IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 25, 2020, accepted March 14, 2020, date of publication March 18, 2020, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981688

Recurrent Neural Networks-Based Collision-Free Motion Planning for Dual Manipulators Under Multiple Constraints

JINGLUN LIANG[1], ZHIHAO XU[2], XUEFENG ZHOU[2], SHUAI LI[3], AND GUOLIANG YE[1]

[1]School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China
[2]Guangdong Institute of Intelligent Manufacturing, Guangzhou 510070, China
[3]School of Engineering, Swansea University, Swansea SA2 8PP, U.K.

Corresponding author: Xuefeng Zhou (xf.zhou@giim.ac.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1300200, in part by the Natural Sciences Foundation of Guangdong China under Grant 2017A030313690, in part by the Guangdong Province Key Areas Research and Development Program under Grant 2019B090919002, in part by the Department of Education of Guangdong China under Grant 2017KZDXM082, in part by the Guangzhou Science and Technology Plan Project under Grant 201803010106, in part by the GDAS Project of Thousand doctors (postdoctors) Introduction under Grant 2019GDASYL-0103078, in part by the GDAS Project of Science and Technology Development under Grant 2020GDASYL-20200402007, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515010631, and in part by the Dongguan Social Science and Technology Development Project under Grant 2019507140209.

ABSTRACT Dual robotic manipulators are robotic systems that are developed to imitate human arms, which shows great potential in performing complex tasks. Collision-free motion planning in real time is still a challenging problem for controlling a dual robotic manipulator because of the overlap workspace. In this paper, a novel planning strategy under physical constraints of dual manipulators using dynamic neural networks is proposed, which can satisfy the collision avoidance and trajectory tracking. Particularly, the problem of collision avoidance is first formulated into a set of inequality formulas, whereas the robotic trajectory is then transformed into an equality constraint by introducing negative feedback in outer loop. The planning problem subsequently becomes a Quadratic Programming (QP) problem by considering the redundancy, the boundaries of joint angles and velocities of the system. The QP is solved using a convergent provable recurrent neural network that without calculating the pseudo-inversion of the Jacobian. Consequently, numerical experiments on 8-DoF modular robot and 14-DoF Baxter robot are conducted to show the superiority of the proposed strategy.

INDEX TERMS Motion planning, dual robotic manipulators, dynamic neural networks, zeroing neural networks, redundant resolution.

I. INTRODUCTION

Robot manipulators are programmable automatic devices with multiple DoFs and are designed to imitate human arms. These robots are built to effectively complete tasks and reduce the workload of human beings [1]. However, it is difficult for a single robot to fulfill complex and flexible tasks due to limited flexibility and payload [2]. Thus, dual robotic manipulator systems have been developed to imitate human arms [3]. Despite the great potential of dual robotic manipulators, the realization of motion planning with acceptable performance in real time remains challenging for several reasons. Firstly, for better flexibility, the mechanical structure of each robot is usually designed to be redundant, which should be solved in the planning process [4]. Secondly, in the coordination process, the physical interference between manipulators should be considered. Because the workspace of two manipulators are overlapped, there is a potential risk of collision. Therefore, collisions must be avoided at all times [5]. Thirdly, the states of manipulators, such as joint angles and velocities, must satisfy their constraints [6], [7]. Although conventional off-line programming methods can achieve collision-free trajectories, they are usually inefficient and inflexible [8]. Therefore, the optimal approach to design an online method that fulfills convergent motion plan while simultaneously avoiding collisions and physical constraints is still unresolved.

The associate editor coordinating the review of this manuscript and approving it for publication was Choon Ki Ahn.

Many schemes have been reported for avoiding collisions in real time. The artificial potential field method, which has the advantage of provable stability, is widely used in industrial applications. The artificial potential field methods use an attractive pole for the position to be reached and repulsive surfaces for the obstacles [9]. Thus, the robot can be programmed to avoid collision. In [10], Csiszar *et al.* improved the method by introducing multiple geometrical forms to characterize obstacles, and this method was verified to be effective in a more generalized environment. In the field of the local minimum problem in the planning process, a dual minima scheme [11]–[14] has been developed by using two attractive poles to avoid the local minima.

In terms of the redundant DOFs, many redundancy resolutions have been proposed to address the redundant resolution problem [15], [16], [34]. They are usually based on calculating the Jacobian-matrix pseudo-inverse (JMPI). Specifically, the velocity command can be formulated as $\dot{\theta} = J^{+}\dot{x}_r + (I - J^{+}J)\gamma$, and it composes a minimum norm solution $J^{+}\dot{x}_r$ and a homogeneous solution $(I - J^{+}J)\gamma$ in the null space [18]. Given that there may exist multiple solutions in the joint space that produce the same end-effector response, γ can be defined according to particular redundancy resolution schemes, such as minimizing the velocity norm or the acceleration norm. Based on the above methods, a self-motion component in the null space can be designed to avoid collisions in [17]. However, in previous studies, boundaries of robot states have been rarely considered.

In redundancy resolution problems, the selection of a homogeneous component follows a particular rule to obtain optimal solutions of predefined objective functions, and the physical constraints can usually be defined by inequality formulas. Thus, it is highly suitable to define the redundancy resolution problem as an optimization problem with constraints [19]. According to the nonlinearity of robotic systems, it is difficult to obtain the solutions of these optimization problems in real time [28]. In [20], a recurrent neural network (RNN) was introduced to calculate the control command online. In [21], a modified RNN was proposed to eliminate the drifting of end-effectors at the speed domain.

Based on the previous research, Cai expanded the RNN to different domains [22], *i.e.*, the velocity domain and acceleration domain. Taking non-convex objective functions into account, Jin studied a manipulability optimization strategy [23] in which the objective function is rebuilt according to its gradient, and a convexification operator was also proposed. Recently, this method has been successfully used in robotic force control [24], [25], adaptive control with model uncertainties [26], [35], noisy systems [29], and visual servo systems [36]. Guo and Zhang have done a lot of work on single manipulators to address the obstacle avoidance problem. For example, in [30], obstacles were described as a group of critical points, and the distance between these points and the robot was calculated on the basis of possible obstacle positions. In [31], Guo improved the method by introducing a novel distance-based smoothing function, and it was verified that the modified method could effectively restrain dithering. In [32], Xu extended the smoothing function to Class-K functions. After discussing the influence of different Class-K functions, the method of function selection was summarized. However, the research mainly focused on motion planning methods for single manipulators, and the obstacles are usually regarded as slow time-varying.

Motivated by the above observations, in this paper, we propose a RNN based motion planning method for dual robotic arms. By understanding the basic principle of obstacle avoidance method, a collision avoidance scheme is developed in form of inequality constraints, where one arm is described as a group of dynamic obstacles to the other arm. By selecting the minimum-velocity-norm scheme for redundancy resolution, a quadratic programming (QP)-type problem formulation in which the physical constraints are also considered. Then, a dynamic neural network is proposed to solve the QP problem online. Numerical results show that the proposed method is efficient. The main contributions of this paper are as:

- The proposed method can simultaneously avoid the collision of dual robots, and ensure convergent planning to predefined trajectories.
- The proposed RNN based method could handle physical constraints online, such as the boundedness of joint angles and velocities, which is of great significance in realtime motion planning.
- Different with traditional JMPI based methods, the scheme in this paper does not require the calculation of pseudo-inversion, which is capable of enhancing computing efficiency.

II. PROBLEM FORMULATION

A. KINEMATIC FORMULATION AND TRAJECTORY TRACKING

In this paper, we consider motion planning problem in velocity level, that is, design joint velocity commands to ensure the convergence of planning errors.

Firstly, the kinematic model of each manipulator of a dual robotic system is given as

$$x_i = f_i(\theta_i), \quad i = 1, 2, \tag{1}$$

where $x_i \in \mathbb{R}^m$ is the positional description of the ith robot in Cartesian space, and $\theta_i \in \mathbb{R}^n$ is the vector of joint angles. Without loss of generality, every single robot is consider as redundant. Therefore, we have $n > m$. The kinematic equation at the speed level can be described by

$$\dot{x}_i = J_i\dot{\theta}_i, \tag{2}$$

where J_i is the Jacobian matrix of the ith robot. In the motion planning problem, a fundamental objective is to generalize joint commands such that the 1^{st} and 2^{nd} arm could track the predefined trajectories $x_{1\mathrm{d}}, x_{2\mathrm{d}}$. In velocity level, the equality constraints about the above problem can be described as

$$J_i\dot{\theta}_i = \dot{x}_{i\mathrm{d}} + k|e_i|^{\rho}\mathrm{sgn}(e_i), \tag{3}$$

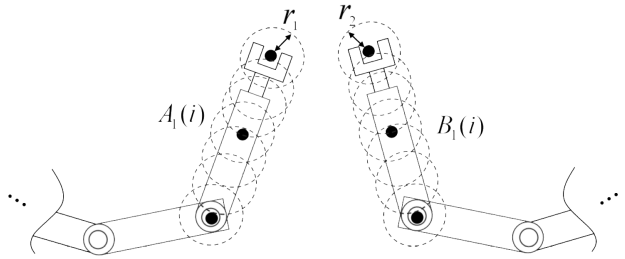


FIGURE 1. The envelope shape-based collision avoidance scheme. When the robots are described by a set of envelope shapes and the distance between critical points is greater than a specified value, collision will be avoided.

where $k|e_i|^\rho \mathrm{sgn}(e_i)$ is a negative feedback of planning errors $e_i = x_{\mathrm{id}} - x_i$, $k > 0$, $0 < \rho < 1$. It can be easily proved that the planning error e_i would converge to 0 in $T_s = 2(||e_i(0)||_2)^{1-\rho}/k(1-\rho)$. That is, if the joint velocity commands $\dot{\theta}_i$ satisfies (3), the fundamental motion planning problem can be realized.

