furthermore, owing to a lack of error feedback, schemes in [14], [17], and [37] have a requirement for the initial position of the end-effector of the manipulator and are unable to help the end-effector return to the given trajectory when it deviates from the given trajectory. In terms of quantitative analysis, maximal steady-state residual error (MSRE) generated by each scheme in the same trajectory tracking task is recorded in the last column of Table II. Note that for each simulation, parameters related to the manipulator (e.g., initial state and joint limits) and the scaling index controlling the convergence rate are the same, and other parameters depend on the corresponding reference of each scheme. As recorded in Table II, although the scheme in [13] achieves the highest accuracy, it does not consider joint limits. Apart from that, the proposed scheme (9) and scheme (16) in [14], with MSRE at the order of 10^{-4} m, outperform other schemes.

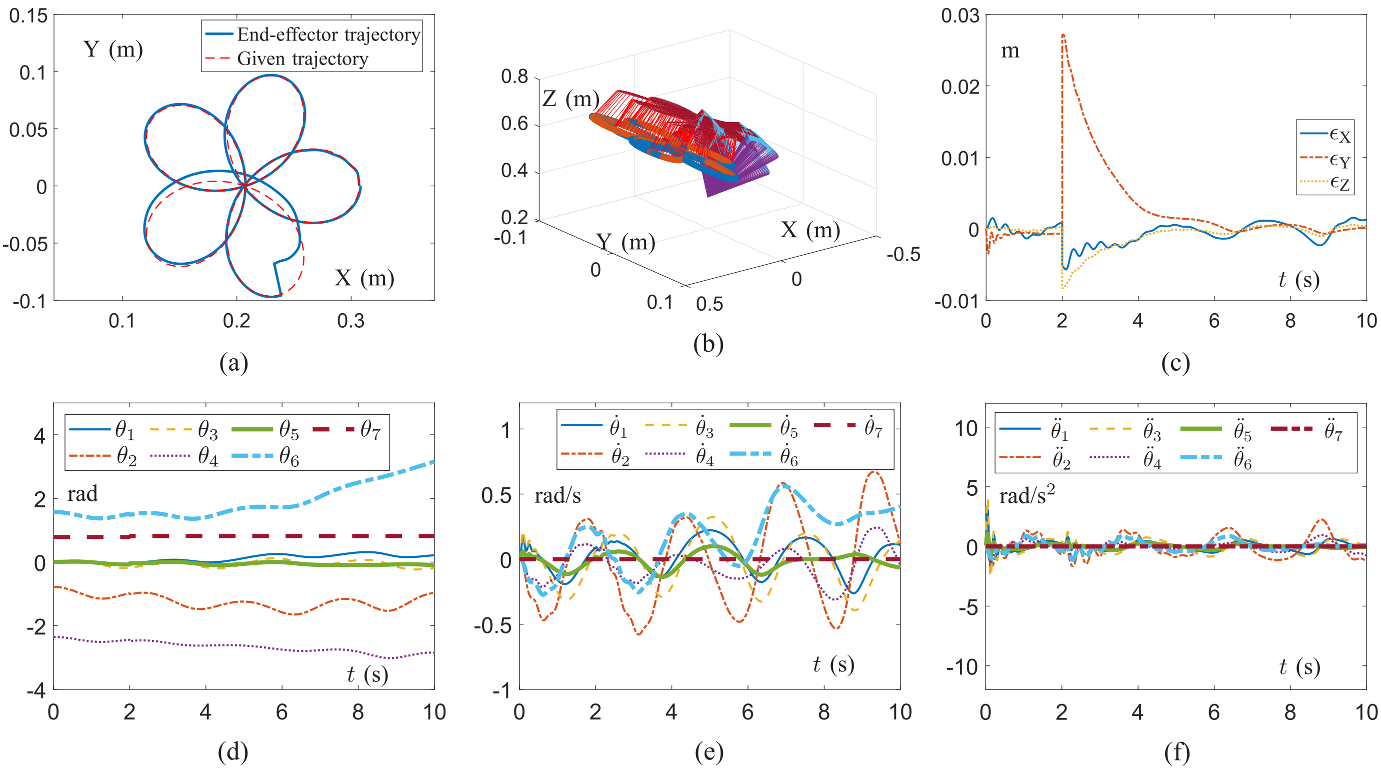


Fig. 10. Simulation results on the Panda manipulator executing the trajectory tracking task with sudden external interference, synthesized by MAN scheme (15) solved by PLPENN model [16]. (a) Top view. (b) Tracking process. (c) Tracking error. (d) Joint angle. (e) Joint velocity. (f) Joint acceleration.

