

## Research Article

# A Synthetic Algorithm for Tracking a Moving Object in a Multiple-Dynamic Obstacles Environment Based on Kinematically Planar Redundant Manipulators

Hongzhe Jin,<sup>1</sup> Hui Zhang,<sup>1</sup> Zhangxing Liu,<sup>1</sup> Decai Yang,<sup>2</sup> Dongyang Bie,<sup>1</sup> He Zhang,<sup>1</sup> Ge Li,<sup>1</sup> Yanhe Zhu,<sup>1</sup> and Jie Zhao<sup>1</sup>

<sup>1</sup>State Key Laboratory of Robotics and Systems, School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, Heilongjiang 150080, China

<sup>2</sup>Aerospace System Engineering Shanghai, Shanghai 201109, China

Correspondence should be addressed to Hongzhe Jin; hongzhejin@hit.edu.cn, Ge Li; lige@hit.edu.cn, Yanhe Zhu; yhzhu@hit.edu.cn, and Jie Zhao; jzhao@hit.edu.cn

Received 3 November 2016; Accepted 14 December 2016; Published 30 April 2017

Academic Editor: Yuri Vladimirovich Mikhlin

Copyright © 2017 Hongzhe Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a synthetic algorithm for tracking a moving object in a multiple-dynamic obstacles environment based on kinematically planar manipulators. By observing the motions of the object and obstacles, Spline filter associated with polynomial fitting is utilized to predict their moving paths for a period of time in the future. Several feasible paths for the manipulator in Cartesian space can be planned according to the predicted moving paths and the defined feasibility criterion. The shortest one among these feasible paths is selected as the optimized path. Then the real-time path along the optimized path is planned for the manipulator to track the moving object in real-time. To improve the convergence rate of tracking, a virtual controller based on PD controller is designed to adaptively adjust the real-time path. In the process of tracking, the null space of inverse kinematic and the local rotation coordinate method (LRCM) are utilized for the arms and the end-effector to avoid obstacles, respectively. Finally, the moving object in a multiple-dynamic obstacles environment is thus tracked via real-time updating the joint angles of manipulator according to the iterative method. Simulation results show that the proposed algorithm is feasible to track a moving object in a multiple-dynamic obstacles environment.

## 1. Introduction

Planar manipulators can accomplish many tasks with high efficiency for a long time in specific/dangerous environments. Many researches have integrated vision technologies into manipulators system to further improve their intelligence, involving object perception, path prediction, action decision, path planning and execution, and so on [1–3]. The planar redundant manipulators with vision technologies can be used to inspect the facilities in complex environments or narrow spaces, such as ITER relevant inspection robots [4, 5] and SAFIRE robots [6]. The control systems of the redundant manipulators can be designed based on dynamics or kinematics [7]. The system based on dynamics has better real-time performance, but it heavily depends on the hardware system and the level of the designer. Besides, the hardware

system is closed, which limits the further development. The system based on kinematics is open and easier to achieve a complicated motion of the manipulator via the upper planning. And such system plays key roles in realizing intelligent control of robot manipulator. So the focus of this paper will be devoted to the study of kinematically planar redundant manipulator.

Typical researches on the manipulators with vision system aim at real-time tracking moving objects in the varying environment. Allen et al. introduce a control algorithm to predict ahead the trajectory of the object in real-time, and the tracking can be performed by kinematic transformation computation [8]. Husain et al. present an automated system with a Microsoft Kinect sensor to track the moving object via reducing the distance between the object and the tripper mounted on the 7 degrees of freedom (DOF) robotic arm

in real-time [9]. A simulation research, developed by Wei et al. [10], predicts the pose of the object in the next period by using the observed feature points at first; then the manipulator is controlled to achieve the real-time tracking. In [11], the authors propose an adaptive neuropredictive control method for redundant robot manipulators, which includes the prediction and obstacles avoidance, but the tracked path is planned offline and defined to be not affected by the dynamic obstacles.

Nevertheless, to track a moving object in a multiple-dynamic obstacles environment, several technologies must be considered at the same time, containing moving path prediction, path planning, and real-time obstacles avoidance. Moreover, these technologies must be synthesized reasonably to ensure that the moving object can be grasped successfully. In the literatures, however, although the existing methods have considered the prediction, path planning, and the obstacles avoidance, the failure of tracking always occurs in a multiple-dynamic obstacles environment. The main reason, on the one hand, is mainly due to the insufficient synthesis of these technologies. On the other hand, the impacts of the dynamic obstacles on the moving path of the end-effector and the behavioral constraints of the manipulator are not taken into account fully, since, for the spatial manipulators, they can bypass obstacles by its own joint rotation in 3D space. But for the planar manipulators, the dynamic obstacles cannot be bypassed because the joints of planar manipulators only rotate in a plane, which may result in the moving object not being grasped. Hence, considering these deficiencies of the planar manipulators in practice, a synthetic algorithm involving path prediction, path planning, and obstacles avoidance need to be further improved to track a moving object in a multiple-dynamic obstacles environment. Furthermore, in order to ensure the effectiveness of the synthetic algorithm, some developments on path prediction, path planning, and obstacles avoidance are also required, respectively.

In path prediction, the future positions of the object are always predicted first to achieve a moving object tracking. The common prediction methods contain Kalman filter [12, 13], learning algorithm [14], biomimetic approach [15], and polynomial fitting method [16]. Kalman filter can predict the current path based on the past information. Learning approach and biomimetic approach need a lot of offline training for many moving curves before prediction. Polynomial fitting method can predict many moving curves in the future by the polynomial calculation, and the error is small, but the motion paths are always considered as regular curves. Spline filter is an advanced method to smooth a large curve and it can ride the curve trend well [17], so Spline filter is raised to ride the trend of the moving paths for the first time in this paper. Then, polynomial fitting is introduced to predict the future moving paths of object and obstacles along their path trends.

On the basis of the predicted moving paths, the tracking path of the manipulator can be planned by A\* search method [18], the artificial potential field method [19], neural networks [20, 21], ant colony algorithm [22], or C-space method [23]. But they all have some limitations. Since the computational amount of A\* search method will increase sharply with the increase of space dimension, A\* search method is difficult

applied to plan paths in real-time. The artificial potential field method is suitable for dealing with dynamic obstacles, but it is easy to fall into local minimum point. Neural network method needs to be trained again when the environment changes. Ant colony algorithm has slow convergence speed and easily falls into local optimum. C-space method needs massive computation, but it is more intuitive and easier to plan a tracking path in a dynamic obstacles environment. So, C-space method is applied to plan a path for the end-effector to track the moving object in this paper. Besides, most of these methods are based on the hypothesis that the positions on the planned path could be reached by manipulators. However, in a multiobstacles environment, some positions on the planned path cannot be reached for the planar manipulator due to obstacles limiting the joint angles, which may result in the moving object not being grasped. That is, no criterion is set to judge whether the future position of the object along the planned path can be reached. Consequently, a feasibility criterion is defined to ensure that the path planned by C-space method is feasible for manipulator to grasp the object.

Although the tracking path is planned appropriately, the real-time obstacle avoidance for the end-effector and the manipulator arms, which is based on local obstacle avoidance approaches [24], still needs to be considered in the varying environment. Most of the local approaches aim at producing motions away from the obstacle for every point or each critical point on the manipulator when they are close to the obstacle in Cartesian space [25–28]. In the varying environment, the obstacles avoidance will be more complex. In [7], switching primary and secondary task and setting priorities are applied to avoid multiple obstacles. This paper will draw on this strategy and set the same priority for these obstacles to avoid the multidynamic obstacles in tracking the moving object at the same time.

Considering that the errors of predicting paths are small in a short time, the tracking path should be as short as possible and the tracking speed must be accelerated. In generally, the gain coefficient of the position error is constant [29]. To achieve fast tracking, a large gain coefficient is usually set, which results in large system gain. Besides, the joint driver overshoot and jitter will occur. Thus an adaptively adjusting method, in which gain coefficient varies with the change of the tracked object speed, is proposed to ease the process.

As previously described, some further studies still need to be done in moving path prediction, path planning, and the real-time obstacles avoidance. Meanwhile, these aspects must be synthesized effectively to track a moving object in a multiple-dynamic obstacles environment. Therefore, based on kinematically 6-DOF planar manipulator, this paper presents a synthetic algorithm to achieve a moving object tracking in a multiple-dynamic obstacles environment, which involves the moving path prediction, path planning, and the real-time obstacles avoidance. The algorithm applies the predicted moving paths of the object and obstacles to plan feasible paths, and an optimized path is selected among these feasible paths based on the shortest principle. A virtual controller is designed for the manipulator to adjust the tracking speed adaptively. Moreover, the local rotation coordinate method (LRCM), which is proposed by the authors, is used to

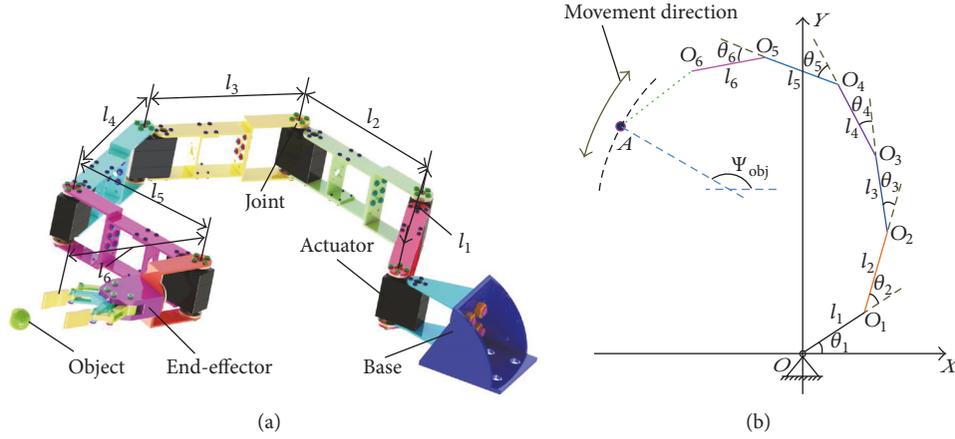


FIGURE 1: Real-time tracking a moving object in absence of obstacles. (a) 3D model of 6-DOF planar manipulator. (b) The simplified model of 6-DOF planar manipulator.

avoid the obstacles that get close to the end-effector, and the null space of inverse kinematics is adapted to avoid obstacles in the vicinity of the manipulator arms. The supplementary material, which can demonstrate the proposed method, is available online (see Supplementary Material available online at <https://doi.org/10.1155/2017/7310105>).

