

Article

A Predictable Obstacle Avoidance Model Based on Geometric Configuration of Redundant Manipulators for Motion Planning

Fengjia Ju, Hongzhe Jin * , Binluan Wang and Jie Zhao

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; 20b908029@stu.hit.edu.cn (F.J.); 18b908042@stu.hit.edu.cn (B.W.); jzhao@hit.edu.cn (J.Z.)

* Correspondence: hongzhejin@hit.edu.cn

Abstract: When a manipulator works in dynamic environments, it may be affected by obstacles and may cause danger to people around. This requires the manipulator to be able to plan the obstacle avoidance motion in real time. Therefore, the problem solved in this paper is dynamic obstacle avoidance with the whole body of the redundant manipulator. The difficulty of this problem is how to model the manipulator to reflect the motion relationship between the manipulator and the obstacle. In order to describe accurately the occurrence conditions of the collision, we propose the triangular collision plane, a predictable obstacle avoidance model based on the geometric configuration of the manipulator. Based on this model, three cost functions, including the cost of the motion state, the cost of a head-on collision, and the cost of the approach time, are established and regarded as optimization objectives in the inverse kinematics solution of the redundant manipulator combined with the gradient projection method. The simulations and experiments on the redundant manipulator and the comparison with the distance-based obstacle avoidance point method show that our method improves the response speed of the manipulator and the safety of the system.

Keywords: motion planning; collision avoidance; kinematics; redundant robots



Citation: Ju, F.; Jin, H.; Wang, B.; Zhao, J. A Predictable Obstacle Avoidance Model Based on Geometric Configuration of Redundant Manipulators for Motion Planning. *Sensors* **2023**, *23*, 4642. <https://doi.org/10.3390/s23104642>

Academic Editor: Yongmin Zhong

Received: 28 March 2023

Revised: 26 April 2023

Accepted: 9 May 2023

Published: 10 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the process of task execution, such as parts assembly, object grasping, and space station construction, the manipulator is often affected by the limitation and change in the operation space and has contact and non-contact interactions with the dynamic environment. Collision detection in contact interactions is usually based on dynamic models, while in non-contact interactions, collision detection is usually based on kinematic geometric models [1]. In non-contact interactions, the manipulator should have the ability to predict whether a collision will occur and react in advance. Therefore, it is necessary to design a predictable geometric collision detection model for the manipulator as the obstacle avoidance model. At present, there are many dynamic obstacle avoidance methods based on manipulator kinematics, and the essential difference between these methods lies in the different ways of constructing the obstacle avoidance model between the manipulator and the obstacle. Depending on the type of information used in the model, we divide the current methods into distance-based methods and velocity-based methods.

The distance-based methods are based on the position of the obstacle and the manipulator, such as some sampling-based methods [2–4] and reinforcement learning [5]. The most typical distance-based method is the obstacle avoidance point method (OAPM) [6]. The closest point on the manipulator body to the obstacle is regarded as the obstacle avoidance point, and the velocity away from the obstacle is assigned to this point. Based on the OAPM, the authors use the human skeleton recognition method to treat a human as an obstacle and realize the obstacle avoidance of the manipulator [7]. In [8], the obstacle avoidance points are obtained from the pre-selected equidistant points on the manipulator links, which ensure that the manipulator body keeps a relatively safe distance from obstacles.

A variety of collision detection methods have been proposed in computer graphics [9] and used to solve trajectory planning problems by combining them with optimization methods [10–16]. In [12], the authors calculate the bound of the obstacle by estimating the trajectory of the moving obstacle over a short time horizon and build the cost function of the signed Euclidean distance transform between the robot sphere and the location boundary of the obstacle to realize the trajectory planning. The sweep-out volume is proposed as the obstacle avoidance model of the manipulator in a continuous time step, and the cost based on the distance between the sweep-out volume and the obstacle is calculated in [13]. A method for strictly convex hull generation is proposed in [14] and used to calculate the continuous gradient proximity distance to generate free-collision optimized postures for humanoid robots. Gilbert–Johnson–Keerthi algorithm is used in [17] to calculate the minimum distance between a manipulator and an obstacle.

