Mutual-Collision-Avoidance Scheme Synthesized by Neural Networks for Dual Redundant Robot Manipulators Executing Cooperative Tasks

Zhijun Zhang, *Senior Member, IEEE*, Lunan Zheng, *Student Member, IEEE*, Zhuoming Chen, Lingdong Kong, *Student Member, IEEE*, and Hamid Reza Karimi, *Senior Member, IEEE*

***Abstract*— Collision between dual robot manipulators during working process will lead to task failure and even robot damage. To avoid mutual collision of dual robot manipulators while doing collaboration tasks, a novel recurrent neural network (RNN)-based mutual-collision-avoidance (MCA) scheme for solving the motion planning problem of dual manipulators is proposed and exploited. Because of the high accuracy and low computation complexity, the linear variational inequality-based primal–dual neural network is used to solve the proposed scheme. The proposed scheme is applied to the collaboration trajectory tracking and cup-stacking tasks, and shows its effectiveness for avoiding collision between the dual robot manipulators. Through network iteration and online learning, the dual robot manipulators will learn the ability of MCA. Moreover, a line-segment-based distance measure algorithm is proposed to calculate the minimum distance between the dual manipulators. If the computed minimum distance is less than the first safe-related distance threshold, a speed brake operation is executed and guarantees that the robot cannot exceed the second safe-related distance threshold. Furthermore, the proposed MCA strategy is formulated as a standard quadratic programming problem, which is further solved by an RNN. Computer simulations and a real dual robot experiment further verify the effectiveness, accuracy, and physical realizability of the RNN-based MCA scheme when manipulators cooperatively execute the end-effector tasks.**

***Index Terms*— Collision avoidance, cooperative task, dual robot manipulators, motion planning, quadratic programming (QP), recurrent neural network (RNN).**

I. Introduction

IN RECENT years, robot manipulators have been widely studied and exploited in different fields, such as mind control [1], humanoid robots [2], and medical instrument [3], [4]. The reason why a robot manipulator can attract so many attentions is that it is capable of carrying heavy objects, doing repetitive and precise works, and working in severe environments. To meet complex requirements of desired tasks, more flexible configurations and optional optimizations are needed. Therefore, redundant robot manipulators equipped with more degrees of freedom (termed DOFs) cater to the demands of more complex tasks than the nonredundant robot manipulators. With more DOFs, the robot manipulator can finish the fundamental task ideally with better optional optimizations and accomplish extra but necessary objectives, such as joint-limit avoidance [5] and position posture control [6]. In addition, dual robot manipulators are becoming more and more popular, because dual robot manipulators can finish cooperative tasks [7], [8] such as working in an automated order-picking system [9] and assisting elderly as well as disabled people to get dressed [10].

However, some challenging problems always exist in the fields of operating dual redundant robot manipulators:

1) *Redundancy Resolution Problem:* With more DOFs, a fundamental problem of redundant manipulators, i.e., inverse-kinematic problem, should be considered first. Due to the nonlinearity of the forward kinematic equation, it is quite difficult to generate the joint trajectories to the desired path in real time. In addition, there exist innumerable solutions to joint trajectories, because the number of the DOFs is more than that of the work space dimensions. A traditional way to solve the fundamental problem and obtain a specific minimum-norm solution is the pseudoinverse method [11]. However, the pseudoinverse method

Manuscript received March 13, 2019; revised August 3, 2019 and March 1, 2020; accepted March 5, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002505, in part by the National Natural Science Foundation of China under Grant 61976096, Grant 61603142, and Grant 61633010, in part by the Guangdong Foundation for Distinguished Young Scholars under Grant 2017A030306009, in part by the Guangdong Special Support Program under Grant 2017TQ04X475, in part by the Science and Technology Program of Guangzhou under Grant 201707010225, in part by the Fundamental Research Funds for Central Universities under Grant 2017MS049, in part by the Scientific Research Starting Foundation of South China University of Technology, in part by the National Key Basic Research Program of China (973 Program) under Grant 2015CB351703, in part by the Italian Ministry of Education, University and Research under the Project "Department of Excellence LIS4.0—Lightweight and Smart Structures for Industry 4.0," in part by the Guangdong Key Research and Development Program under Grant 2018B030339001, and in part by Guangdong Natural Science Foundation Research Team Program 1414060000024. *(Corresponding author: Zhijun Zhang.)*

Zhijun Zhang, Lunan Zheng, and Zhuoming Chen are with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: drzhangzhijun@gmail.com).

Lingdong Kong is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ldkong@ieee.org).

Hamid Reza Karimi is with the Department of Mechanical Engineering, Politecnico di Milano, 20156 Milan, Italy (e-mail: hamidreza.karimi@polimi.it).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2980038

has an obvious disadvantage that it will cost a lot of time to compute the inverse of matrices. In addition, the pseudoinverse method does not contain the inequality constraint for solving the secondary subtasks. Because of the advantages of parallel computation and easy realization, recurrent neural network (RNN) methods for solving the inverse-kinematic problem have attracted numerous researchers' and engineers' interests [12]–[16]. Recently, quadratic programming (termed QP)-based RNNs are widely studied for the convenience of designing the extra optimal criteria as the optimal objective and equality/inequality constraints [15], [17]–[20]. The linear variational inequality-based primal–dual neural network (LVI-PDNN) [13] is a kind of QP-based RNN, which can be applied to solve the inverse-kinematic problems.

2) *Collaboration Task Execution of Dual Robot Manipulators:* The dual redundant robot manipulators are usually expected to finish the end-effector task cooperatively. The method to control the dual robot manipulators simultaneously should be considered. Zhang *et al.* [18] show that an optimization-based method is feasible.
3) *Real-Time Mutual-Collision-Avoidance (MCA) Algorithm:* When the dual robot manipulators are executing an end-effector task, a problem that cannot be negligible is the mutual-collision problem. Once the mutual collision happens, the robot manipulators will be forced to deviate from the desired trajectories and this will also lead to the task failure. Moreover, it may cause serious damage to the vulnerable robot manipulators.

To avoid such a serious case, an MCA scheme (see Fig. 1) is necessary. By retrieving the literatures, we find that there is very few or even no scheme considering such a mutual-collision problem. The MCA problem has the following advantages: 1) MCA can avoid the collision of dual robots when working cooperatively; 2) it is convenient to incorporate the secondary optimal criteria into the proposed QP-based MCA scheme; 3) MCA can reduce the needless sensors to measure the position information of the robot manipulators, because the precise models of the manipulators have been introduced into the MCA scheme; 4) because all the robot formulations and criteria have been taken into one scheme, some global optimization criterion can be considered, e.g., minimizing the global velocity norm and maximizing the global manipulability; and 5) the MCA scheme can be extended to multiple robots case.

Because of the above advantages, an MCA scheme is proposed. It can solve the redundancy resolution problem, control the dual manipulators to work simultaneously, and avoid the mutual collision of dual robot manipulators. Due to the effectiveness of the RNN method, an LVI-PDNN is used to solve the QP-based MCA scheme. The main contributions of this article are described as follows.

1) By converting the MCA problem to form the attractor of the LVI-PDNN and achieving neural network online learning, a novel MCA scheme synthesized by LVI-PDNN of dual redundant robot manipulators to achieve cooperative work is proposed and discussed.
2) The theoretical joint angles of dual manipulators can be solved by an LVI-PDNN-based MCA scheme and used to achieve MCA. Through online learning, the manipulators will obtain the ability of collision avoidance.
3) It is the first time to propose and incorporate the line-segment-based distance measure (termed LSDM) into the QP-based motion planning scheme. LSDM is an iterative minimum-distance calculation method, which obtains the accurate minimum distance between two line segments. This method is different from the traditional distance measurement of the obstacle-avoidance scheme, which can only consider the distance between one line segment and a point.
4) Simulations and comparisons verify the effectiveness, accuracy, and safety of the proposed MCA scheme when dual robot manipulators cooperatively execute an end-effector task.
5) Experiment with real robots further validates the physical realizability and advantage of the proposed MCA scheme.

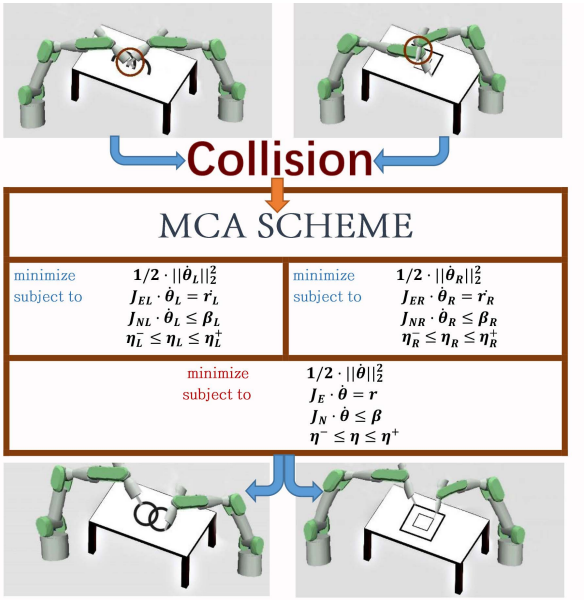


Fig. 1. Structure of an MCA scheme.

The structure of the article is illustrated as follows. Section II retrieves the related work. In Section III, a standard QP problem is formulated to achieve the MCA of the dual robot manipulators. Section IV shows the LVI-PDNN, which is used to solve the QP problem. Section V presents the comparisons between the proposed MCA scheme and some state-of-the-art methods. In Section VI, simulations and comparisons based on the dual PA10$^+$ manipulators with MATLAB and one real-robot experiment based on KINOVA JACO^2 and MICO^2 are presented to verify the correctness of the schemes. Furthermore, the MCA scheme is compared with the some state-of-the-art methods. Finally, conclusions are presented in Section VII.

II. Related Works

To overcome the potential problem that the manipulators and mobile robots may collide with the obstacles, schemes need to figure out how to avoid collisions. In recent years, many studies have been reported to avert the collisions from happening. A significant step to avoid collisions is to measure the minimum distance between the manipulators and the obstacles. The calculated minimum distance is used to achieve real-time and collision-avoidance control, so that the distance needs to be computed quickly and accurately. Some researchers calculated the distance with the relationship between two straight lines in different planes [21]. However, such a method will divide the distance calculation into ten cases and the calculation is time-consuming. Thus, an algorithm to measure the minimum distance between two line segments based on the geometric knowledge is proposed, which can obtain the results quickly.

Artificial potential field (APF) is an effective tool for robot to avoid collision [22]–[24]. Khatib *et al.* [25] proposed a real-time obstacle-avoidance method based on building an APF for manipulators and mobile robots to achieve robot operations in a complicated environment. Volpe *et al.* [26] presented a superquadric-based potential function that can achieve avoidance by generating repulsive forces from obstacles. Nevertheless, due to the inability of dealing with various shapes of obstacles and the massive computation for the complex situations with 3-D obstacles, these APF methods are more suitable for mobile robots and nonredundant manipulators than redundant manipulators.

Under such a situation, Zhang *et al.* [21] proposed a QP problem-based scheme that is combined with the minimum velocity norm, desired path tracking, as well as obstacle avoidance and then solved the QP problem with an RNN. With this scheme, it is able to avoid the massive computation burden caused by the previous pseudoinverse method. In addition, Guo *et al.* [27] improved the scheme by modifying some parameters in the obstacle-avoidance constraint to decline further the sudden saltation of velocities. Furthermore, some effective collision-avoidance schemes have been proposed by some researchers and engineers [28]–[30]. Sun *et al.* [28] first presented a fast method for estimating the probability of collision of a motion planner for a manipulator under the assumptions of Gaussian motion and sensing uncertainty. Oh *et al.* [29] proposed an analytic inverse kinematic solution by considering the joint-limit and the self-collision avoidance for a 7-DOF manipulator with a spherical shoulder and wrist joints. However, the above studies only focused on the single redundant manipulator.