This paper is organized as follows. Section 2 describes the real-time path tracking approaches by utilizing iterative method in the absence of obstacles. The strategies of tracking a moving object in the multiple-dynamic obstacles environment are developed in Section 3, involving moving path prediction, the optimized path planning, obstacles avoidance, and the synthetic object tracking algorithm. And the corresponding simulations are presented in Section 4. The conclusion is drawn in Section 5.

## 2. Obstacles Free Case

**2.1. Based on the Common Method.** As shown in Figure 1(a), the 3D model of 6-DOF planar manipulator is presented. The manipulator can only work in a plane with six rotary joints and the axes of joints are parallel to each other. And it is obvious to see that each joint is driven by an actuator directly. For a given task, the motion path of the end-effector is planned in Cartesian space at first. Then the rotation of each joint can be obtained by the inverse kinematic. Finally, according to the planned path, the movement of the manipulator is realized by the position control of actuators. In this paper, to facilitate the modeling and analysis of kinematics in Cartesian space, a simplified model is established based on the 3D model of 6-DOF planar manipulator as shown in Figure 1(b). Each thin solid line expresses an arm of the manipulator. Also, the line  $O_5O_6$  indicates the end-effector. The Cartesian coordinate  $XOY$  is defined as the base coordinate system of the planar manipulator and the inverse kinematics of the 6-DOF planar manipulator is

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

where  $\dot{\mathbf{X}}$  is the velocity of end-effector in Cartesian space.  $\mathbf{X}$  is the position and attitude of end-effector,  $\mathbf{X} = (x, y, \Psi)^T$ ,  $\Psi$  is

the attitude of end-effector, and  $\Psi = \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6$ .  $\Theta$  is the joint velocity of manipulator in joint space.  $\Theta$  is the joint angle;  $\Theta = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)^T$ .  $J^*$  is the pseudoinverse of Jacobian  $J(\Theta)$  based on the damped least squares (DLS) method,  $J^* = J^T(JJ^T + \lambda I)^{-1}$ , and  $J \in R^{3 \times 6}$ .  $\lambda$  is the damping factor.  $I = \text{diag}(1, 1, 1)$ .  $J(\Theta) = [J_e(\Theta), J_c(\Theta)]^T$ .  $J_e(\Theta) = (\partial x / \partial \Theta, \partial y / \partial \Theta)^T$ ;  $J_c(\Theta) = (\partial \Psi / \partial \Theta)$ . Jacobian  $J$  is a desired matrix; that is, all parameters in  $J$  are accurately identified. In application,  $J$  always contains some errors, but the tasks can still be carried out normally. To prove this view, let  $\hat{J}$  be the Jacobian with some errors and the corresponding simulations are presented in Section 2.3.

In order to track a moving object in real-time, the velocity of end-effector,  $\dot{\mathbf{X}}$  in (1), is defined as

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

where  $\dot{\mathbf{X}}_{obj}$  is the velocity of the moving object.  $\dot{\mathbf{X}}_r$  is the relative velocity and  $\dot{\mathbf{X}}_r$  is expressed as [29]

$$\dot{\mathbf{X}}_r = K \cdot (\mathbf{X}_{obj} - \mathbf{X}), \quad (3)$$

where  $K$  is the gain coefficient of the position error.  $\mathbf{X}_{obj}$  is the position and attitude of the object,  $\mathbf{X}_{obj} = (x_{obj}, y_{obj}, \Psi_{obj})^T$ , and  $\Psi_{obj}$  is the object attitude.

In this paper, the iterative method is used to achieve the manipulator movement [30, 31]. Then (1) is modified as

$$\Delta \Theta \approx J^* \Delta \mathbf{X}, \quad (4)$$

where  $\Delta \mathbf{X}$  is the real-time path;  $\Delta \mathbf{X} = \Delta \mathbf{X}_r + \Delta \mathbf{X}_{obj}$ .  $\Delta \mathbf{X}_{obj}$ ,  $\Delta \mathbf{X}_r$  are the displacement of object and the relative displacement in a sampling time interval  $\Delta T$ , respectively. And (3) becomes

$$\Delta \mathbf{X}_r = \mu \cdot (\mathbf{X}_{obj} - \mathbf{X}), \quad (5)$$

where  $\mu = K \cdot \Delta T$ . Then the moving object can be tracked by updating the joint angles  $\Theta$  in real-time by (6). The control diagram of the tracking process is described in Figure 2(a).

$$\Theta := \Theta + \Delta \Theta. \quad (6)$$

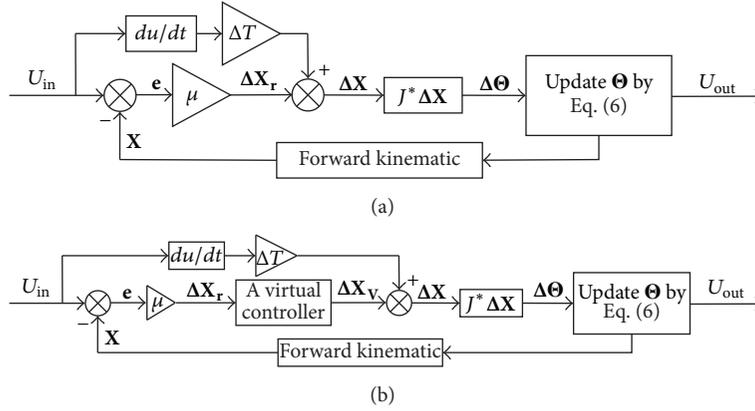


FIGURE 2: The control diagram tracking a moving object.  $U_{in} = \mathbf{X}_{obj}$  and  $U_{out} = \Theta$ . (a) The control diagram based on the common method. (b) The control diagram based on the proposed method.

**2.2. Based on the Virtual Controller.** Based on the common method, the moving object can be tracked, but in order to increase the convergence rate, a large proportional coefficient is usually set and it is not flexible. So, a virtual controller, similar to PD controller, is designed to improve the convergence rate adaptively. Then  $\Delta \mathbf{X}_r$  is modified as  $\Delta \mathbf{X}_v$ .

$$\Delta \mathbf{X}_v = k_p \xi_i + k_I \sum_{j=1}^i \xi_j + k_s \gamma_i, \quad (7)$$

where  $k_p, k_s, k_I$  are the ratio coefficients. Index  $i$  stands for the  $i$ th iterative process.  $\sum \xi_j, \gamma_i$  are used to compensate  $\Delta \mathbf{X}_v$  according to the speed of object.  $\xi_i, \gamma_i$  are calculated by

$$\gamma_i = \left[ \frac{(1 - e^{-\alpha \xi_i^j})}{(1 + e^{-\alpha \xi_i^j})} \right], \quad (8)$$

$$\xi_i = \eta \Delta \mathbf{X}_r + \Delta \dot{\mathbf{X}}_r,$$

where  $\xi_i = \{\xi_i^j\}$ ,  $j = 1, 2, 3$ .  $j = 1$  expresses  $x$  coordinate value.  $j = 2$  expresses  $y$  coordinate value.  $j = 3$  expresses the attitude  $\Psi$ .  $\alpha, \eta$  are the adaptive coefficients defined as

$$\alpha = \alpha_0 + f_1(v_{obj}), \quad (9)$$

$$\eta = \eta_0 + f_2(v_{obj}),$$

where  $\alpha_0, \eta_0$  are the initial coefficients. The velocity correlation functions  $f_1(v_{obj}), f_2(v_{obj})$  can adaptively increase with the increase of speed. When the object speed is low, the values of  $\xi_i$  and  $\gamma_i$  keep small. When the object speed becomes large, the values of  $\xi_i$  and  $\gamma_i$  will increase appropriately. Meanwhile, when the speed becomes too large,  $f_1(v_{obj}), f_2(v_{obj})$  will reach saturation state and  $\xi_i, \gamma_i$  will remain maximum, which keeps  $\Delta \mathbf{X}_r$  not excessive.  $f_1(v_{obj}), f_2(v_{obj})$  are defined as

$$f_1(v_{obj}) = \alpha_p (1 - e^{-p_1 \cdot v_{obj}}), \quad (10)$$

$$f_2(v_{obj}) = \eta_p (1 - e^{-p_2 \cdot v_{obj}}),$$

TABLE 1: Parameters in the virtual controller.

Parameters	$k_p$	$k_I$	$k_s$	$\alpha_0$	$\alpha_p$	$\eta_0$	$\eta_p$	$p_1$	$p_2$
Value	0.016	$10^{-7}$	2	0.05	0.02	300	100	0.2	0.6

TABLE 2: D-H parameters of 6-DOF manipulator.

Parameters	(mm)	Parameters	(mm)	Parameters	(rad)
$l_1$	150	$\hat{l}_1$	140	$\theta_1$	0.3
$l_2$	150	$\hat{l}_2$	150	$\theta_2$	0.3
$l_3$	150	$\hat{l}_3$	160	$\theta_3$	0.6
$l_4$	150	$\hat{l}_4$	141	$\theta_4$	0.5
$l_5$	150	$\hat{l}_5$	155	$\theta_5$	0.5
$l_6$	150	$\hat{l}_6$	149	$\theta_6$	0.8

where  $\alpha_p, \eta_p, p_1, p_2$  are the ratio coefficients.  $v_{obj}$  is the velocity magnitude of the moving object. The control diagram with the virtual controller is shown in Figure 2(b).

### 2.3. Simulation Verification

**Simulation 1.** This simulation is to compare the common method with the proposed method. The object moves back and forth along a curve ( $\Psi_{obj} = 0.7 \times \sin(2(t + 0.95)) + 2.07$ ) as shown in Figure 1. In the simulation,  $\mu$  is 0.048 for common method and  $\mu$  is 0.01 for proposed method, which will make  $k_p \times \eta_0 = 0.048$ . And the  $\Delta \mathbf{X}_r$  in the both cases will be coincident. Since the integral link  $\sum \xi_j$  is accumulated gradually and the adaptive adjusting part  $\gamma_i$  compensates  $\Delta \mathbf{X}_v$ , the planned path  $\Delta \mathbf{X}_v$  will become large and then the convergence rate is increased. In Figure 3, the convergence rate of the proposed method is much quicker than that of the common method in both positions and attitudes, and the error with the virtual controller is smaller. The simulation parameters are shown in Tables 1 and 2.

