

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/333158499>

# Real-time Kinematic Control for Redundant Manipulators in a Time-varying Environment: Multiple-dynamic Obstacle Avoidance and Fast Tracking of a Moving Object

Article in IEEE Transactions on Industrial Informatics · May 2019

DOI: 10.1109/TII.2019.2917392

---

CITATIONS

5

6 authors, including:



Hui Zhang  
Harbin Institute of Technology

6 PUBLICATIONS 12 CITATIONS

[SEE PROFILE](#)

---

READS

237



Hongzhe Jin  
Harbin Institute of Technology

49 PUBLICATIONS 366 CITATIONS

[SEE PROFILE](#)



Zhangxing Liu  
Harbin Institute of Technology

8 PUBLICATIONS 32 CITATIONS

[SEE PROFILE](#)



Yubin Liu  
Fujian Normal University

72 PUBLICATIONS 587 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



self-reconfigurable robot [View project](#)



ITER Project [View project](#)

# Real-time Kinematic Control for Redundant Manipulators in a Time-varying Environment: Multiple-dynamic Obstacle Avoidance and Fast Tracking of a Moving Object

Hui Zhang, Hongzhe Jin, Zhangxing Liu, Yubin Liu, Yanhe Zhu, and Jie Zhao

**Abstract**—This paper presents a real-time kinematic control strategy to realize fast tracking of redundant robot manipulators in a time-varying environment. An obstacle avoidance method based on the law of conservation of energy is proposed to adjust the motion states of robot manipulators in real time. This method defines that the total energy for the end-effector consists of an energy toward object (ETO) and an energy around obstacle (EO), and that the total energy for each critical point on manipulator composes a relative kinematic energy (RKE) and an energy memory (EM). The total energies remain constant at each sampling period, and the conversions between the ETO and the EO or between the RKE and the EM are recognized to obey a distance-related S-function. Such considerations ensure the smooth movement of the manipulator and avoid collisions with obstacles. In real-time planning, an unsupervised single neuron PID model is raised to adaptively increase the convergence ratio of moving object tracking via the online learning of the principal component analysis. Then, combined with the dynamic obstacle avoidance method based on conservation of energy, the kinematic control strategy is established for redundant manipulators to track a moving object rapidly in the presence of multiple-dynamic obstacles. Theory analysis and various contrast experimental results show that the proposed kinematic control strategy is feasible and has fast convergence.

**Index Terms**—Redundant manipulator, principal component analysis, conservation of energy, obstacle avoidance, fast tracking.

## I. INTRODUCTION

THE development of artificial intelligence technology has facilitated the research on the autonomous operation of robot manipulators based on vision. Meanwhile, the increase in requirement of interacting with the time-varying environment (e.g., with humans, dynamic obstacles, or other robots in the industrial production line) [1-3] has made the interactive security in the autonomous operations of manipulators important in industrial application. Various redundant manipulators, such as iiwa [4], Baxter [5], and YUMI [6], have been proposed to satisfy the actual application requirement because of their outstanding capability of obstacle avoidance in such a time-varying environment [7].

The obstacle avoidance of redundant manipulators has been thoroughly studied, not only for the end-effector (EE) [8-11] but also for any other part of the robot [12, 13]. The algorithms for obstacle avoidance can be divided into two categories: global and local strategies. Global strategies rely on planning and can find a collision-free path from the initial position to the goal position if a path exists. The collision-free path is then calculated and mapped to the configuration space of the manipulators to form a control input via inverse kinematics. Global strategies are primarily applied in the path planning problems of a known or static environment [8-11] because of two insufficiencies. One is that the moving path of manipulators may be infeasible when the environment varies, thereby requiring a new plan. The other is that the high complexity of the planning algorithms consumes a large amount of computing resources and causes difficulty in satisfying real-time implementations. Local strategies are applicable to real-time kinematic control problems based on sensor information and can overcome these insufficiencies to a certain extent. Meanwhile, a collision-free movement toward the goal can be generated in accordance with the built relationship between the environment and redundant manipulators [12, 13]. Such advantages benefit the real-time operations of redundant manipulators and make local strategies a research focus in real-time obstacle avoidance for robotic manipulators.

Local strategies for obstacle avoidance are based on mod-

Manuscript submitted July 20, 2018. This work was supported in part by the National Key R&D Program of China under Grant 2017YFF0108000, the Major Research Plan of the National Natural Science Foundation of China under Grant 91648201, the Self-Planned Task of State Key Laboratory of Robotics and System (HIT) under Grant SKLRS201805B, the Provincial Supporting Projects of the National Key R&D Program of China under Grant GX18A017, the Joint Research Fund (U1713201) between the National Natural Science Foundation of China (NSFC) and Shen Zhen, and the National Natural Science Foundation of China under Grant 61473102. (Corresponding author: Hongzhe Jin.) (Hongzhe Jin, Yubin Liu, and Yanhe Zhu are co-correspondence authors.)

The authors are with the School of Mechatronics Engineering, Harbin Institute of Technology (HIT), Harbin, Heilongjiang, China (e-mail: huizh2016@outlook.com; hongzhejin@hit.edu.cn; liuzhangxin-ghit@163.com; liuyubin@hit.edu.cn; yhzhu@hit.edu.cn; jzhao@hit.edu.cn).

el/task/constraints and mostly applied at the kinematic level, including the force/potential model [12, 14-16], magnetic field [17], spring damping model [18, 19], cosine curve model [20], local coordinate model [21, 22], tasks [23, 24], inequality constraints [25, 26], and intelligent methods [27, 28]. Model-based methods are intuitive, simple, and easy to implement and can establish a simple model between the manipulator and obstacles for obstacle avoidance. Task-based methods must halt the current task of robot manipulators to wait for obstacles to depart, which is time-consuming and inefficient. Methods based on constraints/intelligence need to build a mathematical mapping or a cost function in advance and consume much calculation time even if the algorithms are simplified/optimized. Some of these models may be trapped in local optimum, which prevents the EE from moving toward the goal [12]. Several existing methods can be associated with other methods to span the minimum area but have to partially sacrifice the real-time requirements [14, 29]. Most of these methods ignore the attitude adjustment of EE during obstacle avoidance, which may result in failure. Overall, the motion states of the manipulator using local strategies are that the EE moves around the obstacles and that the other part of the manipulator maintains a safe distance from the obstacles to achieve collision-free motion. Thus, this intuitive idea, i.e., the motion states, is adopted into the obstacle avoidance method designed in this paper.

The control system of redundant robot manipulators can be designed using dynamic or kinematic models [20, 21]. The system using dynamic models exhibits excellent tracking performance but relies heavily on the experience of designers and the closed hardware system, thereby limiting its further development. The system using kinematic models, which is considered in this paper, can also exhibit excellent performance if position and velocity commands are applied due to the great tracking capability of the existing commercial drivers. Moreover, the kinematic-based system is open and can be easily integrated into vision technology to realize the autonomous operation of a robot manipulator through high-level intelligent planning [30]. The smooth and continuous motion/path of redundant manipulator needs to be guaranteed for the kinematic-based system to achieve real-time operation. Unfortunately, the real-time path planning methods for the high-level planning of redundant manipulators has been rarely reported and mainly includes fixed proportion- [20], fixed clamping- [31], and virtual controller-based [21]. Fixed proportion- and fixed clamping-based methods are the traditional planning means that set fixed gain or clamping. The virtual controller-based method is advanced and can establish a virtual model of the error between the EE and object to achieve real-time path planning. However, these real-time planning methods are limited to slow tracking tasks. To improve the convergence rate of tracking, a large gain is selected, which may result in the overshoot of the initial motion of redundant manipulators. Nearly no existing research simultaneously considers the tracking performance and obstacle avoidance capability of redundant manipulators.

Therefore, on the basis of previous contributions [12, 14-16, 20-22, 31], this paper proposes a novel real-time kinematic control strategy, which involves a practical obstacle avoidance method and its associated fast tracking algorithm, to improve the motion characteristics of redundant manipulators in a

time-varying environment, especially in a multiple-dynamic obstacle environment. The following aspects differ from those in the existing literature.

On the basis of the law of conservation of energy (LCE) at each sampling period, the proposed obstacle avoidance method considers two kinds of total energies for the movement of manipulators rather than establishing constraints or models. One is for the EE consisting of an energy toward object (ETO) and an energy around obstacle (EO), and the other is for each critical point on a manipulator comprising a relative kinematic energy (RKE) and an energy memory (EM). The ETO holds the primary task of the EE, and the EO allows the circular motion of the EE around the obstacle. The RKE and the EM make each critical point of the manipulator maintain a certain distance from the obstacles. The reciprocal transformation of different energies is accomplished via an ideal S-function, which causes the smooth movement of the manipulator. Meanwhile, the attitude adjustment, which is ignored in existing researches, ensures that the EE passes through the multiple-dynamic obstacles easily. The real-time path planning of a manipulator is regarded as a control problem, and a single neuron PID model is proposed to adjust the convergence rate of moving object tracking adaptively via online learning of the principal component analysis (PCA). This model is an unsupervised learning process and has rapidity and placidity. Then, a real-time kinematic control strategy is designed for fast, smooth, and safe tracking tasks in a time-varying environment. Experiments of multiple-dynamic obstacles in different cases verify the practicability and effectiveness of the proposed kinematic control strategy in time-varying environment applications.

The remainder of this paper is organized as follows. Section II presents the real-time tracking of a redundant manipulator in an obstacle-free case and the convergence analysis of the Hebbian-based PCA. Section III discusses the LCE-based method for avoiding obstacles and the proposed real-time kinematic control strategy. Section IV provides the experimental results for different cases. Section V presents the conclusion.

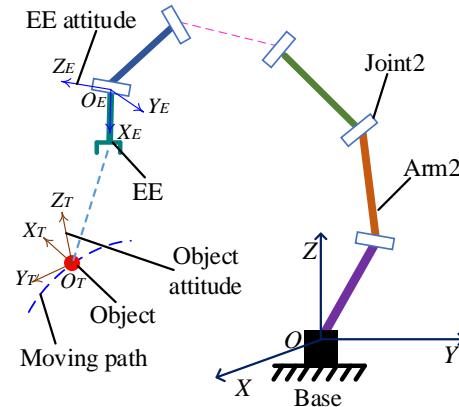


Fig. 1. s-DoF manipulator real-time tracking of a moving object in an obstacle-free case.  $s > 6$ .