B. COLLISION AVOIDANCE SCHEME

In the process of robot cooperation, there is a risk of collisions between two arms. Therefore, another key issue in the motion planning problem of dual robotic manipulators is to avoid potential collisions between robots. In [31], collision detection methods based on envelope modeling have been used to identify whether a collision will happen between a robot and the external environment. The robot manipulator can be simply described by a set of envelope shapes, such as cylindrical or spherical. Thus, a collision avoidance method based on the spherical envelope shapes to describe the robots is proposed in this paper.

Figure 1 shows the basic idea of the collision-free scheme used in this paper. Let $A_1 = [A_1(1), \ldots, A_1(a)] \in \mathbb{R}^{m\times a}$ be a set of centers of spherical envelope shapes of the 1^{st} robot, with the radius being selected uniformly as r_1. The principle of choosing A_1 and r_1 is that the union of spheres could completely surround the 1^{st} robot′s body. Similarly, $A_2 = [A_2(1), \ldots, A_2(a)] \in \mathbb{R}^{m\times a}$ and r_2 are defined for 2^{nd} robot. When no collision occurs, the sufficient condition can be described as follows: for arbitrary point pairs $A_1(k)A_2(l)$ from the 1^{st} and 2^{nd} robot, the distance $||A_1(k)A_2(l)||_2^2$ between $A_1(k)$ and $A_2(l)$ satisfies the inequality $||A_1(k)A_2(l)||_2^2 \geq D$, where $D = r_1 + r_2 + \Delta d$ is a called the safety distance, and $\Delta d > 0$ is the distance allowance.

Remark 1: Although different envelope shapes can be selected, the description methods are similar. In this paper, spherical envelope shapes are used to show the central idea of collision avoidance between robots; the description can be readily extended to other envelope shapes.

The inequality $||A_1(k)A_2(l)||_2^2 \geq D$ defines a basic description of collision avoidance in the position level, which is independent of joint commands. Therefore, by considering the dynamic formula

$$\frac{\mathrm{d}}{\mathrm{d}t}||A_1(k)A_2(l)||_2^2 \geq K(D - ||A_1(k)A_2(l)||_2^2), \tag{4}$$

where K is a positive parameter, it can be readily appreciated that the original inequality $||A_1(k)A_2(l)||_2^2 \geq D$ can be guaranteed by ensuring (4).

It is notable that the kinematic description of $A_1(k)$ can be calculated according to forward kinematics, similar to (1) and (2). Note that $\dot{A}_1(k) = J_{1k}(\theta_1)\dot{\theta}_1$, where J_{1k} is the corresponding Jacobian matrix of the kth spherical center of the ith manipulator. Similarly, $\dot{A}_2(l) = J_{2l}(\theta_2)\dot{\theta}_2$. Calculating the time derivative of $||A_1(k)A_2(l)||_2^2$ yields

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}||A_1(k)A_2(l)||_2^2 \\
&= \frac{\mathrm{d}}{\mathrm{d}t}(\sqrt{(A_1(k)-A_2(l))^{\mathrm{T}}(A_1(k)-A_2(l))}) \\
&= \frac{1}{||A_1(k)A_2(l)||_2^2}(A_1(k)-A_2(l))^{\mathrm{T}}(\dot{A}_1(k)-\dot{A}_2(l)) \\
&= \frac{1}{||A_1(k)A_2(l)||_2^2}(A_1(k)-A_2(l))^{\mathrm{T}}(J_{1k}(\theta_1)\dot{\theta}_1 - J_{2l}(\theta_2)\dot{\theta}_2).
\end{aligned} \tag{5}$$

Substituting (5) to (4) yields

$$\overrightarrow{A_1(k)A_2(l)}^{\mathrm{T}}[J_{1k}, -J_{2l}]\dot{\theta} \geq K(D - ||A_1(k)A_2(l)||_2^2), \tag{6}$$

where $\overrightarrow{A_1(k)A_2(l)}^{\mathrm{T}} = \dfrac{(A_1(k)-A_2(l))^{\mathrm{T}}}{||A_1(k)A_2(l)||_2^2}$ is a unit vector from $A_2(l)$ to $A_1(k)$, and $\dot{\theta} = [\dot{\theta}_1^{\mathrm{T}}, \dot{\theta}_2^{\mathrm{T}}]^{\mathrm{T}}$.

It is noteworthy that the derived inequality in (6) is capable of avoiding a potential collision between $A_1(k)$ and $A_2(l)$. The collision can be avoided effectively by ensuring all inequalities hold for $k = 1, \ldots a$, $l = 1, \ldots b$. Thus, the inequality form in the velocity level can be formulated as

$$J_{ob}\dot{\theta} \leq KD_{dis} \tag{7}$$

where

$$J_{ob} = \begin{bmatrix}
-\overrightarrow{A_1(1)A_2(l)}^{\mathrm{T}}J_{11} & \overrightarrow{A_1(1)A_2(l)}^{\mathrm{T}}J_{21} \\
\vdots & \vdots \\
-\overrightarrow{A_1(1)A_2(b)}^{\mathrm{T}}J_{11} & \overrightarrow{A_1(1)A_2(b)}^{\mathrm{T}}J_{2b} \\
\vdots & \vdots \\
-\overrightarrow{A_1(a)A_2(1)}^{\mathrm{T}}J_{1a} & \overrightarrow{A_1(a)A_2(1)}^{\mathrm{T}}J_{21} \\
\vdots & \vdots \\
-\overrightarrow{A_1(a)A_2(b)}^{\mathrm{T}}J_{1a} & \overrightarrow{A_1(a)A_2(b)}^{\mathrm{T}}J_{2b}
\end{bmatrix} \in \mathbb{R}^{ab\times 2m},$$

$$D_{dis} = \begin{bmatrix}
||A_1(1)A_2(l)||_2^2 - D \\
\vdots \\
||A_1(1)A_2(b)||_2^2 - D \\
\vdots \\
||A_1(a)A_2(l)||_2^2 - D \\
\vdots \\
||A_1(a)A_2(b)||_2^2 - D
\end{bmatrix} \in \mathbb{R}^{ab}.$$

Remark 2: The inequality in (7) describes the cascaded form of the collision avoidance scheme by considering all

pairs of spherical centers. It is noted that the minus sign in the first column in J_{ob} plays an important role in collision avoidance, which shows that the corresponding velocities of the centers of the 1^{st} and 2^{nd} robots are opposite. The proposed collision avoidance schemes considers the relative movement of the two arms, which is an important improvement than methods in [30], [31]. It is also notable that from the description in (7), the self-collision formulation can be similarly derived for the joint-speed level. Since this paper mainly focuses on the collision between robots, it is not addressed here.

C. QP-TYPE PROBLEM FORMULATION AND RNN DESIGN

In previous subsections, we have obtained the equality and inequality constraints about the fundamental plan objective and the collision avoidance scheme. In this subsection, by considering the physical constraints and cost function to be optimized, the original motion planning problem is transferred into a QP one. Then some proper modification is done to formulate the QQP problem in the same level, and finally a RNN is established to solve the QP online.

In real applications, the physical states of manipulators are limited, *i.e.*, the joint angles θ_1 and θ_2 are limited by the mechanical design, and the joint speeds $\dot{\theta}_1$ and $\dot{\theta}_2$ are also constrained by the limitation of actuators. Therefore, during the planning process, both joint angles θ_1, θ_2 and joint velocities $\dot{\theta}_1$ and $\dot{\theta}_2$ must not exceed their limits, in the sense that $\theta_1^- \le \theta_1 \le \theta_1^+$, $\theta_2^- \le \theta_2 \le \theta_2^+$, $\dot{\theta}_1^- \le \dot{\theta}_1 \le \dot{\theta}_1^+$, $\dot{\theta}_2^- \le \dot{\theta}_2 \le \dot{\theta}_2^+$, where θ_1^-, θ_2^-, θ_1^+, and θ_2^+ are the lower and upper bounds of the two manipulators, and $\dot{\theta}_1^-$, $\dot{\theta}_2^-$, $\dot{\theta}_1^+$, and $\dot{\theta}_2^+$ are the limits of the joint velocities.