**Simulation 2.** This paper assumes that the length  $l_p$  ( $p = 1, 2, \dots, 6$ ) of manipulator arms contains some errors, which results in Jacobian  $\mathbf{J}$  being not desired. That is,  $\hat{l}_p = l_p + \Delta l_p$ ;  $\Delta l_p$  is the length error of the  $p$ th arm. The simulations with

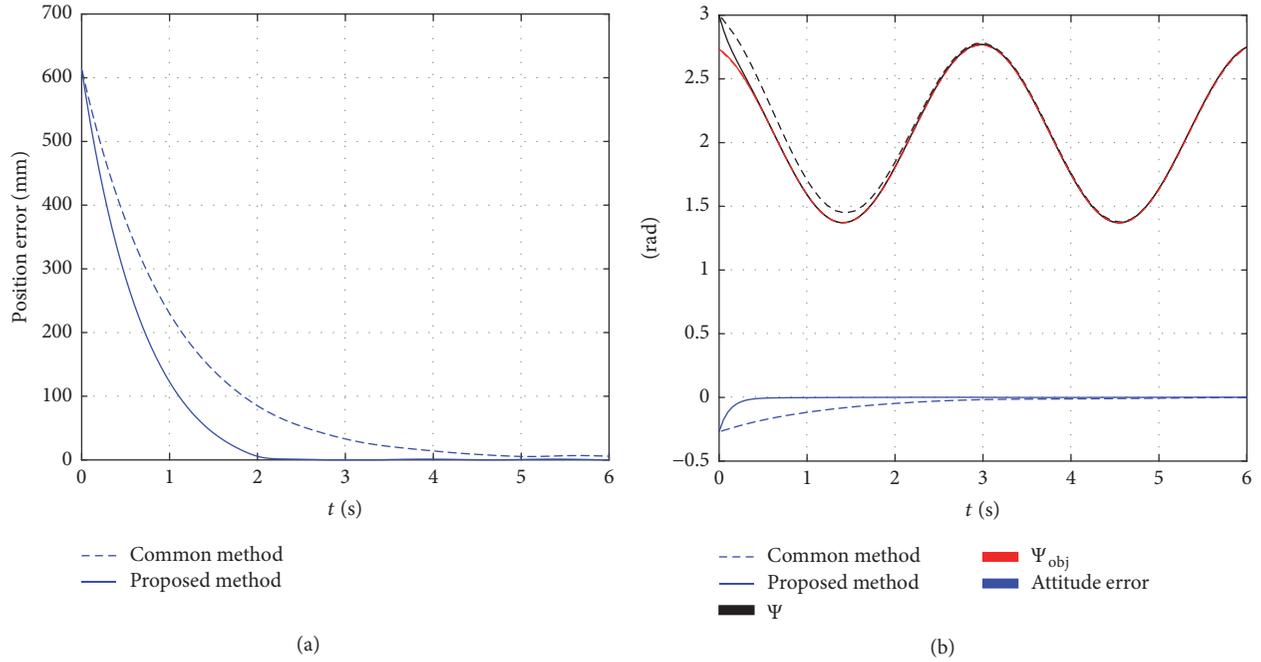


FIGURE 3: The comparison of the common method and the proposed method. (a) The position error between end-effector and object. (b) The attitudes  $\Psi$  and  $\Psi_{obj}$ ; the attitude error between  $\Psi$  and  $\Psi_{obj}$ .

the virtual controller based on  $\hat{\mathbf{J}}$  and  $\mathbf{J}$  are shown in Figure 4 and the corresponding parameters are shown in Tables 1 and 2. In Figure 4, although there are some errors in Jacobian, the moving object is still tracked. The stable error based on  $\hat{\mathbf{J}}$  is slightly larger in positions as shown in Figure 4(a). The attitude error is the same as that based on  $\mathbf{J}$  as shown in Figure 4(b). Based on the simulations, we can conclude that some errors in Jacobian have less impact on the primary tasks and the system has good robustness.

### 3. Multidynamic Obstacles Case

**3.1. Observation, Moving Path Prediction, and Path Planning.**  
 In a multidynamic obstacles environment, avoiding the obstacles will limit the rotation range of the joint angles, which results in some positions within the workspace not being reached. So, before tracking the moving object, the grasped position of object should be predicted, and a reachable path should be planned for the manipulator. To complete the prediction and path planning, the initial motions of the object and obstacles are observed by camera or sensor at first. As shown in Figure 5(a), the dot lines stand for the initial moving paths and the balls  $A, B, C, D, E$  stand for their current positions. Based on their current positions, the obstacles are divided into two kinds of cases. One is inside a closed polygon  $S$  indicated by polygons  $OO_1O_2O_3O_4O_5O_6A$ . The other one is not inside  $S$ . The criterion for judging an obstacle inside  $S$  is based on the method proposed by Hormann and Agathos [32] and the criterion is expressed as

$$\text{in}_i = \begin{cases} 1, & \text{if } S_i \in S \\ 0, & \text{if } S_i \notin S, \end{cases} \quad (11)$$

where  $S_i$  is the  $i$ th obstacle,  $i = 1, 2, \dots, m$ .  $m$  is the number of obstacles within the workspace of manipulator.

*Remark 1.* This paper assumes that the object and the obstacles keep motions within the workspace of the manipulator, and their motions can be observed. Besides, the moving curves of the object and obstacles are smooth.

Consider that the moving paths observed by camera or sensor contain noises. Gaussian filter is always used to suppress the noises, but it suffers from edge effects due to the local weight average [33]. Spline filter is advanced without edge effects, whose amplitude transmission characteristic is similar to that of Gaussian filter. And it can ride the curve trend well. So, Spline filter is attempted to suppress the noises and ride the trend of the moving paths in  $x, y$  direction. Spline filter can be expressed as

$$(\mathbf{I} + \beta^4 \mathbf{Q}) \omega(t) = z(t), \quad (12)$$

where  $\omega(t)$  is the filtered path containing  $n$  points.  $z(t)$  is the observed path with noises.  $\mathbf{Q}$  and  $\mathbf{I}$  are the  $n \times n$  matrix.  $\beta$  is given by

$$\beta = \left( 2 \sin \left( \frac{\pi \Delta x_s}{\lambda_c} \right) \right)^{-1}, \quad (13)$$

where  $\Delta x_s$  is the sample interval and  $\lambda_c$  is the cut-off length. More details about Spline filter are presented in [17].

As shown in Figure 5(b), the black curve is the real path. The orange points stand for the predicted points. The red points, smoothed via Spline filter, indicate initial motion of the object or obstacles, which are used to predict the following

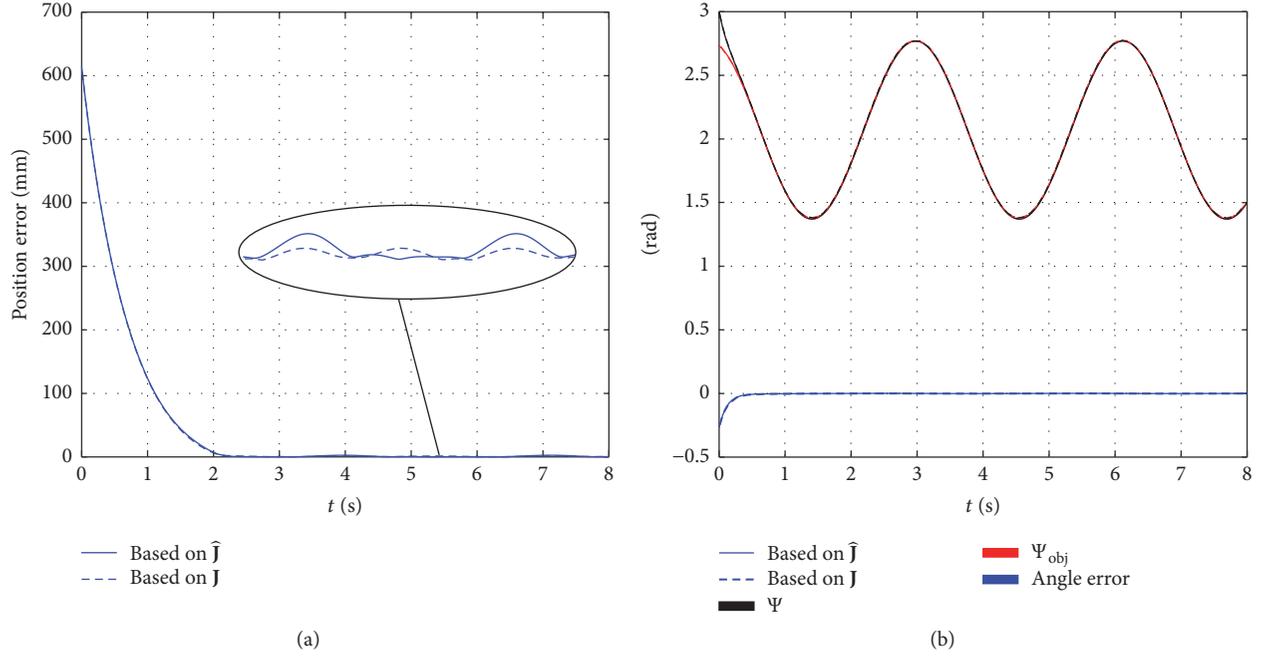


FIGURE 4: The changes of positions and attitude based on the proposed method in tracking a moving object. (a) The position error between end-effector and object. (b) The attitudes  $\Psi$  and  $\Psi_{\text{obj}}$ ; the attitude error between  $\Psi$  and  $\Psi_{\text{obj}}$ .

path.  $n - 1$  vectors  $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \dots, \mathbf{a}_{n-1})$  can be formed by  $n$  points  $(Q_1, Q_2, \dots, Q_n)$ , and the angles of every two adjacent vectors are calculated by

$$\theta_i^* = \arccos \frac{\mathbf{a}_i \cdot \mathbf{a}_{i+1}}{|\mathbf{a}_i| \cdot |\mathbf{a}_{i+1}|}. \quad (14)$$

The angle  $\theta_i^*$  ( $i = 1, 2, 3, \dots, n - 2$ ) can be fitted with a second-order polynomial via least-square methods

$$\theta^*(i) = a_\theta^* + b_\theta^* \cdot i + c_\theta^* \cdot i^2, \quad (15)$$

where  $a_\theta^*, b_\theta^*, c_\theta^*$  are the coefficients. The future angular  $\theta^*(i)$  ( $i \geq n - 1$ ) is calculated by (15). The rotation matrix  $\mathbf{R}$ , used to determine the direction of the next vector  $\mathbf{a}_{i+1}$ , is defined as

$$\mathbf{R} = \begin{bmatrix} \cos \theta^*(i) & -\sin \theta^*(i) \\ \sin \theta^*(i) & \cos \theta^*(i) \end{bmatrix}. \quad (16)$$

The weight  $W_i$ , introduced to calculate the size of the next vector  $\mathbf{a}_{i+1}$ , is expressed as

$$W_i = \frac{|\mathbf{a}_{i+1}|}{|\mathbf{a}_i|}. \quad (17)$$

Similarly, the weights are fitted by a first-order polynomial

$$W(i) = a_W + b_W \cdot i, \quad (18)$$

where  $i$  is the  $i$ th point on the path and  $i = 1, 2, 3, \dots, n - 2$ .  $a_W, b_W$  are the scale coefficients.