In the dynamic obstacle environment, the above distance-based modeling methods lose the velocity information of obstacles and lack the ability to predict whether a collision will occur, which reduces the safety of the system. When humans move, they tend to predict the trajectory of obstacles in their field of vision. If they think the obstacle is moving towards them, they will take action to avoid it in advance, even if the obstacle is far away from them. Therefore, the motion state of obstacles should be considered in order to achieve better dynamic obstacle avoidance. Some methods based on velocity are proposed. The velocity obstacle method is widely used in obstacle avoidance motion planning of multi-agent systems [18–20]. The robot is modeled as a circle or sphere, and the velocity obstacle model is established based on the position and velocity information of the robot. The velocity obstacle method is also applied to the manipulator by modeling the manipulator as a series of discrete spheres along the links [21]. Additionally, dynamic movement primitive is used in [22] to realize obstacle avoidance of the end effector by adjusting the end effector velocity direction according to the position and velocity information of the end effector and the obstacle.

However, the current obstacle avoidance methods are mainly distance-based, and the model of manipulator links is simplified into a point or sphere set. The discrete model may result in the discontinuous joint velocity when the number of sampling points in the discrete model is too small and may cause a large computation when the number of sampling points is too large. Therefore, the current problem in the field of dynamic obstacle avoidance is the lack of a velocity-based continuous obstacle avoidance model to represent the geometric configuration of the manipulator.

From the perspective of topology, the triangular relationship between two adjacent links in the three-dimensional space remains unchanged, so the geometric configuration of the redundant manipulator can be described by the motion and deformation of the triangles. Therefore, the adaptive adjustment of the manipulator in the dynamic environment is transformed into the dynamic obstacle avoidance of the triangles. Instead of being discretized into the point or sphere set, the manipulator link is considered as the side of a triangle. We propose the triangular collision plane model with three kinds of cost functions and combine them with the self-motion term of the gradient projection method [5–8,23] to calculate the inverse kinematics solution of the redundant manipulator, which solves the conflict between obstacle avoidance and task execution.

This paper is organized as follows. Section 2 describes the gradient projection method, which can make the redundant manipulator track objective and avoid obstacles at the same time. In Section 3, we introduce the triangular collision plane model we proposed and the method to realize dynamic obstacle avoidance based on this model. The corresponding simulations and experiments of our method and the contrast method are shown in Section 4. Finally, the conclusion is drawn in Section 5.

2. Problem Formulation

When the dimension of the joint space n exceeds the dimension of the task space m , the manipulator is redundant. There are infinite configurations of redundant manipulators

for the same end effector pose, which ensures the coordination of tracking objectives and obstacle avoidance.

The relationship between the joint angular velocity vector $\dot{\theta}$ of the n degrees of freedom (DOF) redundant manipulator and the end effector velocity vector \dot{X} can be described by the following equation:

$$\dot{X} = J\dot{\theta} \quad (1)$$

where \dot{X} is the end effector velocity vector in m -dimensional task space, $\dot{\theta} = (\dot{\theta}_1, \dots, \dot{\theta}_n)$ is the n -dimensional joint velocity vector, and J is the $m \times n$ Jacobian matrix.

The gradient projection method is widely used to obtain the inverse kinematics of the redundant manipulator. It decouples the motion of the redundant manipulator into the end effector motion and the self-motion. The inverse kinematic solution can be calculated by the gradient projection method [23] as follows,

$$\dot{\theta} = J^+\dot{X} + k(I - J^+J)\dot{\phi} \quad (2)$$

where k is a gain, $J^+ = J^T(JJ^T)^{-1}$ is the pseudoinverse of J , I is an $n \times n$ identity matrix, $J^+\dot{X}$ is the minimum norm solution of (1), $(I - J^+J)$ is the projection operator, $\dot{\phi}$ is an arbitrary vector in $\dot{\theta}$ -space used to achieve some additional goals, and $(I - J^+J)\dot{\phi}$ is the general solution of the homogeneous linear equations, which constitutes the null space of the redundant manipulator Jacobi matrix at the same time.