Due to the redundancy of the manipulators, in addition to completing the original task, it can also achieve the secondary target simultaneously. Therefore, without loss of generality, the secondary task is selected to minimize the norm of joint velocities, *i.e.*, $\dot{\theta}_1^{\mathrm{T}}\dot{\theta}_1 + \dot{\theta}_2^{\mathrm{T}}\dot{\theta}_2$. On the basis of the previous descriptions, the QP-type problem formulation is derived as

$$\begin{aligned}
&\min\ \dot{\theta}_1^{\mathrm{T}}\dot{\theta}_1/2 + \dot{\theta}_2^{\mathrm{T}}\dot{\theta}_2/2, && (8a)\\
&s.t.\ J_1\dot{\theta}_1 = \dot{x}_{1\mathrm{d}} + k|e_1|^{\rho}\mathrm{sgn}(e_1), && (8b)\\
&\quad J_2\dot{\theta}_2 = \dot{x}_{2\mathrm{d}} + k|e_2|^{\rho}\mathrm{sgn}(e_2), && (8c)\\
&\quad \theta_1^- \le \theta_1 \le \theta_1^+, && (8d)\\
&\quad \theta_2^- \le \theta_2 \le \theta_2^+, && (8e)\\
&\quad \dot{\theta}_1^- \le \dot{\theta}_1 \le \dot{\theta}_1^+, && (8f)\\
&\quad \dot{\theta}_2^- \le \dot{\theta}_2 \le \dot{\theta}_2^+, && (8g)\\
&\quad J_{ob}\dot{\theta} \le KD_{dis}. && (8h)
\end{aligned}$$

By letting $\theta = [\theta_1^{\mathrm{T}}, \theta_2^{\mathrm{T}}]^{\mathrm{T}}$, $\dot{\theta} = [\dot{\theta}_1^{\mathrm{T}}, \dot{\theta}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\theta^- = [(\theta_1^-)^{\mathrm{T}}, (\theta_2^-)^{\mathrm{T}}]^{\mathrm{T}}$, $\theta^+ = [(\theta_1^+)^{\mathrm{T}}, (\theta_2^+)^{\mathrm{T}}]^{\mathrm{T}}$, $\dot{\theta}^- = [(\dot{\theta}_1^-)^{\mathrm{T}}, (\dot{\theta}_2^-)^{\mathrm{T}}]^{\mathrm{T}}$, $\dot{\theta}^+ = [(\dot{\theta}_1^+)^{\mathrm{T}}, (\dot{\theta}_2^+)^{\mathrm{T}}]^{\mathrm{T}}$, $e = [(e_1^-)^{\mathrm{T}}, (e_2^-)^{\mathrm{T}}]^{\mathrm{T}}$, $x_{\mathrm{d}} = [x_{\mathrm{d1}}^{\mathrm{T}}, x_{\mathrm{d2}}^{\mathrm{T}}]^{\mathrm{T}}$, and define $J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix}$ as a block diagonal matrix of J_1 and J_2, (8)can be reformulated as

$$\begin{aligned}
&\min\ \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, && (9a)\\
&s.t.\ J\dot{\theta} = \dot{x}_{\mathrm{d}} + k|e|^{\rho}\mathrm{sgn}(e), && (9b)\\
&\quad \theta^- \le \theta \le \theta^+, && (9c)\\
&\quad \dot{\theta}^- \le \dot{\theta} \le \dot{\theta}^+, && (9d)\\
&\quad J_{ob}\dot{\theta} \le KD_{dis}. && (9e)
\end{aligned}$$

The original QP (9) gives the basic idea to transfer the original trajectory tracking problem into an equality constraint, which is essential to rebuild the trajectory tracking into QP type problem. In this paper, (9) is capable of ensuring fast convergence of tracking error.

Given the angular constraints in (9c) and (9d), a unified description can be obtained according to the escape velocity method as below

$$\dot{\theta}_r^- \le \dot{\theta} \le \dot{\theta}_r^+, \tag{10}$$

where $\dot{\theta}_r^- = \max(\dot{\theta}^-, \alpha(\theta^- - \theta))$ and $\dot{\theta}_r^+ = \min(\dot{\theta}^+, \alpha(\theta^+ - \theta))$ play important roles in ensuring the inequality constraints in (9c) and (9d). Here, α is a positive constant for weighting the feedback gain to ensure (9c). Now, the final form of the QP-type formulation is formulated as

$$\begin{aligned}
&\min\ \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, && (11a)\\
&s.t.\ J\dot{\theta} = \dot{x}_{\mathrm{d}} + k|e|^{\rho}\mathrm{sgn}(e), && (11b)\\
&\quad \dot{\theta}_r^- \le \dot{\theta} \le \dot{\theta}_r^+, && (11c)\\
&\quad J_{ob}\dot{\theta} \le KD_{dis}. && (11d)
\end{aligned}$$

According to the Karush-Kuhn-Tucker conditions, the optimal solution of the QP-type formulation in (11) satisfies

$$\begin{aligned}
&\dot{\theta} = P_{\Omega}(\dot{\theta} - \frac{\partial L}{\partial \dot{\theta}}), && (12a)\\
&J\dot{\theta} = \dot{x}_{\mathrm{d}} + k|e|^{\rho}\mathrm{sgn}(e), && (12b)\\
&\begin{cases} \lambda_2 > 0 & \text{if } J_{ob}\dot{\theta} = KD_{dis}, \\ \lambda_2 = 0 & \text{if } J_{ob}\dot{\theta} \le KD_{dis}, \end{cases} && (12c)
\end{aligned}$$

where L is a Lagrange function, which is defined by

$$\begin{aligned}
L = \dot{\theta}^{\mathrm{T}}\dot{\theta}/2 + \lambda_1^{\mathrm{T}}(\dot{x}_{\mathrm{d}} + k|e|^{\rho}\mathrm{sgn}(e) - J\dot{\theta}) \\
+ \lambda_2^{\mathrm{T}}(J_{ob}\dot{\theta} - KD_{dis}), \qquad (13)
\end{aligned}$$

in which $\lambda_1 \in \mathbb{R}^{2m}$ and $\lambda_2 \in \mathbb{R}^{ab}$ are dual state variables corresponding to the constraints in (11b) and (11c), respectively. P_{Ω} is a projection operator to a closed set $\Omega = [\dot{\theta}_r^-, \dot{\theta}_r^+]$ and $P_{\Omega}(x) = \mathrm{argmin}_{y\in\Omega}||y - x||$ is the saturation function defined as

$$P_{\Omega}(x) = \begin{cases} \dot{\theta}_r^- & x < \dot{\theta}_r^- \\ x & \dot{\theta}_r^- \le x \le \dot{\theta}_r^+ \\ \dot{\theta}_r^+ & x > \dot{\theta}_r^+ \end{cases}.$$

Equation (12) is typically nonlinear, and it is time-consuming to solve. Moreover, because of the time-varying characteristic, the solution of (12) also differs with time. Therefore, in this paper, a dynamic neural network which has the ability of parallel computing is proposed as

$$\begin{aligned}
&\epsilon\ddot{\theta} = -\dot{\theta} + P_{\Omega}(J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2), && (14a)\\
&\epsilon\dot{\lambda}_1 = \dot{x}_{\mathrm{d}} + k|e|^{\rho}\mathrm{sgn}(e) - J\dot{\theta}, && (14b)
\end{aligned}$$

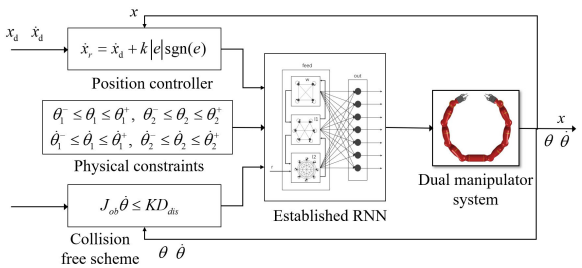


FIGURE 2. The dual robotic manipulator system. The blue points indicate the key points of the manipulators.

$$\epsilon\dot{\lambda}_2 = -\lambda_2 + (\lambda_2 + J_{ob}\dot{\theta} - KD_{dis})^+, \tag{14c}$$

where the operator $(\bullet)^+$ is a defined as $y^+ = [y(1)^+, \cdots, y(m)^+] = [\max(y(1), 0), \cdots, \max(y(m), 0)]$.

The structure diagram of the developed planning method is shown in figure 2. Given the desired task x_{d} and $\dot{x}_{\mathrm{d}}$, the scheme can be built based on the sensor feedbacks $\theta, \dot{\theta}$ of the dual manipulator system. The collision avoidance scheme has been established by setting a proper safety distance considering the joint configurations of both manipulators. With the consideration of physical constraints and optimization metrics, the motion planning problem is transferred into a QP type description, and an RNN is developed to obtain the joint command $\dot{\theta}$ in real time.

Remark 3: Compared to conventional pure data driven neural networks that are used to approximate unknown nonlinear items in reference [19], [27], the RNN established in this paper uses model information to determine weights of the neural network, and thus it has a simpler structure than conventional ones.

Remark 4: Using the proposed RNN based controller, the joint velocity command $\dot{\theta}$ can be obtained by the updating law Eq.(14). It can be shown in the latter section that the output of Eq.(14) would ensure the convergent tracking of the dual manipulator systems, and avoiding the possible collisions. It is notable that based on the proposed controller Eq.(14), dynamic controllers in the inner loop can be further developed, in which the robot dynamics such as inertia, Coriolis-centripetal force, external disturbances can be considered. Relative studies can be found in references such as [39]–[41].

D. STABILITY ANALYSIS

In this subsection, the stability of the proposed RNN based planning method is discussed. Firstly, several lemmas are given.