Based on (15)–(18), the following predicted vector,  $\mathbf{a}_{i+1}$ , can be calculated by

$$\mathbf{a}_{i+1}^T = W(i) \cdot \mathbf{R} \cdot \mathbf{a}_i^T, \quad i > n - 2. \quad (19)$$

Then the following point  $Q_{i+1}(x_{i+1}, y_{i+1})$  can be obtained by (20). The predicted path is accurate and the simulation is presented in Section 4.

$$(x_{i+1}, y_{i+1}) = (x_i, y_i) + \mathbf{a}_i, \quad (20)$$

$$i = n, n + 1, \dots, n + k - 1,$$

where  $k$  is the number of the predicted points. As shown in Figure 5(c), the black dash curve,  $AA'$ , is the predicted path of the moving object. The dash lines,  $BB', CC', DD',$  and  $EE'$ , are the predicted paths of obstacles, respectively. The predicted point  $A'$  is the grasped position of the object. The points  $B', C', D', E'$  are the predicted positions of obstacles when the object  $A$  is grasped. According to the predicted positions of object and obstacles, we can plan the paths  $C_1 = \text{path}\{O_6-A\}$ ,  $C_2 = \text{path}\{O_6-B-A\}$ ,  $C_3 = \text{path}\{O_6-C-A\}$ , and  $C_4 = \text{path}\{O_6-D-A\}$  as the tracked paths as shown in Figure 5(d). However, the convex polygon  $OA'B'C'D'$  will limit the joint rotation range, which may result in the object not being grasped along  $C_1$ . Similarly, polygons  $OA'C'D', OA'D'$  have the same influence on  $C_2, C_3$  as shown in Figure 5(c).

So, the convex polygons, which have influence on the paths, should be found out. Then, point  $A'$  is defined as the rotation center.  $A'O_6$  rotates clockwise around point  $A'$ . Line  $A'O$  has set the finishing line for rotation. The obstacles within  $S$  and the object can form polygons  $OA'B'C'D', OA'C'D', OA'D'$ . For a convex polygon, each interior angle

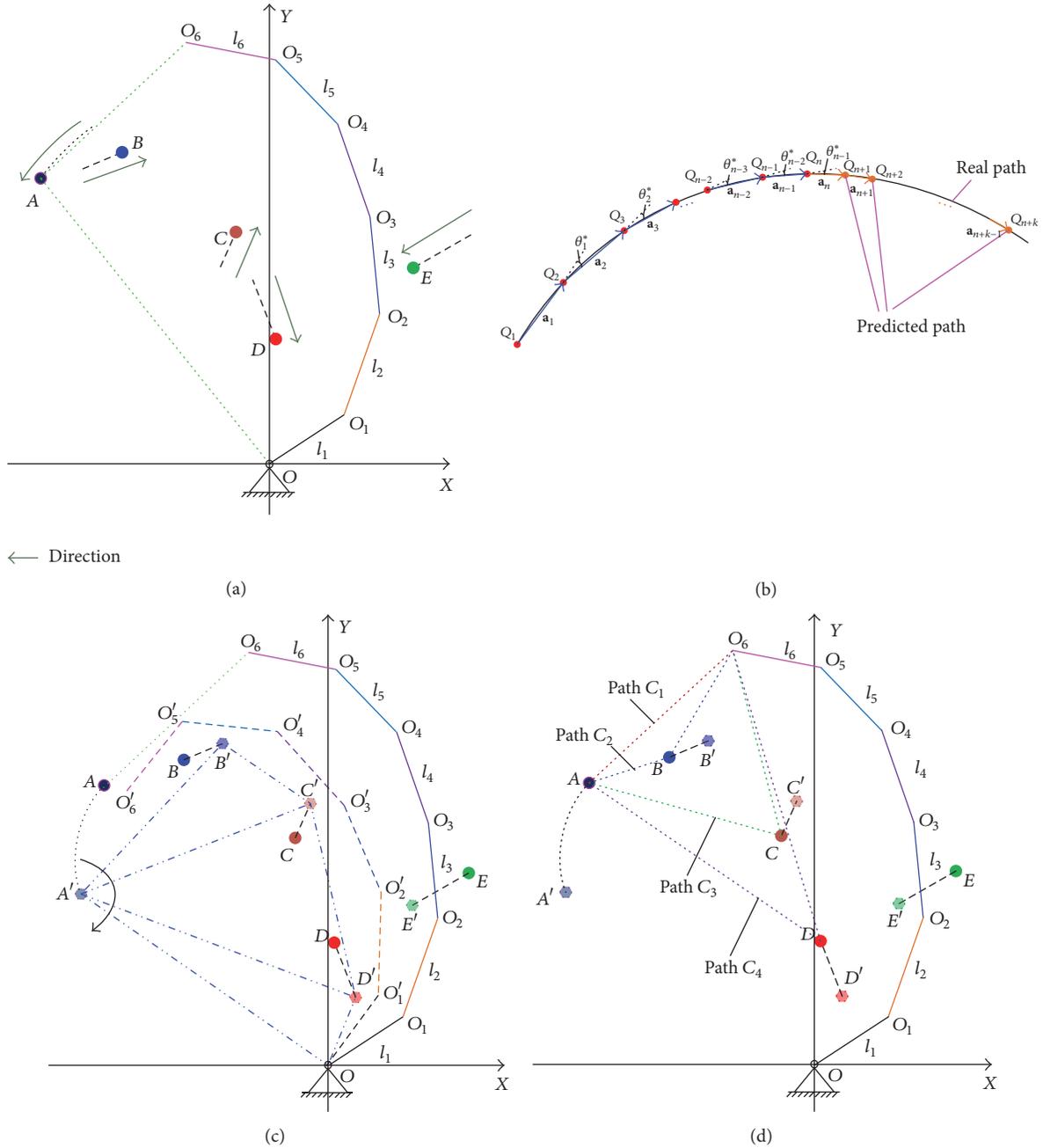


FIGURE 5: Observation, path prediction, and path planning. *A* is the object. *B*, *C*, *D*, and *E* are the obstacles. (a) Motions observation and region division of the obstacles. (b) Path prediction based on observed points. (c) Path judgment and selection based on convex polygons formed by the object *A'* and obstacles *B'*, *C'*, and *D'*. (d) Planning feasible paths and the optimized path.

of the polygon is less than  $180^\circ$ . So, let  $P_j$  be the evaluation function for the convexity-concavity of an arbitrary polygon

$$P_j = \sum_{k=1}^{K_j} \min \{ \mathbf{b}_k \times \mathbf{b}_{k+1}, 0 \}, \quad (21)$$

where  $j = 1, 2, \dots, N$ ,  $N$  is the number of polygons.  $K_j$  is the number of angles for the  $j$ th polygon.  $\mathbf{b}_k, \mathbf{b}_{k+1}$  are the adjacent edges of the  $j$ th polygon along clockwise or anticlockwise direction. If  $P_j$  is 0, the  $j$ th polygon is convex. Otherwise, it

is not convex. As shown in Figure 5(c), polygons  $OA'B'C'D'$ ,  $OA'C'D'$ ,  $OA'D'$  are convex. Then, let  $F_j$  be the feasibility criterion for  $C_1, C_2, C_3, C_4$ .

$$F_j = L_j - \overline{OA'} - \sum_{p=1}^6 l_p + \delta, \quad (22)$$

where  $L_j$  is the perimeter of the  $j$ th convex polygon which has an influence on the corresponding path.  $l_p$  is the length of the  $p$ th link.  $\delta$  is the estimated safe distance ( $\delta = 220$  mm),

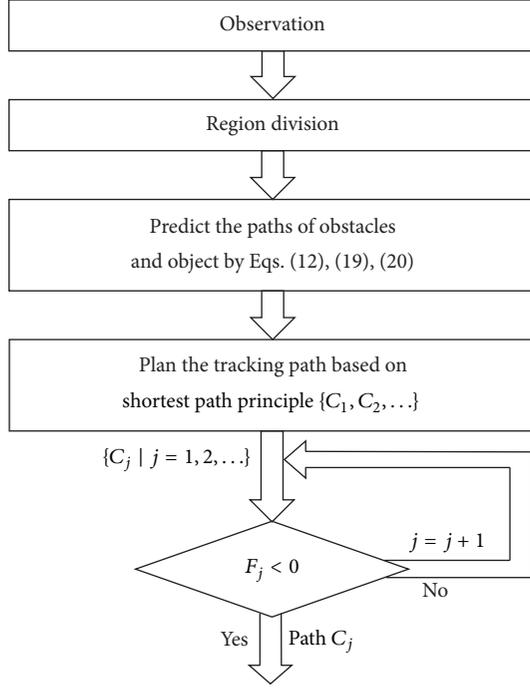


FIGURE 6: The algorithm diagram of the optimized path.

which ensures that no collision will happen when grasping the object. And  $\delta$  can be determined by simulation or experiment. If  $F_j$  is less than 0, the path is feasible. Otherwise, the path is infeasible.

Based on (22), the feasible paths can be obtained ( $C_2, C_3, C_4$ ). And among these paths, the shortest path is selected as the optimized path for the manipulator to track the moving object in real-time. As shown in Figure 5(d),  $C_2$  is the optimized path due to  $C_2 < C_3 < C_4$ . Based on the above analysis, the *algorithm of the optimized path* can be briefly described in Figure 6.