## II. OBSTACLE-FREE CASE

### A. Based on the Common Method

In Fig. 1, the Cartesian coordinate  $O\text{-}XYZ$  is the base

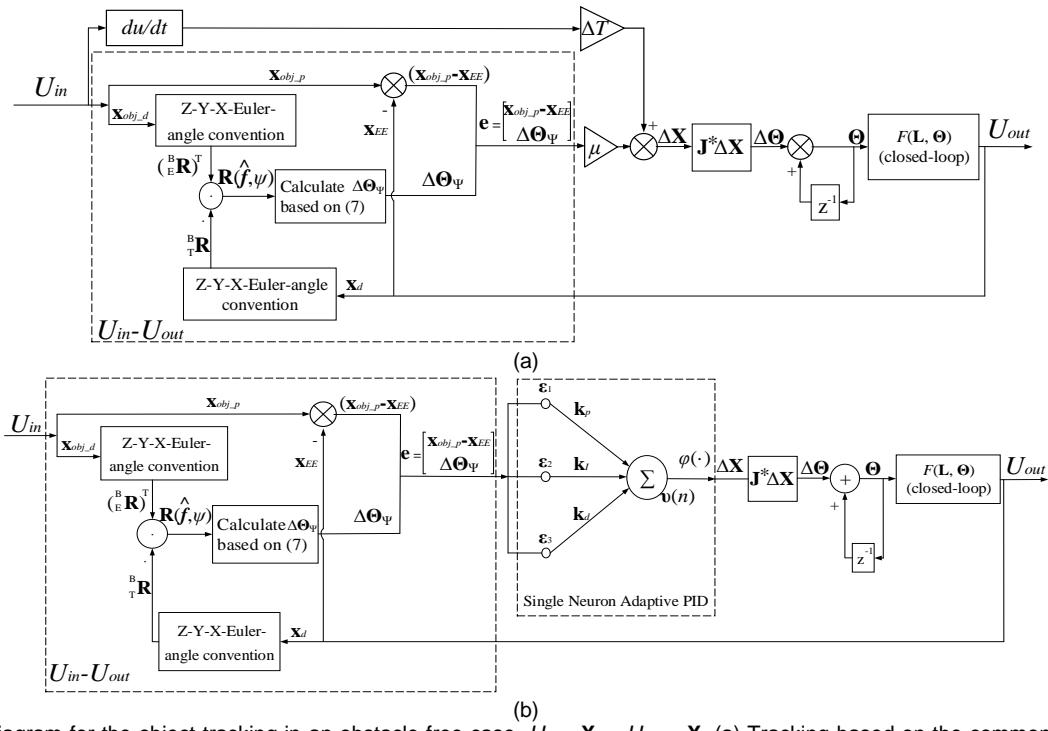


Fig. 2. Control diagram for the object tracking in an obstacle-free case.  $U_{in} = \mathbf{X}_{obj}$ ,  $U_{out} = \mathbf{X}$ . (a) Tracking based on the common method. (b) Fast tracking algorithm based on the single neuron adaptive PID model.

coordinate system of the redundant manipulator. The coordinate system  $\{X_T, Y_T, Z_T\}$  is the attitude of the object and regarded as the tracking attitude of the EE, which makes the EE face the direction of the coordinate axis  $X_T$ . The coordinate system  $\{X_{EE}, Y_{EE}, Z_{EE}\}$  is the attitude of the EE. The general equation of inverse kinematics for redundant manipulators is as follows:

$$\dot{\Theta} = \mathbf{J}^* \dot{\mathbf{X}}, \quad (1)$$

where  $\dot{\mathbf{X}}$  is the velocity of the EE in the Cartesian space.  $\mathbf{X}$  denotes the pose,  $\mathbf{X} \in R^{N \times 1}$ , and  $N$  represents the rows of  $\mathbf{X}$ .  $\mathbf{X} = (\mathbf{x}_{EE}, \mathbf{x}_d)^T$ , where  $\mathbf{x}_{EE}$  is the position.  $\mathbf{x}_d$  indicates the attitude and can be represented by the Z-Y-X Euler angle about the base coordinate system.  $\dot{\Theta}$  refers to the joint velocity of the manipulator in joint space.  $\Theta$  signifies the joint angle,  $\Theta = (\theta_1, \theta_2, \dots, \theta_s)^T$ , and the degree of freedom (DoF) is  $s$ .  $\mathbf{J}^*$  denotes the pseudo-inverse of Jacobian matrix  $\mathbf{J}(\Theta)$  based on the damped least squares (DLS) method [32], and  $\mathbf{J}^* = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda\mathbf{I})^{-1}$ .  $\mathbf{J} \in R^{N \times s}$ .  $\lambda$  ( $\lambda > 0$ ) represents the damping factor that can handle the ill-conditioned  $\mathbf{J}$  in the neighborhood of singular configurations for redundant manipulators and guarantee the EE with the minimum possible deviation at all configurations.  $\mathbf{I}$  is a unit matrix with the dimension  $N \times N$ .

The traditional fixed proportion-based method [20] for the real-time tracking of an object defines  $\dot{\mathbf{X}}$  in (1) as follows:

$$\dot{\mathbf{X}} = \dot{\mathbf{X}}_{obj} + K_p \cdot (\mathbf{X}_{obj} - \mathbf{X}), \quad (2)$$

where  $\dot{\mathbf{X}}_{obj}$  is the velocity of the object,  $\mathbf{X}_{obj}$  denotes the pose, and  $\mathbf{X}_{obj} = (\mathbf{x}_{obj\_p}, \mathbf{x}_{obj\_d})^T$ .  $\mathbf{x}_{obj\_p}$  and  $\mathbf{x}_{obj\_d}$  represent the position and attitude of the object, respectively.  $K_p$  is the gain coefficient.

The iterative method is utilized in this paper to realize the manipulator motion [31]. Thus, (1) becomes

$$\Delta\Theta \approx \mathbf{J}^* \Delta\mathbf{X}, \quad (3)$$

where  $\Delta\mathbf{X}$  is defined as the real-time path and becomes

$$\Delta\mathbf{X} = \Delta\mathbf{X}_{obj} + \mu \cdot (\mathbf{X}_{obj} - \mathbf{X}), \quad (4)$$

where  $\mu = K_p \cdot \Delta T$ .  $\Delta\mathbf{X}$  is the planned pose for the EE, and  $\Delta\mathbf{X}_{obj}$  represents the changing pose of the object at a sampling time interval  $\Delta T$ . Accordingly, the moving object can be tracked by updating the joint angles  $\Theta$  in real time through

$$\Theta(p) := \Theta(p-1) + \Delta\Theta, \quad (5)$$

where  $\Theta(p)$  is the current joint angles at the  $p$ -th iteration.  $\Theta(p-1)$  denotes the past joint angles at the  $(p-1)$ -th iteration ( $p \geq 1$ ). The control diagram of the tracking process is described in Fig. 2(a).

**Remark 1.** Unlike calculating the position error in Fig. 2, the attitude error cannot be directly obtained by two kinds of attitudes. Thus, the rotation matrix  ${}^B_E \mathbf{R}$  is defined to describe the EE coordinate system  $\{X_E, Y_E, Z_E\}$  relative to the base coordinate system  $\{X, Y, Z\}$ . The rotational operator is similar to that of the object coordinate system  $\{X_T, Y_T, Z_T\}$ ,  ${}^B_E \mathbf{R}$ .  $\mathbf{R}(\hat{f}, \psi)$  is the rotational operator about the axis direction  $\hat{f}$  by  $\psi$  radians,  $\hat{f} = (\hat{f}_x, \hat{f}_y, \hat{f}_z)$ .  ${}^B_E \mathbf{R}$ ,  ${}^B_T \mathbf{R}$ ,  $\hat{f}$ , and  $\psi$  can be easily obtained following [33].  $\mathbf{R}(\hat{f}, \psi)$  is defined as

$$\mathbf{R}(\hat{f}, \psi) = ({}^B_E \mathbf{R})^T \cdot {}^B_T \mathbf{R}. \quad (6)$$

The attitude error described in the base coordinate system is

$$\Delta\Theta_\Psi = {}^B_E \mathbf{R} \cdot (\hat{f}_x \cdot \psi, \hat{f}_y \cdot \psi, \hat{f}_z \cdot \psi)^T. \quad (7)$$

### B. Based on the Single Neuron Adaptive PID Model

In the traditional fixed proportion-based method for real-time path planning, the gain in the initial stage of motion is large when fast tracking a moving object, that is,  $\mu$  in (4) is set large. With the path planning as a control problem, the single neuron PID model is introduced to suppress the large initial gain and achieve fast and smooth tracking. Meanwhile, the unsupervised Hebbian-based PCA is proposed to adjust the PID

parameters via online learning (Fig. 2(b)).  $F(\mathbf{L}, \Theta)$  expresses the forward kinematics of the redundant manipulator.  $\mathbf{L} = (l_1, l_2, \dots, l_s)$  is the length of the arms. The input  $\mathbf{e}_i$  is defined as

$$\begin{cases} \mathbf{e}_1(p) = \mathbf{e}(p) - \mathbf{e}(p-1) \\ \mathbf{e}_2(p) = \mathbf{e}(p) \\ \mathbf{e}_3(p) = \mathbf{e}(p) - 2\mathbf{e}(p-1) + \mathbf{e}(p-2) \end{cases}, \quad (8)$$

where  $\mathbf{e}_i = (\varepsilon_{i1}, \varepsilon_{i2}, \dots, \varepsilon_{iN})^T$ ,  $i = 1, 2, 3$ .  $\mathbf{e}(p)$  is the pose error. To simplify the expression of the pose error,  $\mathbf{e}(p)$  is defined as  $\mathbf{e}(p) = \mathbf{U}_{in} - \mathbf{U}_{out} = \mathbf{X}_{obj} - \mathbf{X} = (e_1, e_2, \dots, e_N)^T$  as shown in Fig. 2. In accordance with the PCA based on the Hebbian-based learning rule [34-36], the synapse weights in the single neuron are updated using the following nonlinear stochastic difference equations:

$$\begin{cases} \mathbf{w}_1(p) = \mathbf{w}_1(p-1) + \boldsymbol{\eta}_p \mathbf{v}(p-1) [\mathbf{e}_1(p-1) - \mathbf{v}(p-1) \mathbf{w}_1(p-1)] \\ \mathbf{w}_2(p) = \mathbf{w}_2(p-1) + \boldsymbol{\eta}_I \mathbf{v}(p-1) [\mathbf{e}_2(p-1) - \mathbf{v}(p-1) \mathbf{w}_2(p-1)] \\ \mathbf{w}_3(p) = \mathbf{w}_3(p-1) + \boldsymbol{\eta}_d \mathbf{v}(p-1) [\mathbf{e}_3(p-1) - \mathbf{v}(p-1) \mathbf{w}_3(p-1)] \end{cases} \quad (9)$$

where  $\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{iN})^T$  and  $i = 1, 2, 3$ .  $\mathbf{v}, \boldsymbol{\eta}_p, \boldsymbol{\eta}_I, \boldsymbol{\eta}_d \in R^{N \times N}$ .  $\mathbf{w}_{ij}$ ,  $\mathbf{v}$ ,  $\boldsymbol{\eta}_p$ ,  $\boldsymbol{\eta}_I$ , and  $\boldsymbol{\eta}_d$  are defined as

$$\begin{cases} w_{ij}(p) = k_{pj}, w_{2j}(p) = k_{pj}, w_{3j}(p) = k_{dj}; \\ \boldsymbol{\eta}_p = diag(\eta_{p1}, \eta_{p2}, \dots, \eta_{pN}); \\ \boldsymbol{\eta}_I = diag(\eta_{I1}, \eta_{I2}, \dots, \eta_{IN}); \\ \boldsymbol{\eta}_d = diag(\eta_{d1}, \eta_{d2}, \dots, \eta_{dN}); \\ \mathbf{v}(p) = diag \left[ \sum_{h=1}^3 w_{h1}(p) \varepsilon_{h1}(p), \sum_{h=1}^3 w_{h2}(p) \varepsilon_{h2}(p), \dots, \sum_{h=1}^3 w_{hN}(p) \varepsilon_{hN}(p) \right]; \end{cases} \quad (10)$$

where  $j = 1, 2, \dots, N$ .  $\mathbf{k}_p = (k_{p1}, k_{p2}, \dots, k_{pN})^T$ .  $\mathbf{k}_I = (k_{I1}, k_{I2}, \dots, k_{IN})^T$ .  $\mathbf{k}_d = (k_{d1}, k_{d2}, \dots, k_{dN})^T$ .  $\mathbf{k}_p$ ,  $\mathbf{k}_I$ , and  $\mathbf{k}_d$  are shown in Fig. 2(b). The output  $\Delta\mathbf{X}$  of the neuron is as follows:

$$\Delta\mathbf{X} = \varphi(\cdot) = \boldsymbol{\kappa} \cdot [\mathbf{v}(p) \cdot \mathbf{b}], \quad (11)$$

where  $\Delta\mathbf{X} \in R^{N \times 1}$ .  $\mathbf{b} = (1, 1, \dots, 1)^T$ , and  $\mathbf{b} \in R^{N \times 1}$ .  $\boldsymbol{\kappa}$  is the adaptive coefficient matrix of the single neuron,  $\boldsymbol{\kappa} \in R^{N \times N}$ .