TABLE II
COMPARISONS AMONG DIFFERENT PLANNING SCHEMES FOR TRAJECTORY TRACKING OF REDUNDANT MANIPULATORS

	Joint limits level♯	Joint limits manner‡	Minimizing velocity norm or acceleration norm	Initial position	Returning to the given trajectory§	MSRE◇
The proposed scheme (9)	Multiple	Directly	Both	Any	Yes	2.789×10^{-4} m
Scheme (15) in [16]	Multiple	Indirectly	Acceleration	Any	Yes	3.315×10^{-3} m
Scheme (16) in [14]	Two	Indirectly	Velocity	Limited†	No	3.142×10^{-4} m
Scheme in [12]	Single	Directly	Velocity	Any	Yes	1.234×10^{-3} m
Scheme in [13]	None	NA⋆	Velocity	Any	Yes	3.028×10^{-7} m
Scheme in [17]	Multiple	Indirectly	Acceleration	Limited†	No	1.048×10^{-1} m
Scheme in [37]	Two	Indirectly	Velocity	Limited†	No	6.907×10^{-2} m
Scheme in [38]	Multiple	Indirectly	Acceleration	Any	Yes	7.199×10^{-2} m
Scheme in [39]	Multiple	Indirectly	Velocity	Any	Yes	2.174×10^{-2} m

♯ "Joint limits level" refers to the joint limits at single or multiple levels in the scheme.
‡ "Joint limits manner" refers to how the scheme deals with joint limits at different levels.
⋆ "NA" means that the scheme does not consider joint limits.
† "Limited" means that the scheme requires the initial position of the end-effector of the manipulator to be on the given trajectory.
§ "Returning to the given trajectory" refers to whether the scheme helps the end-effector of manipulator return to the given trajectory when it deviates from the given trajectory.
◇ "MSRE" refers to the maximal steady-state residual error generated by the scheme in the same trajectory tracking task.

V. Conclusion

This article constructed an RHC scheme for the trajectory tracking of redundant manipulators, directly considering joint limits at three levels and minimizing the velocity norm and the acceleration norm simultaneously. For the online solution of the constructed RHC scheme, an RNN model was devised on the strength of the technique of converting inequality constraints into equality constraints. The RHC scheme solved by the RNN model was employed in the control of the manipulator through computer simulations and experiments. Overall, the high efficiency, quick-response capability, and strong robustness of the proposed RHC scheme solved by the RNN model were verified by simulation and experiment results, and its superiority to other existing schemes was also demonstrated. For future directions of this study, the authors would further improve the RHC scheme and consider some additional and practical functions of redundant manipulators, such as obstacle avoidance.

References

[1] C. Wang, H. Tnunay, Z. Zuo, B. Lennox, and Z. Ding, "Fixed-time formation control of multi-robot systems: Design and experiments," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6292–6301, Aug. 2019.

[2] L. Jin, J. Yan, X. Du, X. Xiao, and D. Fu, "RNN for solving time-variant generalized Sylvester equation with applications to robots and acoustic source localization," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6359–6369, Oct. 2020.

[3] L. Jin *et al.*, "Perturbed manipulability optimization in a distributed network of redundant robots," *IEEE Trans. Ind. Electron.*, to be published, doi: 10.1109/TIE.2020.3007099.

[4] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8112–8123, Oct. 2018.

[5] 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. Mechatronics*, vol. 26, no. 1, pp. 90–101, Feb. 2021.

[6] Z. Xie, L. Jin, X. Luo, Z. Sun, and M. Liu, "RNN for repetitive motion generation of redundant robot manipulators: An orthogonal projection-based scheme," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2020.3028304.

[7] 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.

[8] G. Peng, C. Yang, W. He, and C. L. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3138–3148, Apr. 2020.

[9] L. Jin, Y. Liufu, H. Lu, and Z. Zhang, "Saturation-allowed neural dynamics applied to perturbed time-dependent system of linear equations and robots," *IEEE Trans. Ind. Electron.*, to be published, doi: 10.1109/TIE.2020.3029478.

[10] C. Yang, G. Peng, L. Cheng, J. Na, and Z. Li, "Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2019.2920870.

[11] C. Yang, D. Huang, W. He, and L. Cheng, "Neural control of robot manipulators with trajectory tracking constraints and input saturation," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2020.3017202.

[12] 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.

[13] H. Lu, L. Jin, J. Zhang, Z. Sun, S. Li, and Z. Zhang, "New joint-drift-free scheme aided with projected ZNN for motion generation of redundant robot manipulators perturbed by disturbances," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2019.2956961.

[14] J. Yang, L. Chen, Y. Qi, M. Liu, and C. Cui, "Discrete perturbation-immunity neural network for dynamic constrained redundant robot control," *IEEE Access*, vol. 8, pp. 84490–84500, 2020.

[15] W. Li, "Predefined-time convergent neural solution to cyclical motion planning of redundant robots under physical constraints," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10732–10743, Dec. 2020.

[16] L. Jin and Y. N. Zhang, "G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 153–164, Feb. 2015.

[17] L. Xiao and Y. N. Zhang, "Dynamic design, numerical solution and effective verification of acceleration-level obstacle-avoidance scheme for robot manipulators," *Int. J. Syst. Sci.*, vol. 47, no. 4, pp. 932–945, Dec. 2016.