In the studies from [31]–[35], they focused on the collision between the manipulators and the external obstacles without considering the mutual collision of the dual arms. Wang *et al.* [34] proposed a self-identification and collision-prediction method that can avoid the collision, even if the obstacle is close to the robot and, at the same time, prevent unnecessary robot movements when the obstacle is far enough from the robot. Nicolis *et al.* [35] proposed a unified real-time optimization controller for a visual-servo dual-arm teleoperation system that allows robust collision avoidance in cluttered environments. However, the above studies focused on the obstacle collision-avoidance problems of dual manipulators more than the MCA. Therefore, an MCA scheme is proposed to meet the requirements of obtaining an optimal solution that contains joint-limit avoidance, collision avoidance, and finishing the end-effector task.

III. Problem Formulation and Scheme Design

First, an inverse-kinematic model and a minimum velocity norm (MVN) formulation of dual redundant manipulators are presented. Then, MCA constraints and joint constraints are considered and integrated into QP problem for the left-hand-side robot manipulator. After that, two QPs of the left-hand-side and right-hand-side manipulators are formulated, respectively. Finally, a combined standard QP problem unified by the two QPs is proposed to achieve the MCA of the dual manipulators.

A. Traditional Minimum Velocity Norm Scheme

The forward kinematic equation of the manipulators [18] is

$$f_{\mathrm{L/R}}(\theta_{\mathrm{L/R}}) = r_{\mathrm{L/R}} \tag{1}$$

where $\theta_{\mathrm{L/R}} \in \mathbb{R}^n$ and $r_{\mathrm{L/R}} \in \mathbb{R}^m$ denote the joint angles and desired end-effector paths of the left-hand-side and right-hand-side robot manipulators, respectively, and $f_{\mathrm{L/R}}(\cdot)$ is a continuous nonlinear mapping function. The forward kinematic problem is to obtain $r_{\mathrm{L/R}}$ when $\theta_{\mathrm{L/R}}$ is known. On the contrary, the inverse kinematic problem is to obtain the joint space solutions while the desired end-effector path $r_{\mathrm{L/R}}$ is known. Due to the nonlinearity, it is difficult to obtain directly the following solutions $\theta_{\mathrm{L/R}}$, that is:

$$\theta_{\mathrm{L/R}} = f_{\mathrm{L/R}}^{-1}(r_{\mathrm{L/R}}). \tag{2}$$

A common way to solve the inverse kinematic problem is to handle it at the velocity level [18]. By differentiating the forward kinematic equation (1) with respect to time t, the following equation can be obtained, that is,

$$J_{\mathrm{L/R}}\dot{\theta}_{\mathrm{L/R}} = \dot{r}_{\mathrm{L/R}} \tag{3}$$

where $J_{\mathrm{L/R}} \in \mathbb{R}^{m\times n}$ denotes the end-effector Jacobin matrix of the left-hand-side and right-hand-side robot manipulators, which is defined as $J_{\mathrm{L/R}}=\partial f(\theta_{\mathrm{L/R}})/\partial\theta_{\mathrm{L/R}}$. The traditional approach to solve (3) is the pseudoinverse method [11], that is:

$$\dot{\theta}_{\mathrm{L/R}} = J_{\mathrm{L/R}}^{\dagger}\dot{r}_{\mathrm{L/R}} + (I - J_{\mathrm{L/R}}^{\dagger}J_{\mathrm{L/R}})\Phi \tag{4}$$

where $J_{\mathrm{L/R}}^{\dagger}$ denotes the generalized inverse (or called pseudoinverse) of $J_{\mathrm{L/R}}$ and I is a unit matrix. $I - J_{\mathrm{L/R}}^{\dagger}J_{\mathrm{L/R}}$ is the projectional operator that projects arbitrary vectors Φ onto the null space of $J_{\mathrm{L/R}}$. The term $I - J_{\mathrm{L/R}}^{\dagger}J_{\mathrm{L/R}}$ denotes the general solution to the homogeneous equation $J_{\mathrm{L/R}}\dot{\theta}_{\mathrm{L/R}} = 0$. The traditional method (4) needs to compute the pseudoinverse of the matrices, which is a time-consuming work. Recently, many researchers prefer a QP-based method. For instance,

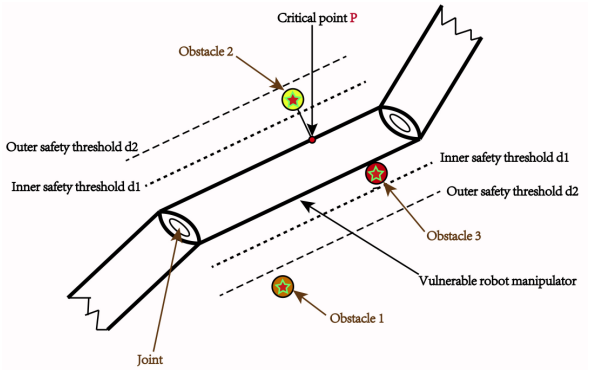


Fig. 2. Inner and outer safety thresholds of a vulnerable robot manipulator.

an MVN scheme [27] of the left-hand-side and right-hand-side robot manipulators is designed as

$$\min \ \frac{1}{2}\|\dot{\theta}_{\mathrm{L/R}}\|_2^2 \tag{5}$$

$$\text{s.t.} \ \ J_{\mathrm{L/R}}\dot{\theta}_{\mathrm{L/R}} = \dot{r}_{\mathrm{L/R}} \tag{6}$$

where $\|\cdot\|_2$ stands for the Euclidean norm.

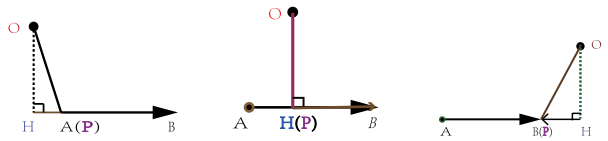
B. Collision-Avoidance Constraint

For simplicity, we take the left-hand-side robot manipulator as an example to present the collision avoidance in detail and then directly give the result of the right one due to space limitation.

As is shown in Fig. 2, assume that there are three obstacles near a vulnerable robot manipulator. In the algorithms, two safety thresholds are applied, i.e., inner safety threshold d_1 and outer safety threshold d_2, and there are three cases.

1) If the minimum value between one obstacle (like obstacle 1) and the vulnerable manipulator is larger than d_2, it means that the robot would not collide with the obstacle and is safe.
2) If the minimum value is between d_1 and d_2 (like obstacle 2), it means that the manipulator is easy to collide with the obstacle, and the robot should decelerate toward or keep off obstacle 2. Therefore, an escape velocity of the critical point P on the manipulator needs to be assigned. When the dual manipulators are working cooperatively, one manipulator needs to be treated as the obstacle versus the other manipulator. Thus, the escape velocity is designed for the vulnerable link to drive away from the critical point on the other manipulator.
3) If the minimum distance is less than the safety threshold d_1 (like obstacle 3), it means that the manipulator has already collided with the obstacle, and we need to avoid this before the minimum distance decreases to the value less than d_1.

A key step to achieve MCA is to treat one of the dual manipulators as an obstacle and measure the minimum distance between the dual manipulators accurately. A novel LSDM at geometric level is proposed and illustrated as follows.



(a) (b) (c)

Fig. 3. Different situations between point O and line AB. (a) H on the left-hand side. (b) H on the line segment. (c) H on the right-hand side.

Step 1: Assume there is a point O outside line segment $\overrightarrow{\mathrm{AB}}$, as shown in Fig. 3. Point H is the intersection point on the perpendicular line through Point O of the line $\overrightarrow{\mathrm{AB}}$.

Since the minimum distance $|\overrightarrow{\mathrm{OP}}|$ and the location of point P are the key factors in the algorithm, they should be first considered and calculated. From Fig. 3(b), we know that

$$\overrightarrow{\mathrm{OH}}^2 = \overrightarrow{\mathrm{OA}}^2 - \overrightarrow{\mathrm{AH}}^2 = \overrightarrow{\mathrm{OB}}^2 - \overrightarrow{\mathrm{HB}}^2 \tag{7}$$

$$\overrightarrow{\mathrm{AH}} = \overrightarrow{\mathrm{AB}} - \overrightarrow{\mathrm{HB}}. \tag{8}$$

Substituting (8) into (7), we can obtain

$$\overrightarrow{\mathrm{OA}}^2 - (\overrightarrow{\mathrm{AB}} - \overrightarrow{\mathrm{HB}})^2 = \overrightarrow{\mathrm{OB}}^2 - \overrightarrow{\mathrm{HB}}^2 \tag{9}$$

and then $\overrightarrow{\mathrm{HB}} = (\overrightarrow{\mathrm{OB}}^2 + \overrightarrow{\mathrm{AB}}^2 - \overrightarrow{\mathrm{OA}}^2)/(2\overrightarrow{\mathrm{AB}})$. Based on the different locations of point H, the following three cases are considered.

1) If point H locates on the left-hand side of the line segment $\overrightarrow{\mathrm{AB}}$, i.e., $\overrightarrow{\mathrm{HB}} > \overrightarrow{\mathrm{AB}}$, the minimum distance from point O to line segment $\overrightarrow{\mathrm{AB}}$ is $\overrightarrow{\mathrm{OA}}$ and the critical point P is point A.
2) If point H locates on the line segment $\overrightarrow{\mathrm{AB}}$, i.e., $0 \leqslant \overrightarrow{\mathrm{HB}} \leqslant \overrightarrow{\mathrm{AB}}$, the minimum distance from point O to line segment $\overrightarrow{\mathrm{AB}}$ is $\overrightarrow{\mathrm{OH}}$ and the critical point P is point H.
3) If point H locates on the right-hand-side of line segment $\overrightarrow{\mathrm{AB}}$, i.e., $\overrightarrow{\mathrm{HB}} < 0$, the minimum distance from point O to line segment $\overrightarrow{\mathrm{AB}}$ is $\overrightarrow{\mathrm{OB}}$ and the critical point P is point B.

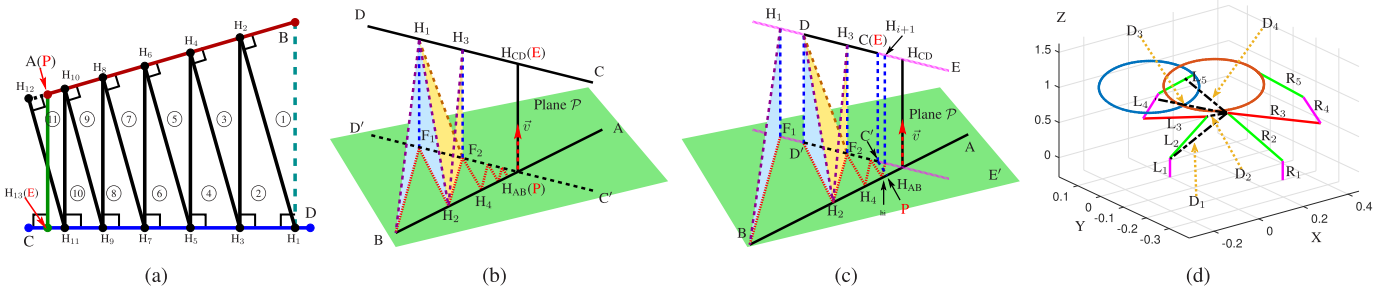
Step 2: For the left-hand-side manipulator, we define one link vector $\overrightarrow{\mathrm{AB}}$ as $\overrightarrow{\mathrm{AB}} = [x_{\overrightarrow{\mathrm{AB}}}, y_{\overrightarrow{\mathrm{AB}}}, z_{\overrightarrow{\mathrm{AB}}}]$. Similarly, for the right-hand-side manipulator, we define one link vector $\overrightarrow{\mathrm{CD}}$ as $\overrightarrow{\mathrm{CD}} = [x_{\overrightarrow{\mathrm{CD}}}, y_{\overrightarrow{\mathrm{CD}}}, z_{\overrightarrow{\mathrm{CD}}}]$.

Step 3: In order to find the minimum distance between two line segments (such as line segment $\overrightarrow{\mathrm{AB}}$ and $\overrightarrow{\mathrm{CD}}$ in Fig. 4), the main idea of the LSDM is presented.

First, select B as the initial point and then calculate the minimum distance from point B to $\overrightarrow{\mathrm{CD}}$ with the method illustrated in Step 1. After this, as shown in Fig. 4, the location of the corresponding point H_1 on $\overrightarrow{\mathrm{CD}}$ and the minimum distance between point B to $\overrightarrow{\mathrm{CD}}$ are simultaneously determined.