$$\boldsymbol{\kappa} = diag \left\{ \left[ 1 - \left( \frac{\alpha_1 - |e_1(p)|}{\alpha_1} \right)^2 \right] \cdot \beta_1 + \delta_1, \left[ 1 - \left( \frac{\alpha_2 - |e_2(p)|}{\alpha_2} \right)^2 \right] \cdot \beta_2 + \delta_2, \dots, \left[ 1 - \left( \frac{\alpha_N - |e_N(p)|}{\alpha_N} \right)^2 \right] \cdot \beta_N + \delta_N \right\}, \quad (12)$$

where  $\alpha_i, \beta_i$ , and  $\delta_i$  are constant,  $i = 1, 2, 3, \dots, N$ . Then, the fast tracking of an object can be achieved placidly via the online learning model as shown in Fig. 2(b), and the contrast experiments are presented in Part B of Section IV.

### C. Convergence Analysis of the Hebbian-based PCA

The stability analysis of Hebbian-based PID models realizes the convergence of synapse weights due to the system errors gradually decreasing along the negative gradient direction of the synapse weights [36]. For supervised Hebbian-based models, the convergence of synapse weights can be easily analyzed because the errors are directly used as the tutor signal, and the Lyapunov (or estimation) function can be easily established. For unsupervised Hebbian-based models, the convergences cannot be directly analyzed via Lyapunov theory because the unsupervised models are developed on the basis of neurobiology and statistics theories, and no system error is directly used as the tutor signal for the online learning of synapse weights in the optimization algorithm of Eq. (9). Fortunately, the proposed Hebbian-based PCA aims to maximize the output variance or

the information entropy of Hebbian-based PID models through the self-organization learning principles of neural networks, especially their competition and collaboration principles. Thus, by utilizing these characteristics, the nonlinear stochastic difference equations in (9) for the error  $e_\chi$  can be rewritten in a unified form following the Kushner direct average method [36]:

$$\mathbf{W}_\chi(p) = \mathbf{W}_\chi(p-1) + \boldsymbol{\eta}_\chi [\mathbf{R}_\chi - \mathbf{W}_\chi^T(p-1) \mathbf{R}_\chi \mathbf{W}_\chi(p-1) \mathbf{I}] \mathbf{W}_\chi(p-1), \quad (13)$$

where  $\chi = 1, 2, \dots, N$ .  $\mathbf{R}_\chi = \tilde{\boldsymbol{\epsilon}}_\chi \tilde{\boldsymbol{\epsilon}}_\chi^T$  is the correlation matrix and used to replace the output covariance matrix of the Hebbian-based PID model when  $\boldsymbol{\eta}_\chi$  is a small value.  $\tilde{\boldsymbol{\epsilon}}_\chi = (\varepsilon_{1\chi}, \varepsilon_{2\chi}, \varepsilon_{3\chi})^T$ ,  $\boldsymbol{\eta}_\chi = diag(\eta_{p\chi}, \eta_{I\chi}, \eta_{d\chi})$ , and  $\mathbf{W}_\chi = (w_{1\chi}, w_{2\chi}, w_{3\chi})^T$ .

Let  $\Delta\mathbf{W}_\chi(p-1) = \mathbf{W}_\chi(p) - \mathbf{W}_\chi(p-1)$ ,  $\boldsymbol{\eta}_\chi \cdot \frac{d\mathbf{W}_\chi(t)}{dt} = \Delta\mathbf{W}_\chi(p-1)$ . Then  $\mathbf{W}_\chi(t)$  can be expanded as

$$\mathbf{W}_\chi(t) = \sum_{g=1}^3 \zeta_g(t) \mathbf{q}_g, \quad (14)$$

where  $\mathbf{q}_g$  is the  $g$ -th normalized eigenvector of  $\mathbf{R}_\chi$  and  $\zeta_g(t)$  denotes the projection coefficient. Then, the convergence analysis of (13) can be attributed to the system stability analysis of the ordinary differential equations containing  $\zeta_g(t)$  [36]:

$$\frac{d\zeta_g(t)}{dt} = \xi_g \zeta_g(t) - \zeta_g(t) \sum_{i=1}^3 \xi_i(t) \zeta_i^2(t), \quad (15)$$

where  $\xi_g$  is the eigenvalue of  $\mathbf{q}_g$ ,  $\xi_1 > \xi_2 > \xi_3 > 0$ . Finally, the features of Hebbian-based PCA can be obtained as follows:

$$\lim_{p \rightarrow \infty} \sigma^2(p) = \xi_1, \lim_{p \rightarrow \infty} \mathbf{W}_\chi(p) = \mathbf{q}_1, \lim_{p \rightarrow \infty} \|\mathbf{W}_\chi(p)\| = 1, \quad (16)$$

where  $\sigma^2$  is the output variance of the PID model. In (16),  $\sigma^2$ ,  $\mathbf{W}_\chi$ , and  $\|\mathbf{W}_\chi\|$  tend to stabilize via the competition and collaboration among the synapse weights. Therefore, the single neuron PID model using Hebbian-based PCA is asymptotically stable. Additional theoretical derivations and explanations can be found in [36].

## III. OBSTACLE CASE

### A. Obstacle Avoidance Planning for EE

In this paper, an obstacle avoidance strategy is proposed based on the LCE at each sampling period. The LCE states that the total energy of an isolated system remains constant, which indicates that energy can only be transformed from one form to another. This paper defines that the EE and the obstacle form an isolated system at each sampling period, and that the end of the EE is hypothetically a point with a quality of a unit mass of 1 (kg). In accordance with the LCE, the total energy of the point at the end of the EE is instantaneously constant at time  $t$ . Then, we define the total energy as follows:

$$E_{Total} \triangleq Const = \underbrace{f(d) \cdot E_k}_{\tilde{E}_k} + \underbrace{[1 - f(d)] \cdot E_c}_{\tilde{E}_c}, \quad (17)$$

where  $E_{Total}$  is the total energy,  $E_k$  denotes the constant kinetic energy toward the object, and  $E_c$  represents the constant kinetic energy of circular motion around the obstacle in real time.  $E_k \triangleq E_c$ .  $\tilde{E}_k$  and  $\tilde{E}_c$  refer to the ETO and the EAO, respectively.

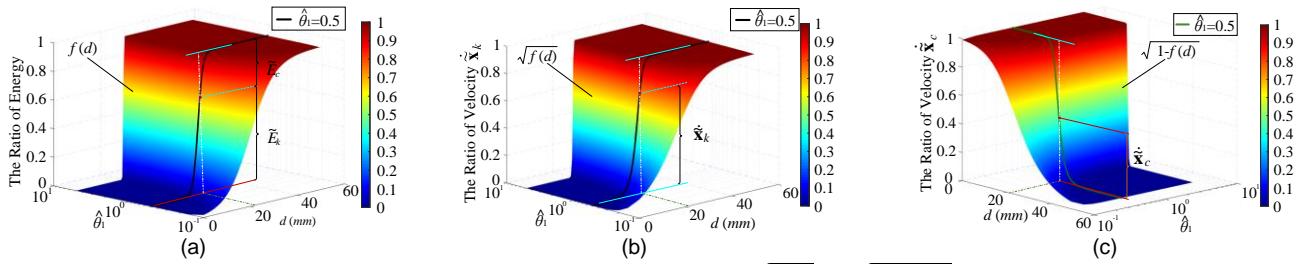


Fig. 3. Transformation based on an S-function  $f(d)$  at sampling time  $t$ . (a)  $f(d)$ . (b)  $\sqrt{f(d)}$ . (c)  $\sqrt{1-f(d)}$ .

When EE moves close to the obstacle,  $\tilde{E}_k$  gradually decreases and  $\tilde{E}_c$  increases accordingly. The transformation between  $\tilde{E}_k$  and  $\tilde{E}_c$  is considered to obey the S-function  $f(d)$ , thereby resulting in an ideal and smooth transformation.  $E_k$ ,  $E_c$ , and  $f(d)$  are defined as follows:

$$E_k = \frac{1}{2} \dot{\mathbf{x}}_k^T \dot{\mathbf{x}}_k, E_c = \frac{1}{2} \dot{\mathbf{x}}_c^T \dot{\mathbf{x}}_c; \quad (18)$$

$$f(d) = \frac{1}{2} \{ \tanh[\hat{\theta}_1 \cdot (d(\mathbf{x}_{EE}, \mathbf{x}_{Obs}) - d_0(\mathbf{x}_{EE}, \mathbf{x}_{Obs}))] + 1 \}, \quad (19)$$

where  $\dot{\mathbf{x}}_k$  and  $\dot{\mathbf{x}}_c$  are defined as the constant velocities toward the object and around the obstacle at each sampling period, respectively.  $\dot{\mathbf{x}}_k$  and  $\dot{\mathbf{x}}_c$  have the same magnitude since  $E_k \triangleq E_c$ .  $d_0(\mathbf{x}_{EE}, \mathbf{x}_{Obs})$  represents the critical distance and limits the minimum proximity between the EE and the obstacles.  $\hat{\theta}_1$  is defined as the control parameter of the transformation rate for the conversion between the ETO and the EAO, as shown in Fig. 3.  $d(\mathbf{x}_{EE}, \mathbf{x}_{Obs})$  indicates the distance between the EE and the obstacles, and is calculated as

$$d(\mathbf{x}_{EE}, \mathbf{x}_{Obs}) = \|\mathbf{x}_{EE} - \mathbf{x}_{Obs}\| = \left[ \sum_{k=1}^M (\mathbf{x}_{EE,k} - \mathbf{x}_{Obs,k}) \right]^{\frac{1}{2}}, \quad (20)$$

where  $\mathbf{x}_{EE}$  is the EE position.  $\mathbf{x}_{Obs}$  denotes the obstacle position.  $M = 2$  for planar manipulators and  $M = 3$  for spatial manipulators.