The motion of the joints generated in the null space does not change the pose of the end effector, so the motion of the redundant manipulator can be decoupled into the motion of the end effector by $J^+\dot{X}$ in (2) and the self-motion of the manipulator by $(I - J^+J)\dot{\phi}$ in (2). In order to track the objective, the desired velocity of the end effector \dot{X} is defined as

$$\dot{X} = \dot{X}_{\text{obj}} + K_p(X_{\text{obj}} - X) \quad (3)$$

where K_p is a gain for the pose error $X_{\text{obj}} - X$, X_{obj} is the objective pose of the end effector, \dot{X}_{obj} is the velocity vector of the objective, and X is the pose of the end effector.

The collision cost function $E(\theta, A)$, composed of the joint angle vector θ and the obstacle A , can be built to describe the relationship between the manipulator and the obstacle. In order to reduce the collision cost $E(\theta, A)$, $\dot{\phi}$ in the gradient projection method can be defined as

$$\dot{\phi} = -\nabla E(\theta, A) \quad (4)$$

where $\nabla E(\theta, A)$ is the gradient of the collision cost function $E(\theta, A)$.

The iterative method we used to control the motion of the manipulator is

$$\theta_{d+1} = \theta_d + \Delta\theta_d \quad (5)$$

where $\Delta\theta_d = \dot{\theta}_d \cdot \Delta T$ is the joint angle increment, and ΔT is the sampling time interval.

Remark 1. *The collision cost function is the key to solving the obstacle avoidance problem because it reflects the building way of the dynamic obstacle avoidance model. The traditional methods only use the distance to build the obstacle avoidance model, while our method applies the triangle element and velocity information into the model to establish the corresponding collision cost to improve the system's safety.*

The gradient projection method introduced in Section 2 is the framework used in our dynamic obstacle avoidance method. The focus of this paper is on the innovation of the dynamic obstacle avoidance model reflected in the collision cost function. For the sake of logical rationality and coherence, we introduce our dynamic obstacle avoidance model in Section 3 after introducing the gradient projection method in Section 2.

3. Obstacle Avoidance Method Design

In this section, we will introduce the proposed triangular collision plane model and 3 kinds of cost functions based on this model, obtain the obstacle avoidance optimization objective by the collision cost neural network, and design the control method for dynamic obstacle avoidance of the redundant manipulator.

3.1. Predictable Obstacle Avoidance Model

First of all, we simplify the redundant manipulator, which is connected by the pitch joint and yaw joints alternately, as shown in Figure 1a. The obstacle is regarded as a point, and the link of the manipulator is simplified as a line segment.

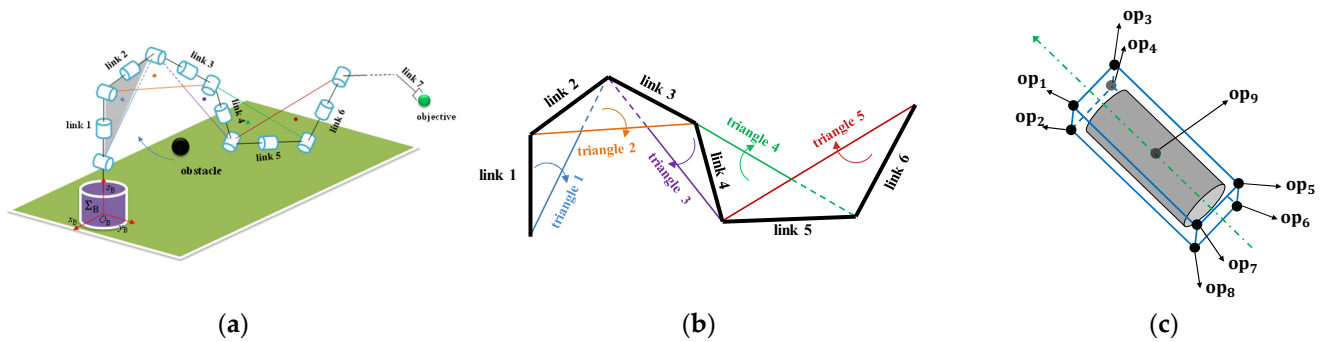


Figure 1. The simplified manipulator and obstacle in the triangular collision plane method. (a) The simplified manipulator and obstacle models; (b) the triangles in the same plane for demonstration; (c) the obstacle model.