Then, take the critical point H_1 as another initial point and apply the method again to locate the corresponding critical point H_2 on $\overrightarrow{\mathrm{AB}}$. In addition, the updated temporary minimum distance between $\overrightarrow{\mathrm{AB}}$ and $\overrightarrow{\mathrm{CD}}$ is simultaneously obtained.

Third, apply the aforementioned method iteratively until the minimum distance stops declining. We consider the latest updated value as the minimum distance between $\overrightarrow{\mathrm{AB}}$ and $\overrightarrow{\mathrm{CD}}$.



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

Fig. 4. Minimum distance calculation. (a) Two segments are coplanar. (b) H_{AB} is within $\overrightarrow{C'D'}$. (c) H_{AB} is on the extension line of $\overrightarrow{C'D'}$. (d) Minimum distance between two manipulators.

Theorem 1: Assume that A, B, C, and D are the end points, and P and E are the critical points of the two line segments $\overrightarrow{AB}$ and $\overrightarrow{CD}$. If the LSDM method is used, the minimum distance d is found when $d = d_{Min1} = d_{Min2}$, where d_{Min1} denotes the minimum distance between point P on $\overrightarrow{AB}$ and line segment $\overrightarrow{CD}$, and d_{Min2} denotes the minimum distance between point E on $\overrightarrow{CD}$ and line segment $\overrightarrow{AB}$.

Proof: According to the spatial location relation between the dual robot manipulators, two cases should be considered.

1) When two segments are coplanar, which is shown in Fig. 4(a). Without loss of generality, taking point B on line $\overrightarrow{AB}$ as the initial point, we can create a line perpendicular to $\overrightarrow{CD}$, and the intersection is H_1. From point D, we take a line perpendicular to $\overrightarrow{AB}$, and the intersection is H_2. We can get a right-angled triangle with three sides, i.e., BD, DH_2, and H_2D. Similarly, from point H_2, we can create next right-angled triangle, and the iterative process will stop until the minimum distance d_{Min1} between point P on $\overrightarrow{AB}$ to line $\overrightarrow{CD}$ is equal to the minimum distance d_{Min2} from H_{13} to $\overrightarrow{AB}$. It means that there is no smaller right-angled triangle with the minimum distance |PE| as the hypotenuse any more. Therefore, |PE| is the minimum distance between the line segments $\overrightarrow{AB}$ and $\overrightarrow{CD}$.
2) Consider lines $\overrightarrow{AB}$ and $\overrightarrow{CD}$ in different noncoplanar planes. For calculating the minimum |PE|, the normal vector $\vec{v}$ of $\overrightarrow{AB}$ and $\overrightarrow{CD}$ is used. Move $\overrightarrow{CD}$ along the direction of $\vec{v}$ to intersect line $\overrightarrow{AB}$ with intersection H_{AB} (i.e., line $\overrightarrow{C'D'}$). The corresponding point in $\overrightarrow{CD}$ is H_{CD}. Then, two situations should be considered.
   A) If H_{AB} is within $\overrightarrow{C'D'}$: In Fig. 4(b), $\overrightarrow{C'D'}$ is parallel to $\overrightarrow{CD}$. Moreover, lines $\overrightarrow{C'D'}$ and $\overrightarrow{AB}$ form a plane $\mathcal{P}$ and $\overrightarrow{PE}$ is a vertical line perpendicular to plane $\mathcal{P}$. That is, |PE| is the minimum distance between $\overrightarrow{AB}$ and $\overrightarrow{CD}$. Similar to the coplanar situation, taking B as the initial point, create a line perpendicular to $\overrightarrow{CD}$ with intersection H_1. Furthermore, a line $\overrightarrow{F_1H_1}$, which is parallel to $\overrightarrow{PE}$ and perpendicular to $\overrightarrow{CD}$, can be found. That is, $\overrightarrow{CD}$ is perpendicular to plane BH_1F_1 and $\angle H_1F_1B = 90°$. Similarly, taking H_1 as the initial point, create a line perpendicular to $\overrightarrow{AB}$ with intersection H_2. Then, $\overrightarrow{AB}$ is perpendicular to plane $H_1H_2F_1$ and $\angle H_1F_1H_2 = 90°$. Since $\angle F_1H_2B = 90°$, according to triangle inequality, we have $|BF_1| > |F_1H_1|$. It implies that $d_{Min1} = |BH_1| > |H_1H_2| = d_{Min2}$. Repeating the above steps, through gradual iteration, d_{Min1} and d_{Min2} will converge toward |PE|, i.e., the target minimum $d = |PE| = d_{Min1} = d_{Min2}$ will be obtained finally through iteration. It is worth pointing out that if the footpoint H_i is on the extension line of $\overrightarrow{CD}$ [such as H_1 in Fig. 4(c)], then, the next starting point will be set as the end point of $\overrightarrow{CD}$ [such as D in Fig. 4(c)].
   B) If H_{AB} is on the extension line of $\overrightarrow{C'D'}$: Take Fig. 4(c) as an example. Similar to the steps in (1), a series of points H_i will be found. Repeating the iteration steps in (1), the final starting point is C (i.e., point E) and the minimum distance is CP, where point P is the footpoint at line $\overrightarrow{AB}$. Thus, the target minimum |PE| will be obtained.

■

To achieve the LSDM algorithm of the MCA scheme, the minimum distances are computed by the following progress. Take Fig. 4(d) as an example, where L_i and R_i are the links of two manipulators and D_i are the distances from R_2 to each link of the left-hand-side manipulator. First, for each link (e.g., link R_2) of a manipulator, minimum distances between this link and all the links of another manipulator (e.g., $D_1 \sim D_4$) are calculated. Second, we choose the smallest value of the minimum distances from link R_2 to the left-hand-side manipulator as the global minimum value of R_2. For example, D_2 (the minimum distance from R_2 to L_2, or from R_2 to L_3) is the smallest among $D_1 \sim D_4$. Third, the collision-avoidance constraints of all the global minimum values of each link and their critical points are incorporated into the QP-based MCA scheme.

Without loss of generality, considering line segment AB as the first link of the left-hand-side manipulator and CD as the obstacle (i.e., one of the links of the right-hand-side manipulator), the collision-avoidance constraint of critical point P_1 can be formulated as

$$\overrightarrow{r_{P_1E_1}} \diamond J_{P_1L}\dot{\theta}_L \leqslant \mathbf{0} \in \mathbb{R}^m \tag{10}$$

where $\mathbf{0}$ denotes the full-zero column vector; $\overrightarrow{r_{P_1E_1}} = [x_{E_1} - x_{P_1}, y_{E_1} - y_{P_1}, z_{E_1} - z_{P_1}]^T$ with $(x_{P_1}, y_{P_1}, z_{P_1})$ and $(x_{E_1}, y_{E_1}, y_{E_1})$ denoting locations of the critical points P_1 and E_1 in Cartesian space, respectively; operator $\diamond$ is defined as $\iota \diamond \chi =$

$[\iota_1\chi_1, \iota_2\chi_2, \ldots, \iota_u\chi_u]^{\mathrm{T}}$ with $\iota := [\iota_1, \iota_2, \ldots, \iota_u] \in \mathbb{R}^u$ as a column vector, and $\chi := [\chi_1, \chi_2, \ldots, \chi_u]^{\mathrm{T}} \in \mathbb{R}^{u\times i}$ as a matrix, where χ_i denotes the i row of the matrix χ; $J_{\mathrm{P_1L}} \in \mathbb{R}^{m\times n}$ denotes the Jacobian matrix of the critical points on the robot manipulator. In this article, since the robot has 8-DOFs, and the robot works in three-dimensional space, $n = 8$ and $m = 3$.

As for all the critical points $\mathrm{P}_1 \sim \mathrm{P}_\sigma$, the overall collision-avoidance constraint is

$$\overrightarrow{r_{\mathcal{PE}}} \diamond J_{\mathcal{P}\mathrm{L}}\dot{\theta}_{\mathrm{L}} \leqslant \mathbf{0} \in \mathbb{R}^{\sigma m} \tag{11}$$

where $\overrightarrow{r_{\mathcal{PE}}} = [\overrightarrow{r_{\mathrm{P_1E_1}}}^{\mathrm{T}}, \ldots, \overrightarrow{r_{\mathrm{P}_\sigma\mathrm{E}_\sigma}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{\sigma m}$ and $J_{\mathcal{P}\mathrm{L}} = [J_{\mathrm{P_1L}}, \ldots, J_{\mathrm{P}_\sigma\mathrm{L}}]^{\mathrm{T}} \in \mathbb{R}^{\sigma m\times n}$. Symbol σ stands for the number of the critical points on the left-hand-side vulnerable robot manipulator.

To decrease the suddenly saltation of the position and velocity of the end-effector when the manipulator moves close enough to the obstacle, the vector $\mathbf{0}$ in (11) should be replaced by a gradually changing variable parameter $\mathcal{B}_{\mathrm{L}}$

$$\mathcal{B}_{\mathrm{L}} = S(d)\max\{\overrightarrow{r_{\mathcal{PE}}} \diamond J_{\mathcal{P}\mathrm{L}}\dot{\theta}_{\mathrm{L}}|_{d=d_2}, \mathbf{0}\} \tag{12}$$

$$S(d) = \begin{cases} 1, & \text{if } d \geqslant d_2 \\ \sin^2\left(\dfrac{\pi}{2}\cdot\dfrac{d-d_1}{d_2-d_1}\right), & \text{if } d_1 < d < d_2 \\ 0, & \text{if } d \leqslant d_1 \end{cases} \tag{13}$$

where d_1 and d_2 denote the inner and outer safety thresholds.

Therefore, inequality (11) can be simply replaced by

$$J_{\mathrm{NL}}\dot{\theta}_{\mathrm{L}} \leq \mathcal{B}_{\mathrm{L}} \tag{14}$$

and the QP problem (5) and (6) of the left-hand-side manipulator becomes

$$\min\ \frac{1}{2}\|\dot{\theta}_{\mathrm{L}}\|_2^2 \tag{15}$$

$$\text{s.t. } J_{\mathrm{L}}\dot{\theta}_{\mathrm{L}} = \dot{r}_{\mathrm{L}} \tag{16}$$

$$J_{\mathrm{NL}}\dot{\theta}_{\mathrm{L}} \leq \mathcal{B}_{\mathrm{L}} \tag{17}$$

where $J_{\mathrm{NL}} = \overrightarrow{r_{\mathcal{PE}}} \diamond J_{\mathcal{P}\mathrm{L}}$ is the diamond multiplication result between $\overrightarrow{r_{\mathcal{PE}}}$ and the Jacobin matrix $J_{\mathcal{P}\mathrm{L}}$.

When the minimum distance between the dual manipulators is more than d_2, i.e., the dual robot manipulators work in the safe situation, the dual manipulators will work normally. When the minimum distance is larger than d_1 and less than d_2, if the dual manipulators are moving closer, their relative speed will decrease to zero, even to a negative value; if the dual manipulators are moving in opposite directions, their relative speed would track their original directions with the same speed. When the minimum distance is getting close enough to d_1, inequality (17) can ensure that the left-hand-side manipulator stops moving toward the right-hand-side one to avoid collision.

C. Joint-Limit-Avoidance Constraint

In practical applications, it is necessary to consider the joint limits of a robot. The joint limits can be described as

$$\theta_{\mathrm{L}}^- \leqslant \theta_{\mathrm{L}} \leqslant \theta_{\mathrm{L}}^+, \quad \dot{\theta}_{\mathrm{L}}^- \leqslant \dot{\theta}_{\mathrm{L}} \leqslant \dot{\theta}_{\mathrm{L}}^+ \tag{18}$$

where $\theta_{\mathrm{L}}^{\pm}$ and $\dot{\theta}_{\mathrm{L}}^{\pm}$ denote the upper and lower limits of the joints angles and velocities of the left-hand-side manipulator, respectively.