**Remark 2.** Considering that the robot system in this paper cannot directly feed back speed information, all velocities are calculated by the ratio of the position difference to the sampling time interval  $\Delta T$ . In (18),  $\dot{\mathbf{x}}_k = \frac{\Delta \mathbf{x}_{EE}}{\Delta T} = \frac{\mathbf{x}_{EE}(p) - \mathbf{x}_{EE}(p-1)}{\Delta T}$ , where  $\mathbf{x}_{EE}(p)$  is the value at the  $p$ -th iteration. The magnitude of  $\dot{\mathbf{x}}_c$  is calculated by  $\|\dot{\mathbf{x}}_c\| = \|\dot{\mathbf{x}}_k\| = \left\| \frac{\Delta \mathbf{x}_{EE}}{\Delta T} \right\|$  (since  $E_k \triangleq E_c$ ), and the direction of  $\dot{\mathbf{x}}_c$  is obtained by (22) and (23).

In (17),  $\tilde{E}_k$  and  $\tilde{E}_c$  are defined as

$$\tilde{E}_k = \frac{1}{2} \dot{\mathbf{x}}_k^T \dot{\mathbf{x}}_k, \tilde{E}_c = \frac{1}{2} \dot{\mathbf{x}}_c^T \dot{\mathbf{x}}_c, \quad (21)$$

where  $\dot{\mathbf{x}}_k$  and  $\dot{\mathbf{x}}_c$  are the instantaneous velocities of the ETO and the EAO, respectively. The relationship between  $\dot{\mathbf{x}}_k$  (and  $\dot{\mathbf{x}}_c$ ) and  $\dot{\mathbf{x}}_k$  (and  $\dot{\mathbf{x}}_c$ ) is derived as

$$\dot{\mathbf{x}}_k = \sqrt{f(d)} \dot{\mathbf{x}}_k, \dot{\mathbf{x}}_c = \sqrt{1-f(d)} \dot{\mathbf{x}}_c. \quad (22)$$

Meanwhile,  $\dot{\mathbf{x}}_c$  should satisfy the constraints in (23) because the EE will move around an obstacle when close to it.

$$\begin{cases} \mathbf{T}_{Obs} \cdot \dot{\mathbf{x}}_c = 0 \\ (\mathbf{T}_{Obs} \times \mathbf{T}_{Obj}) \cdot \dot{\mathbf{x}}_c = 0 \end{cases} \quad (23)$$

where the vector  $\mathbf{T}_{Obs}$  is from the EE to the obstacle and the vector  $\mathbf{T}_{Obj}$  is from the EE to the object, as shown in Fig. 4(a).

**Remark 3.** To ensure that (23) has a unique solution for  $\dot{\mathbf{x}}_c$ , that is, ensuring that  $\mathbf{T}_{Obs} \times \mathbf{T}_{Obj} \neq 0$ , when the object and the

obstacle are in the same direction from the EE, the calculation of  $\mathbf{T}_{Obs} \times \mathbf{T}_{Obj}$  in (23) is defined as

$$\mathbf{T}_{Obs} \times \mathbf{T}_{Obj} = \varphi_1 + \varphi_3 + \varphi_4, \quad (23-a)$$

$$\varphi_1 = \mathbf{T}_{Obs}(p) \times \mathbf{T}_{Obj}(p), \quad (23-b)$$

$$\varphi_2 = \mathbf{T}_{Obs}(p-1) \times \mathbf{T}_{Obj}(p-1), \quad (23-c)$$

$$\varphi_3 = [1 - sgn(|\varphi_1|)] \cdot \varphi_2, \quad (23-d)$$

$$\varphi_4 = [1 - sgn(|\varphi_1|)] \cdot [1 - sgn(|\varphi_2|)] \cdot (\varrho_x \mathbf{e}_{Ix} + \varrho_y \mathbf{e}_{Iy} + \varrho_z \mathbf{e}_{Iz}), \quad (23-e)$$

where  $\mathbf{T}_{Obs}(p)$  and  $\mathbf{T}_{Obj}(p)$  indicate the calculated vectors at the  $p$ -th iteration.  $\mathbf{e}_{Ix} = (1, 0, 0)$ ,  $\mathbf{e}_{Iy} = (0, 1, 0)$ , and  $\mathbf{e}_{Iz} = (0, 0, 1)$ .  $\varrho_x$ ,  $\varrho_y$ , and  $\varrho_z$  are defined as the numbers between -0.1 and 0.1.

### B. Attitude Adjustment of the EE for Obstacle Avoidance

In Fig. 4, the changes in the EE attitude are beneficial for the EE of the manipulator to avoid and pass through multiple obstacles when the distance between obstacle1 and obstacle2 becomes close. However, the adjustment of the EE attitude is nearly neglected in the existing literature. Thus, the coordinate system  $\{\mathbf{e}_{\dot{\mathbf{x}}_c}, \mathbf{e}_{-\mathbf{T}_{Obs}}, \mathbf{e}_{\mathbf{T}_{Obs} \times \mathbf{T}_{Obj}}\}$  (or  $\{X_s, Y_s, Z_s\}$ ) is temporarily defined as the target orientation of the EE when avoiding obstacles, which makes the EE face the direction of  $\dot{\mathbf{x}}_c$ . The rotation matrix  ${}^B_S \mathbf{R}$  describes the coordinate system  $\{X_s, Y_s, Z_s\}$  relative to the base coordinate system  $\{X, Y, Z\}$ . The vectors  $\mathbf{e}_{\dot{\mathbf{x}}_c}$ ,  $\mathbf{e}_{-\mathbf{T}_{Obs}}$ , and  $\mathbf{e}_{\mathbf{T}_{Obs} \times \mathbf{T}_{Obj}}$  are the unit vectors of  $\dot{\mathbf{x}}_c$ ,  $-\mathbf{T}_{Obs}$ , and  $\mathbf{T}_{Obs} \times \mathbf{T}_{Obj}$ , respectively. After avoiding the obstacle, the coordinate system  $\{X_t, Y_t, Z_t\}$  is then regarded as the target orientation of the EE, and makes the EE face the direction of the coordinate axis  $X_t$ . Therefore,  $\mathbf{R}(\hat{f}, \psi)$  in (6) becomes

$$\mathbf{R}(\hat{f}, \psi) = \begin{cases} ({}^B_E \mathbf{R})^T \cdot {}^B_S \mathbf{R}, & \text{when avoiding obstacles} \\ ({}^B_E \mathbf{R})^T \cdot {}^B_T \mathbf{R}, & \text{no avoiding obstacles} \end{cases}. \quad (24)$$

### C. Obstacle Avoidance for the Manipulator Arm

The obstacles in the workspace of manipulator and not on the motion path of the EE influence the motion of arms. To avoid these obstacles, five critical points are set at equal intervals in each arm of the manipulator. Meanwhile, the null space of the redundant manipulator is combined to avoid the obstacles. As described in Part A of Section III, the obstacle avoidance method for the arms of the manipulator is also based on the LCE. To avoid multiple obstacles simultaneously, equation (3) is modified as follows [20]:

$$\Delta \Theta = \mathbf{J}^* \Delta \mathbf{X} + (\mathbf{I} - \mathbf{J}^* \mathbf{J}) \cdot (\mathbf{K} \cdot \nabla \mathbf{H}), \quad (25)$$

where  $(\mathbf{I} - \mathbf{J}^* \mathbf{J}) \cdot (\mathbf{K} \cdot \nabla \mathbf{H})$  is the null space.  $\mathbf{K}$  is defined as

$$\mathbf{K} = (k_1, k_2, \dots, k_W), \quad (26)$$

where  $W$  is the number of obstacles.  $\nabla \mathbf{H}$  denotes the motion

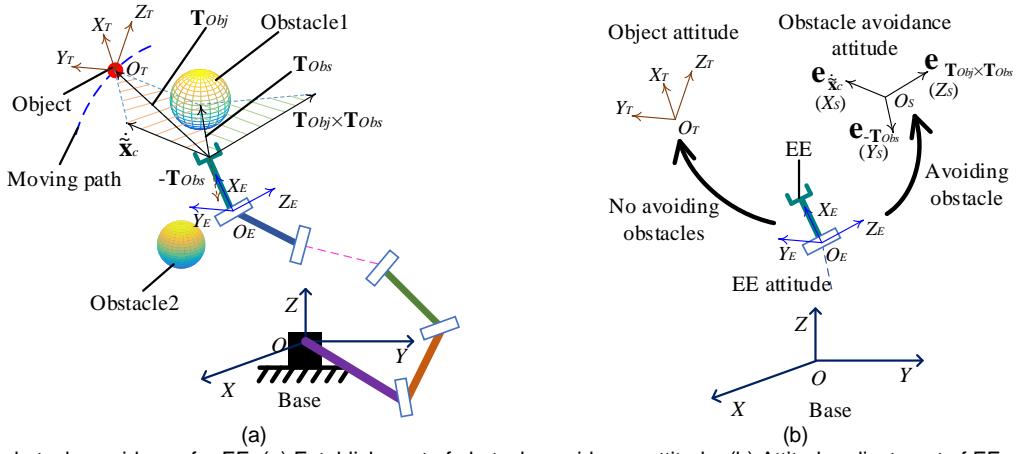


Fig. 4. Principle of obstacle avoidance for EE. (a) Establishment of obstacle avoidance attitude. (b) Attitude adjustment of EE.

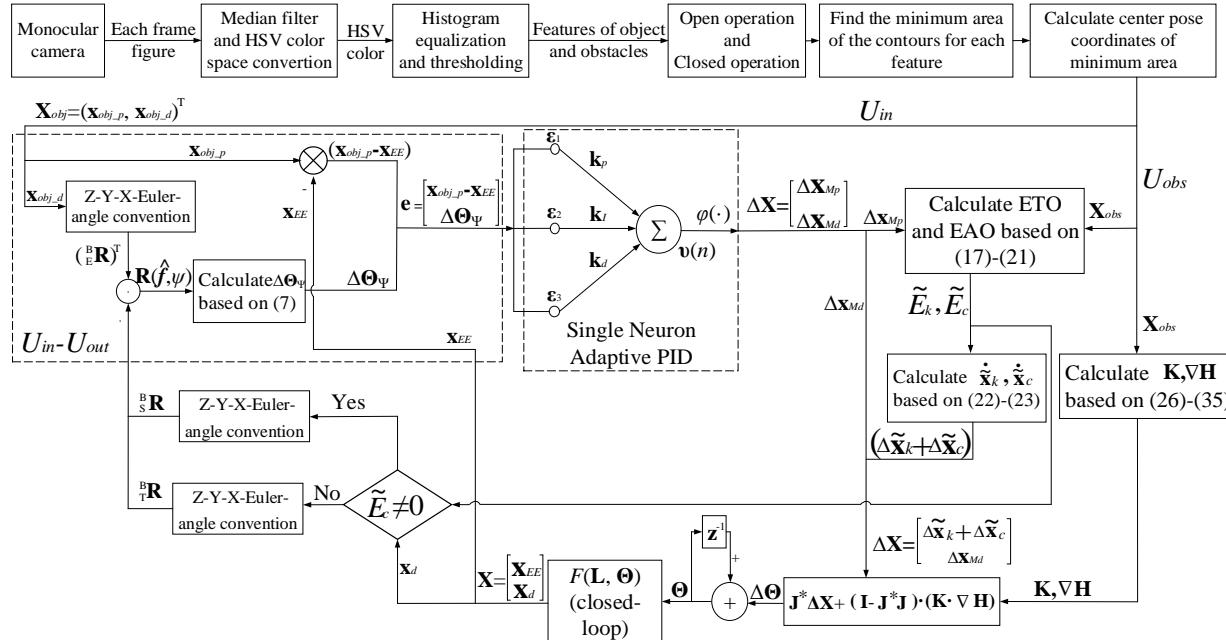


Fig. 5. Real-time kinematic control strategy for object tracking in an obstacle case. \$U\_{in} = \mathbf{X}\_{obj}\$, \$U\_{out} = \mathbf{X}\$, \$U\_{obs} = \mathbf{X}\_{obs}\$.