**3.2. Obstacles Included in the Planned Path.** Based on *algorithm of the optimized path*,  $C_2$  is tracked by the end-effector of manipulator as illustrated in Figure 7(a). However, the end-effector cannot move along the path completely due to the obstacle being on the planned path. As shown in Figures 7(a) and 7(b), the obstacle  $B$  is tracked at first in tracking the path  $C_2$ . And then when the end-effector  $O_6$  moves on the edge of safe area, the object is tracked in real-time. Similarly, if any one of paths  $C_3$  and  $C_4$  is the optimized path, their tracked processes are the same to the path  $C_2$ . Besides, when the end-effector  $O_6$  moves on the edge of safe area, the motion of obstacle avoidance for end-effector should be produced; that is, the moving direction of end-effector needs to be changed as shown in Figure 7(b). For this purpose, the local rotation coordinate method (LRCM) is proposed for end-effector to avoid the obstacles that are located in the moving path and LRCM can be described as follows. Firstly, according to the iterative method presented in Section 2, the real-time path is planned  $\Delta \mathbf{X} = (\Delta x, \Delta y, \Delta \Psi)^T$ . A local coordinate based

on the point  $O_6$  is established and the  $X_B$ -axis is along line  $O_6A$  as shown in Figure 7(b). In the local coordinate, the planned path  $(\Delta x, \Delta y)^T$  can be expressed as  $(\Delta x_B, 0)^T$ ,  $\Delta x_B = (\Delta x^2 + \Delta y^2)^{1/2}$ . To avoid the obstacle,  $\Delta y_B$  is defined along  $Y_B$ -axis in (23).

$$\begin{aligned} \Delta x_B &= \sqrt{\Delta x^2 + \Delta y^2}, \\ \Delta y_B &= (-1)^\tau \cdot K_B \cdot \Delta x_B \\ &\quad \cdot \left[ \cos \left( \pi \cdot \frac{d_B - d_{\min}}{R_{\text{safe}} - d_{\min}} \right) + 1 \right]^2, \end{aligned} \quad (23)$$

where  $K_B$  is the scale factor.  $R_{\text{safe}}$  is the radius of safe area.  $d_B$  is the distance between point  $O_6$  and obstacle  $B$  in the safe area.  $d_{\min}$  is the minimum distance.  $\tau$  decides the direction around the obstacle (along clockwise or counter clockwise direction).  $\tau$  is calculated as

$$\tau = \begin{cases} 1, & \text{if } x_A - x_{O_6} > 0 \\ 0, & \text{Otherwise,} \end{cases} \quad (24)$$

where  $x_A, x_{O_6}$  are the positions of object  $A$  and end-effector along  $x$ -axis in Cartesian space. Considering that the object is tracked all the time when path  $C_1$  is the optimized path as shown in Figure 7(c). To reduce the length of obstacle avoidance path in safe area,  $\tau$  is calculated by

$$\tau = \begin{cases} 1, & \text{if } \mathbf{d} \cdot (\mathbf{u} \times \mathbf{w}) \geq 0 \\ 0, & \text{Otherwise,} \end{cases} \quad (25)$$

where vector  $\mathbf{d} = (0, 0, 1)$ . Vectors  $\mathbf{u}$  and  $\mathbf{w}$  are shown in Figure 7(c). Then the planned path  $(\Delta x_B, \Delta y_B)^T$  in local coordinate  $X_B O_B Y_B$  is mapped to Cartesian coordinate  $XOY$  by the rotation matrix  $\mathbf{R}_o$ .

$$\mathbf{R}_o = \begin{bmatrix} \cos \varphi_o & -\sin \varphi_o \\ \sin \varphi_o & \cos \varphi_o \end{bmatrix}, \quad (26)$$

where  $\varphi_o$  is the rotation angle calculated by

$$\varphi_o = \arccos \frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}}. \quad (27)$$

Then the planned path  $(\Delta x, \Delta y)^T$  will be updated by

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} := \mathbf{R}_o \cdot \begin{pmatrix} \Delta x_B \\ \Delta y_B \end{pmatrix}. \quad (28)$$

The real-time planned path  $\Delta \mathbf{X}$  is thus modified. And the LRCM can be described by a diagram as shown in Figure 7(d).

**3.3. Obstacles Close to Manipulator Arms.** In the case that the obstacles are close to manipulator arms, the principle of avoiding obstacles is based on the critical points and there

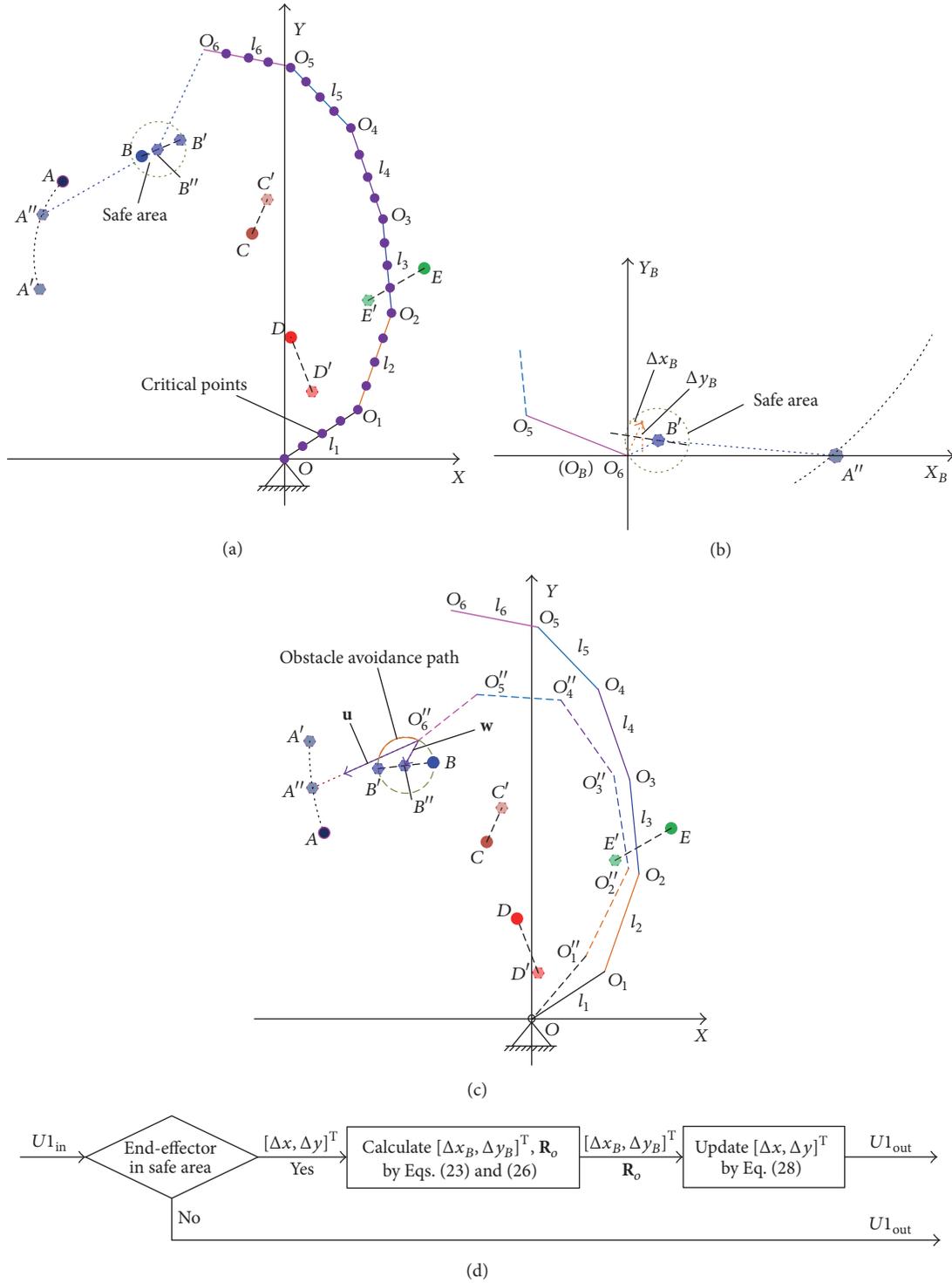


FIGURE 7: The principle of obstacles avoidance on the path. (a) The overall layout. (b) The end-effector moves around the obstacle  $B$  in real-time. (c) In the case that path  $C_1$  is the optimized path. (d) The LRCM diagram.  $U_{1in}$  is  $\Delta X$ ;  $U_{1out}$  is the updated path  $\Delta X$ .

are five critical points on each arm as shown in Figure 7(a). In this paper, there are four obstacles in the workspace of manipulator. To avoid multiobstacles, the method by setting task priority is proposed by Petrić et al. [7]. When these obstacles move to the manipulator at the same time, the

priorities of tasks are the same. So the obstacles avoiding formula are modified as follows:

$$\Delta\Theta = J^* \Delta X + k (\mathbf{I} - J^* J) \sum_{j=1}^M k_j \nabla H_j, \quad (29)$$

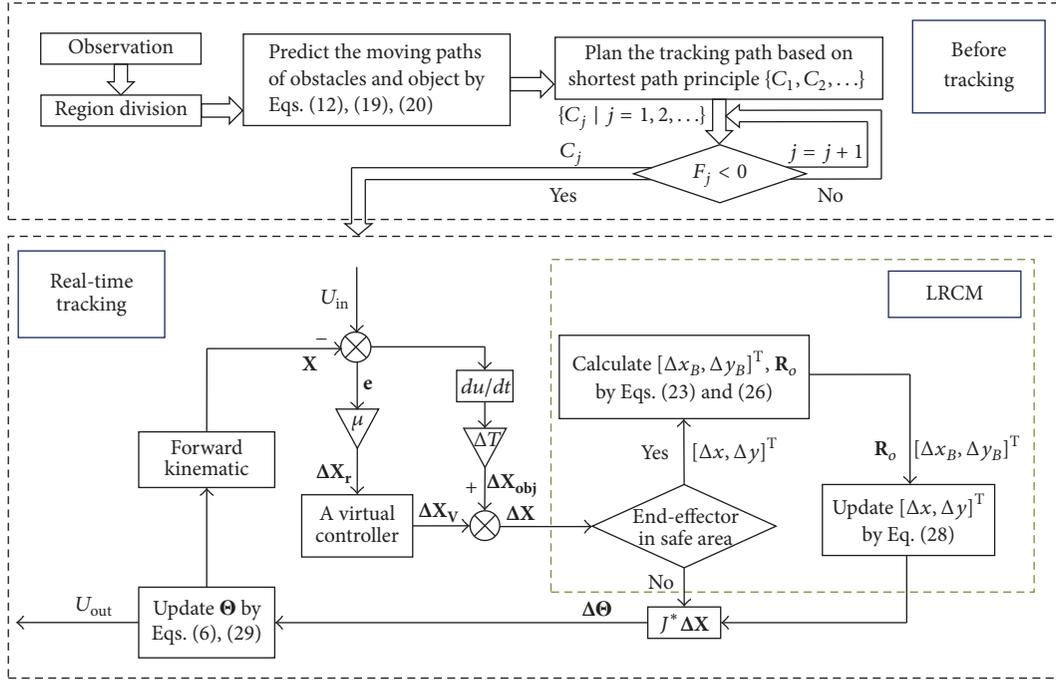


FIGURE 8: The synthetic algorithm diagram of tracking a moving object in multiple-dynamic obstacles environment.  $U_{in} = \mathbf{X}_{obj}$ ;  $U_{out} = \Theta$ .

where  $M$  is the number of obstacles.  $k(\mathbf{I} - \mathbf{J}^*)k_j \nabla \mathbf{H}_j$  is the null space, which is used to avoid the  $j$ th obstacle.  $k_j$  is a scale coefficient.  $k$  is defined as

$$k = \begin{cases} 2, & \text{for } d \leq d_{\min} \\ \cos\left(\pi \frac{d - d_{\min}}{R_{\text{safe}} - d_{\min}}\right) + 1, & \text{for } d_{\min} < d < R_{\text{safe}} \\ 0, & \text{for } d \geq R_{\text{safe}}, \end{cases} \quad (30)$$

where  $R_{\text{safe}}$  is the radius of safe area.  $d$  is the distance between critical point and obstacles in the safe area and  $d_{\min}$  is defined as the minimum distance between the obstacle and the critical points.