Since the inverse kinematic problem is solved at the velocity level [18], [27], [36], the joint angle variable needs to be represented by the joint velocities, that is,

$$\varepsilon_{\mathrm{L}}^- = \varepsilon\left(\theta_{\mathrm{L}}^- - \theta_{\mathrm{L}}\right) \leqslant \dot{\theta}_{\mathrm{L}} \leqslant \varepsilon\left(\theta_{\mathrm{L}}^+ - \theta_{\mathrm{L}}\right) = \varepsilon_{\mathrm{L}}^+ \tag{19}$$

where ε is a parameter based on the structure of the manipulator. Considering (18) and (19), the bound constraint is

$$\eta_{\mathrm{L}}^- = \max\left\{\varepsilon_{\mathrm{L}}^-, \dot{\theta}_{\mathrm{L}}^-\right\} \leqslant \dot{\theta}_{\mathrm{L}} \leqslant \min\left\{\varepsilon_{\mathrm{L}}^+, \dot{\theta}_{\mathrm{L}}^+\right\} = \eta_{\mathrm{L}}^+. \tag{20}$$

D. MCA Schemes of Left-Hand-Side and Right-Hand-Side Manipulators

Considering the mutual avoidance, joint-limit avoidance and optimization, and end-effector path tracking, the QP-based schemes of the left-hand-side and right-hand-side manipulators are formulated as

$$\min\ \frac{1}{2}\dot{\theta}_{\mathrm{L/R}}^{\mathrm{T}}\dot{\theta}_{\mathrm{L/R}} \tag{21}$$

$$\text{s.t. } J_{\mathrm{L/R}}\dot{\theta}_{\mathrm{L/R}} = \dot{r}_{\mathrm{L/R}} \tag{22}$$

$$J_{\mathrm{NL/R}}\dot{\theta}_{\mathrm{L/R}} \leqslant \mathcal{B}_{\mathrm{L/R}} \tag{23}$$

$$\eta_{\mathrm{L/R}}^- \leqslant \dot{\theta}_{\mathrm{L/R}} \leqslant \eta_{\mathrm{L/R}}^+. \tag{24}$$

Therefore, the requirements of accomplishing the end-effector task and MCA of the left-hand-side and right-hand-side manipulators have been transferred into (21)–(24).

In addition, since the spatial position relation of the dual manipulators changes all the time, the minimum distances between them need to be measured at each instant in real time. After that, we can obtain the matrices J_{NL}, J_{NR} and vectors $\mathcal{B}_{\mathrm{L}}, \mathcal{B}_{\mathrm{R}}$, which constitute (23). Moreover, other decision variables and the coefficient matrices $J_{\mathrm{L}}, J_{\mathrm{R}}, r_{\mathrm{L}}, r_{\mathrm{R}}, \eta_{\mathrm{L}}^-, \eta_{\mathrm{L}}^+, \eta_{\mathrm{R}}^-$, and η_{R}^+ can be obtained from the trajectories of the end-effectors and the joint angular limits of the manipulators.

When (21)–(24) are solved, it means that the dual manipulators can finish the end-effector task while accomplishing the subtasks including the MCA.

E. Overall Mutual-Collision-Avoidance Scheme

To formulate further a standard QP problem to adapt to the format of the QP solver, the QP problems (21)–(24) need to be integrated as the following MCA scheme:

$$\min\ \frac{1}{2}\phi^{\mathrm{T}}\phi \tag{25}$$

$$\text{s.t. } G\phi = \alpha \tag{26}$$

$$\phi^- \leqslant \phi \leqslant \phi^+ \tag{27}$$

$$U\phi \leqslant \beta \tag{28}$$

where the variables, matrices, and vectors are defined as $\phi = [\dot{\theta}_{\mathrm{L}}, \dot{\theta}_{\mathrm{R}}]^{\mathrm{T}} \in \mathbb{R}^{2n}$, $\alpha = [\dot{r}_{\mathrm{L}}, \dot{r}_{\mathrm{R}}]^{\mathrm{T}} \in \mathbb{R}^{2m}$, $\beta = [\mathcal{B}_{\mathrm{L}}, \mathcal{B}_{\mathrm{R}}]^{\mathrm{T}} \in \mathbb{R}^{2\sigma m}$, $G = [J_{\mathrm{L}}, 0; 0, J_{\mathrm{R}}] \in \mathbb{R}^{2m\times 2n}$, $U = [J_{\mathrm{NL}}, 0; 0, J_{\mathrm{NR}}] \in \mathbb{R}^{2\sigma m\times 2n}$, $\phi^- = [\eta_{\mathrm{L}}^-, \eta_{\mathrm{R}}^-]^{\mathrm{T}} \in \mathbb{R}^{2n}$, and $\phi^+ = [\eta_{\mathrm{L}}^+, \eta_{\mathrm{R}}^+]^{\mathrm{T}} \in \mathbb{R}^{2n}$.

The MCA scheme (25)–(28) would degrade to tractional MVN scheme with (25)–(27). Next, what we need to do is to find a method to solve the unified standard QP problem.

IV. Neural Network Solution

According to the previous study [36], the QP problem (25)–(28) can be converted into linear variational inequalities, which is further converted into piecewise linear projection equations

$$P_{\Omega}(\mathcal{Y}-(M\mathcal{Y}+q))-\mathcal{Y}=0 \tag{29}$$

where $P_{\Omega}(\cdot)$ is a piecewise linear projection operator with

$$P_{\Omega}(\mathcal{Y}_i)=\begin{cases}\mathcal{Y}_i^-, & \text{if } \mathcal{Y}_i<\mathcal{Y}_i^-\\ \mathcal{Y}_i, & \text{if } \mathcal{Y}_i^-\leqslant\mathcal{Y}_i\leqslant\mathcal{Y}_i^+\\ \mathcal{Y}_i^+, & \text{if } \mathcal{Y}_i>\mathcal{Y}_i^+\end{cases} \tag{30}$$

where $\mathcal{Y}=[\phi,\nu_e,\nu_i]^{\mathrm{T}}\in\mathbb{R}^h$ is the primal–dual decision vector, $h=2n+2m+2\sigma m$, $\phi\in\mathbb{R}^{2n}$ is the joint angles of the manipulators, and $\nu_e\in\mathbb{R}^{2m}$ and $\nu_i\in\mathbb{R}^{2\sigma m}$ are the dual decision vectors for constraints (26) and (28), respectively. In addition, the convex set $\boldsymbol{\Omega}$ is defined as

$$\boldsymbol{\Omega}=\{\mathcal{Y}\in\mathbb{R}^h|\mathcal{Y}^-\leqslant\mathcal{Y}\leqslant\mathcal{Y}^+\} \tag{31}$$

where $\mathcal{Y}^-=[\phi^-,-\varpi\cdot\mathbf{1}_v,\mathbf{0}\cdot\mathbf{1}_u]$ and $\mathcal{Y}^+=[\phi^+,\varpi\cdot\mathbf{1}_v,\varpi\cdot\mathbf{1}_u]$ with ϖ denoting a huge number that can replace the $+\infty$ numerically. $\mathbf{1}_v=[1,1,\ldots,1]^{\mathrm{T}}\in\mathbb{R}^{2m}$ and $\mathbf{1}_u=[1,1,\ldots,1]^{\mathrm{T}}\in\mathbb{R}^{2\sigma m}$. In addition, $M\in\mathbb{R}^{h\times h}$ and $q\in\mathbb{R}^h$ are defined as

$$M=\begin{bmatrix}I & -G^{\mathrm{T}} & U^{\mathrm{T}}\\ G & 0 & 0\\ -U & 0 & 0\end{bmatrix},\quad\text{and}\quad q=\begin{bmatrix}\tau\\ -\alpha\\ \beta\end{bmatrix}$$

where $\tau\in\mathbb{R}^{2n\times 1}$ is a zero matrix. Other vectors and matrices are defined in Section III.

Then, the PLPE problem can be solved by the LVI-PDNN

$$\dot{\mathcal{Y}}=\lambda(I+M^{\mathrm{T}})(P_{\Omega}(\mathcal{Y}-(M\mathcal{Y}+q))-\mathcal{Y}) \tag{32}$$

where λ is a parameter related to the convergence rate.

Remark 1: LVI-PDNN is a kind of RNN originating from the Hopfield neural network. Through designing the attractor and the dynamic system, this kind of RNN has associative memory ability. The target of this kind of neural network is to actuate the neural output from the initial position to a target position and achieve network learning. Through network iteration, the network output will converge to the theoretical state and find the optimal solution. Then, the robot will obtain the design abilities (e.g., collision avoidance).

Take the motion planning problem in this article as an example. The target of the task is to find a time-varying joint-angle series signal, which can solve the mutual-collision problem, i.e., to find the theoretical solution to (29). Then, setting the solution of (32) as the attractor, the LVI-PDNN (32) will be built. Through online training, the LVI-PDNN is stable and the output will converge to the theoretical solution. Furthermore, both the solution $\mathcal{Y}$ and the weight matrix M will converge to the theoretical joint-angle series signal and the target time-varying weight matrix, respectively. Finally, the solution $\mathcal{Y}$ can achieve the MCA.

To illustrate the RNN clearly, the standard RNN form of the LVI-PDNN is written as follows.

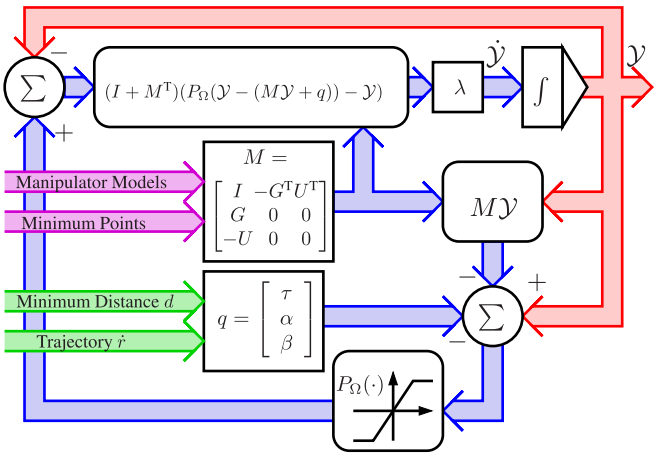


Fig. 5. Block diagram realization of the LVI-PDNN-based MCA scheme.

Consider the general equations of the RNN

$$\begin{cases} s_t=F_1(W_1\ s_{t-1}+W_2\mathfrak{x}_{t-1}) & (33)\\ o_t=F_2(W_3\ s_t+W_4\mathfrak{x}_{t-1}) & (34)\end{cases}$$

where s_t is the state of the hidden neural layer at time t, o_t is the output of the RNN at time t, and $\mathfrak{x}_t$ is the input of the RNN at time t. $W_1\sim W_4$ represent the weights of the RNN. Considering the differential of (33), the above general equations (33) and (34) can be rewritten as

$$\begin{cases} \rho\dot{s}_t=s_{t+1}-s_t=F_1(W_1\ s_t+W_2\mathfrak{x}_t)-s_t & (35)\\ o_t=F_2(W_3\ s_t+W_4\mathfrak{x}_{t-1}) & (36)\end{cases}$$

where ρ is a proportion matrix. If the weights are defined as $W_1=(I-M)$, $W_2=I$, $W_3=I$, and $W_4=\mathbf{0}$ as well as $s_t=\mathcal{Y}$, $\mathfrak{x}_t=q$, and $\rho=1/\lambda(I+M^{\mathrm{T}})$, the general RNN equations (35) and (36) with $F_1(\cdot)=P_{\Omega}(\cdot)$ and $F_2(a)=a$ can be rewritten as

$$\begin{cases} \dot{\mathcal{Y}}=\lambda(I+M^{\mathrm{T}})(P_{\Omega}(\mathcal{Y}-(M\mathcal{Y}+q))-\mathcal{Y}) & (37)\\ o_t=\mathcal{Y} & (38)\end{cases}$$

that is, the LVI-PDNN satisfied the general equation of the RNN.