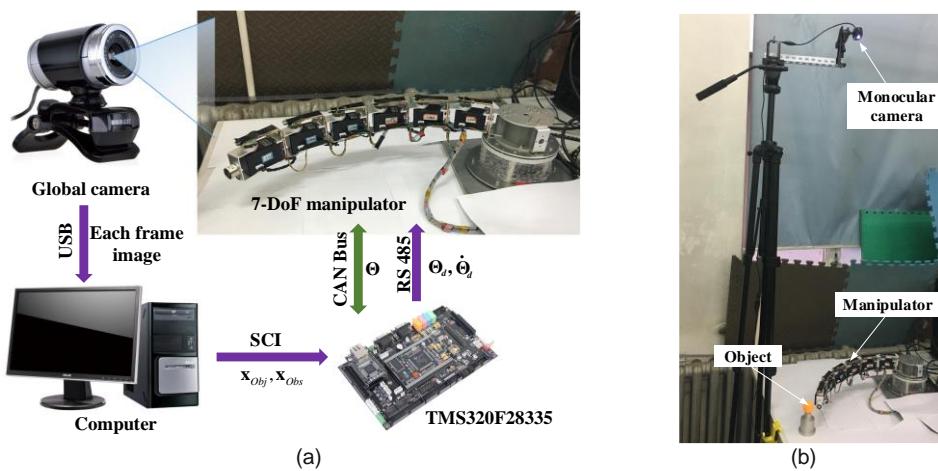


Fig. 6. Experimental setup. (a) Principle. (b) Experimental platform.

vector of obstacle avoidance expressed as follows:

$$\nabla \mathbf{H} = (\mathbf{J}_{o1}^* \dot{\mathbf{x}}_{ok1}^A, \mathbf{J}_{o2}^* \dot{\mathbf{x}}_{ok2}^A, \dots, \mathbf{J}_{ow}^* \dot{\mathbf{x}}_{okW}^A)^T, \quad (27)$$

where  $\mathbf{J}_{oj}^*$  is the pseudo-inverse of the critical point Jacobian for

avoiding the  $j$ -th obstacle based on the DLS method,  $\mathbf{J}_{oj}^* = \mathbf{J}_{oj}^T (\mathbf{J}_{oj} \mathbf{J}_{oj}^T + \lambda \mathbf{I}_o)^{-1}$ ,  $j = 1, 2, \dots, W$ .  $\dot{\mathbf{x}}_{okj}^A$  is the specified obstacle avoidance point velocity.

In each sampling period, the total energy for the manipulator arms will become

$$\mathbf{E}_{Total}^A \triangleq \mathbf{\Lambda}_{Const} = diag(\Lambda_1, \Lambda_2, \dots, \Lambda_W), \quad (28)$$

where  $\Lambda_j$  is the total energy to avoid the  $j$ -th obstacle for the manipulator arms and defined as

$$\Lambda_j = \underbrace{\vartheta(d_\alpha) \cdot E_{kj}^A}_{\tilde{E}_{kj}^A} + \underbrace{[1 - \vartheta(d_\alpha)] \cdot E_{sj}^A}_{\tilde{E}_{sj}^A}, \quad (29)$$

where  $E_{kj}^A$  and  $E_{sj}^A$  are defined as the constant energies in real time, where  $E_{sj}^A \triangleq E_{kj}^A$ .  $\tilde{E}_{kj}^A$  is the RKE toward the  $j$ -th obstacle.

$\tilde{E}_{sj}^A$  denotes the EM.  $\tilde{E}_{kj}^A$  and  $\tilde{E}_{sj}^A$  are expressed as

$$\tilde{E}_{kj}^A = \frac{1}{2} (\dot{\mathbf{x}}_{i,m}^A - \dot{\mathbf{x}}_{obs,j}^A)^T (\dot{\mathbf{x}}_{i,m}^A - \dot{\mathbf{x}}_{obs,j}^A), \quad (30)$$

$$\|\dot{\mathbf{x}}_{i,m}^A - \dot{\mathbf{x}}_{obs,j}^A\| = \sqrt{2\vartheta(d_\alpha) \cdot E_{kj}^A}, \quad (31)$$

$$E_{kj}^A = \frac{1}{2} (\dot{\mathbf{x}}_{i,m}^A - \dot{\mathbf{x}}_{obs,j}^A)^T (\dot{\mathbf{x}}_{i,m}^A - \dot{\mathbf{x}}_{obs,j}^A), \quad (32)$$

$$\tilde{E}_{sj}^A = (1 - \vartheta(d_\alpha)) E_{sj}^A, \quad (33)$$

where  $(\dot{\mathbf{x}}_{i,m}^A - \dot{\mathbf{x}}_{obs,j}^A)$  is the constant relative velocity between the  $i$ -th critical point of  $m$ -th arm and the  $j$ -th obstacle without considering obstacle avoidance at each sampling period,  $i = 1, 2, \dots, 5$ .  $(\dot{\mathbf{x}}_{obs,j}^A - \dot{\mathbf{x}}_{i,m}^A) = \dot{\mathbf{x}}_{obj}^A$  represents the actual relative (or specified) velocity to avoid the  $j$ -th obstacle.  $\vartheta(d_\alpha)$  refers to an S-function similar to  $f(d)$ , and  $\hat{\theta}_2$  in  $\vartheta(d_\alpha)$  indicates the control parameter of the transformation rate for the conversion between the RKE and the EM.  $d_\alpha$  denotes the distance between the critical points of arm and the obstacles.  $\vartheta(d_\alpha)$  and  $d_\alpha$  are defined as

$$\vartheta(d_\alpha) = \frac{1}{2} \left\{ \tanh \left[ \hat{\theta}_2 \cdot \left( d_\alpha (\mathbf{x}_{i,m}^A, \mathbf{x}_{obs,j}^A) - d_\beta (\mathbf{x}_{i,m}^A, \mathbf{x}_{obs,j}^A) \right) \right] + 1 \right\}, \quad (34)$$

$$d_\alpha (\mathbf{x}_{i,m}^A, \mathbf{x}_{obs,j}^A) = \|\mathbf{x}_{i,m}^A - \mathbf{x}_{obs,j}^A\| = \left[ \sum_{k=1}^M (\mathbf{x}_{i,m}^{A,k} - \mathbf{x}_{obs,j}^{A,k})^2 \right]^{\frac{1}{2}} \quad (35)$$

where  $d_\beta$  is the limited distance between the critical points of arm and the obstacles.  $\mathbf{x}_{i,m}^A$  is the position of the  $i$ -th critical point of the  $m$ -th arm.  $\mathbf{x}_{obs,j}^A$  is the position of the  $j$ -th obstacle. Combined with the single neuron adaptive PID model, the *real-time kinematic control strategy* is built to achieve the fast, smooth, and collision-free tracking of redundant robot manipulators in a time-varying environment, as shown in Fig. 5.

#### IV. EXPERIMENTAL VERIFICATION

##### A. Experimental Setup

The principle of the experimental setup in Fig. 6(a) can be briefly described as follows. 1) The global camera transfers the observed each frame image to the computer by using a universal serial bus (USB) in real time. The computer extracts the position information of the object and obstacles from each frame image and transfers these position information to the TMS320F28335 controller through a serial communication interface (SCI) bus. The vision processing procedures in the computer are developed based on morphology using the Open Source Computer Vision Library, and the detailed processing

flow is provided in Fig. 5. 2) The state feedback data  $\Theta$  of the manipulator run on a controller area network (CAN) bus. The command data  $\Theta_d$  and  $\dot{\Theta}_d$  control the manipulator motion by using the RS485 bus. The baud rates of the USB, CAN bus, SCI bus, and RS485 bus are set as 1 MHz. 3) The TMS320F28335 controller receives the data ( $\mathbf{x}_{obj}$ ,  $\mathbf{x}_{obs}$ , and  $\Theta$ ). Meanwhile, the controller with a dominant frequency of 150 MHz carries and executes the program, including the forward kinematics, inverse kinematics, motion planning, and obstacle avoidance algorithm. Figure 6(b) shows the experimental platform.

TABLE I

PARAMETERS IN THE 7-DOF MANIPULATOR

Parameters	$l_1$	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$	$l_7$
Value (mm)	118.0	88.0	88.0	88.0	88.0	88.0	57.85

##### B. Static Object-Tracking Experiment in Obstacle-Free Case

The parameters of 7-DoF manipulator are presented in Table I. The maximum velocity of the EE is 3.22 m/s at the designed performance limit of the manipulator. For the safety of the experimental process, the maximum velocities of the EE and the tracked object are artificially limited to 0.7 m/s and 0.4 m/s, respectively. To compare the real-time tracking performance of different methods fairly in path planning, the initial configurations of the manipulator and the tracked states are set with the same values as shown in Fig. 7(a), and all parameter settings of the methods used for comparison with the proposed unsupervised single neuron adaptive PID model have the same  $\Delta X$  at the first iteration,  $\Delta X = (-20.8 \text{ mm}, 6.34 \text{ mm}, 0.03 \text{ rad})^T$ . The convergence analysis of the Hebbian-based PCA posits that the learning rates  $\eta$  in the proposed method should be set small. Meanwhile, the smooth movement of the manipulator from the static state is guaranteed by conducting online learning prior to the experiment to determine the relevant parameters of the adaptive PID in Table II. Thus, the initial  $\Delta X$  of the proposed method is near 0,  $\Delta X = (-0.000041 \text{ mm}, 0.00024 \text{ mm}, 0.028 \text{ rad})^T$ .

Comparative planning methods include the traditional fixed proportion-based, classical fixed clamping-based, advanced virtual controller-based, and proposed methods. Considering that path planning is regarded as a control problem in this paper, the classical PID and sliding mode controls [37], which are widely used in the robotic field, are also considered the comparative control methods to further show the performance of the adaptive single neuron PID in path planning. The fixed proportion-based method can generate the real-time tracking path by using (2). The detailed descriptions for the fixed clamping-and virtual controller-based methods are offered in [31] and [21], respectively. The different methods are considered to be convergent when the position error is less than 4 mm and the attitude error is less than 0.02 rad. The reason is twofold: one is that the tolerable errors are far less than the initial pose errors between EE and the object, and the other is that the errors are difficult to completely converge to 0 on the actual experimental platform.

The comparative experiments for tracking a static object are implemented in an obstacle-free case. Figure 7(b) shows the tracking process and final configuration of the manipulator based on the proposed method in the obstacle-free case. In Fig.