**3.4. The Synthetic Object Tracking Algorithm in Multiple-Dynamic Obstacle Environment.** On the basis of above analysis, an algorithm can be synthesized to track the moving object in a multidynamic obstacle environment, and the algorithm diagram is shown in Figure 8. Before tracking, a reachable path for real-time tracking is planned via the motions observation, moving paths prediction, and path planning, which ensures that the object can be grasped if the end-effector moves along the planned path. Then the tracking can be performed in real-time.

## 4. Simulations

**4.1. Simulation of the Predicted Path.** The first part of the simulation is to prove the exactness and the stability of the predicted path for the paths of the moving body in the future. Based on the initial 40 points, the future 20 points of three

kind curves are predicted as shown in Figure 9. The sample interval,  $\Delta x_s$ , is 1 mm and cut-off length,  $\lambda_c$ , is 40 mm for Spline filter. The similarity degree between the real moving path and predicted path can be calculated as

$$\text{SSI} = \frac{1}{k_{\text{SSI}}} \sum_{j=1}^{k_{\text{SSI}}} \text{SSI}_j = \frac{1}{k_{\text{SSI}}} \sum_{j=1}^{k_{\text{SSI}}} \left[ 1 - \left( \frac{d_{sj}}{r_s} \right)^2 \right], \quad (31)$$

where SSI ranges from 0 to 1. If the value of SSI is closer to 1, the predicted path is more accurate.  $\text{SSI}_j$  is similarity degree of the  $j$ th predicted point.  $k_{\text{SSI}}$  is the number of predicted points.  $d_{sj}$  is the distance between the predicting point and the real point at each instant  $t$  or position.  $r_s$  ( $r_s = 20$  mm) is the maximum radius of allowable deviation.

According to (31), the similarity degrees of the different motions are calculated and 20 experiments of each curve are shown in Figure 9. Figures 9(d), 9(e), and 9(f) show the average similarity degrees of 20 experiments, and Figures 9(g), 9(h), and 9(i) show the similarity degree of the last predicted point. Figures 9(d) and 9(g) are corresponding to Figure 9(a). Figures 9(e) and 9(h) are corresponding to Figure 9(b). Figures 9(f) and 9(i) are corresponding to Figure 9(c). In Figures 9(a)–9(c), it is obvious to see that the moving path, via Spline filter, is smoothing and can ride the desired path well. Besides, the deviation of predicted path related to the desired path is small. As shown in Table 3, the average value of the similarity degrees of 20 experiments is calculated. In this paper, the similarity degrees, more than 0.7, are considered to be effective. And based on Table 3, it shows that the proposed method for predicting path is stable and exact. So the proposed method is suitable to predict the moving paths in this paper.

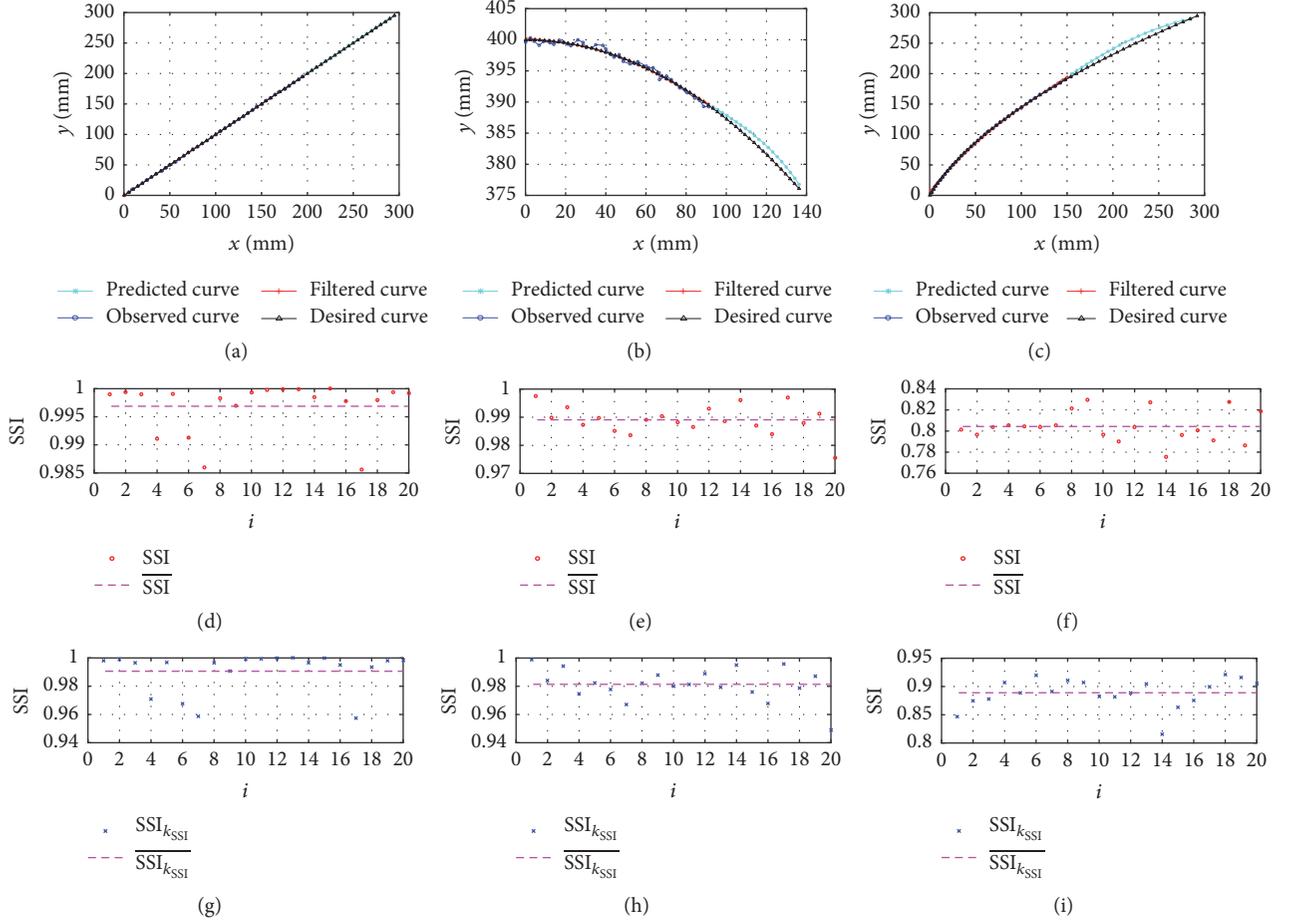


FIGURE 9: Comparison of the predicted path and the real moving path with noises for 20 points in the future. (a) Straight line:  $x = 5j$  and  $y = 5j$ . (b) Circle curve:  $x = R \cdot \sin(0.03\pi j/16)$ ,  $y = R \cdot \cos(0.03\pi j/16)$ , and  $R = 400$ . (c) Combined curve:  $x = 0.05j^2 + 2j$  and  $y = 5j$ ,  $j = 0, 1, 2, \dots, 39$ . Subfigures (d), (e), (f), (g), (h), and (i) show the similarity degrees of 20 experiments for each prediction. In subfigures, the value of  $SSI_{k_{SSI}}$  is the similarity degree of the last predicted point.  $\bar{SSI}$  and  $\overline{SSI_{k_{SSI}}}$  are the average values of SSI and  $SSI_{k_{SSI}}$ , respectively.

TABLE 3: The similarity degrees of the different motions for Figure 9.

Similarity degree	Type		
	Straight line	Circle curve	Combined curve
$\bar{SSI}$	0.9969	0.9891	0.8043
$\overline{SSI_{k_{SSI}}}$	0.9906	0.9814	0.8890

**4.2. Simulation of LRCM.** The second part of the simulation is to verify the feasibility of LRCM. LRCM is used to avoid the obstacles that are close to the end-effector of manipulator. In Figures 10(a) and 10(e), the movement of avoidance obstacles will be produced when the end-effector enters the safe area, and the static obstacle is avoided based on LRCM. The motion of obstacle avoidance will not work when the end-effector moves out of the safe area. As shown in Figures 10(b) and 10(f), the LRCM also shows the feasibility for the moving obstacle to track the moving object. When end-effector moves over the obstacle, the null space of inverse kinematics is used to avoid the obstacle for other arms of manipulator. Besides, based on LRCM, the end-effector of the manipulator can achieve a large turn to avoid the obstacle that is located in

TABLE 4: Parameters for Figures 9 and 10.

Parameters	$R_{safe}$	$V_{obstacle}$	$V_{object}$	$K_B$	$d_{min}$	$k_r$
Value	80 mm	8 mm/s	63 mm/s	5	30 mm	$1.0 \times 10^7$

TABLE 5: Parameters in the virtual controller.