The block diagram realization of the LVI-PDNN (32) is presented in Fig. 5. Through splitting the matrix equation of the LVI-PDNN, the ith neuron ($i=1,\ldots,k$) equation of the LVI-PDNN can be rewritten as

$$\dot{\mathcal{Y}}_i=\lambda\sum_{j=1}^{h}\zeta_{ij}\left(p_j\left(\sum_{k=1}^{h}\psi_{jk}\mathcal{Y}_k+q_j\right)-\mathcal{Y}_j\right) \tag{39}$$

where $\mathcal{Y}_i$ is the ith element of $\mathcal{Y}$, q_j is the jth element of q, ζ_{ij} denotes the (i,j)th element of $(I+M^{\mathrm{T}})$, ψ_{ij} denotes the (i,j)th element of $(I-M^{\mathrm{T}})$, and $p_i(\cdot)$ is the ith subfunction of $P_{\Omega}(\cdot)$. Thus, according to (39), a more detailed neural network architecture of the LVI-PDNN is shown in Fig. 6. In Fig. 6, neurons $\mathfrak{A}_1\sim\mathfrak{A}_h$, $\mathfrak{B}_1\sim\mathfrak{B}_h$, and $\mathfrak{C}_1\sim\mathfrak{C}_h$ are the intermediate neurons of the LVI-PDNN.

As for the LVI-PDNN, because of $P_{\Omega}(\cdot)$, the bound constraint (27) can be removed, and the dimension of the LVI-PDNN (h) is equal to the sum of the dimensions of the primal–dual decision vector, the equality constraint, and

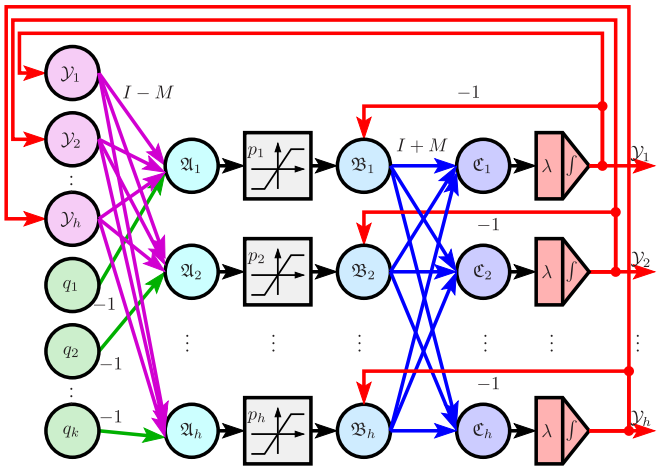


Fig. 6. Network architecture of the LVI-PDNN-based MCA scheme.

the inequality constraint of the QP problem (25)–(28) (i.e., $h = 2n + 2m + 2\sigma m$). As for practical realization, this continuous LVI-PDNN can be realized by using the electronic circles. Moreover, activation function $P_\Omega(\cdot)$ can be implemented by using a limiter. Based on (39), the complexity of the scale of the network circuits is that it consists of h integrators, h limiters, $2h^2 + h$ multipliers, and $2h^2$ summers. According to [37], the structural and computational complexities of the LVI-PDNN are at least half reduced compared with the dual neural network method.

A graphical illustration including all the steps of the MCA scheme is presented in Fig. 7. The overall design process of the MCA scheme is presented as follows.

Step 1: According to the control task requirements and the practical robot manipulators, the trajectories of the end-effectors and the types of manipulators are determined.

Step 2: Based on the trajectories and types of manipulators, the mathematical model of the whole scenario is established.

Step 3: The optimization objective (minimum energy, repetitive motion, and so on), domain of definition (joints limitation), equality (control task equation), and inequality constraints (collision avoidance) for achieving control task are chosen.

Step 4: Put all the elements in Step 3 into a QP problem [see (25)–(28)].

Step 5: According to the design formula of LVI-PDNN (32), the RNN for solving the QP problem in Step 4 is formed.

Step 6: Through the LVI-PDNN in Step 5, a series of joint angles is solved. Then, put them into the practical system for achieving the practical control task.

V. Compared With the State-of-the-Art Methods

A. Distance Calculation

To illustrate the advantages and disadvantages of the LSDM method, a different plane straight line (DPSL) method [21] is used for comparison. Since both the methods can achieve collision avoidance, comparison criteria change from task realization to calculation error, runtime, and code complexity.

1) Calculation Error: Since the DPSL makes full use of the position relationship between two straight lines in different planes, the error between the calculation result and the real distance is relatively smaller than LSDM. LSDM is an iteration method, and the calculation will stop if the stopping criteria (e.g., the difference between two adjacent minimum distances is smaller than 10^{-5} m) have been satisfied.

2) Runtime: According to [21] and [38], DPSL should calculate the parameters by using the inversion of matrix. Furthermore, ten situations should be considered and some of them demand to recompute the footpoint for achieving further control. These procedures are time-consuming. Correspondingly, the LSDM changes the straight lines in different plane problems to a point-to-line problem. Although the LSDM should iterate many times, it needs less runtime in most cases.

3) Code Complexity: The programming of the DPSL is more complicated than the LSDM due to the consideration of the difference position relationship between the straight lines in different planes. However, the LSDM makes full use of the point-to-line situation and the programming is clearer. Each iteration procedure can call the same function implementation.

B. Mutual-Collision-Avoidance Scheme

1) Compared With MVN Scheme: Actually, MVN do not consider the collision avoidance between the dual manuscripts. When facing the collision-avoidance problem, MVN is not effective enough.

2) Compared With Artificial Potential Field: Comparisons between the MCA scheme and the APF method [22]–[24] for controlling the robot manipulators are presented.

1) Attractive Velocity: According to the APF method [22], we define the attractive velocity based on the target position

$$\mathcal{V}_{\mathrm{P}} = \delta_{\mathrm{P}} \frac{\overrightarrow{\mathfrak{P}_{\mathrm{E}}\mathfrak{P}_{\mathrm{G}}}}{\|\overrightarrow{\mathfrak{P}_{\mathrm{E}}\mathfrak{P}_{\mathrm{G}}}\|_2} \tag{40}$$

with $\delta_{\mathrm{P}} = K_{\mathrm{P}} e_{\mathrm{P}} + D_{\mathrm{P}} \dot{e}_{\mathrm{P}}$ and $e_{\mathrm{P}} = \|\overrightarrow{\mathfrak{P}_{\mathrm{E}}\mathfrak{P}_{\mathrm{G}}}\|_2$, where K_{P} and D_{P} are the control parameters in the position layer, and $\mathfrak{P}_{\mathrm{G}}$ and $\mathfrak{P}_{\mathrm{E}}$ are the target and end-effecter points in the Cartesian coordinates.

Since the target point of the end-effecter is time-varying, for realizing more accurate tracking control, the attractive velocity based on the target velocity is defined as

$$\mathcal{V}_{\mathrm{V}} = \delta_{\mathrm{V}} \frac{\mathfrak{S}_{\mathrm{G}} - \mathfrak{S}_{\mathrm{E}}}{\|\mathfrak{S}_{\mathrm{G}} - \mathfrak{S}_{\mathrm{E}}\|_2} \tag{41}$$

with $\delta_{\mathrm{V}} = K_{\mathrm{V}} e_{\mathrm{V}} + D_{\mathrm{V}} \dot{e}_{\mathrm{V}}$ and $e_{\mathrm{V}} = \|\mathfrak{S}_{\mathrm{G}} - \mathfrak{S}_{\mathrm{E}}\|_2$, where $\mathfrak{S}_{\mathrm{G}}$ is the target velocity, $\mathfrak{S}_{\mathrm{E}}$ is the velocity of the end-effecter, and K_{V} and D_{V} are the control parameters in the velocity layer.

The resultant attractive velocity is the weighted sum of the attractive velocities in the position and velocity layers

$$\mathcal{V}_{\mathrm{Att}} = \varrho \mathcal{V}_{\mathrm{P}} + (1 - \varrho)\mathcal{V}_{\mathrm{V}} \tag{42}$$

where $0 \le \varrho \le 1$ is a positive weighted parameter.

2) Repulsive Velocity: According to the APF design method [22], we can define the following repulsive velocity

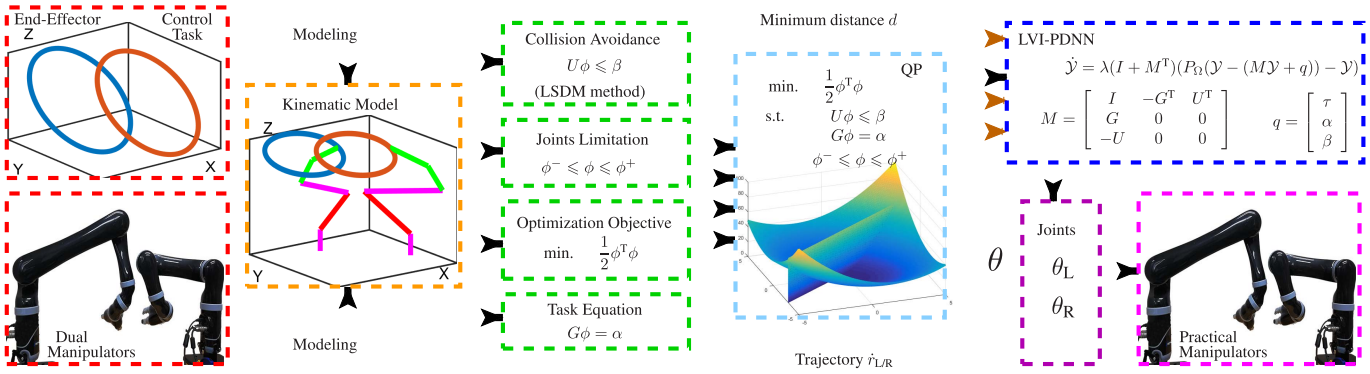


Fig. 7. Graphical illustration including all the steps of the MCA scheme.

function:

$$\mathcal{V}_{\mathrm{Rep}} = \delta_{\mathrm{Rep}} \frac{\overrightarrow{\mathfrak{P}_{\mathrm{O}}\mathfrak{P}_{\mathrm{R}}}}{\|\overrightarrow{\mathfrak{P}_{\mathrm{O}}\mathfrak{P}_{\mathrm{R}}}\|_2} \text{ with } \delta_{\mathrm{Rep}} = \frac{\upsilon_{\text{Rep-max}}}{1 + \exp\left(\frac{2d_{\min}\vartheta}{d}\right)} \tag{43}$$

where $\mathfrak{P}_{\mathrm{O}}$ and $\mathfrak{P}_{\mathrm{R}}$ are the obstacle and critical points in the Cartesian coordinates, $\upsilon_{\text{Rep-max}}$ is the maximum repulsive velocity that the manipulator can create, ϑ is a factor for adjusting the changing rate of the repulsive velocity, and $d_{\min}$ is the minimum distance between the manipulator and the obstacle.

3) Resultant Velocity: $\mathcal{V}_{\mathrm{Att}}$ is constructed from the end-effector and the $\mathcal{V}_{\mathrm{Rep}}$ is constructed from the critical point. These velocities cannot be synthesized to one velocity in the Cartesian space. Therefore, through inverse kinematics of the manipulator, we map $\mathcal{V}_{\mathrm{Att}}$ and $\mathcal{V}_{\mathrm{Rep}}$ to the joint-angle space

$$\dot{\theta}_{\mathrm{Att}} = J^{\dagger}(\mathfrak{P}_{\mathrm{E}})\mathcal{V}_{\mathrm{Att}} \tag{44}$$

$$\dot{Q}_{\mathrm{Rep}} = \sum_{i=1}^{n} J^{\dagger}(\mathfrak{P}_{\mathrm{R}}^{(i)})\mathcal{V}_{\mathrm{Rep}}^{(i)} \tag{45}$$

where $J^{\dagger}(\mathfrak{P}_{\mathrm{E}})$ is the pseudoinverse of the Jacobian matrix $J(\mathfrak{P}_{\mathrm{E}})$, and $J^{\dagger}(\mathfrak{P}_{\mathrm{R}}^{(i)})$ and $\mathcal{V}_{\mathrm{Rep}}^{(i)}$ are the pseudoinverse of the Jacobian matrix and repulsive velocity from the ith obstacle.

The resultant joint-angle velocity is

$$\dot{\theta} = \dot{\theta}_{\mathrm{Att}} + \dot{Q}_{\mathrm{Rep}}. \tag{46}$$

Thus, the joint angle θ of the robot manipulator is solved.