TABLE II  
PARAMETERS IN THE PROPOSED METHOD

Parameters	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\beta_1$	$\beta_2$	$\beta_3$	$\delta_1$	$\delta_2$	$\delta_3$	$\eta_{p1}$	$\eta_{p2}$	$\eta_{p3}$	$\eta_{d1}$	$\eta_{d2}$
Value	400	280	2.0	0.0025	0.015	1.0	0.0005	0.001	0.5	0.008	0.05	1.3	0.009	0.05
Parameters	$k_{p1}$	$k_{p2}$	$k_{p3}$	$k_{l1}$	$k_{l2}$	$k_{l3}$	$k_{d1}$	$k_{d2}$	$k_{d3}$	$\eta_{l1}$	$\eta_{l2}$	$\eta_{l3}$	$\eta_{d3}$	$\Delta T$
Value	$10^{-4}$	$10^{-4}$	0.05	$10^{-5}$	$10^{-4}$	0.06	$10^{-5}$	$2 \times 10^{-4}$	0.05	$3.2 \times 10^{-4}$	0.006	1.6	1.3	0.02

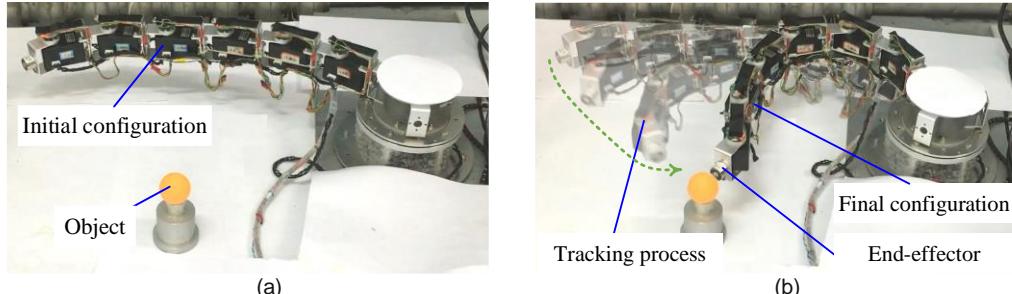


Fig. 7. Real-time tracking of a static object in an obstacle-free case. (a) Initial configuration,  $\Theta_{initial} = (-10^\circ, 10^\circ, 10^\circ, 10^\circ, 0^\circ, 20^\circ, 20^\circ)^T$ .  $\mathbf{X}_{obj} = (203\text{ mm}, 296\text{mm}, 1.535\text{ rad})^T$ . (b) Tracking process and final configuration of the manipulator based on the unsupervised single neuron adaptive PID model.

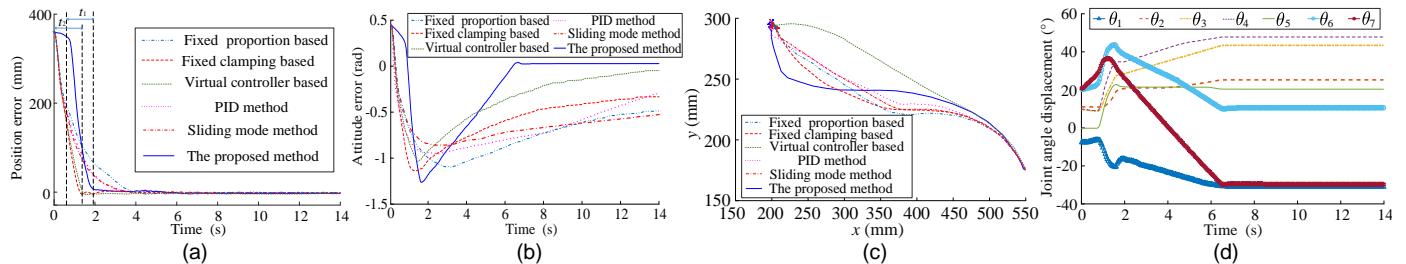


Fig. 8. Comparison of methods based on fixed proportion, fixed clamping, virtual controller, PID control, and sliding mode control. (a) Position error of the EE. (b) Attitude error of the EE. (c) Motion paths of the EE. (d) Joint angle displacement of the manipulator.

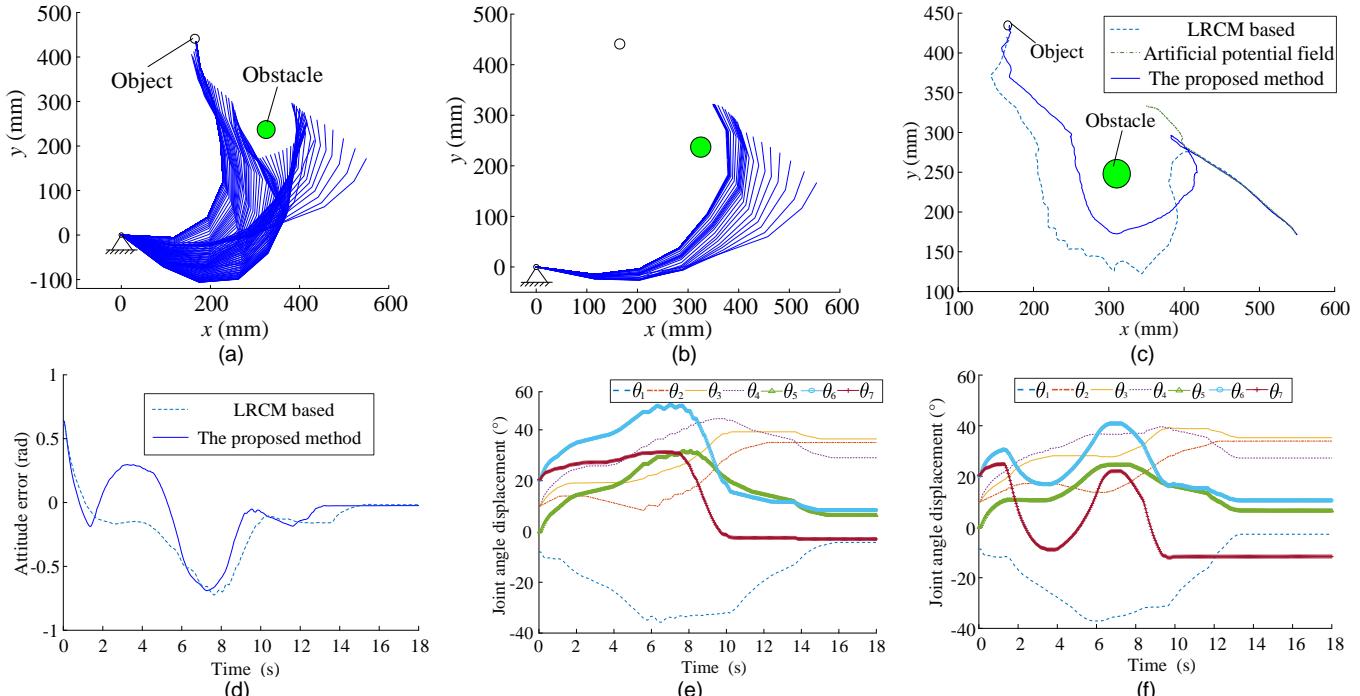


Fig. 9. Comparison of obstacle avoidance methods based on artificial potential field, LRCM, and the proposed LCE-based method.  $\hat{\theta}_1 = 0.6$ ,  $d_0 = 40\text{ mm}$ .  $\Theta_{initial} = (-10.0^\circ, 10.0^\circ, 10.0^\circ, 10.0^\circ, 0.0^\circ, 20.0^\circ, 20.0^\circ)^T$ . (a) Motion of the manipulator based on LCE. (b) Motion of the manipulator based on artificial potential field. (c) Motion paths. (d) Attitude changes. (e) Joint trajectories by LRCM. (f) Joint trajectories by LCE.

8(a), six methods can plan a path for the EE to arrive to the object position. For the EE position, the order of the convergence rates is as follows: virtual controller-based method, fixed

clamping-based method, proposed method, PID control, sliding mode control, and fixed proportion-based method. The required total time is 2.5, 3.5, 4, 4.5, 5.1, and 6.1 s, respectively. The

tracking speed of the fixed clamping-based method is constant, and will gradually decrease when the errors are less than the clamping value. For the fixed proportion- and sliding model-based methods, the convergence rates of the pose errors mainly depend on the fixed gain coefficient (Eq. (2) and Reference [37]). Considering the existence of integral and differential terms in the PID control and virtual controller, the error is continuously accumulated that results in their fast convergence speeds. For the proposed method, the convergence rate is slightly slow in terms of the total required time. However, in the initial stage, the slope of the proposed method is gentler than those of other methods. Then the slope increases gradually through the online learning of the Hebbian-based PCA. When the position error is small, the convergence rate decreases. Such a convergence rate benefits the smooth motions of the EE and joints in real time as shown in Figs. 8(a) and (d), and ensures safe operations in practical applications. On the contrary, the time used in fast tracking of the EE is short in the proposed method, that is,  $t_1 < t_2$  ( $t_1=1.3$  s,  $t_2=1.64$  s (virtual controller-based method)). This condition is due to the adjustment of parameters online rather than fixed gains in other methods despite the existence of integral and differential links for improving convergence.

In Fig. 8(b), the attitude error of the proposed method can converge to nearly 0 within 7.4 s. Nevertheless, the other five methods cannot achieve the convergence of attitude error within 14 s. The attitude in the proposed method can be adjusted adaptively through the online learning of the parameters. Consequently, the attitude converges quickly.

In Fig. 8(c), the motion path of the EE based on the online learning of the PCA is slightly longer than those of the other methods, which is caused by the smooth movement of joint trajectories, as shown in Fig. 8(d). The slightly long path has nearly no effect on its convergence rate, and the proposed method can enable the manipulator to achieve maximum speed smoothly in the initial stage. Such characteristics benefit the steady operation and rapid tracking of redundant manipulators in practical application.

The comprehensive analysis of Figs. 8(a)–(d) indicates that the proposed planning method can realize the smooth motion and fast convergence of a manipulator, and has a certain application value.

### C. Static Object-Tracking Experiments in One Static Obstacle Case

#### 1) Obstacle on the Moving Path of the EE

In Fig. 9, the obstacle avoidance methods for the EE, including the artificial potential field-based method [12], local rotation coordinate method (LRCM) [21], and proposed LCE-based method, are compared by tracking a static object. The object pose is  $\mathbf{X}_{obj} = (165 \text{ mm}, 441 \text{ mm}, 1.592 \text{ rad})^T$ . The position of the obstacle is  $\mathbf{X}_{obs} = (325 \text{ mm}, 237 \text{ mm})^T$ . In Figs. 9(a)–(c), the proposed method and the LRCM can achieve a large angle turn to achieve the successful tracking of the object by the manipulator. However, the artificial potential field method can only realize a small angle turn in such a situation. In Fig. 9(c), the obstacle avoidance path planned by the proposed method is shorter and smoother than that using the LRCM. For the curves of the paths and joint trajectories, the fluctuations on the path planned by LRCM result in the concussion of the joint, as il-

lustrated in Figs. 9(e) and (f). Meanwhile, the convergence rate of the attitude based on the proposed LCE method is considerably faster than those of other methods as shown in Fig. 9(d). Therefore, the proposed method can generate a smooth and large angle-turning path for the obstacle avoidance of the EE.