Definition 1: A continuously differentiable function $F \in \mathbb{R}^k$ is monotonic if its gradient ∇F satisfies the following skew symmetry property:

$$\xi^{\mathrm{T}}(\nabla F + \nabla F^{\mathrm{T}})\xi \geq 0. \quad \forall \xi \in \mathbb{R}^k \tag{15}$$

Lemma 1 (About the Convergence of Neural Networks Based on Projection Operators): [33] A dynamic neural network is ensured to converge to its equilibrium point if its dynamics satisfies the following equation:

$$\kappa\dot{x} = -x + P_S(x - F(x)), \tag{16}$$

where $\kappa > 0$ is a positive constant that describes the convergence speed of the network, and $F(x)$ is a monotonic function, which is introduced later. $P_S(x) = \mathrm{argmin}_{y\in S}||y - x||$ is a projection operator to the convex set S.

The stability is analyzed and proved by three parts. Firstly, the neural network is able to converge to its equilibrium point. Then this equilibrium point is proved to be equivalent to the optimal solution of the QP-type description of the problem. Finally, the convergence of the planning error is analyzed.

Theorem 1: The proposed recurrent neural network (14) will converge to its optimal solution globally.

Proof: If a new augmented vector ξ composed of $\dot{\theta}$, λ_1, and λ_2 is defined as

$$\xi = [\dot{\theta}^{\mathrm{T}}, \lambda_1^{\mathrm{T}}, \lambda_2^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2n+2m+ab}, \tag{17}$$

then equation (14) is reformulated as

$$\epsilon\dot{\xi} = \epsilon\begin{bmatrix}\ddot{\theta}\\ \dot{\lambda}_1\\ \dot{\lambda}_2\end{bmatrix} = \begin{bmatrix}-\dot{\theta}\\ -\lambda_1\\ -\lambda_2\end{bmatrix} + \begin{bmatrix}P_\Omega(\dot{\theta} - \dot{\theta} + J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2)\\ \lambda_1 + (\dot{x}_{\mathrm{d}} + k|e|^\rho \mathrm{sgn}(e) - J\dot{\theta})\\ (\lambda_2 + J_{ob}\dot{\theta} - KD_{dis})^+\end{bmatrix}, \tag{18}$$

which can be further presented as

$$\epsilon\dot{\xi} = -\xi + P_{\bar{\Omega}}(\xi - F(\xi)), \tag{19}$$

where

$$F(\xi) = \begin{bmatrix}\dot{\theta} - J^{\mathrm{T}}\lambda_1 + J_o^{\mathrm{T}}\lambda_2\\ -\dot{x}_{\mathrm{d}} - k|e|^\rho \mathrm{sgn}(e) + J\dot{\theta}\\ -J_{ob}\dot{\theta} + KD_{dis}\end{bmatrix} \tag{20}$$

Algorithm 1 Collision-Free Motion Planning Scheme for Dual Robotic Manipulators

Require: The expected trajectories $x_{1\mathrm{d}}$ and $x_{2\mathrm{d}}$. Feasible range of joint states θ_1^-, θ_2^-, θ_1^+, θ_2^+, $\dot{\theta}_1^-$, $\dot{\theta}_2^-$, $\dot{\theta}_1^+$, $\dot{\theta}_2^+$. Positive parameters k, K, ρ, α, and ϵ. Information of key points A_i and B_j and the corresponding Jacobian matrices J_{1i}, J_{2j}. Feedback of the end-effector's positional description x_1, x_2 in Cartesian space. Task duration T.

Ensure: To realize motion planning of dual manipulators and avoid collisions

Initialize $\lambda_1(0)$, $\lambda_2(0)$
repeat
  Get feedback information x, θ_1, θ_2 from sensors
  Calculate tracking error $e \leftarrow x_{\mathrm{d}} - x$
  Calculate matrices J_{ob} and D_{dis} in the collision avoidance scheme according to (7)
  Calculate the reference velocity limits $\dot{\theta}_r^-, \dot{\theta}_r^+$
  Update joint command $\dot{\theta}$ by $\ddot{\theta}$ according to (14a)
  Update state variable λ_1 according to (14b)
  Update state variable λ_2 according to (14c)
until $(t > T)$

It is noteworthy that for (20), $P_R(\bullet)$ can be regarded as a projection operator of λ_1 to a special convex R, with the lower and upper bounds being $\pm\infty$. $P_+(\bullet) = (\bullet)^+$ is also a special projection operator to a non-negative real set. For example, the lower and upper bounds of the set are 0 and $+\infty$, respectively.

The gradient of F is calculated as

$$\nabla F(\xi) = \begin{bmatrix} I & -J^{\mathrm{T}} & J_{ob}^{\mathrm{T}} \\ J & 0 & 0 \\ -J_{ob}^{\mathrm{T}} & 0 & 0 \end{bmatrix}. \tag{21}$$

Thus $\nabla F(\xi) + \nabla F^{\mathrm{T}}(\xi) = \begin{bmatrix} 2I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, and $F(\xi)$ is monotonic according to definition 1.

It can be concluded that the dynamics of the established dynamic neural network in (14) satisfies the property described in (21). This is equivalent to the sufficient condition for the convergence of neural networks based on projection operators, such as (15). According to Lemma 1, the convergence of the proposed planning method is guaranteed, *i.e.*, let $\xi^* = [\dot{\theta}^*; \lambda_1^*; \lambda_2^*]$ be the equilibrium point, $\xi \to \xi^*$ as $t \to \infty$.

Theorem 2: The optimal solution ξ^* is equivalent to the QP problem in (11) and ensures the convergence of the planning error.

Proof: The optimal solution of the proposed network in (14) satisfies

$$0 = -\dot{\theta} + P_\Omega(J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2), \tag{22a}$$
$$0 = \dot{x}_{\mathrm{d}} + k|e|^{\rho}\mathrm{sgn}(e) - J\dot{\theta}, \tag{22b}$$
$$0 = -\lambda_2 + (\lambda_2 + J_{ob}\dot{\theta} - KD_{dis})^+. \tag{22c}$$

From the definition of $(\bullet)^+$, the following conditions are discussed as

- When $\lambda_2 + J_{ob}\dot{\theta} - KD_{dis} > 0$, $(\lambda_2 + J_{ob}\dot{\theta} - KD_{dis})^+ = \lambda_2 + J_{ob}\dot{\theta} - KD_{dis}$, which leads to $J_{ob}\dot{\theta} - KD_{dis} = 0$ and $\lambda_2 > 0$ according to (22c).
- When $\lambda_2 + J_{ob}\dot{\theta} - KD_{dis} \le 0$, $(\lambda_2 + J_{ob}\dot{\theta} - KD_{dis})^+ = 0$, and then $\lambda_2 = 0$. thus, $J_{ob}\dot{\theta} - KD_{dis} \le 0$.

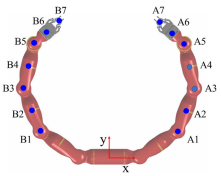
Next, a deep comparison between (22) and (12) shows that the solution of (22) is equivalent to that of (12). □

From theorem 1, it can be concluded that the output $\dot{\theta}$ of the established dynamic neural network will eventually satisfy the equality constraint in (22b). From the definition of the planning error e, its gradient $\dot{e}$ can be formulated as $\dot{e} = \dot{x}_{\mathrm{d}} - \dot{x} = \dot{x}_{\mathrm{d}} - J\dot{\theta}$. By defining a Lyapunov function V as $V = e^{\mathrm{T}}e/2$, it can be readily observed that

$$\begin{aligned} \dot{V} &= e^{\mathrm{T}}\dot{e} \\ &= -ke^{\mathrm{T}}|e|^{\rho}\mathrm{sgn}(e) = -k|e|^{\rho+1} \le 0, \end{aligned} \tag{23}$$

and then the convergence of the planning error is ensured. □

Remark 5: It is notable that the RNN based planning method could only ensure the global convergence of planning error, rather than finite convergent. Since the proposed Eq. (14) is global convergent, therefore, the planning error considering the whole system is global convergent. However, whether the planning error is finite-convergent or global convergent is no the key issue of the proposed method. The main contributions of the paper is that the method could ensure convergence of planning error and avoid collision between robots, while it is capable of handling physical constraints in real time.



(a)

	Joint	$a_i(m)$	$\alpha_i(rad)$	$d_i(m)$	offset(rad)
Robot A	1	0.296	0	0	0
	2	0.296	0	0	0
	3	0.296	0	0	0
	4	0.212	0	0	0
	Joint	$a_i(m)$	$\alpha_i(rad)$	$d_i(m)$	offset(rad)
Robot B	1	0.296	0	0	π
	2	0.296	0	0	0
	3	0.296	0	0	0
	4	0.212	0	0	0

(b)

FIGURE 3. The architecture of 8-DOF modular robot. (a) Physical structure of 8-DOF modular robot. (b) D-H parameters.

Remark 6: In this section, the stability of the proposed control scheme is proved. Compared with other neural network-based methods in which neural networks are used to learn nonlinear items, the method proposed in this paper is much simpler. This is mainly because robotic models are introduced to the neural networks, while conventional networks only use error information.