However, when collision avoidance is considered, the accuracy of the manipulator when using the APF method will decrease. The reason why the accuracy reduces is that the repulsive velocity mapping to the joint-angle space ($\dot{Q}_{\mathrm{Rep}}$) may conflict to the attractive velocity mapping to joint-angle space ($\dot{\theta}_{\mathrm{Att}}$). $\dot{\theta}_{\mathrm{Att}}$ and $\dot{Q}_{\mathrm{Rep}}$ are simply added in the joint-angle space. Although θ_{Att} is the solution to (44) and θ_{Rep} is the solution to (45), θ is not the solution that satisfied both (44) and (45).

As for the MCA scheme, the inverse kinematics of the manipulators and the collision-avoidance constraint are considered as the constraint conditions of a QP problem. MCA scheme is to find a solution to the QP problem in the feasible solution space satisfying all the constraint conditions.

VI. Simulations and Real Robot Experiment

In this section, dual manipulators (i.e., two PA10$^+$) are conducted to verify the correctness of the MCA scheme (25)–(28). PA10$^+$ is a manipulator equipped with a long tool of 1-DOF based on the PA10 robot arm of 7-DOFs. Thus, the DOF of a PA10$^+$ is 8, and the dimension of the end-effector in the Cartesian space is 3. Moreover, the parameter λ in LVI-PDNN (32) is set to 1×10^5. The desired paths of the dual manipulators are tracking two cross circles or a Chinese character "Hui," which means "Go home." The simulations are performed with MATLAB 2016a on an Asus Z170-AR with an Intel Core i7-6700K CPU at 4.00 GHz with 32 GB of 2400-MHz RAM.

In addition, in the real robot experiment, we use KINOVA JACO^2 and MICO^2 with 6-DOFs, which have some differences in shape. Furthermore, JACO^2 and MICO^2 are controlled by a VS2013 program by using the API provided by KINOVA company, and the hybrid programming with the M language of MATLAB and C language of VS2013 is built.

The applications of the RNN-based MCA scheme are the collaboration trajectory tracking and cup-stacking tasks in Sections VI-A–VI-E. These two applications are common tasks for robot manipulators, and they show the effectiveness of the MCA scheme for controlling the dual robot manipulators to avoid collision with each other. The collaboration trajectory tracking task is an important kind of applications to robot manipulators in many fields, such as painting, welding, and engraving. The collaboration stacking task is usually used in carrying and assembling in human daily life. As for some complex collaboration works, usage of the dual robot manipulators is necessary and avoiding mutual collision is usually a significant step. Thus, the MCA scheme is applied to the trajectory tracking and cup-stacking tasks for verifying its practicability.

A. Simulation of Drawing Cross Circles

The MCA scheme is performed on dual PA10$^+$ to draw the cross circles of radius 0.2 m and the slope angle $\pi/3$ referred to as the xy plane. We initialize the joint angles of dual PA10$^+$ as $\theta_{\mathrm{L0}} = [0, \pi/4, 0, -\pi/2, 0, \pi/4, 0, \pi/4]^{\mathrm{T}}$rad and $\theta_{\mathrm{R0}} = [-\pi/2, -\pi/4, 0, \pi/2, 0, -\pi/4, 0, -\pi/4]^{\mathrm{T}}$rad, respectively. The task-execution period T is 10 s. The safety thresholds d_1 and d_2 are set to 0.05 and 0.12 m, respectively.

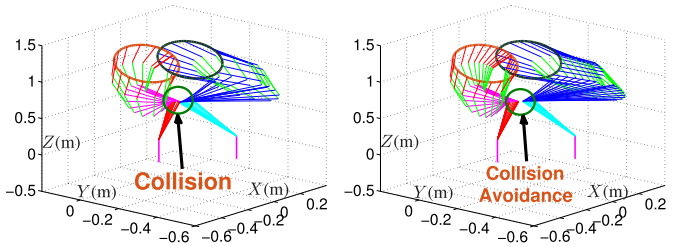


Fig. 8. Comparisons of trajectories synthesized by (a) MVN scheme (25)–(27) and (b) MCA scheme (25)–(28) of the dual manipulators.

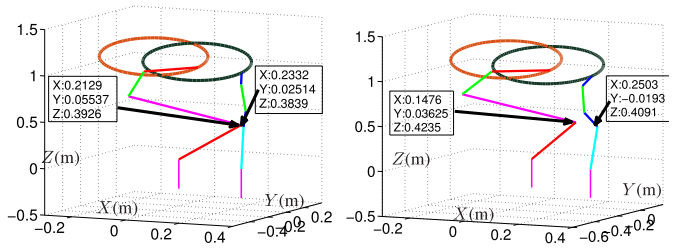


Fig. 9. Comparisons of the position of PA10$^+$ at 8.2 s synthesized by (a) MVN scheme (25)–(27) and (b) MCA scheme (25)–(28) of the dual manipulators.

TABLE I

COMPARISONS OF MINIMUM DISTANCE BETWEEN DUAL MANIPULATORS OF SAMPLING INSTANTS SYNTHESIZED BY THE MVN AND MCA SCHEMES

t	d_1	d_2	MVN scheme	MCA scheme
0s	0.05m	0.12m	0.08730m	0.08730m
2s	0.05m	0.12m	0.09563m	0.09784m
4s	0.05m	0.12m	0.10910m	0.12120m
6s	0.05m	0.12m	0.09208m	0.11540m
8s	0.05m	0.12m	0.03820m	0.11969m
10s	0.05m	0.12m	0.03900m	0.11600m

According to the physical parameters of the manipulators, the initial joint-angle state, and the equation of the target tracking trajectory, the weight matrix M and the bias vector q in (32) can be formed. Substituting M, q, and the initial values into (32) and programming the LVI-PDNN in MATLAB by M language, a series of joint angles will be obtained. These time-series signals are the theoretical joint angles for the dual robot manipulators for completing the collaboration tasks. Then, substituting these signals into the robot manipulators, the overall simulation is complete.

First, the trajectories synthesized by the MVN scheme (25)–(27) and the MCA scheme (25)–(28) of dual PA10$^+$ are shown in Fig. 8. From Fig. 8(a), two PA10$^+$ would collide if the MCA constraint is not considered, which may lead to task failure and even robot damage. Evidently, this situation is not expected in practical applications. Contrastively, after applying the MCA scheme (25)–(28), two manipulators keep a proper distance during the task execution period, as shown in Fig. 8(b).

Second, Fig. 9(a) shows the distance d at time instant 8.2 s between dual PA10$^+$ before applying the MCA scheme.

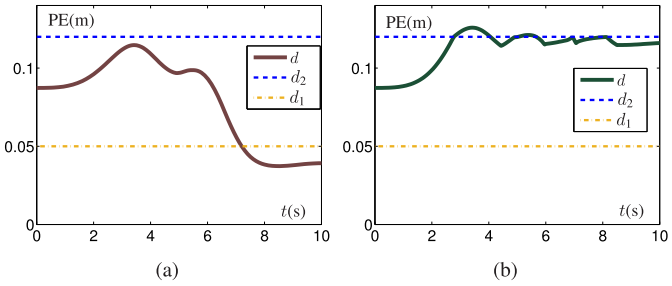


Fig. 10. Comparisons of minimum distance between dual PA10$^+$ synthesized by (a) MVN scheme (25)–(27) and (b) MCA scheme (25)–(28).

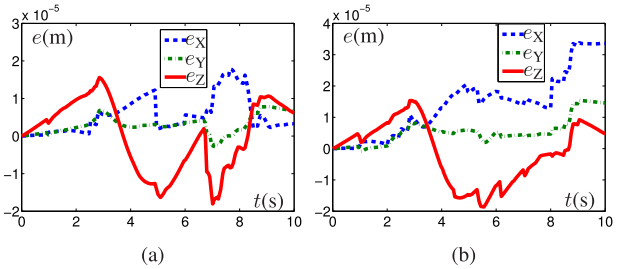


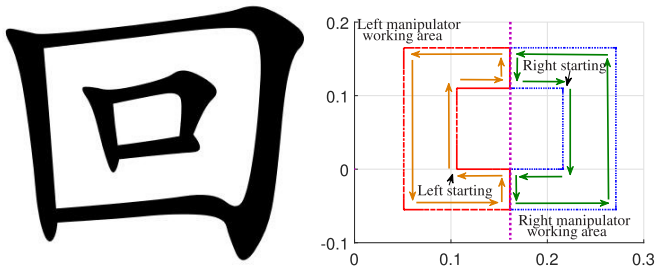
Fig. 11. Position errors synthesized by the MCA scheme (25)–(28). Position errors of (a) left and (b) right manipulators.

The result $((0.292-0.2332)^2+(0.055-0.025)^2+(0.392-0.384)^2)^{1/2}=0.03744$ m values less than d_1, which means that the collision has happened. Contrastively, the distance d after applying the MCA scheme shown in Fig. 8(b) is $((0.147-0.250)^2+(0.0363+0.0193)^2+(0.424-0.409)^2)^{1/2}=0.11765$ m, whose values are bigger than d_1. In addition, the minimum distance between dual PA10$^+$ during the task-execution period is shown in Fig. 10. The minimum distance between the dual PA10$^+$ synthesized by the MVN scheme (25)–(27) becomes less than d_1 after 7.2 s, as shown in Fig. 10(a), which means that the collision has happened. Conversely, the result of the MCA scheme (25)–(28) is much optimistic, as shown in Fig. 10(b) that the distance is larger than d_1. That is to say, the dual manipulators succeed in avoiding mutual collision. In addition, Table I shows the specific numerical comparisons of the minimum distances between dual PA10$^+$ at time instant 0 s, 2 s, 4 s, ..., 10 s. From Table I, the minimum distance between dual PA10$^+$ synthesized by the MVN scheme values is less than 0.05 m (d_1), but the result synthesized by the MCA scheme has a tendency of increasing in the first few seconds and then keeps a safety distance. The results further illustrate the effectiveness and advantage of the MCA scheme in solving the mutual-collision problem.

Third, the end-effector position error of the dual manipulators synthesized by the MCA scheme (25)–(28) is shown in Fig. 11. The error keeps within a small range of 10^{-5} m, and thus, it can ensure the accuracy while executing the cooperative tasks. This guarantees the effectiveness of the MCA scheme when apply it to track the trajectories accurately while achieving MCA.

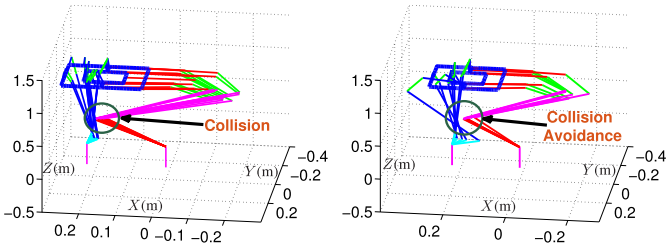
B. Simulation of Writing East Asian Word "Hui"

To illustrate further the effectiveness and wide extendibility, the MCA scheme is performed on dual PA10$^+$ to track word



(a) (b)

Fig. 12. (a) Chinese character "Hui." (b) Target trajectory.



(a) (b)

Fig. 13. Comparisons of the trajectories synthesized by (a) MVN scheme (25)–(27) and (b) MCA scheme (25)–(28) of dual PA10$^+$.

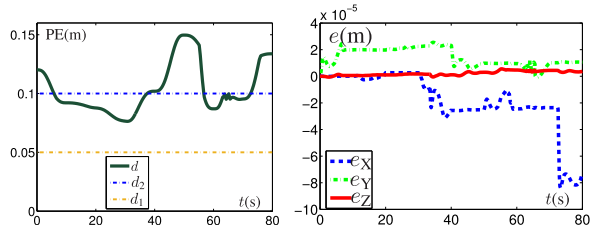
TABLE II

COMPARISONS OF MINIMUM DISTANCE BETWEEN DUAL MANIPULATORS OF SAMPLING INSTANTS SYNTHESIZED BY THE MVN AND MCA SCHEMES

t	d_1	d_2	MVN scheme	MCA scheme
0s	0.05m	0.10m	0.12010m	0.12010m
20s	0.05m	0.10m	0.07584m	0.08774m
40s	0.05m	0.10m	0.08779m	0.10200m
60s	0.05m	0.10m	0.08077m	0.08701m
80s	0.05m	0.10m	0.11370m	0.13380m

"Hui" collaboratively and the initial state of the joint angles is set similar to the previous experiment. In addition, d_1 and d_2 are set as 0.05 and 0.10 m, respectively. The picture of "Hui" is presented in Fig. 12(a). Furthermore, the target trajectories of the manipulators are presented in Fig. 12(b).