We choose two adjacent links as the two sides of a triangle and connect the starting point of the first side and the endpoint of the second side to form a triangle. The motion and deformation of the triangle are determined by joint angles, so the configuration of the manipulator can be described by the pose and shape of the triangles. Therefore, the triangles established according to the manipulator configuration can be used to replace the manipulator body for obstacle avoidance motion planning. As shown in Figure 1b, for the convenience of displaying the model of the triangles built based on the manipulator configuration, an appropriate configuration is selected so that all triangles in Figure 1a are located in the same plane.

Considering the volume and different shapes of the real obstacle, we first use a bounding box surrounding it by the oriented bounding box (OBB) method. Then, we choose 8 vertices of the box and the geometric center of the box as the obstacle points to reflect the motion of the obstacle, as shown in Figure 1c. The i th triangle t_i matches the j th obstacle point op_j to form a pair, $p_{i,j}(t_i, op_j)$.

In Figure 2, we choose the i th ($i = 1$) triangle, shown in Figure 1a as the grey one, to introduce our obstacle avoidance model. The points and vectors in Figure 2 are relative to the base coordinate Σ_B of the manipulator. We propose 3 kinds of cost functions, including the *cost of motion state* (CMS), the *cost of head-on collision* (CHOC), and the *cost of approach time* (CAT). W is the unit vector of the relative velocity between the obstacle and the centroid O_i of the triangle, and the CMS is built by the angle γ_i between W and the unit normal vector N_i of the triangle. The obstacle along the direction of W has an intersection point C_i with the plane P where the triangular collision plane is located. There is a position offset δ_i between C_i and O_i . The CHOC is calculated by δ_i and constrained by the 3 sides of the triangle. d_i is the approach distance between the position of the obstacle X_{obs} and the plane P . The CAT is composed of d_i and the relative velocity between the obstacle velocity \dot{X}_{obs} and the centroid velocity \dot{O}_i . Based on the above 3 costs, we establish the *triangular collision plane* as the geometric obstacle avoidance model of the manipulator. More details about the model are shown as follows.

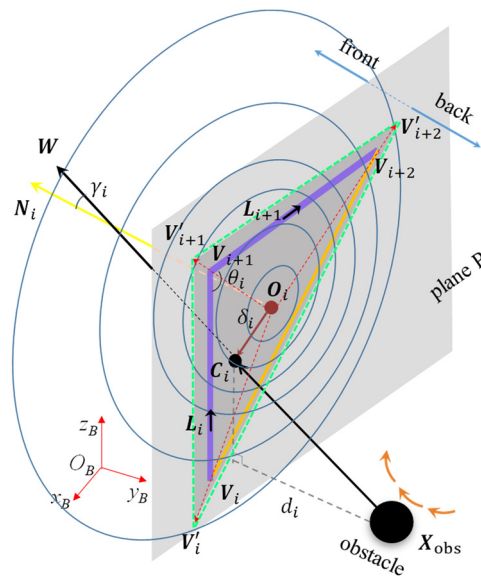


Figure 2. The triangular collision plane model.

As shown in Figure 2, we define the direction of the unit vector L_i of the i th link as pointing from the vertex V_i to the vertex V_{i+1} , and the unit vector L_{i+1} is defined in the same way. N_i is determined by L_i and L_{i+1} according to the right-hand rule. The plane P divides the space into two parts. The plane P where the triangular collision plane is located is expressed as

$$A_i x + B_i y + C_i z + D_i = 0 \quad (6)$$

where A_i , B_i , and C_i are the 3 components of N_i in x , y , and z coordinates, respectively, and $D_i = -(A_i x_i + B_i y_i + C_i z_i)$. (x_i, y_i, z_i) is the coordinate of V_i . We define the part that N_i points to as the front of the space, and the other part is the back. We substitute the coordinate of the obstacle $(x_{obs}, y_{obs}, z_{obs})$ into (6). If the value is positive, it means that the obstacle is in the front part; if the value is negative, it means that the obstacle is in the back part; and if the value is zero, it means the obstacle is on the plane P .