Remark 7: So far, the proposed method is proved to be capable of ensuring the convergent planning of the end-effectors of both manipulators, and the collision between robots is avoided. By perceiving the configuration of the manipulators, collision avoidance is achieved by adjusting the robots online, which results in better performance than methods with offline programming.

III. NUMERICAL RESULTS

In this section, numerical simulations are presented to show the effectiveness of the proposed method. The coordination tasks of dual manipulator systems require robots to cooperate with each other. In other words, the manipulators receive control commands that end-effectors x_1 and x_2 track their reference trajectories $x_{1\mathrm{d}}$ and $x_{2\mathrm{d}}$, respectively, where $x_{1\mathrm{d}}$ are $_{2\mathrm{d}}$ can be the same or different in real applications. Firstly, simulations on a 8DOF planar modular robot system are done. An interesting case is analyzed in which one end-effector is required to stay at a fixed point while the other end-effector is planned to track a certain trajectory. Then a case in which the robots are planed to track different trajectories is presented. Secondly, simulation on a 14DOF dual arm Baxter is carried out. Finally, comparisons are made with previous methods to demonstrate the superiority of the proposed motion planning method.

A. NUMERICAL RESULTS ON MODULAR ROBOT

Firstly, two simulations are carried out on a 8DOF planar modular robot system. The physical structure and the DH parameters of single arm are given in Fig. 3. The ground coordinate system is marked in red arrows in Fig. 3(a). The

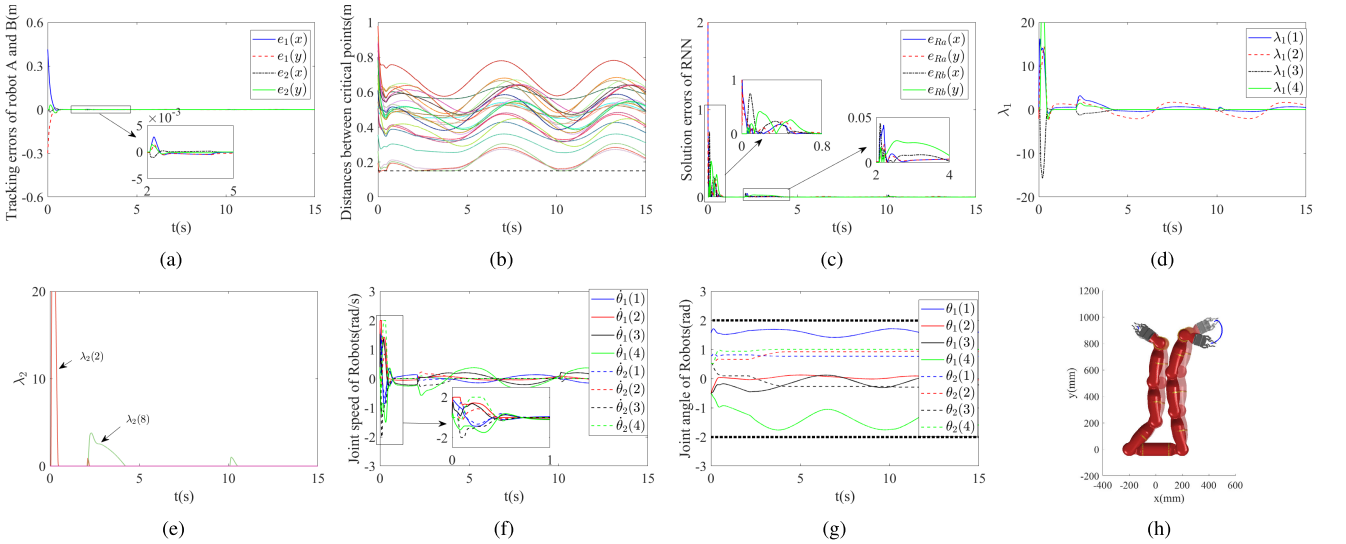


FIGURE 4. Numerical results of the motion planning method with a Regulation Hand. (a) Profile of planning errors. (b) Profile of distances between the critical points. (c) Profile of solution errors of the established neural network. (d) Profile of state variable λ_1. (e) Profile of state variable λ_2. (f) Profile of joint velocities. (g) Profile of joint angles. (h) Snapshots of the system with a regulation robot.

key points A_i, $i = 1, \cdots, 6$ and B_i, $i = 1, \cdots, 6$, also marked in Fig. 3(a), are important points for achieving collision avoidance. The 1^{st}, 3^{rd}, and 5^{th} key points are located in the 2^{nd}, 3^{rd}, and 4^{th} joint, respectively, while the 2^{nd}, 4^{th}, and 6^{th} key points are located at the center of mass of the 2^{nd}, 3^{rd}, and 4^{th} link. The 7^{th} key point exactly marks the end-effector. The DH parameters are given in Fig. 3(b).

In this section, the safety distance D is defined as $D = 0.15$ m to eliminate collisions, and the parameter in the collision avoidance scheme in (6) is selected as $K = 200$. In the outer loop, the control gain of the position controller is selected as $k = 8$ and $\rho = 1$. As an important parameter that scales the convergence of the established dynamic neural network, ϵ is selected as $\epsilon = 0.001$. For the physical constraints, the angular limits of both manipulators are defined as $\theta_1^- = \theta_2^- = [-2, -2, -2, -2]^{\mathrm{T}}$ rad, $\theta_1^+ = \theta_2^+ = [2, 2, 2, 2]^{\mathrm{T}}$ rad, $\dot{\theta}_1^- = \dot{\theta}_2^- = [-2, -2, -2, -2]^{\mathrm{T}}$ rad/s, and $\dot{\theta}_1^+ = \dot{\theta}_2^+ = [2, 2, 2, 2]^{\mathrm{T}}$ rad/s.

1) MOTION PLAN WITH A REGULATION HAND

In this part, "fixed point-trajectory tracking planning" verification is carried out. The end-effector of right arm is expected to stay at the initial position, depending on the initial joint configurations. The end-effector of left arm is expected to track the time-varying trajectory $x_{1\mathrm{d}} = [0.45 + 0.1sin(t); 0.9 + 0.1cos(t)]^{\mathrm{T}}$ m. The initial states of the robots are set to $\theta_2(0) = [\pi/4, \pi/6, \pi/6, \pi/6]^{\mathrm{T}}$ rad, $\theta_1(0) = [\pi/2, -\pi/6, -\pi/6, -\pi/6]^{\mathrm{T}}$ rad, $\dot{\theta}_2(0) = [0, 0, 0, 0]^{\mathrm{T}}$ rad/s, and $\dot{\theta}_1(0) = [0, 0, 0, 0]^{\mathrm{T}}$ rad/s.

Numerical results are as shown in Fig. 4. The planning errors of the end-effectors in the desired trajectories are shown in Fig. 4(a). The stable errors of left and right arm are about 4×10^{-4} and 1×10^{-5} m, respectively. The profiles of the joint angles and velocities are shown in Fig. 4 (f)(g). It is remarkable that at the beginning stage, joint velocities of the robots reach the maximum value, and then slow down quickly. Correspondingly, the planning errors converge to 0. It can be also readily observed from Fig. 4 (g) that the joints are ensured not to exceed the upper and lower bounds (black dotted line). During $t = 0$–0.5s and $t = 3$–5 s, the robots approach each other, accordingly, the state variable λ_2 become positive, which guarantees the inequality (11d), and the minimum safety distance is obviously ensured through (14c), as shown in Fig. 4 (b). The proposed planning method ensures that the safety distance between the two robots is maintained, and the collision is thus avoided. The solution errors of the established neural network are shown in Fig. 4 (c). The errors converge to zero in less than 0.5 seconds. At $t = 5$ s, it reaches its maximum value of 3×10^{-3} m. The snapshots during the planning process are given in Fig. 4 (h). It can be observed that the right arm adjusts its joint configuration while maintaining the position of end-effector, and the collision is avoided in realtime.

2) MOTION PLAN OF TIME-VARYING TRAJECTORIES

In this scenario, the motion planning problem to two different time-varying trajectories will be discussed. The desired trajectories are selected as $x_{1\mathrm{d}} = [0.4 - 0.1sin(t); 0.9 + 0.1cos(t)]$ and $x_{2\mathrm{d}} = [-0.4 + 0.1sin(t); 0.9 + 0.1cos(t)]$ m, respectively. Joint angles at $t = 0$ are set to $\theta_1(0) = [\pi/2; -\pi/6; -\pi/6; -\pi/6]$ and $\theta_2(0) = [\pi/2; \pi/6; \pi/6; \pi/6]$ rad.