Fig. 13 shows different trajectories synthesized by the MVN and MCA schemes of dual PA10$^+$, respectively. When the MCA constraint is not considered, from Fig. 13(a), two manipulators would collide. On the contrary, the dual PA10$^+$ succeed in avoiding mutual collision while executing the task after applying the MCA scheme, as shown in Fig. 13(b). In addition, the minimum distance during the task execution process is shown in Fig. 14(a), from which we can see that it keeps larger than d_1. That is to say, the dual PA10$^+$ are able to execute the tasks safely. In addition, Table II shows the minimum distances between dual PA10$^+$ at time instants 0 s, 20 s, ..., 80 s. From Table II, the minimum distance synthesized by the MCA scheme keeps larger than d_1, and thus, it succeeds in avoiding collision. Last but not least, the tiny position error of the right-hand-side robot manipulator shown in Fig. 14(b) keeps in the range of 10^{-5} m, which can ensure the accuracy of tracking the desired path.



(a) (b)

Fig. 14. (a) Minimum distance and (b) position error of the MCA scheme (25)–(28).

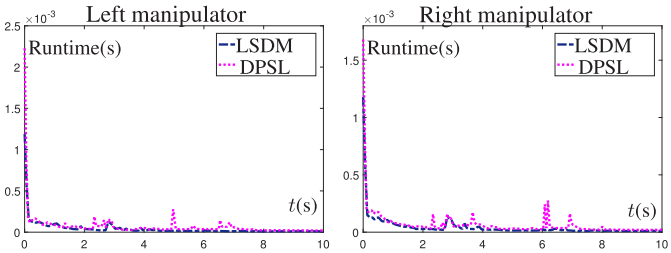


Fig. 15. Runtime of the proposed LSDM method and the DPSL method.

TABLE III

PARAMETERS OF THE APF METHOD USED IN THE SIMULATION

Parameters	K_{Pos}	D_{Pos}	K_{Vel}	D_{Vel}	ϱ	d(m)	$v_{\text{Rep-max}}$(rad/s)	ϑ
Values	100	0.1	5	0.1	0.8	0.12	0.001	1

C. Compared With the State-of-the-Art Method

1) Distance Calculation: A simulation of drawing cross circles by using two PA10$^+$ with the mentioned two distance computation methods is shown in Figs. 15 and 16. Fig. 15 shows the runtime between two methods. From Fig. 16(a), the minimum distance calculated by the DPSL is similar to the minimum distance calculated by the LSDM.

a) Calculation error: From Fig. 16, the RMSE between d_2 and the minimum distance calculated by the LSDM is 0.1130 m. The RMSE between d_2 and the minimum distance of LSDM is 0.1119 m. Furthermore, the RMSE between the minimum distances calculated by the LSDM and DPSL methods is 2.5×10^{-3} m. That is to say, the minimum distances of these methods are similar and do not interfere with the MCA scheme.

b) Runtime: Fig. 15 shows the runtime of the mentioned two methods when applying for drawing cross circles. The runtime of the LSDM is smaller than the DPSL. The root-mean-square errors (RMSEs) of the LSDM are 1.16×10^{-4} s (left) and 1.16×10^{-4} s (right), which are smaller than that of the DPSL (left: 2.04×10^{-4} s; right: 1.67×10^{-4} s).

2) Mutual-Collision-Avoidance Scheme: All the parameters used in the simulation when using the APF method are shown in Table III. The minimum distances calculated by the APF and MCA methods are shown in Fig. 16. We can see that the MCA and APF methods can achieve the control task.

The position errors of the APF method with and without considering collision avoidance are shown in Fig. 17. When the APF method without considering the collision avoidance (APF-NCA) is applied for drawing two circles, the position errors of the manipulators are within

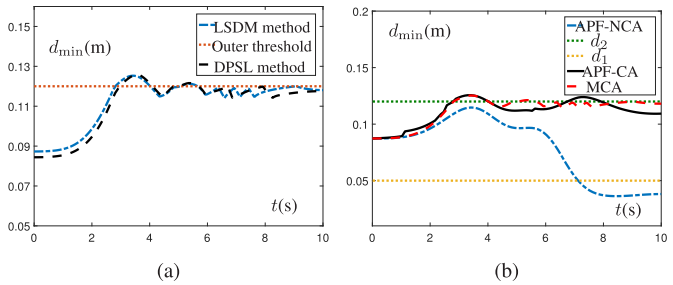


Fig. 16. Minimum distance comparisons. (a) Minimum distances calculated by the LSDM and DPSL methods. (b) Minimum distances calculated by the APF-NCA, APF-CA, and MCA methods.

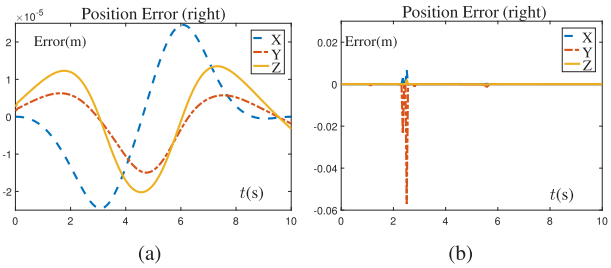


Fig. 17. Position and velocity errors between the APF-NCA and APF-CA. (a) LSDM and (b) DPSL methods.

$[-3 \times 10^{-5}, 3 \times 10^{-5}]$. The accuracy of the manipulators satisfies the requirement of the control task. However, when collision avoidance is considered, the accuracy of the manipulators decreases. Furthermore, the accuracy when using the MCA scheme will not decrease. The above results verify the analysis presented in Section V.

D. Experiment With Real Robots to Put the Cups in Order

An experiment with real robot manipulators KINOVA JACO^2 and MICO^2 is designed to demonstrate the physical realizability of the MCA scheme (25)–(28). Considering the structure of the manipulators KINOVA JACO^2 and MICO^2, we set d_1 as 0.07 m and d_2 as 0.12 m for safety. The parameter n is set as 6, since the DOFs of JACO^2 and MICO^2 are 6. Other experimental parameters are set the same as the above simulations. The arm lengths of JACO^2 and MICO^2 are 90 and 70 cm. The distance between the centers of the base of two manipulators is 78 cm. The radii of the working areas of JACO^2 and MICO^2 on the desk are 69.3 and 46.6 cm. To make it easy for the reader to understand the working area of the experiment, an equivalent illustration of the working area of the manipulators in simulation is shown in Fig. 18. The red lines denote the left-hand-side JACO^2 manipulator and the blue lines denote the right-hand-side MICO^2 manipulator. The inside of the yellow arc is the working area of JACO^2, and the inside of the green arc is the working area of MICO^2. The black dashed block is the target area of the stacking cups.

Dual manipulators, i.e., a JACO^2 and a MICO^2, are controlled to finish cooperatively a cup-stacking task. Fig. 19(a) and (b) shows the minimum distances between the two robots controlled by the MVN and MCA schemes, respectively. From Fig. 19(a), the manipulators would collide at 97.32 s without considering the MCA criterion and the minimum distance

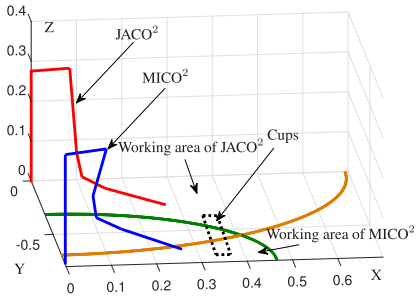


Fig. 18. Illustration of the working areas in the real-world experiment.

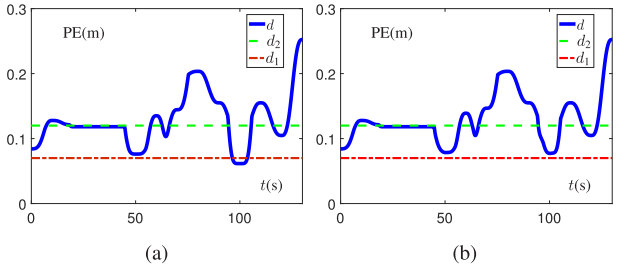


Fig. 19. Comparisons of minimum distance between JACO^2 and MICO^2 synthesized by (a) MVN scheme (25)–(27) and (b) MCA scheme (25)–(28).

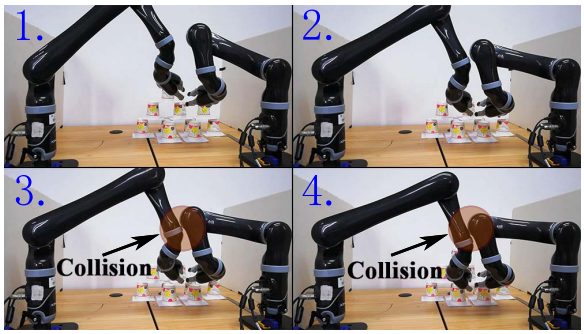


Fig. 20. Real-robot experiment of cup stacking synthesized by the MVN scheme.

decreases to 0.06153 m at 100 s. By contrast, Fig. 19(b) illustrates the effectiveness of the MCA scheme by keeping the distance in the safe distance and the global minimum distance increases to 0.07715 m at 100 s. The snapshots of the real robot working process are shown in Figs. 20 and 21. Accordingly, we can find that JACO^2 and MICO^2 collide with each other in Fig. 20 (3) and (4) and we stop the manipulators right away for safety. Obviously, from Fig. 20, we know that if the MCA criterion is not considered, JACO^2 and MICO^2 would then get closer step by step and collide finally. By contrast, since the MCA scheme is applied in Fig. 21, JACO^2 and MICO^2 cooperatively complete well the cup-stacking task. This comparative real-robot experiment verifies the effectiveness and safety of the MCA scheme synthesized by LVI-PDNN when dual redundant manipulators execute the cooperative tasks.

In summary, the above simulations and one experiment with real robots validate the physical realizability, effectiveness, and accuracy of the MCA scheme (25)–(28) for MCA while completing the end-effector task.

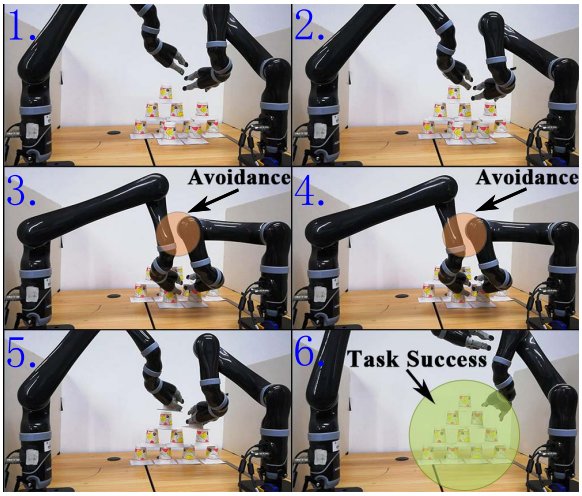


Fig. 21. Real-robot experiment of cup stacking synthesized by the MCA scheme.

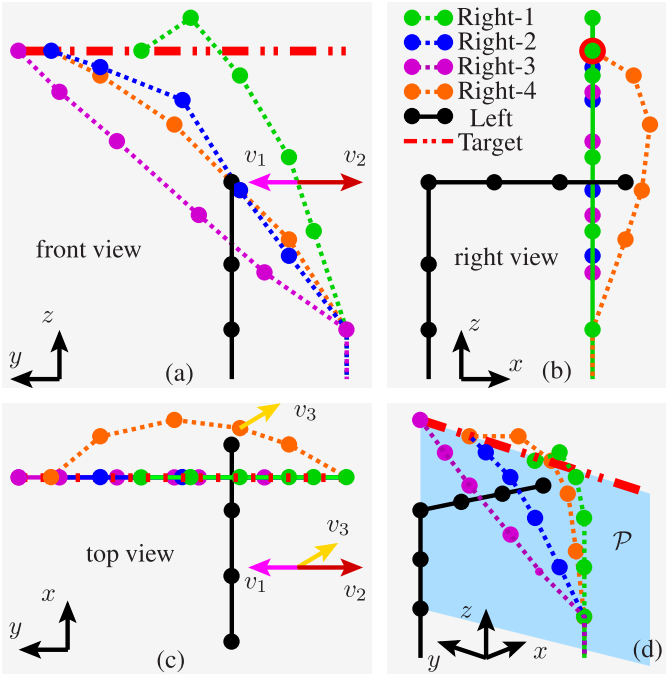


Fig. 22. (a) Front view, (b) right view, (c) top view, and (d) general view of the negative example.