The CMS is different in the two parts. The inner product η_i is calculated by

$$\eta_i = W \cdot (L_i \times L_{i+1}) \quad (7)$$

The motion states of the obstacle relative to the triangular collision plane in each part are defined as 3 kinds based on the value of η_i , including the *head-on state*, the *parallel state*, and the *escaping state*. When the obstacle is in the front part, if η_i is between -1 and 0 (γ_i is between 90° and 180°), the obstacle is in the head-on state, and we consider that this state can increase the likelihood of the collision. If η_i is between 0 and 1 (γ_i is between 0° and 90°), the obstacle is in the escaping state, and we consider the likelihood of the collision occurrence to be equal to 0. If η_i is 0 (γ_i is 90°), the obstacle is in the parallel state and will not collide with the triangular collision plane as well. We classify the parallel state into the escaping state in the definition of the CMS. The CMS of the front part is expressed as

$$\text{CMS}_f = \begin{cases} \alpha \cdot \eta_i^2, & \eta_i \in [-1, 0) \\ 0, & \eta_i \in [0, 1] \end{cases} \quad (8)$$

where α is a gain and is usually equal to 1.

It is the opposite when the obstacle is in the back part. If η_i is between -1 and 0 (γ_i is between 0° and 90°), the obstacle is in the escaping state. If η_i is between 0 and 1 (γ_i is between 90° and 180°), the obstacle is in the head-on state. The CMS of the back part is given as

$$\text{CMS}_b = \begin{cases} 0, & \eta_i \in [-1, 0] \\ \alpha \cdot \eta_i^2, & \eta_i \in (0, 1] \end{cases} \quad (9)$$

When the value of the CMS is not 0, it does not indicate that the collision must occur. Next, we introduce the CHOC into our model to restrict the likelihood of the collision.

The line which passes the position of the obstacle along the direction of W is expressed as

$$(x - x_{\text{obs}})/m = (y - y_{\text{obs}})/n = (z - z_{\text{obs}})/q \quad (10)$$

where $(x_{\text{obs}}, y_{\text{obs}}, z_{\text{obs}})$ is the coordinate of the obstacle, and m, n, q are the 3 components of W . Combing (6) and (10), we can obtain C_i and determine whether it is in the triangle.

Considering the geometric volume of the real manipulator and the obstacle, we define the slack factor ϵ (≥ 1), which enlarges the triangle appropriately and proportionally. The slack vertices V'_i, V'_{i+1} , and V'_{i+2} are calculated by

$$\overrightarrow{O_i V'_r} = \epsilon \cdot \overrightarrow{O_i V_r} \quad (11)$$

where r is equal to $i, i + 1, i + 2$. We use the slack triangle to replace the initial triangle in the following. ϵ can directly affect the response time of the obstacle avoidance by controlling the size of the triangle, which determines the influence range of our model. The sides of the slack triangle are shown by the green dashed line in Figure 2, and the dark gray part is the slack triangular collision plane.

For dynamic obstacle avoidance, the manipulator is replaced by triangles to plan obstacle avoidance motion. Considering the geometric volume of the manipulator links, we use a slack factor ϵ to enlarge the size of the triangle in order to make the triangles cover the manipulator links. Therefore, the manipulator link is located between the center and boundary of the triangle actually. The goal is to ensure that both the center and boundary of the triangle do not collide with obstacles. Therefore, when constructing the CHOC, set the value of CHOC at the center of the triangle to the maximum. It makes the obstacle have a motion direction from the center of the triangle to the outside of the boundaries relative to the triangle.

In each control cycle, O_i has a distance to each vertex of the slack triangle, and among them, the largest one is defined as the maximum distance, μ_i . The closer O_i and C_i are, the higher the possibility of collision occurrence. The CHOC is defined as

$$\text{CHOC} = \begin{cases} \beta \cdot (\delta_i / \mu_i - 1)^2 \cdot e^{-\lambda \cdot \Delta t}, & \text{if } C_{i0} \text{ is in the triangle} \\ \beta \cdot (\delta_i / \mu_i - 1)^2 \cdot (1 - e^{-\lambda \cdot \Delta t}), & \text{otherwise} \end{cases} \quad (12)$$

where β is a gain, Δt is the duration beginning from the obstacle entering or leaving the triangle boundary and is equal to 0 initially, λ is a smooth factor, and C_{i0} is C_i in the initial state of the obstacle. The calculation of CHOC is chosen by the relative position between C_{i0} and the triangle. If C_i is in the triangle, it is possible for the obstacle to collide with the triangular collision plane, and CHOC is not equal to 0; otherwise, if C_i is outside the triangle or C_i does not exist when the obstacle is in a parallel state relative to the triangular collision plane, the collision will not happen, and CHOC is equal to 0. The circular contour makes the calculation simple in (12), but the shape difference between the circular cost contour and triangular boundary causes a great change in the CHOC value at the triangular boundary on the space scale. As for this, $e^{-\lambda \cdot \Delta t}$ is added to smooth the cost at the triangular boundary on the time scale and only changes when C_i crosses the boundary.