Parameters	$k_p$	$k_I$	$k_s$	$\alpha_0$	$\alpha_p$	$\eta_0$	$\eta_p$	$p_1$	$p_2$
Value	0.005	$10^{-7}$	1	$10^{-3}$	$10^{-2}$	300	100	0.02	0.02

the tracking path. Figures 10(c), 10(d), 10(g), and 10(h) show that the artificial potential field cannot work well in this case. The reason is that the repulsive force  $F_{ref}$  is along the radial direction of the obstacle so that the end-effector cannot turn to the right. The force  $F_{ref}$  of artificial potential field can be calculated by (32) [16]. Tables 4 and 5 show the simulation parameters of LRCM and the artificial potential field.

$$F_{ref} = \begin{cases} k_r \left( \frac{1}{d} - \frac{1}{R_{safe}} \right) \frac{1}{d^2} \frac{\partial d}{\partial \mathbf{x}} & \text{if } d \leq R_{safe} \\ 0 & \text{if } d > R_{safe}, \end{cases} \quad (32)$$

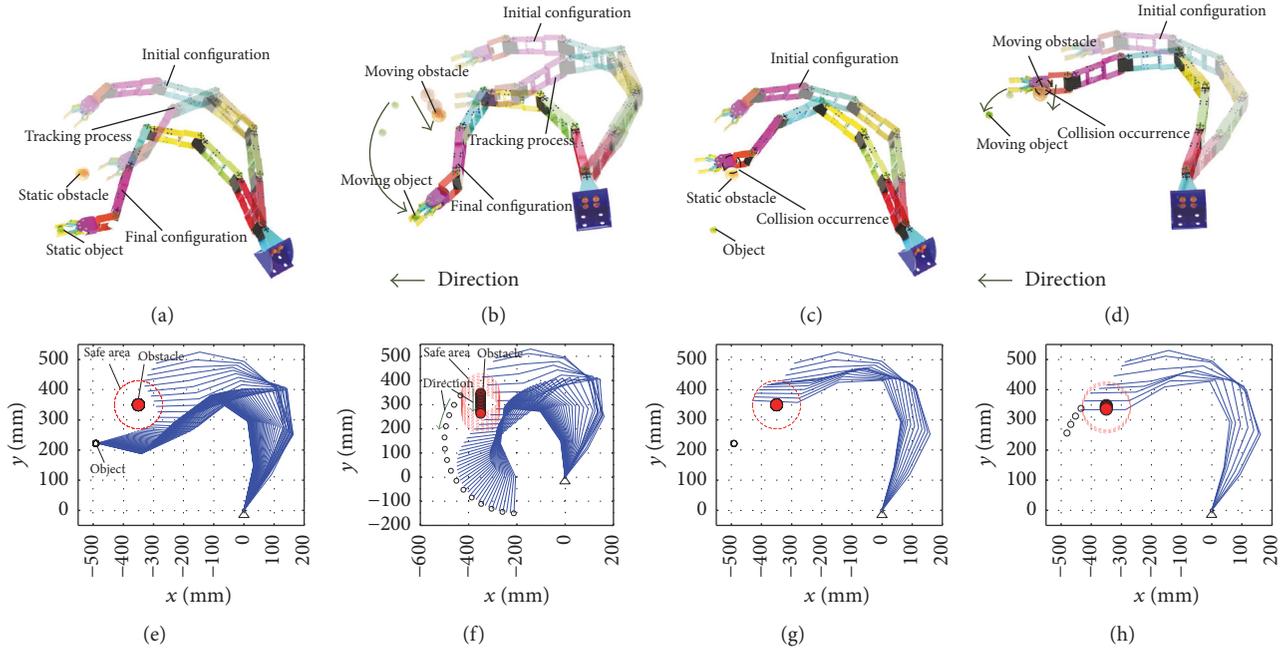


FIGURE 10: Avoiding the obstacle close to the end-effector by using LRCM in real-time. The initial joint angles  $\Theta = (0.9, 0.2, 0.8, 0.5, 0.5, 0.5)^T$ . (a), (e) Static obstacle by using LRCM. (b), (f) Moving obstacle by using LRCM. (c), (g) Static obstacle by using artificial potential field. (d), (h) Moving obstacle by using artificial potential field. Subfigures (a)–(d) show the movement of 3D model. Subfigures (e)–(h) show the movement of the simplified model.

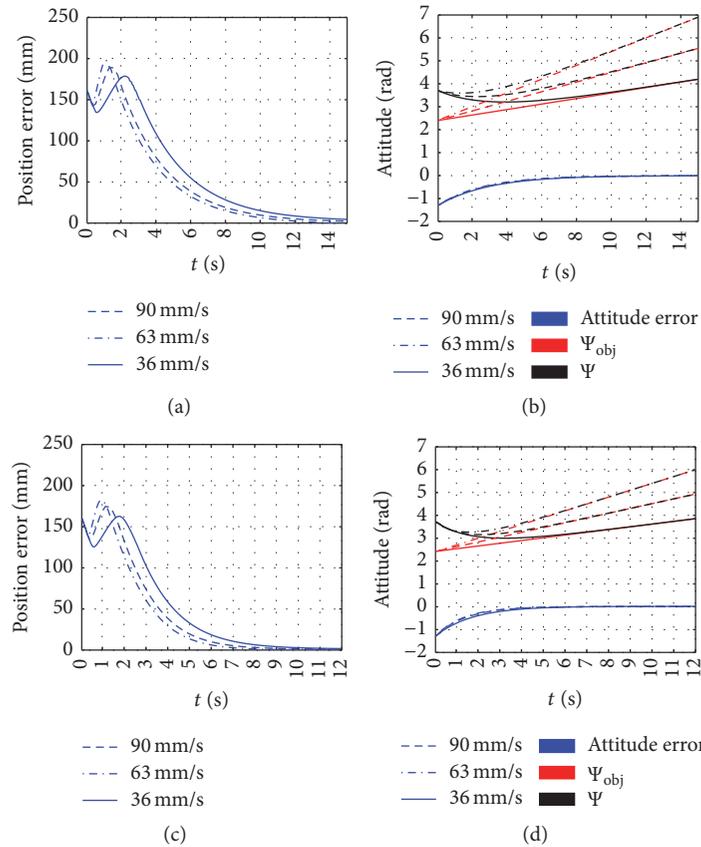


FIGURE 11: Tracking the object with different moving speed.  $\Psi$  is the attitude of end-effector;  $\Psi_{obj}$  is the attitude of the object. (a), (b) In the case without the visual controller. (c), (d) In the case with the visual controller without the visual controller.

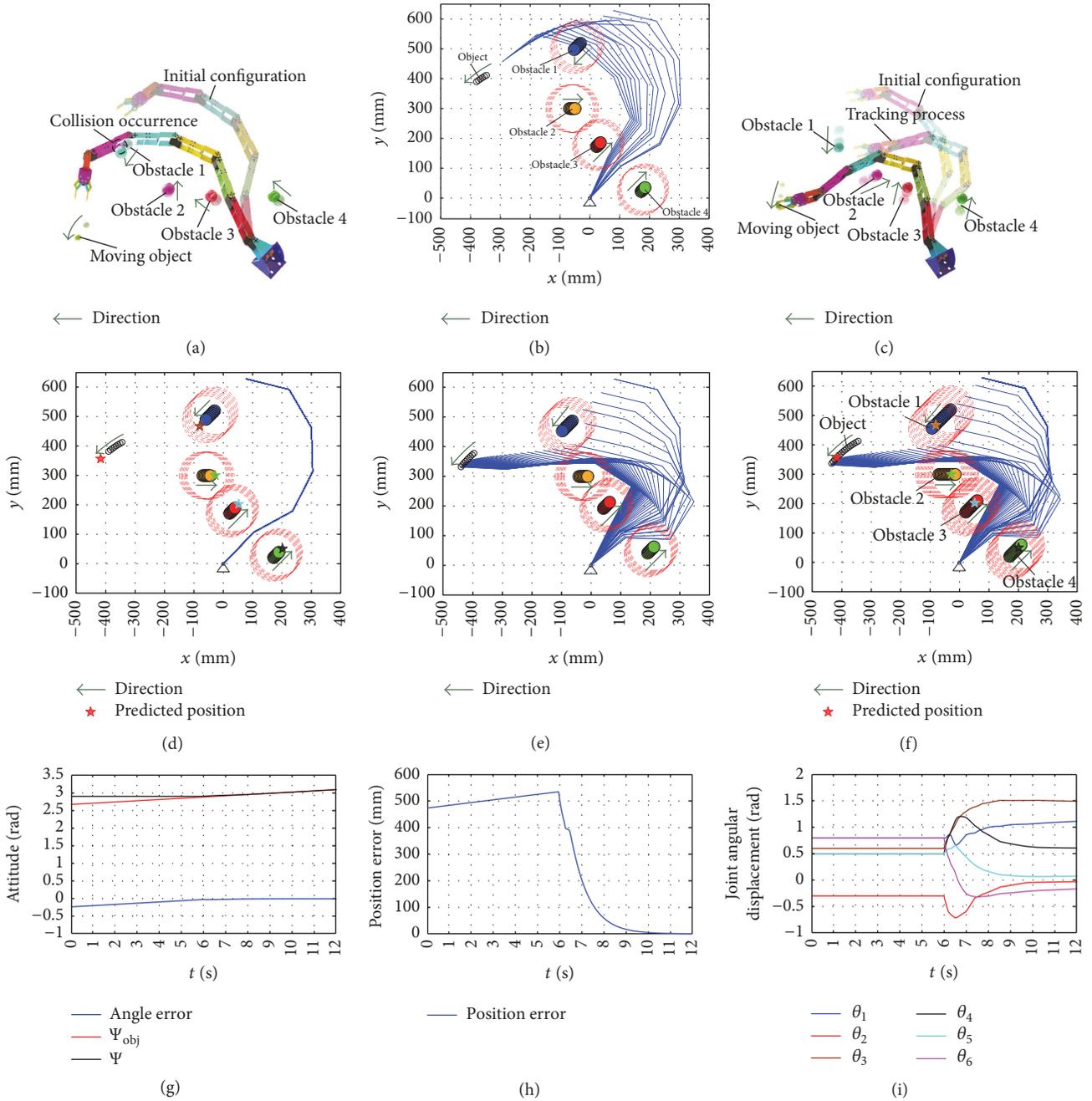


FIGURE 12: The simulation of tracking the moving object in multiple-dynamic obstacles environment.  $\Psi$  is the attitude of end-effector;  $\Psi_{obj}$  is the attitude of the object. The initial joint angles  $\Theta = (0.8, -0.3, 0.6, 0.5, 0.5, 0.8)^T$ . (a), (b) Tracking the moving object without observation and moving path prediction. (d) Observation and prediction. (e) Real-time tracking the object and avoiding obstacles. (c), (f) The whole process of tracking the object. (g) The attitudes  $\Psi$ ,  $\Psi_{obj}$  and the error between  $\Psi$  and  $\Psi_{obj}$ . (h) The position error between end-effector and object. (i) The joint angular displacement. Subfigures (a), (c) show the movement of 3D model. Subfigures (b), (d)–(f) show the movement of the simplified model.

where  $\mathbf{x}$  is the position vector of the end-effector,  $\mathbf{x} = (x, y)$ .  $d$  is the distance between critical point and obstacles in safe area.