#### 2) Obstacle Not on the Moving Path of the EE

To avoid the obstacles that move toward any other part of the manipulator, the cosine curve model [20] and the proposed method are compared by means of the null space of the redundant manipulator. In Figs. 10(a) and (b), the motions of the manipulator are nearly the same based on the cosine curve model and the proposed method. The curves of the joint trajectories are similar as shown in Figs. 10(c) and (d). Thus, the LCE-based method can endow the redundant manipulator with nearly the same performance of obstacle avoidance for the arms of manipulators. However, the proposed method theoretically belongs to a continuous smooth process for the conversion between different energies, which attributes to the continuous derivable function  $\theta(d_\alpha)$  in (34) rather than the piecewise functions of the cosine curve model. In the cosine curve model, two distances need to be clearly defined. One determines where the obstacle avoidance motion should be started, and the other determines where the maximum is reached. The changing rate of obstacle avoidance motion between 0 and the maximum is simultaneously determined by the difference between the values of the two distances [20]. By contrast,  $\hat{\theta}_2$  in the proposed method individually determines the transformation rate for the conversion between different energies, and  $d_\beta$  limits the minimum proximity between the critical points of arms and the obstacles. The program can also automatically calculate the obstacle avoidance motion without considering where to start. Therefore, the proposed method with a similar performance to that of the cosine curve model is concise and simple in theory and application.

### D. Static Object-Tracking Experiments in Multiple-Static Obstacles Case

In Figs. 11(a) and (b), the LRCM and the proposed method are used for the real-time tracking of a static object in a narrow space. For the motion of the manipulator, the proposed method makes the EE move around the obstacles and any other part of the manipulator not contact with the obstacles. In the initial motion of the manipulator, the path planned by the proposed method maintains a certain distance between the EE and obstacle1 and is smooth. However, the path planned by the LRCM moves the EE near obstacle1, and the arms are near obstacle4, which remains unsafe for the manipulator although the obstacles can be bypassed. When the parameters are set unsuitably, the motion of avoiding the obstacles is also potentially dangerous. The adjustment of the attitude produces small fluctuations in tracking and contributes to the motion of the manipulator in such a space, as shown in Figs. 11(b)–(d).

### E. Moving Object-Tracking Experiments in Multiple-Dynamic Obstacles Case

The final experiment is used to verify the feasibility of the proposed *real-time kinematic control strategy* in a time-varying environment. The initial layout of obstacles and object are shown in Fig. 12(a). The object remains static and the obstacles move continuously in the first 9 s. When the static object is

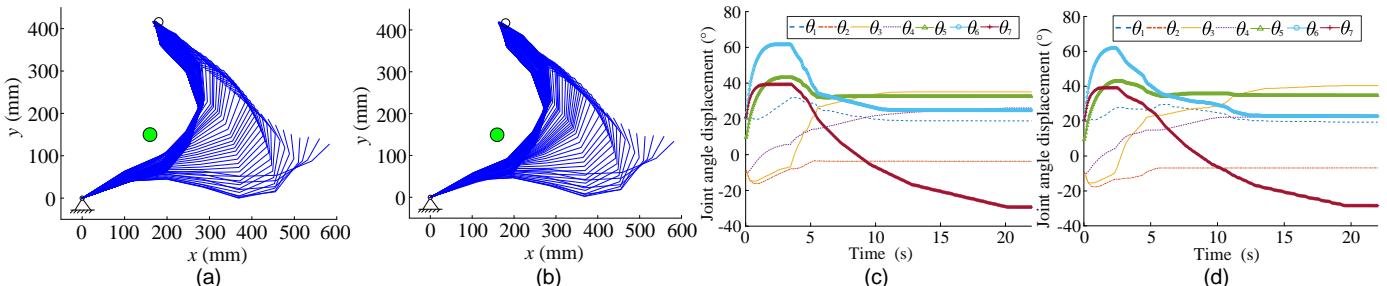


Fig. 10. Comparison of obstacle avoidance methods for the arms of the manipulator.  $\Theta_{initial} = (20.0^\circ, -10.0^\circ, -10.0^\circ, -10.0^\circ, 10.0^\circ, 10.0^\circ, 20.0^\circ, 20.0^\circ)^T$ ,  $\mathbf{X}_{obj} = (182 \text{ mm}, 416 \text{ mm}, 1.744 \text{ rad})^T$ ,  $\mathbf{X}_{obs} = (160 \text{ mm}, 150 \text{ mm})^T$ .  $\hat{\theta}_2 = 0.2$ ,  $d_\beta = 90 \text{ mm}$ . (a) Motion of null space for obstacle avoidance based on cosine curve model. (b) Motion of null space for obstacle avoidance based on LCE. (c) Joint trajectories by cosine curve model. (d) Joint trajectories by LCE.

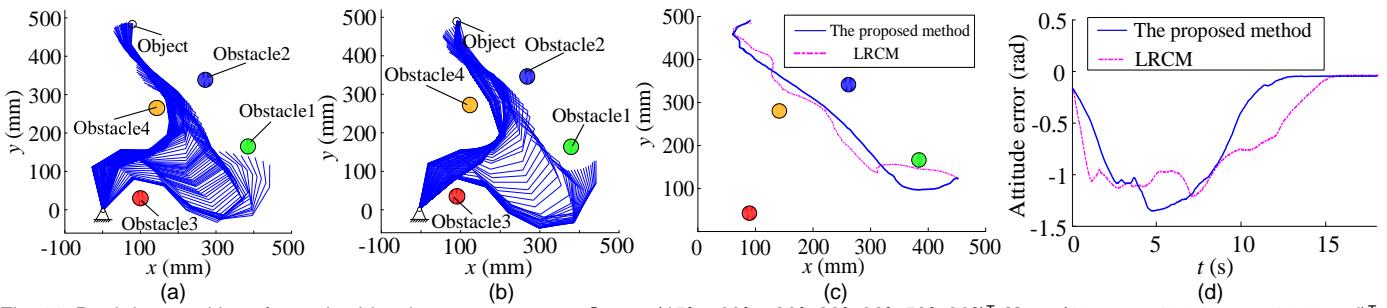


Fig. 11. Real-time tracking of a static object in a narrow space.  $\Theta_{initial} = (45^\circ, -60^\circ, -30^\circ, 20^\circ, 30^\circ, 50^\circ, 30^\circ)^T$ ,  $\mathbf{X}_{obj} = (105 \text{ mm}, 483.13 \text{ mm}, 1.396 \text{ rad})^T$ ,  $\mathbf{X}_{obs1} = (386 \text{ mm}, 169 \text{ mm})^T$ ,  $\mathbf{X}_{obs2} = (272.7 \text{ mm}, 345 \text{ mm})^T$ ,  $\mathbf{X}_{obs3} = (95 \text{ mm}, 33.8 \text{ mm})^T$ ,  $\mathbf{X}_{obs4} = (132.63 \text{ mm}, 272.6 \text{ mm})^T$ .  $\hat{\theta}_1 = 0.6$ ,  $d_0 = 40 \text{ mm}$ .  $\hat{\theta}_2 = 0.2$ ,  $d_\beta = 90 \text{ mm}$ .  $\mathbf{K} = (12, 38, 74, 38)$ . (a) Tracking of the manipulator based on LRCM. (b) Tracking of the manipulator based on the proposed LCE-based method. (c) Motion paths of the EE. (d) Attitude adjustment during tracking.

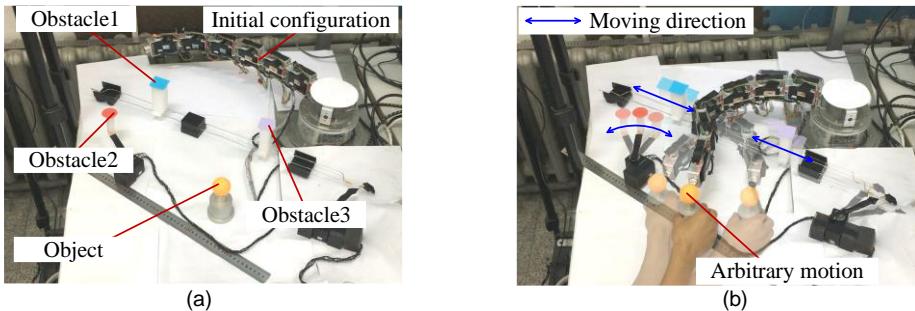


Fig. 12. Real-time tracking based on the *real-time kinematic control strategy* in a multiple-dynamic obstacle environment. (a) Initial layout. (b) Motions of the object, the obstacles, and the manipulator.

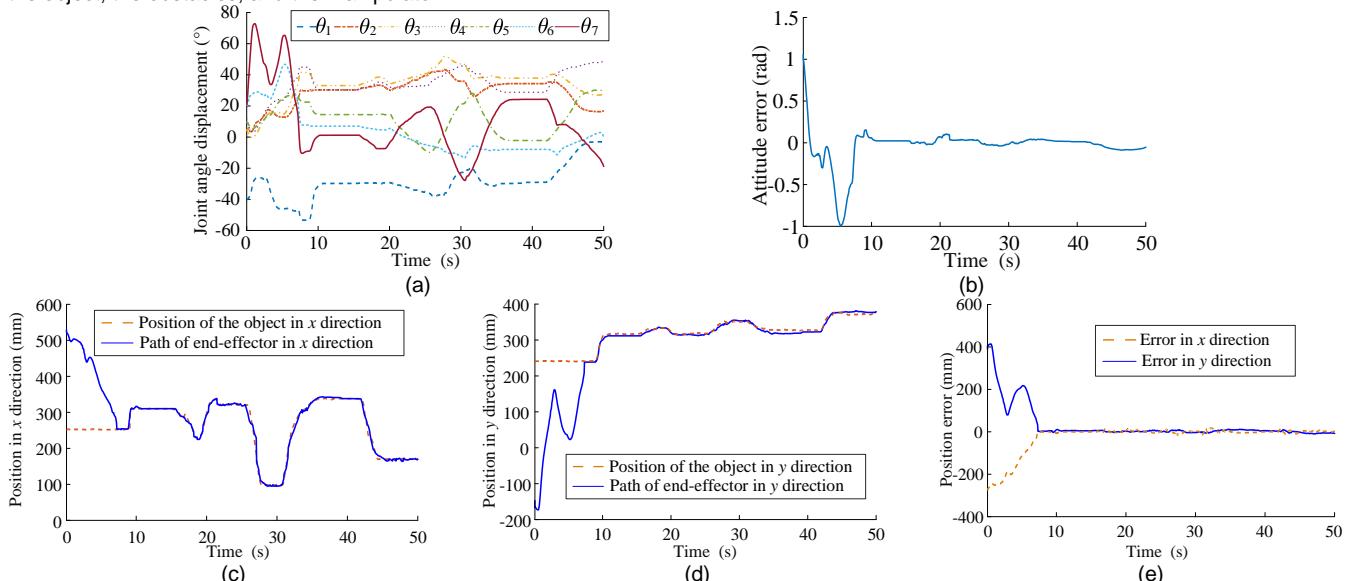


Fig. 13. Joint trajectories and EE pose changes of the manipulator in real-time tracking.  $\Theta_{initial} = (-40.0^\circ, 5.0^\circ, 5.0^\circ, 10^\circ, 10^\circ, 20^\circ, 20^\circ)^T$ . Attitude angle is  $1.535 \text{ rad}$ .  $\hat{\theta}_1 = 0.6$ ,  $d_0 = 40 \text{ mm}$ .  $\hat{\theta}_2 = 0.2$ ,  $d_\beta = 90 \text{ mm}$ .  $\mathbf{K} = (38, 74, 38)$ . (a) Joint trajectories. (b) Attitude error of EE. (c) Position in x direction. (d) Position in y direction. (e) Position error.

tracked successfully, the object is then moved arbitrarily, as shown in Fig. 12(b). In the tracking process, the attitude of the EE is constant (i.e., 1.535 rad). Figure 13(a) shows the joint trajectories, and the adjustment of  $\theta_7$  is fast and in a large range. The other angles ( $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ ) vary with minor adjustment. Such changes indicate that the EE is moving around obstacle1 while adjusting the attitude, as shown in Fig. 13(b). The attitude error is 1.011 rad in the initial tracking of the moving object. Figures 13(c) and (d) present the position tracking of the EE. Figure 13(e) displays the corresponding position errors. In the initial tracking of the moving object, the error in  $x$  direction is -269.7 mm, and the error in  $y$  direction is 406.75 mm. Although the pose errors of the EE are large at the initial stage of tracking, they are gradually diminished or near zero through online learning based on the PCA. Therefore, the proposed *real-time kinematic control strategy* based on the LCE and PCA is feasible and can be used in the rapid real-time tracking of a moving object in the time-varying environment.