E. Negative Example

Since the QP problem (25)–(28) is a convex optimization problem, there are no other local solutions except the global optimal solution. However, the solution of (25)–(28) is a solution satisfying the design rules. This solution will be within all the feasible solutions to achieve the task if (25)–(28) is reasonable. For example, if the end-effector task equality constraint has conflicts with the collision-avoidance inequality constraint (i.e., there is no intersection region), the solution of (29) is a least square solution but not exact solution. It means that the solution of (29) will become a fake attractor to the motion planning problem and will cause task failure. The above situation will happen, as shown in Fig. 22.

As shown in Fig. 22, assume that the left-hand-side manipulator is fixed in the xoz plane and the right-hand-side manipulator needs to track the target trajectory in the yoz plane. As for such a practical problem, the end-effector equality constraint gives an attraction along v_1 and the collision-avoidance inequality constraint gives another attraction along v_2 [Fig. 22(c)]. This implies that the attractor will be along the y-axis. By using the proposed scheme, the planning task will become failure due to the fake attractor caused by the contradiction between constraints (26) and (28). For example, Right-1→Right-2→Right-3. The corresponding simulation results of this negative example are presented in Fig. 23. Two manipulators crash with each other and the collision avoidance is failure.

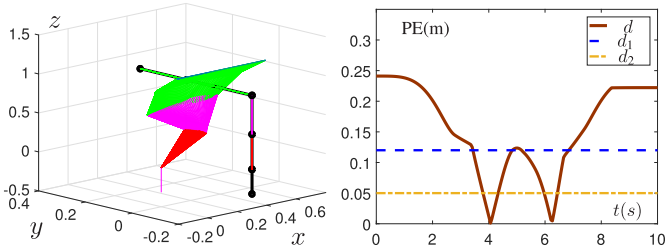


Fig. 23. Simulation results of the negative example.

To solve this problem, the attraction along v_3 is more suitable, i.e., Right-1→Right-4→Right-3 presented in Fig. 22. Thus, the collision-avoidance scheme can be further improved by using some improved design rule. This will become one of our future works.

VII. Conclusion

To avoid collisions between the dual robot manipulators, a novel MCA scheme has been proposed and analyzed in detail in this article. It has been proven by simulation and real robot experiments that the MCA scheme cannot only ensure the high tracking accuracy while tracking the desired paths, but also avoid the collision of the dual manipulators by keeping the manipulators in a safe distance. To achieve the desired goal, a new measure strategy of the minimum distance between the two line segments has been designed and discussed. In addition, the MCA scheme has been then formulated as a standard quadratic program, which has been solved by an RNN in real time. Finally, the simulations and comparisons based on PA10$^+$ and an experiment with real robots based on KINOVA JACO^2 and MICO^2 have verified the physical realizability, effectiveness, accuracy, and advantage of the MCA scheme.

References

[1] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, "Mind control of a robotic arm with visual fusion technology," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3822–3830, Sep. 2018.

[2] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, and Y. Li, "Neural-dynamic-method-based dual-arm CMG scheme with time-varying constraints applied to humanoid robots," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3251–3262, Dec. 2015.

[3] K. C. Lau, E. Y. Y. Leung, P. W. Y. Chiu, Y. Yam, J. Y. W. Lau, and C. C. Y. Poon, "A flexible surgical robotic system for removal of early-stage gastrointestinal cancers by endoscopic submucosal dissection," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2365–2374, Dec. 2016.

[4] S. Yang, R. A. MacLachlan, and C. N. Riviere, "Manipulator design and operation of a six-degree-of-freedom handheld tremor-canceling microsurgical instrument," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 761–772, Apr. 2015.

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

[6] X. Lai, P. Zhang, Y. Wang, and M. Wu, "Position-posture control of a planar four-link underactuated manipulator based on genetic algorithm," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4781–4791, Jun. 2017.

[7] Y. Li, C. Yang, W. Yan, R. Cui, and A. Annamalai, "Admittance-based adaptive cooperative control for multiple manipulators with output constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3621–3632, Dec. 2019.

[8] Z. Li and C.-Y. Su, "Neural-adaptive control of single-master–multiple-slaves teleoperation for coordinated multiple mobile manipulators with time-varying communication delays and input uncertainties," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1400–1413, Sep. 2013.

[9] N. Kimura *et al.*, "Mobile dual-arm robot for automated order picking system in warehouse containing various kinds of products," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2016, pp. 332–338.

[10] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, "Reinforcement learning of clothing assistance with a dual-arm robot," in *Proc. 11th IEEE-RAS Int. Conf. Humanoid Robots*, Oct. 2011, pp. 733–738.

[11] L. Beiner and J. Mattila, "An improved pseudoinverse solution for redundant hydraulic manipulators," in *Robotica*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[12] 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.

[13] 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.

[14] 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.

[15] 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. 2017.

[16] 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.

[17] Y. Zhang, H. Gong, M. Yang, J. Li, and X. Yang, "Stepsize range and optimal value for Taylor–Zhang discretization formula applied to zeroing neurodynamics illustrated via future equality-constrained quadratic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 959–966, Mar. 2019.

[18] 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. 2017.

[19] B. Liao, Y. Zhang, and L. Jin, "Taylor $O(h^3)$ discretization of ZNN models for dynamic equality-constrained quadratic programming with application to manipulators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 225–237, Jun. 2016.

[20] L. Xiao, K. Li, and M. Duan, "Computing time-varying quadratic optimization with finite-time convergence and noise tolerance: A unified framework for zeroing neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3360–3369, Nov. 2019.

[21] 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.

[22] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Auton. Robot.*, vol. 13, no. 3, pp. 207–222, 2002.

[23] 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, Mar. 2017.

[24] L. Palacios, M. Ceriotti, and G. Radice, "Close proximity formation flying via linear quadratic tracking controller and artificial potential function," *Adv. Space Res.*, vol. 56, no. 10, pp. 2167–2176, Nov. 2015.

[25] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. Newbury Park, CA, USA: Sage, 1986.

[26] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: Theory and experiments," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1423–1436, 1990.

[27] 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.

[28] W. Sun, L. G. Torres, J. V. D. Berg, and R. Alterovitz, "Safe motion planning for imprecise robotic manipulators by minimizing probability of collision," in *Robotics Research*. Cham, Switzerland: Springer, 2016.

[29] J. Oh, H. Bae, and J.-H. Oh, "Analytic inverse kinematics considering the joint constraints and self-collision for redundant 7DOF manipulator," in *Proc. 1st IEEE Int. Conf. Robot. Comput. (IRC)*, Apr. 2017, pp. 123–128.

[30] T. Kivelä, J. Mattila, J. Puura, and S. Launis, "Redundant robotic manipulator path planning for real-time obstacle and self-collision avoidance," in *Proc. Int. Conf. Robot. Alpe-Adria Danube Region*, 2017, pp. 208–216.

[31] Z. Zhang, S. Chen, X. Zhu, and Z. Yan, "Two hybrid end-effector posture-maintaining and obstacle-limits avoidance schemes for redundant robot manipulators," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 754–763, Feb. 2019.

[32] Y. Choi, D. Kim, S. Hwang, H. Kim, N. Kim, and C. Han, "Dual-arm robot motion planning for collision avoidance using B-spline curve," *Int. J. Precis. Eng. Manuf.*, vol. 18, no. 6, pp. 835–843, Jun. 2017.

[33] N. M. Ceriani, A. M. Zanchettin, and P. Rocco, "Collision avoidance with task constraints and kinematic limitations for dual arm robots," in *Intelligent Autonomous Systems*. Cham, Switzerland: Springer, 2016.

[34] X. Wang, C. Yang, Z. Ju, H. Ma, and M. Fu, "Robot manipulator self-identification for surrounding obstacle detection," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6495–6520, Mar. 2016.

[35] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 796–803, Apr. 2018.

[36] Z. Zhang and Y. Zhang, "Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 674–686, Apr. 2013.

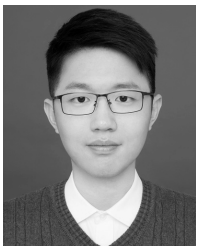
[37] Y. Zhang, W. Ma, X.-D. Li, H.-Z. Tan, and K. Chen, "MATLAB simulink modeling and simulation of LVI-based primal–dual neural network for solving linear and quadratic programs," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1679–1687, Mar. 2009.

[38] F.-T. Cheng, Y.-T. Lu, and Y.-Y. Sun, "Window-shaped obstacle avoidance for a redundant manipulator," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 28, no. 6, pp. 806–815, Dec. 1998.

Zhijun Zhang (Senior Member, IEEE) received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2012.

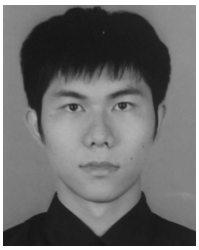
He was a Post-Doctoral Research Fellow with the Institute for Media Innovation, Nanyang Technological University, Singapore, from 2013 to 2015. From 2015 to 2019, he worked as an Associate Professor, and since 2020, he has been a Full Professor with the School of Automation Science and Engineering, South China University of Technology, Guangzhou. He has obtained Guangdong Science Fund for Distinguished Young Scholars and Youth Talents in Science and Technology Innovation of Guangdong Special Support Plan. His current research interests include neural networks, robotics, machine learning, and interaction. He has authored or coauthored more than 80 research articles on the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON MECHATRONICS, the IEEE TRANSACTIONS ON CYBERNETICS, and top international conferences.

Dr. Zhang is currently a member of the Committee on Visual Cognition and Computation of Chinese Society of Image Graphics and the Hybrid Intelligence Professional Committee of Chinese Association of Automation. He is currently the Executive Editor-in-Chief of *Global Journal of Neuroscience*, Associate Editor of the *International Journal of Robotics and Control*, Review Editor of *Frontiers in Robotics and AI*, and a reviewer of over 20 international journals.



Lunan Zheng (Student Member, IEEE) received the B.S. degree in automation from the South China University of Technology, Guangzhou, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Automation Science and Engineering, with a major in pattern recognition and intelligence system.

His current research interests include neural networks, machine learning, and robotics.



Zhuoming Chen is currently pursuing the B.S. degree with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China, with a major in artificial science and technology.

His current research interests include machine learning and robotics.

Lingdong Kong (Student Member, IEEE) received the B.Eng. degree in intelligent science and technology from the South China University of Technology, Guangzhou, China, in 2019, and the M.Sc. degree in computer control and automation from Nanyang Technological University (NTU), Singapore, in 2020, where he is currently pursuing the M.Eng. degree in computer science and engineering with the School of Computer Science and Engineering.

His current research interests include object detection, image captioning, and machine learning.

Hamid Reza Karimi (Senior Member, IEEE) received the B.Sc. degree (Hons.) in power systems from the Sharif University of Technology, Tehran, Iran, in 1998, and the M.Sc. and Ph.D. degrees (Hons.) in control systems engineering from the University of Tehran, Tehran, in 2001 and 2005, respectively.

He is currently a Professor of applied mechanics with the Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy. His current research interests include control systems and mechatronics.

Dr. Karimi is a member of Agder Academy of Science and Letters and also the IEEE Technical Committee on Systems with Uncertainty, the Committee on Industrial Cyber-Physical Systems, the IFAC Technical Committee on Mechatronic Systems, the Committee on Robust Control, and the Committee on Automotive Control. He was awarded as the 2016–2019 Web of Science Highly Cited Researcher in Engineering. He is currently the Editor-in-Chief of the *Journal of Cyber-Physical Systems*, *Machines*, the *International Journal of Aerospace System Science and Engineering*, and the *Journal of Designs*, the Section Editor-in-Chief of the *Journal of Electronics* and the *Journal of Science Progress*, the Subject Editor for the *Journal of The Franklin Institute*, and an associate editor for some international journals.