4.3. Simulation of the Virtual Controller. The third part of the simulation is to test the applicability of tracking the moving object with different speeds. As shown in Figure 11, the objects

at the speed of 90 mm/s, 63 mm/s, and 36 mm/s are tracked, respectively, and the moving process is similar to Figures 10(b) and 10(f). In Figure 11, it is obvious to see that the convergence rate of the tracking with the virtual controller is more fast than that without the virtual controller, which is caused by the integral link  $\sum \xi_j$  and the adaptive adjusting part  $\gamma_i$ . Besides, the time-consuming becomes less with the

increasing of the speed of the object, due to the adaptive adjusting part  $\gamma_i$ . What is more, based on the slope of the position error curve, the convergence rate will decrease when the distance between the object and the end-effector becomes small. So it is more safe to track the moving object within the predicted time. The simulation parameters are shown in Table 5.

#### 4.4. Simulation of the Synthetic Object Tracking Algorithm.

The last part of the simulation is to validate the feasibility of the synthetic tracking algorithm proposed by the authors. Firstly, the simulation of tracking the moving object without observation and moving path prediction is presented as shown in Figures 12(a) and 12(b). Based on the shortest principle, path  $C_1$  is tracked in real-time as shown in Figure 5(d). In Figures 12(a) and 12(b), although the end-effector can avoid the obstacle, it cannot reach the position of object due to the obstacles limiting the joint angles of manipulator. So, the observation and moving path prediction are introduced into the proposed algorithm in this paper as shown in Figures 12(c)–12(f). Figure 12(d) presents that the motions of the object and the obstacles are observed, and the future position is predicted at first before tracking. Meanwhile, the optimized path  $C_2$  is planned. Figure 12(e) shows that the moving object is tracked in real-time along the optimized path  $C_2$ . Figures 12(c) and 12(f) display the whole process of tracking in the dynamic environment based on the proposed tracking algorithm. The tracking pose of manipulator and the pose error are shown in Figures 12(g) and 12(h). In Figure 12(i), we can see that the changes of the joint angles are continuous and smooth. And during the first six seconds, the manipulator remains static, and the object and obstacles keep moving. At 9th second, the moving object is almost grasped. Although there still is some position error, it is relatively small comparing with the sizes of the object or the obstacles as shown in Figure 12(h). The simulation results show that the reachability of the object position can be guaranteed by introducing the observation and moving path prediction, and the proposed algorithm is suitable to tracking a moving object in the multidynamic obstacle environment. The parameters of simulation are shown in Table 5 and the simulation process of the proposed algorithm can be downloaded from [34].

## 5. Conclusion

A synthetic algorithm, involving observation, moving path prediction, path planning, and real-time tracking, is proposed to track a moving object in a multidynamic obstacles environment for kinematically planar redundant manipulators. Spline filter combining with polynomial fitting is developed to predict the moving paths of the object and obstacles. The predicted moving paths and the defined feasibility criterion are used to plan paths of the end-effector in Cartesian space. Then the optimized path algorithm is raised for the manipulator to plan a tracking path. In tracking, the LRCM method is proposed for end-effector to avoid obstacles, and the null space of inverse kinematic is adopted for the manipulator arms to avoid obstacles. A virtual controller based on PD

controller is designed to achieve the adaptive fast tracking. The moving object is thus tracked by updating the joint angles based on iterative method in real-time. Simulation results show that the predicting method of path is stable and accurate, the LRCM method is feasible for the end-effector to avoid obstacles, and the convergence rate of tracking can increase adaptively by using the virtual controller. Besides, the proposed synthetic algorithm is feasible to track a moving object in a multidynamic obstacle environment. Therefore, the synthetic algorithm is recommended by the authors.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

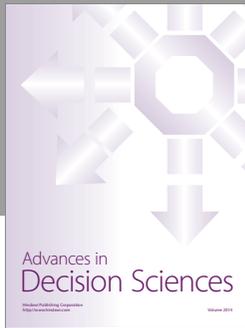
## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61473102, 61503098, and 91648201 and supported by Self-Planned Task of State Key Laboratory of Robotics and System (HIT), SKLRS201702A.

## References

- [1] V. V. Riabchenko, A. V. Nikitin, V. N. Khomenko et al., "Application of the computer vision technology to control of robot manipulators," in *Theoretical and Applied Aspects of Cybernetics*, 2012.
- [2] M. R. Akella, "Vision-based adaptive tracking control of uncertain robot manipulators," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 748–753, 2005.
- [3] C. Cai, N. Somani, and A. Knoll, "Orthogonal image features for visual servoing of a 6-DOF manipulator with uncalibrated stereo cameras," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 452–461, 2016.
- [4] L. Gargiulo, P. Bayetti, V. Bruno et al., "Operation of an ITER relevant inspection robot on Tore Supra tokamak," *Fusion Engineering and Design*, vol. 84, no. 2–6, pp. 220–223, 2009.
- [5] L. Gargiulo, J. J. Cordier, J. P. Friconneau et al., "Towards operations on Tore Supra of an ITER relevant inspection robot and associated processes," *Fusion Engineering and Design*, vol. 82, no. 15–24, pp. 1996–2000, 2007.
- [6] R. O. Buckingham and A. C. Graham, "Dexterous manipulators for nuclear inspection and maintenance—case study," in *Proceedings of the 1st International Conference on Applied Robotics for the Power Industry (CARPI '10)*, pp. 1–6, October 2010.
- [7] T. Petrič, A. Gams, N. Likar, and L. Žlajpah, "Obstacle avoidance with industrial robots," *Mechanisms and Machine Science*, vol. 29, pp. 113–145, 2015.
- [8] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Trajectory filtering and prediction for automated tracking and grasping of a moving object," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '92)*, pp. 1850–1856, May 1992.
- [9] F. Husain, A. Colome, B. Dellen, G. Alenya, and C. Torras, "Realtime tracking and grasping of a moving object from range video," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 2617–2622, June 2014.
- [10] W. Bing and L. Xiang, "A simulation research on 3D visual servoing robot tracking and grasping a moving object," in

- Proceedings of the 15th International Conference on Mechatronics and Machine Vision in Practice (M2VIP '08)*, pp. 362–367, December 2008.
- [11] A. M. Jasour and M. Farrokhi, “Adaptive neuro-predictive control for redundant robot manipulators in presence of static and dynamic obstacles: a Lyapunov-based approach,” *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 3–5, pp. 386–411, 2014.
- [12] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, “Relative end-effector control using cartesian position based visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.
- [13] D. B. Westmore and W. J. Wilson, “Direct dynamic control of a robot using an end-point mounted camera and Kalman filter position estimation,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '91)*, pp. 2376–2384, April 1991.
- [14] M. Matsushima, T. Hashimoto, and F. Miyazaki, “Learning to the robot table tennis task-ball control & rally with a human,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2962–2969, October 2003.
- [15] K. Mülling, J. Kober, and J. Peters, “A biomimetic approach to robot table tennis,” in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 1921–1926, October 2010.
- [16] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, “A learning approach to robotic table tennis,” *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 767–771, 2005.
- [17] J. Raja, B. Muralikrishnan, and S. Fu, “Recent advances in separation of roughness, waviness and form,” *Precision Engineering*, vol. 26, no. 2, pp. 222–235, 2002.
- [18] R. Kala, A. Shukla, and R. Tiwari, “Fusion of probabilistic A\* algorithm and fuzzy inference system for robotic path planning,” *Artificial Intelligence Review*, vol. 33, no. 3–4, pp. 307–327, 2010.
- [19] M. Park, J. Jeon, and M. Lee, “Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing,” in *Proceedings of the IEEE International Symposium on Industrial Electronics Proceedings (ISIE '01)*, pp. 1530–1535, June 2001.
- [20] J. Wang, X. Hu, and B. Zhang, “Real-time motion planning of kinematically redundant manipulators using recurrent neural networks,” *Recent Advances in Intelligent Control Systems*, pp. 169–193, 2009.
- [21] M. Duguleana, F. G. Barbuceanu, A. Teirelbar, and G. Mogan, “Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 132–146, 2012.
- [22] D. Zhou, L. Wang, and Q. Zhang, “Obstacle avoidance planning of space manipulator end-effector based on improved ant colony algorithm,” *SpringerPlus*, vol. 5, no. 1, article 509, 2016.
- [23] G. Ping, B. Wei, X. Li, and X. Luo, “Real time obstacle avoidance for redundant robot,” in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '09)*, pp. 223–228, August 2009.
- [24] R. Mayorga, K. Ma, and A. Wong, “A robust local approach for the obstacle avoidance of redundant robot manipulators,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1727–1734, Raleigh, NC, USA, July 1992.
- [25] A. Maciejewski, “Obstacle avoidance for kinematically redundant manipulators,” *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.
- [26] V. Perdereau, C. Passi, and M. Drouin, “Real-time control of redundant robotic manipulators for mobile obstacle avoidance,” *Robotics and Autonomous Systems*, vol. 41, no. 1, pp. 41–59, 2002.
- [27] Z. Mao and T. C. Hsia, “Obstacle avoidance inverse kinematics solution of redundant robots by neural networks,” *Robotica*, vol. 15, no. 1, pp. 3–10, 1997.
- [28] R. V. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, “A collision-avoidance scheme for redundant manipulators: theory and experiments,” *Journal of Robotic Systems*, vol. 22, no. 12, pp. 737–757, 2005.
- [29] A. S. Deo and I. D. Walker, “Overview of damped least-squares methods for inverse kinematics of robot manipulators,” *Journal of Intelligent & Robotic Systems*, vol. 14, no. 1, pp. 43–68, 1995.
- [30] R. Samuel and J. Buss, “Selectively damped Least squares for inverse kinematics,” *Journal of Graphics GPU & Game Tools*, vol. 10, no. 3, pp. 37–49, 2005.
- [31] S. Buss, “Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Transactions on Robotics & Automation*, vol. 17, 2004.
- [32] K. Hormann and A. Agathos, “The point in polygon problem for arbitrary polygons,” *Computational Geometry*, vol. 20, no. 3, pp. 131–144, 2001.
- [33] Y.-B. Yuan, X.-F. Qiang, J.-F. Song, and T. V. Vorburger, “Fast algorithm for determining the Gaussian filtered mean line in surface metrology,” *Precision Engineering*, vol. 24, no. 1, pp. 62–69, 2000.
- [34] <https://drive.google.com/file/d/0B2EkpfXB3kFqUIVYs0RGbE13ZIE/view?usp=sharing>.




**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