We consider that when the obstacle is in the head-on state, and C_i is in the slack triangle, the obstacle will collide with the triangular collision plane as long as the obstacle keeps the same velocity direction. Then, we use the CAT to describe the time urgency of the collision when the obstacle is approaching the triangular collision plane. It is defined as

$$\text{CAT} = \rho \|\dot{\mathbf{X}}_{\text{obs}} - \dot{\mathbf{O}}_i\| / d_i \quad (13)$$

where ρ is a gain. When the distance is large, the CAT is not 0. The distance-based obstacle avoidance methods usually require setting a safe distance and do not avoid obstacles beyond a certain safe distance. However, for dynamic obstacle avoidance, the safe distance is often difficult to determine, and factors such as the velocity of the obstacle need to be considered. We do not need to set a safe distance in our obstacle avoidance model by using the distance between the obstacle and the triangular collision plane to calculate CAT. The farther the distance, the lower the value of CAT, which ensures the obstacle avoidance model works without limitation by the safe distance.

The above 3 cost functions represent the occurrence conditions of the collision. Only if all 3 conditions are satisfied will the collision occur; that is, CMS, CHOC, and CAT are not equal to 0 at the same time. Therefore, the collision cost function $E_{i,j}$ between the i th triangular collision plane and the j th obstacle point should be composed of the multiplication of the above 3 cost functions and is defined as

$$E_{i,j}(\boldsymbol{\theta}, \text{op}_j) = \text{CMS} \cdot \text{CHOC} \cdot \text{CAT} \quad (14)$$

The calculation process of $E_{i,j}$, shown in the flow chart in Figure 3a, can be considered as a collision cost neuron $\Gamma(\mathbf{p}_{i,j})$ with the input $\mathbf{p}_{i,j}(t_i, \text{op}_j)$ and the output $E_{i,j}$.

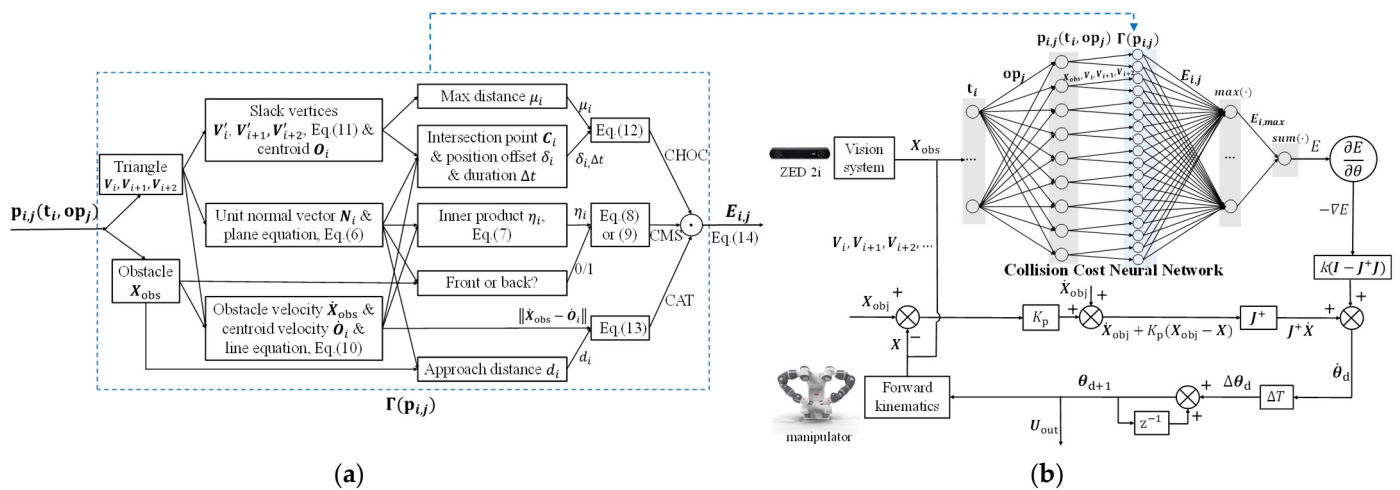


Figure 3. The triangular collision plane method for dynamic obstacle avoidance. (a) The collision cost neuron; (b) the control diagram of the manipulator system.