Simulation results are shown in Fig. 5. The planning errors of both manipulators converge to 0 successfully, and stable errors are less than 2×10^{-4} m, showing that the designed scheme can achieve stable tracking(Fig. 5(a)). The changing curve of the distance between robots is given in Fig. 5(b). Fig. 5(f) and (g) illustrate the states of the manipulators, and it

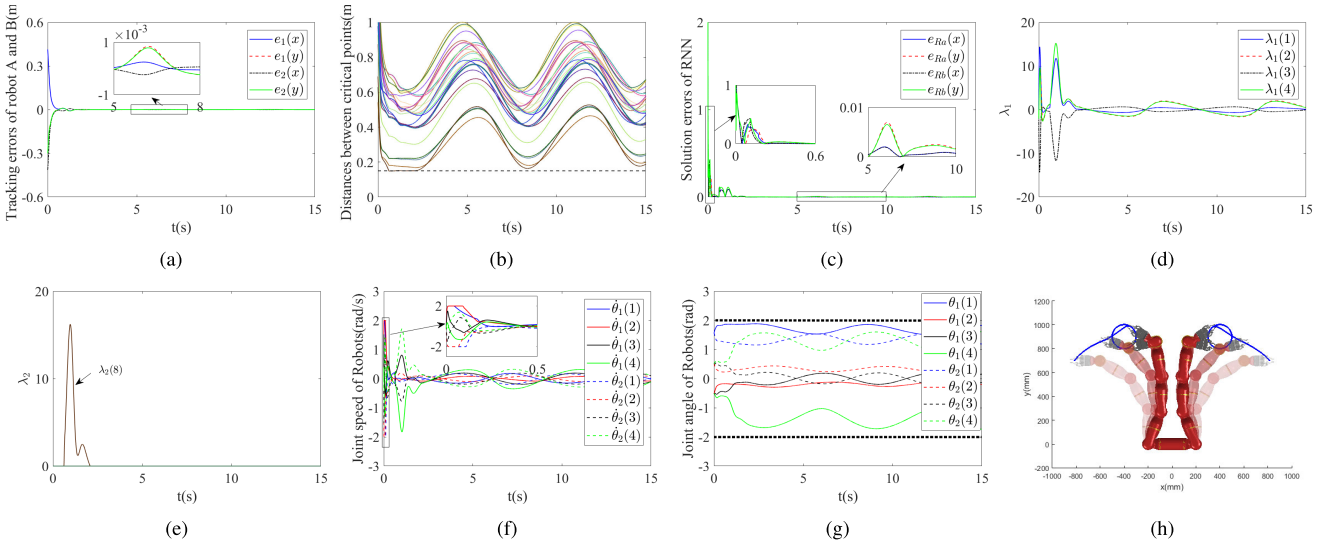


FIGURE 5. Numerical results of the motion planning method for time-varying trajectories. (a) Profile of planning errors. (b) Profile of distances between the critical points. (c) Profile of solution errors of the established neural network. (d) Profile of state variable λ_1. (e) Profile of state variable λ_2. (f) Profile of joint velocities. (g) Profile of joint angles. (h) Snapshots of the system when tracking time-varying trajectories.

can be readily observed that the boundary of the joint angles and velocities is guaranteed. And the speed curves uplift at $t = 3$ s, which is because of the influence of (14c). As proved in Section 2.4, the RNN-based scheme globally converges, which can be verified from the quick convergence of solution errors in Fig. 5(c). Snapshots of the planning process are given in Fig. 5(h). The end-effectors move smoothly toward the desired paths, and then they track them successfully.

3) COMPARISON WITH JMPI BASED METHOD

In this part, comparative simulations are carried out to show the superiority of the proposed controller. We compare the proposed strategy with JMPI based methods, in which the joint velocity command is obtained by calculating the pseudo-inverse of Jacobian matrix in every control period, and the collision avoidance is achieved in the null space. Comparative results are shown in Fig. (6). Fig. 6(a) shows the Euclidean norm of the planning errors by methods based on RNN and JMPI, respectively, and the corresponding curves of minimum distance between the left and right arm are as given in Fig. 6(b). Both methods could guarantee the convergence of planning errors and collision avoidance. However, by comparing the curves of joint velocities (Fig. 6(c) and Fig. 5(f)) and joint angles (Fig. 6(d) and Fig. 5(g)), it can be observed that JMPI based method failed in handling physical constraints. It is remarkable that in real applications, the satisfaction of physical constraints in real time is of great significance in ensuring the safety of dual-manipulator systems.

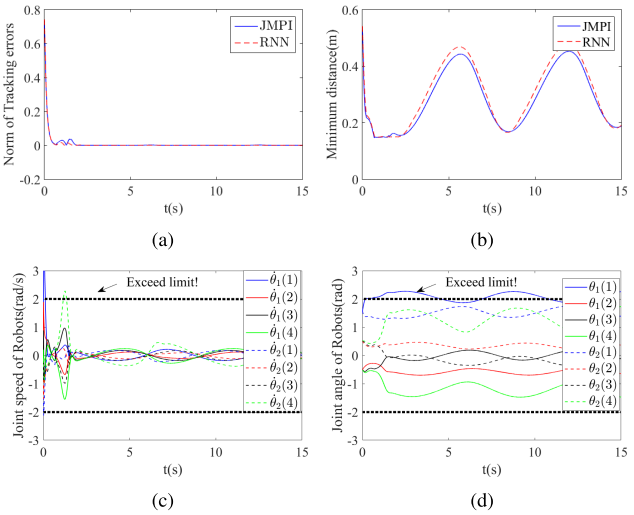


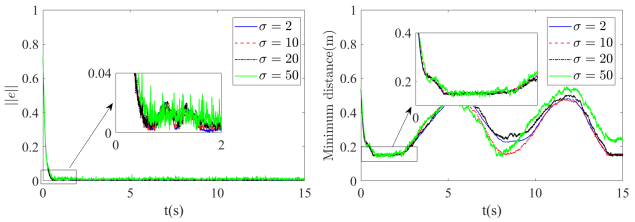
FIGURE 6. Comparative results under JMPI based method. (a) Comparison of planning error using methods based on JMPI and RNN. (b) Comparison of minimum distance using methods based on JMPI and RNN. (c) Profile of joint velocity using method based on JMPI. (d) Profile of joint angles using method based on JMPI.

4) WITH NOISE

In this section, we evaluate the impact of noise on neural controllers. It is notable that the angle feedback accuracy of robot joint is very high, and the noise mainly exists in the angle velocity. Therefore, we consider white Gaussian noise in the feedback of joint speed. In this simulation, four cases are considered, *i.e.,* covariance is selected as 2, 10, 20, 50, respectively. Simulation results are shown in Fig. 7. Fig. 7 (a) shows the Euclidean norm planning errors, with the maximum value in stable state is 0.02. Fig. 7 (b) shows the minimum distance between the robots. It can be found that under the disturbance of white Gaussian noise, the proposed planning method could achieve satisfactory results.

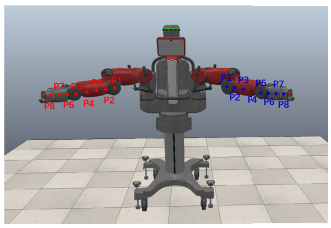
B. NUMERICAL RESULTS ON 14DOF BAXTER

In this section, numerical results on a 14-DOF robot Baxter are presented. The physical structure and the DH parameters



(a) (b)

FIGURE 7. Numerical results of the proposed motion planning method for time-varying trajectories under white Gaussian noise. (a) Planning errors. (b) Minimum distance between left and right arm.



	Joint	$\alpha_i(m)$	$\alpha_i(rad)$	$d_i(m)$	offset(rad)
Robot A	1	0.069	-π/2	0.2703	0
	2	0	π/2	0	π/2
	3	0.069	-π/2	0.3644	0
	4	0	π/2	0	0
	5	0.01	-π/2	0.3743	0
	6	0	π/2	0	0
	7	0	0	0.28	0
	Joint	$\alpha_i(m)$	$\alpha_i(rad)$	$d_i(m)$	offset(rad)
Robot B	1	0.069	-π/2	0.2703	0
	2	0	π/2	0	π/2
	3	0.069	-π/2	0.3644	0
	4	0	π/2	0	0
	5	0.01	-π/2	0.3743	0
	6	0	π/2	0	0
	7	0	0	0.28	0

(a) (b)

FIGURE 8. The architecture of 14-DOF Baxter. (a) Physical structure of 14-DOF Baxter. (b) D-H parameters.

of Baxter are given in Fig. 8. The key points $A_i, i = 1, \cdots, 8$ and $B_i, i = 1, \cdots, 8$, also marked in Fig. 8(a), are important points for achieving collision avoidance. The 1^{st} and 5^{th} key points are located in the 4^{th}, 6^{th}joint, respectively, while the 2^{nd}, 3^{rd}, 4^{th}, 6^{th}, 7^{th}, 8^{th} are evenly distributed on the 3^{rd}, 4^{th} link. The DH parameters are given in Fig. 8(b).