## V. CONCLUSION

This paper presents a simple and practical kinematic control strategy for redundant manipulators to track a moving object rapidly in a time-varying environment. An obstacle avoidance method based on the LCE is proposed to avoid contact between the manipulator and the obstacles. This method transforms the energies smoothly between the ETO and the EAO or between the RKE and the EM at each sampling period. Such transformations make the manipulator maintain a safe distance from obstacles placidly and the EE produce a large angle-turning steadily in accordance with the experimental comparison results. In real-time path planning, a single neuron adaptive PID model using the unsupervised Hebbian-based PCA is proposed to increase the fast tracking performance of redundant manipulators via online learning. Combined with the proposed obstacle avoidance method, a real-time kinematic control strategy is established to achieve moving object tracking in the presence of multiple-dynamic obstacles. The experimental comparison results confirm that the proposed strategy is feasible and has fast convergence.

## REFERENCES

- [1] L. Jin, S. Li, X. Luo, Y. M. Li, and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 3812-3821, Sept. 2018.
- [2] G. Du, M. Chen, C. B. Liu, B. Zhang, and P. Zhang, "Online robot teaching with natural human-robot interaction," *IEEE Trans. Ind. Electron.*, no. 99, pp. 1-1, Dec. 2018.
- [3] C. G. Yang, H. W. Wu, Z. J. Li, W. He, N. Wang, and C. Y. Su, "Mind Control of A Robotic Arm with Visual Fusion Technology," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 3822-3830, Sept. 2018.
- [4] V. Chawda and G. Niemeyer, "Toward torque control of a KUKA LBR IIWA for physical human-robot interaction," in *Proc. IROS*, Sept. 24-28 2017, pp. 6387-6392.
- [5] A. D. Wilson, J. A. Schultz, A. R. Ansari, and T. D. Murphey, "Dynamic task execution using active parameter identification with the Baxter research robot," *IEEE Trans. Autom. Sci. & Eng.*, no. 99, pp. 1-7, Jan. 2017.
- [6] A. Wahrburg, S. Zeiss, B. Matthias, J. Peters, and H. Ding, "Combined pose-wrench and state machine representation for modeling Robotic Assembly Skills," in *Proc. IROS*, Sept. 28-Oct. 2 2015, pp. 852-857.
- [7] H. P. N. Dong, M. Hoffmann, A. Roncone, U. Pattacini, and G. Metta, "Compact real-time avoidance on a humanoid robot for human-robot interaction," in *Proc. ACM/HRI*, Jan. 2018, pp. 416-424.
- [8] T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, vol. C-32, no. 2, pp. 108-120, 2006.
- [9] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. ICRA*, Apr. 24-28 2002, pp. 995-1001.
- [10] B. Burns, O. Brock, "Toward optimal configuration space sampling," *Robot. Sci. Syst.*, pp. 105-112, 2005.
- [11] A. Reiter, A. Müller, and H. Gatringer, "On higher-order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators," *IEEE Trans. Ind. Inform.*, vol. 14, no. 99, pp. 1-1, Apr. 2018.
- [12] Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. ICRA*, Jan. 2003, pp. 90-98.
- [13] D. Guo and Y. Zhang, "Acceleration-level inequality-based man scheme for obstacle avoidance of redundant robot manipulators," *IEEE Trans. on Ind. Electron.*, vol. 61, no. 12, pp. 6903-6914, Dec. 2014.
- [14] J. O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Trans. Rob. Autom.*, vol. 8, no. 3, pp. 338-349, Jun. 1992.
- [15] H. J. S. Feder and J. J. E. Slotine, "Real-time path planning using harmonic potentials in dynamic environments," in *Proc. ICRA*, Apr. 25-25 1997, pp. 874-881, vol. 1.
- [16] M. Ouyang, W. Sun, S. N. Liu, and P. F. Yang, "On joint obstacle avoidance based on artificial potential field for duct cleaning robot," in *Proc. IEICE*, Apr. 2012, pp. 1045-1049.
- [17] F. Wu, A. Vibhute, G. Song Soh, and K. L. Wood, "A compact magnetic field-based obstacle detection and avoidance system for miniature spherical robots," *Sensors*, vol. 17, no. 6, May 2017.
- [18] A. Mclean, "The virtual springs method: Path planning and collision avoidance for redundant manipulators," *Int. J. Robot. Res.*, vol. 15, no. 4, pp. 300-319, Aug. 1996.
- [19] Q. Liu, Y. Yu, Z. H. Xu, and L. Y. Su, "Obstacle avoidance of underactuated robots based on virtual spring-damper model," in *Proc. WCICA*, 2008, pp. Jun. 25-27 3213-3217.
- [20] T. Petrić, A. Gams, N. Likar, and L. Zlajpah, "Obstacle avoidance with industrial robots," *Motion and Operation Planning of Robotic Systems*, pp. 113-145, Mar. 2015.
- [21] H. Z. Jin, H. Zhang, Z. X. Liu, D. C. Yang, D. Y. Bie, H. Zhang, G. Li, Y. H. Zhu, and J. Zhao, "A synthetic algorithm for tracking a moving object in a multiple-dynamic obstacles environment based on kinematically planar redundant manipulators," *Math. Probl. Eng.*, vol. 2017, no. 3, pp. 1-15, Apr. 2017.
- [22] G. Ping, B. Wei, X. L. Li, and X. Luo, "Real time obstacle avoidance for redundant robot," in *Proc. ICMA*, Aug. 9-12 2009, pp. 223-228.
- [23] Y. Li, H. Liu, and D. Ding, "Motion planning for robot manipulators among moving obstacles based on trajectory analysis and waiting strategy," in *Proc. Sice Conference*, Aug. 20-22 2008, pp. 3020-3025.
- [24] O. Brock, O. Khatib, and S. Viji, "Task-consistent obstacle avoidance and motion behavior for mobile manipulation," in *Proc. ICRA*, May 11-15 2002, pp. 388-393, vol. 1.
- [25] R. Colbaugh, K. Glass, and H. Seraji, "Obstacle avoidance for redundant robots using configuration control," *J. Field Robot.*, vol. 6, no. 6, pp. 721-744, 2010.
- [26] J. S. Arévalo, G. Poisson, and P. Vieyres, "A new kinematic formulation of the RCM constraint for redundant torque-controlled robots," in *Proc. IROS*, Sept. 24-28 2017.
- [27] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, "Simultaneous trajectory planning and tracking using an MPC method for cyber-physical systems: a case study of obstacle avoidance for an intelligent vehicle," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4273-4283, Sept. 2018.
- [28] M. Duguleana, F. G. Barbuceanu, A. Teirelbar, and G. Mogan, "Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning," *Robot. Comput. Integrat. Manuf.*, vol. 28, no. 2, pp. 132-146, Apr. 2012.
- [29] W. Guan, Z. Weng, and J. Zhang, "Obstacle avoidance path planning for manipulator based on variable-step artificial potential method," in *Proc. CDC*, May 23-25 2015, pp. 4325-4329.
- [30] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. R. Walter, "Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms," in *Proc. IROS*, Sept. 25-30 2011, pp. 4307-4313.
- [31] R. Samuel Buss and J.S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics Gpu & Game Tools*, vol. 10, no. 3, pp. 37-49, Jan. 2005.

- [32] A. S. Deo and I. D. Walker, "Overview of damped least-squares methods for inverse kinematics of robot manipulators," *J. Intell. Robot. Syst.*, vol. 14, no. 1, pp. 43-68, Mar. 1995.
- [33] J. J. Craig, "Introduction to robotics: Mechanics and control," *Pearson Education, Inc.*, 2005.
- [34] E. Oja, "A Simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, no. 3, pp. 267-273, Nov. 1982.
- [35] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Netw.*, vol. 2, no. 6, pp. 459-473, Dec. 1989.
- [36] S. Haykin, "Neural networks and learning machines," *Prentice-Hall*, 2008.
- [37] J. J. Slotine and S. S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators," *Internat. J. Control.*, vol. 38, no. 2, pp. 465-492, Jun. 22-24 1982.



**Hui Zhang** received the B.S. in mechanical engineering from the Zhongyuan University of Technology, Zhengzhou, China, in 2013, and the M.S. degrees in mechanical engineering from Harbin Institute of Technology (Shenzhen), Shenzhen, China, in 2015.

He is currently working towards the PhD degree at the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His research interests include robotic automated planning and cooperative manipulation.



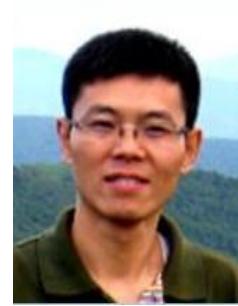
**Hongzhe Jin** (M'10) received the B.S. degree in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 1999, and the Ph.D. degree in electronic engineering from the Pusan National University, Busan, South Korea, in 2009.

He is currently an Associate Professor with the School of Mechatronics Engineering, HIT. His research interests include modeling, analysis, control, identification of complex systems, and robotics.



**Zhangxing Liu** received the B.S. in mechanical engineering from the Chongqing University, Chongqing, China, in 2014, and the M.S. degrees in mechanical engineering from Harbin Institute of Technology, Harbin, China, in 2016.

He is currently working towards the PhD degree at the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His research interests include system identification, robotic control and manipulation.



**Yubin Liu** received the B.S. and M.S. degrees in mechanical engineering from the Harbin University of Commerce, Harbin, China, in 1999 and 2002, and received the Ph.D. degree in mechanical engineering from Harbin Institute of Technology (HIT), Harbin, China, in 2008.

He is currently an Associate Professor with the State Key Laboratory of robotics and systems, HIT, Harbin, China. His research interests include bionic robots and flexible joints with variable stiffness.



**Yanhe Zhu** (M'08) received the B.S. and Ph.D. degrees in Mechatronics Engineering from Harbin Institute of Technology (HIT), Harbin, China, in 1998 and 2004, respectively.

He is currently a Professor and a Ph.D. supervisor of the School of Mechatronics Engineering, HIT. He also is the deputy director of the State Key Laboratory of Robotics and System. His research interests include reconfigurable modular robots and robotic exoskeletons.



**Jie Zhao** (M'05) received the B.S. and Ph.D. degrees in mechatronics engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 1990 and 1996, respectively.

He is currently a Professor with the School of Mechatronics Engineering, HIT. He is the director of the State Key Laboratory of Robotics and System, HIT.

His research interests include industrial robots and bionic robots. Dr. Zhao is the leader of the Subject Matter Expert Group of Intelligent Robot in National 863 Program supervised by the Ministry of Science and Technology of China.