[18] N. Van Duijkeren, T. Faulwasser, and G. Pipeleers, "Dual-objective NMPC: Considering economic costs near manifolds," *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3788–3795, Sep. 2019.

[19] A. A. Ahmed, B. K. Koh, and Y. I. Lee, "A comparison of finite control set and continuous control set model predictive control schemes for speed control of induction motors," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1334–1346, Apr. 2018.

[20] K. Miyazaki, K. Kobayashi, S. Azuma, N. Yamaguchi, and Y. Yamashita, "Design and value evaluation of demand response based on model predictive control," *IEEE Trans. Ind. Informat.*, vol. 15, no. 8, pp. 4809–4818, Aug. 2019.

[21] Y. Pan and J. Wang, "Nonlinear model predictive control using a recurrent neural network," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 2296–2301.

[22] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1505–1511, Jul. 2017.

[23] G. A. Garcia, S. S. Keshmiri, and T. Stastny, "Robust and adaptive nonlinear model predictive controller for unsteady and highly nonlinear unmanned aircraft," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1620–1627, Jul. 2015.

[24] P. F. Lima, M. Nilsson, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Spatial model predictive control for smooth and accurate steering of an autonomous truck," *IEEE Trans. Intell. Veh.*, vol. 2, no. 4, pp. 238–250, Dec. 2017.

[25] O. Gulbudak and M. Gokdag, "Finite control set model predictive control approach of nine switch inverter-based drive systems: Design, analysis, and validation," *ISA Trans.*, to be published, doi: 10.1016/j.isatra.2020.10.037.

[26] L. Wei, L. Jin, C. Yang, K. Chen, and W. Li, "New noise-tolerant neural algorithms for future dynamic nonlinear optimization with estimation on Hessian matrix inversion," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2019.2916892.

[27] Y. Qi, L. Jin, Y. Wang, L. Xiao, and J. Zhang, "Complex-valued discrete-time neural dynamics for perturbed time-dependent complex quadratic programming with applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3555–3569, Sep. 2020.

[28] S. Li, M. Zhou, and 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.*, vol. 29, no. 10, pp. 4791–4801, Oct. 2018.

[29] L. Jin, S. Li, B. Hu, and M. Liu, "A survey on projection neural networks and their applications," *Appl. Soft Comput.*, vol. 76, pp. 533–544, Jan. 2019.

[30] H. Xiao *et al.*, "Robust stabilization of a wheeled mobile robot using model predictive control based on neurodynamics optimization," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 505–516, Jan. 2017.

[31] S. Li, J. He, Y. Li, and M. U. Rafique, "Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 415–426, Feb. 2017.

[32] S. Li, Y. N. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2243–2254, Oct. 2017.

[33] S. Li, H. Wang, and M. U. Rafique, "A novel recurrent neural network for manipulator control with improved noise tolerance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1908–1918, May 2018.

[34] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[35] B. Chen, X. Chen, and C. Kanzow, "A penalized Fischer-Burmeister NCP-function," *Math. Program.*, vol. 88, pp. 211–216, Jun. 2000.

[36] S. Li, L. Jin, and M. A. Mirza, *Kinematic Control of Redundant Robot Arms Using Neural Networks*. Hoboken, NJ, USA: Wiley, 2019.

[37] 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. Mechatronics*, vol. 22, no. 3, pp. 1423–1434, Jun. 2017.

[38] L. Xiao and Y. N. Zhang, "Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 906–914, May 2013.

[39] Y. 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.

Jingkun Yan received the B.E. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2018. She is currently working toward the Ph.D. degree in computer applications technology with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China.

She is currently conducting collaborative research with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing. Her research interests include neural networks and robotics.

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.

He received postdoctoral training from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, from 2016 to 2017. His current research interests include neural networks, robotics, optimization, and intelligent computing.

Dr. Jin was the recipient of the Excellent Doctoral Dissertation Award of the Chinese Association for Artificial Intelligence.

Zhanting Yuan received the B.E. degree in automation and the M.E. degree in automatic control theory and application from Xi’an Jiaotong University, Xi’an, China, in 1983 and 1989, respectively.

He is currently a Full Professor with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. His main research interest includes automatic control theory.

Zhiyi Liu received the Master of Science degree in experimental nuclear physics from the China Institute of Atomic Energy, Beijing, China, in 2002, and the Ph.D. degree in particle and nuclear physics from Simon Fraser University, Burnaby, BC, Canada, in 2009.

He is currently a Distinguished Professor with the School of Nuclear Science and Technology, Lanzhou University, Lanzhou, China. Before joining Lanzhou University in 2018, he was with TRIUMF, Canada’s national laboratory in particle and nuclear physics, as a Research Scientist for about nine years. His primary research fields include applications of neural networks, robotics, and intelligent information processing in applied nuclear science and technology.