The safety distance D is defined as $D = 0.2$ m to eliminate collisions, and the parameter in the collision avoidance scheme in (6) is selected as $K = 200$. In the outer loop, the control gain of the position controller is selected as $k = 8$ and $\rho = 1$. As an important parameter that scales the convergence of the established dynamic neural network, ϵ is selected as $\epsilon = 0.001$. For the physical constraints, the angular limits of both manipulators are defined as $\theta_1^- = \theta_2^- = [-90, -110, -160, 0, -160, -80, -160]^{\mathrm{T}}$ degree, $\theta_1^+ = \theta_2^+ = [90, 50, 160, 130, 150, 110, 160]^{\mathrm{T}}$ degree, *i.e.*, $\theta_1^- = \theta_2^- = [-1.57, -1.92, -2.79, 0, -2.79, -1.40, -2.79]^{\mathrm{T}}$ rad, $\theta_1^+ = \theta_2^+ = [1.57, 0.87, 2.79, 2.27, 2.62, 1.92, 2.79]^{\mathrm{T}}$ rad. $\dot{\theta}_1^- = \dot{\theta}_2^- = [-2, -2, -2, -2, -2, -2, -2]^{\mathrm{T}}$ rad/s, and $\dot{\theta}_1^+ = \dot{\theta}_2^+ = [2, 2, 2, 2, 2, 2, 2]^{\mathrm{T}}$ rad/s. In this simulation, the left and right arms are required to track a circular path, and the phase difference is π: $x_{1d} = [0.4 - 0.15cos(0.5t), -0.15sin(0.5t), 1]^{\mathrm{T}}$, $x_{2d} = [0.4 + 0.15cos(0.5t), 0.15sin(0.5t), 1]^{\mathrm{T}}$. The initial joint angles are set to be $\theta_1(0) = [-0.18, -0.17, -0.89, 2.0, 0.09, 0.89, 0]^{\mathrm{T}}$ rad; $\theta_2(0) = [1.56, -0.61, -0.781.51, 0.72, 1.10, 0]^{\mathrm{T}}$ rad.

Numerical results are as shown in Fig. 9–11. The planning errors are shown in Fig. 9(a), the stable errors of robots A and B are about 1×10^{-3} and 6×10^{-4} m, respectively. At the beginning stage, joint velocities of the robots reach the maximum value, and then slow down quickly. Correspondingly, the planning errors converge to 0. During $t = 1.5$–7s, the robots approach each other, and the distance reaches the minimum value (0.2m). Using the proposed motion plan method, joint angular velocity curve jumps (up to about 1 rad/s), and the planning error of the robot changes slightly (about 1×10^{-3}), but the minimum safety distance robots is guaranteed. It is notable that the state variables λ_1 and λ_2 change according to the planning error and minimum distance, respectively. The solution errors of the established neural network are shown in Fig. 9(g). The errors converge to zero in less than 0.3 seconds. The profile of joint angles and the corresponding limits are shown in Fig. 10, in which the red filled part represents the accessible range of the joint angles. It can be readily observed that the joint angles are ensured not to exceed the limits (the 2^{nd} joint of left arm and 1^{st} joint reach the upper bound). Although existing JMPI-based online motion planning methods may obtain better performance in the suppression of regulation errors, the proposed method in this paper is still meaningful. In conventional methods, the self-motion is calculated in the null space of the Jacobian matrix: pseudo-inversion is calculated to decouple the movement of the end-effector from self-motion. In this paper, with the dynamic neural network in (14), rather than calculating pseudo-inversion, the joint command is obtained in a recurrent manner that is not decoupled. The elimination of the pseudo-inversion calculation reduces the computational cost in the planning process. Furthermore, the boundedness of robot joint states is ensured, which is difficult to do with conventional methods. The snapshots during the planning process are given in Fig. 11. It can be observed that the manipulators adjust joint configuration while maintaining the position of end-effector, and the collision is avoided in realtime.

TABLE 1. Comparison of different collision-free motion planning methods.

Features	This paper	[17]	[37]	[31] [32]	[38]
Globally stable	Yes	Yes	Yes	Yes	Yes
Secondary task	Yes	No	No	Yes	No
Real-time	Yes	Yes	Yes	Yes	Yes
Physical constraints	Yes	No	No	Yes	No
Pseudo-inversion required	No	Yes	Yes	Yes	Yes
Continuous trajectory	Yes	Yes	Yes	Yes	No

C. COMPARISONS

In this part, comparisons with other motion planning methods are presented to show the superiority of the proposed collision-free planning strategy, as shown in Table 1. In [17], [37], JMPI-based motion planners were introduced in which collision is avoided in the null space. In [37], the JMPI method was improved by introducing a damped least-squares algorithm to solve the singularity problem. In [31], [32],

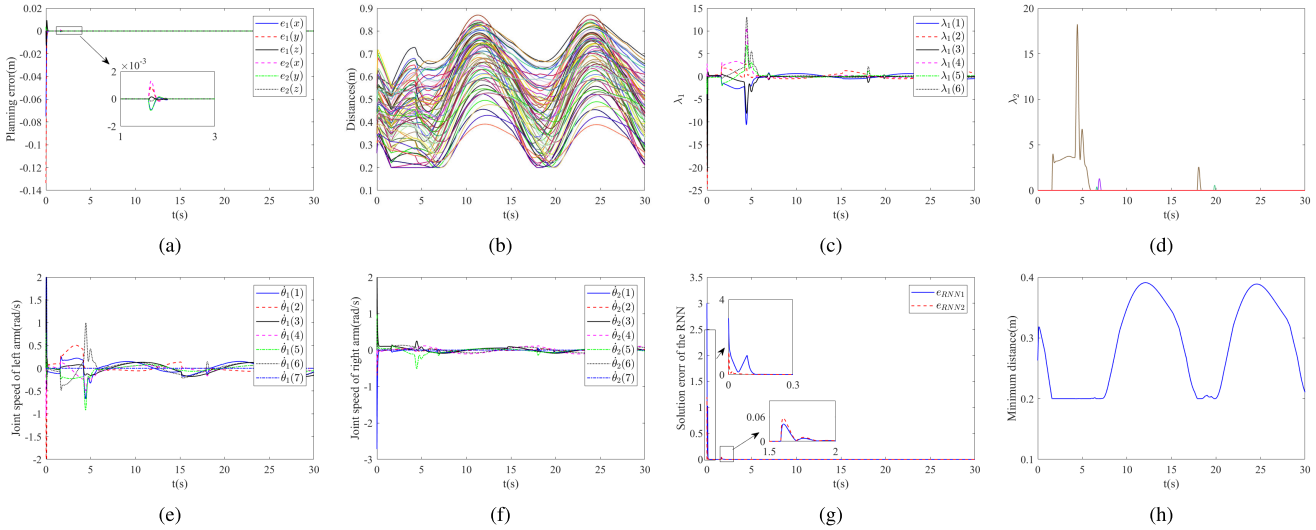


FIGURE 9. Numerical results of the motion planning method for time-varying trajectories. (a) Profile of planning errors. (b) Profile of distances between the critical points. (c) Profile of state variable λ_1. (d) Profile of state variable λ_2. (e) Profile of joint velocities of left arm. (f) Profile of joint velocities of right arm. (g) Profile of solution errors of the established neural network. (h) Profile of minimum distance between left and right arm.

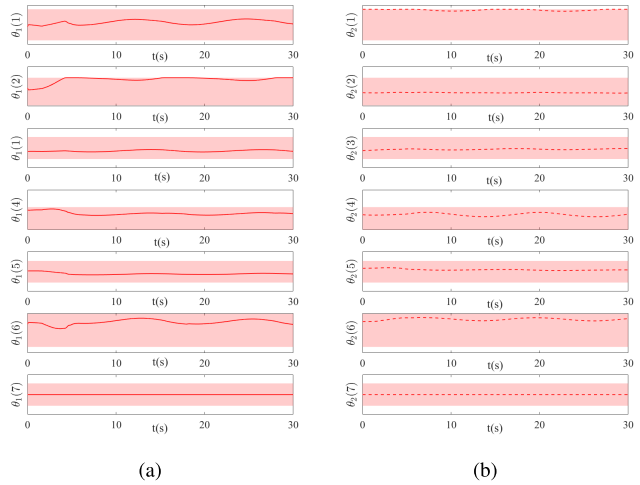


FIGURE 10. Profile of joint angles and the corresponding accessible range(red filled part). (a) Left arm. (b) Right arm.

FIGURE 11. Snapshots of Baxter when tracking time-varying trajectories. (a) $t = 0$s. (b) $t = 5$s. (c) $t = 10$s. (d) $t = 15$s.

optimization-based motion planners were designed for single robots, and they are able handle physical constraints. In [38], a configuration space-based motion planning method was developed. This method uses a reachable manifold and contact manifold to realize collision avoidance. However, the research mainly focused on point-to-point collision-free planning. In this paper, the motion planning problem for dual manipulator systems with continuous trajectories in Cartesian space and physical constraints are taken into consideration. It is noteworthy that, compared with the JMPI-based methods, the proposed scheme does not require the pseudo-inversion calculation, which could enhance real-time performance.

IV. CONCLUSION

This paper proposes a novel RNN-based motion planning strategy to avoid collisions for dual manipulator systems in real time. An outer-loop controller was built as an equality constraint, and the physical constraints are described as inequality constraints. Consequently, the collision avoidance can be defined by the inequality constraints using the envelope model-based collision detection methods, and the constraints can be reformulated in the speed level. A QP-type problem is obtained to describe the implicit solution of the motion planning problem through constraint optimization. Finally, a dynamic neural network is used to optimize the solution. This work is a significant extension of the RNN-based framework for robotic systems, and it can handle both collision avoidance and physical constraints in real time.

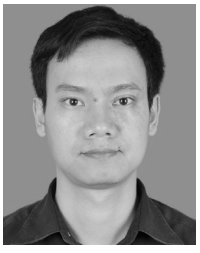
REFERENCES

[1] 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.