The collision cost neural network to calculate the optimization objective E is constructed, as shown in Figure 3b. The inputs of the network are triangles and obstacle points, and the output is the total cost. The maximal $E_{i,j}$ between one triangle plane collision and all obstacle points is regarded as $E_{i,\text{max}}$. We add all $E_{i,\text{max}}$ by weights w_i to obtain the total cost E ,

$$E = \sum_{i=1}^s w_i E_{i,\text{max}} \quad (15)$$

where s is the number of the triangular collision plane. The weights w_i are set to 1 in our model, which means that the weight of the impact of each triangular collision plane on the obstacle avoidance motion is the same.

The control diagram of the manipulator system with tracking and obstacle avoidance is also shown in Figure 3b. The input of the system is the obstacle point position \mathbf{X}_{obs} and the objective pose \mathbf{X}_{obj} of the end effector, and the output U_{out} is the command of the joint angles.

For our model, the setting of triangles for the configuration of the manipulator is also important. The more triangles divided, the more accurate the obstacle avoidance model. However, an increase in the number of triangles can lead to a decrease in computational efficiency. Therefore, there is a trade-off between the model accuracy and the computational efficiency. In the process of modeling, we consider the length of manipulator links, and if the link is short enough, it can be ignored in order to improve computational efficiency. However, the ignored link can be covered by the adjustment of the slack factor ϵ in (11).

3.2. Modeling Strategy for Singularity

When the manipulator and the obstacle are in one of the following two situations, we consider that the obstacle avoidance model reaches the singular state:

- (1) The obstacle is nearly moving on plane P in Figure 2; that is, d_i is close to 0, and the obstacle is in parallel state.
- (2) The two adjacent links are close to collinear; that is, θ_i is close to 180° . In this case, the triangular collision plane cannot be formed.

In order to make our model work in the singular state, we simplify the triangular collision plane based on two adjacent links into the *collision line segment* based on each link shown as the green dashed line in Figure 4. The gray plane P' is composed of the i th link and the obstacle position, and all variables in the model are represented on this plane. N_i is the unit normal vector of the link on the plane now. W is determined by the projection of the obstacle velocity and centroid velocity on the plane P' . The slack vertices V'_i and V'_{i+1} replace the initial vertices, and μ_i is half of the slack link length. C_i is the intersection point between the line which passes the obstacle along the direction of W and the line where the collision line segment is located. d_i is the approach distance between the position of the obstacle and the collision line segment. The CMS and CAT of the collision line segment are the same as that of the triangular collision plane. There is no shape difference, so $e^{-\lambda \cdot \Delta t}$ can be removed in CHOC. The obstacle avoidance motion is performed on the plane P' , and this model is used only when the triangular collision plane model is close to the singular state.

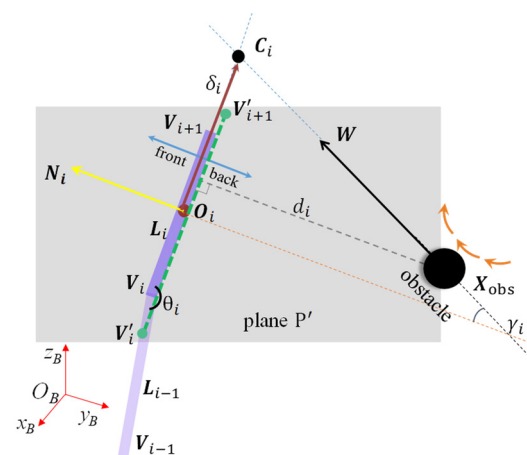


Figure 4. The collision line segment model.