[2] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Trans Ind. Informat.*, vol. 15, no. 2, pp. 1153–1161, Feb. 2019.

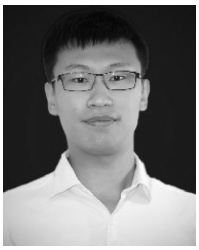
[3] Z. Zhang, Y. Lin, S. Li, Y. Li, Z. Yu, and Y. Luo, "Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1345–1357, Jul. 2018.

[4] Z. Zhang, S. Chen, and S. Li, "Compatible convex–nonconvex constrained QP-based dual neural networks for motion planning of redundant robot manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1250–1258, May 2019.
[5] S. Dalibard, A. El Khoury, F. Lamiraux, A. Nakhaei, M. Taïx, and J.-P. Laumond, "Dynamic walking and whole-body motion planning for humanoid robots: An integrated approach," *Int. J. Robot. Res.*, vol. 32, nos. 9–10, pp. 1089–1103, Aug. 2013.
[6] Y. Pan, X. Li, H. Wang, and H. Yu, "Continuous sliding mode control of compliant robot arms: A singularly perturbed approach," *Mechatronics*, vol. 52, pp. 127–134, Jun. 2018.
[7] Y. Pan, H. Wang, X. Li, and H. Yu, "Adaptive command-filtered backstepping control of robot arms with compliant actuators," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 3, pp. 1149–1156, May 2018.
[8] Y. Zhang, X. Lv, Z. Li, Z. Yang, and K. Chen, "Repetitive motion planning of PA10 robot arm subject to joint physical limits and using LVI-based primal dual neural network," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 9, pp. 475–485, 2008.
[9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, pp. 90–98, Mar. 1986.
[10] A. Csiszar, M. Drust, T. Dietz, A. Verl, and C. Brisan, "Dynamic and interactive path planning and collision avoidance for an industrial robot using artificial potential field based method," in *Mechatronics*, vol. 1. Berlin, Germany: Springer, 2011, pp. 413–421.
[11] A. Badawy, "Dual-well potential field function for articulated manipulator trajectory planning," *Alexandria Eng. J.*, vol. 55, no. 2, pp. 1235–1241, Jun. 2016.
[12] C.-H. Tsai, J.-S. Lee, and J.-H. Chuang, "Path planning of 3-D objects using a new workspace model," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 3, pp. 405–410, Aug. 2001.
[13] T. Tsuji, Y. Tanaka, P. G. Morasso, V. Sanguineti, and M. Kaneko, "Bio-mimetic trajectory generation of robots via artificial potential field with time base generator," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 4, pp. 426–439, Nov. 2002.
[14] G. Wen, S. S. Ge, F. Tu, and Y. S. Choo, "Artificial potential-based adaptive H_∞ synchronized tracking control for accommodation vessel," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5640–5647, Jul. 2017.
[15] D. Chen, Y. Zhang, and S. Li, "Tracking control of robot manipulators with unknown models: A Jacobian-matrix-adaption method," *IEEE Trans Ind. Informat.*, vol. 14, no. 7, pp. 3044–3053, Jul. 2018.
[16] L. Jin, Y. Zhang, S. Li, and Y. Zhang, "Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 6978–6988, Nov. 2016.
[17] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, "Task-oriented motion planning for multi-arm robotic systems," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 5, pp. 569–582, Oct. 2012.
[18] K. Tchoń and J. Ratajczak, "Dynamically consistent Jacobian inverse for non-holonomic robotic systems," *Nonlinear Dyn.*, vol. 85, no. 1, pp. 107–122, Jul. 2016.
[19] Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, and D. Huang, "Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties," *Neurocomputing*, vol. 329, pp. 255–266, Feb. 2019.
[20] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 34, no. 5, pp. 2126–2132, Oct. 2004.
[21] S. Li, Y. 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.
[22] B. Cai and Y. Zhang, "Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and Zhang's neural-dynamic methods," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3146–3155, Aug. 2012.
[23] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, Jun. 2017.
[24] Z. Xu, S. Li, X. Zhou, and T. Cheng, "Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties," *Neurocomputing*, vol. 357, pp. 271–281, Sep. 2019.
[25] Z. Xu, S. Li, X. Zhou, S. Zhou, and T. Cheng, "Dynamic neural networks for motion-force control of redundant manipulators: An optimization perspective," *IEEE Trans. Ind. Electron.*, early access, doi: 10.1109/TIE.2020.2970635.
[26] 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.
[27] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 3, pp. 574–588, Mar. 2019.
[28] P. S. Stanimirović and M. D. Petković, "Gradient neural dynamics for solving matrix equations and their applications," *Neurocomputing*, vol. 306, pp. 200–212, Sep. 2018.
[29] P. Stanimirovic, V. Katsikis, and S. Li, "Integration enhanced and noise tolerant ZNN for computing various expressions involving outer inverses," *Neurocomputing*, vol. 329, pp. 129–143, Feb. 2019.
[30] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 34, no. 1, pp. 752–759, Feb. 2004.
[31] D. Guo and Y. Zhang, "A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1326–1340, Nov. 2012.
[32] Z. Xu, X. Zhou, and S. Li, "Deep recurrent neural networks based obstacle avoidance control for redundant manipulators," *Frontiers Neurorobotics*, vol. 13, pp. 1–13, Jul. 2019.
[33] X.-B. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 426–431, Mar. 2003.
[34] L. Yu, J. Huang, and S. Fei, "Robust switching control of the direct-drive servo control systems based on disturbance observer for switching gain reduction," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 8, pp. 1366–1370, Aug. 2019.
[35] L. Yu, S. Fei, L. Sun, J. Huang, and G. Yang, "Design of robust adaptive neural switching controller for robotic manipulators with uncertainty and disturbances," *J. Intell. Robot. Syst.*, vol. 77, nos. 3–4, pp. 571–581, Mar. 2015.
[36] L. Yu, C. Li, and S. Fei, "Any-wall touch control system with switching filter based on 3-D sensor," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4697–4703, Jun. 2018.
[37] W. Xu, H. Liu, C. Li, J. Zhang, and B. Liang, "Autonomous path planning of dual-arm space robot for capturing moving target," *Robot*, vol. 34, no. 6, pp. 704–714, 2012.
[38] Y. Fei, D. Fuqiang, and Z. Xifang, "Collision-free motion planning of dual-arm reconfigurable robots," *Robot. Comput.-Integr. Manuf.*, vol. 20, no. 4, pp. 351–357, Aug. 2004.
[39] J. Na, X. Ren, and D. Zheng, "Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 370–382, Mar. 2013.
[40] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, "Robust adaptive finite-time parameter estimation and control for robotic systems," *Int. J. Robust Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, Nov. 2015.
[41] Y.-J. Liu, J. Li, S. Tong, and C. L. P. Chen, "Neural network control-based adaptive learning design for nonlinear systems with full-state constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1562–1571, Jul. 2016.



JINGLUN LIANG was born in Meizhou, China, in 1985. He received the B.E. degree in mechatronic engineering and the Ph.D. degree in mechanical manufacturing and automation from the South China University of Technology, Guangzhou, China, in 2008 and 2013, respectively.

From 2008 to 2013, he was a Research Assistant with the Guangdong Key Laboratory of Precision Equipment and Manufacturing Technique, Guangzhou. He is currently a Lecturer with the School of Mechanical Engineering, Dongguan University of Technology, Dongguan, China.



ZHIHAO XU received the B.E. and Ph.D. degrees from the School of Automation, Nanjing University of Science and Technology, Nanjing, China, in 2010 and 2016, respectively.

He is currently a Researcher with the Robotics Team, Guangdong Institute of Intelligent Manufacturing, Guangzhou, China. His main research interests include neural networks, force control, and intelligent information processing.

XUEFENG ZHOU received the M.E. and Ph.D. degrees in mechanical engineering from the South China University of Technology, Guangzhou, China, in 2006 and 2011, respectively.

He is currently the Director of the Robotics Team, Guangdong Institute of Intelligent Manufacturing, Guangzhou. His research interests include neural networks, force control, and intelligent information processing. He received the Best Student Paper Award at the 2010 IEEE International Conference on Robotics and Biomimetics.

SHUAI LI received the B.E. degree in precision mechanical engineering from the Hefei University of Technology, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, China, in 2008, and the Ph.D. degree in electrical and computer engineering from the Stevens Institute of Technology, USA, in 2014.

He is currently an Associate Professor with the School of Engineering, Swansea University, Swansea, U.K. His current research interests include dynamic neural networks, robotic networks, and other dynamic problems defined on a graph.

GUOLIANG YE received the M.S. degree in information and intelligence engineering from the University of Liverpool, U.K., in 2002, and the Ph.D. degree in information and biomedical engineering from the University of Oxford, U.K., in 2008.

From 2005 to 2008, he was a Research Assistant with the Department of Engineering Science, University of Oxford. From 2009 to 2012, he was a Postdoctoral Research Associate with the Department of Engineering, University of Cambridge. From 2012 to 2014, he was a Research Fellow and a Project Leader with the Brunel Innovation Centre, Brunel University joint with The Welding Institute, Cambridge, U.K. He is currently a Full Professor with the School of Mechanical Engineering, Dongguan University of Technology, Dongguan, China.