4. Simulation and Experiment

We perform simulations and experiments on ABB Yumi with two arms with seven DOF. In order to verify our model and compare it with distance-based dynamic obstacle avoidance methods, we choose the obstacle avoidance point method (OAPM) [7], which is representative of the contrast method. For the sake of fairness, we design the same obstacle avoidance experiment as the one in [7]. Two sets of simulations and experiments are performed to compare the triangular collision plane method (TCPM) with the OAPM. Because of the limitation of the degrees of freedom of the ABB Yumi, it cannot maintain both the position and the orientation of the end-effector and avoid obstacles at the same time. In

order to better show the obstacle avoidance effect, in the one-triangle case, we make the end-effector maintain the position, while in the two-triangle case, we make the end-effector maintain the orientation. The objective velocity \dot{X}_{obj} in the one-triangle/two-triangle case is set to the three-dimensional zero vector. A moving objective does not have an effect on obstacle avoidance. Based on the redundancy and the gradient projection method from Equation (2), the motion of obstacle avoidance generated in the null space does not influence the motion of task execution for tracking a moving objective. The purpose of setting the objective velocity to zero is just convenient to show the obstacle avoidance effect.

In order to compare clearly, we use a yellow ping-pong ball as the dynamic obstacle. It can be considered as only one obstacle point. In the simulation, the obstacle with accelerated rectilinear motion approaches the manipulator. In the experiment, the vision system uses ZED 2i camera to identify the obstacle in the environment and uses the Kalman filter to process the obstacle position information.

In the OAPM, the obstacle avoidance point is defined as the closest point on the manipulator body to the obstacle. The distance between them is called critical distance, d_o . Using the unit vector n_o , which points from the obstacle to the obstacle avoidance point, and combining with the gradient projection method to solve inverse kinematics, the exact solution with reduced operational space is proposed in [7] as

$$\dot{\theta} = J^+ \dot{X} + \alpha_\eta [J_{d_o} (\mathbf{I} - J^+ J)]^+ (\dot{x}_o - J_{d_o} J^+ \dot{X}) \quad (16)$$

where J^+ is the pseudoinverse of the J associated with the end effector, \dot{X} is the $m \times 1$ desired velocity of the end effector calculated from (3), $J_{d_o} = n_o^T J_o$ is a $1 \times n$ matrix, J_o is the Jacobian matrix associated with the obstacle avoidance point, $\dot{x}_o = \alpha_o \cdot v_o$ is the desired obstacle avoidance point velocity, v_o is the nominal velocity, and $\alpha_o(d_o)$ and $\alpha_\eta(d_o)$ are the gains related to the critical distance and limited by the distance of influence range d_i and the unity gain distance d_m .

4.1. One-Triangle Case

The right arm of ABB Yumi without a hand is used in the one-triangle case. We select the shoulder, elbow, and wrist, namely the coordinate origins of the first, third, and fifth joints, as the three vertices of the triangle. The lines connecting the selected vertices are considered as the triangle sides to form a triangle. In this simulation and experiment, the manipulator avoids obstacles and maintains the position of the end effector at the same time. The screenshots per second of the simulations and experiments are shown in Figure 5a,b.

Comparing the starting time when the joint angles, velocities, and accelerations change by the two methods in Figure 5c,e, we can see that the time of obstacle avoidance motion generation by the TCPM is earlier than that by the OAPM. It proves that the TCPM can predict, take action in advance and improve the safety of the system. The changes in the joint angles, velocities, and accelerations in experiments shown in Figure 5d,f are the same as the simulations. From the comparison, it can be seen that the accelerations of the TCPM are smoother than the ones of the OAPM and with fewer oscillations. When the obstacle is captured by the vision system, the triangular collision plane model is used to determine whether the obstacle will collide in the current motion state. If a collision occurs, obstacle avoidance action will be taken; otherwise, there is no action. In the beginning, due to the long distance and low obstacle velocity, the effect of obstacle avoidance is weak. However, as the obstacle approaches, the effect is gradually enhanced. It can be seen from Figure 5g-i that when the intersection point C_i is out of the slack triangle, the CHOC and total cost smoothly drop to 0 by the effect of $e^{-\lambda \cdot \Delta t}$, and the manipulator completes the obstacle avoidance motion. Then, as long as the obstacle maintains the direction of velocity, collision will not happen, even if the approach distance is shortening and the CAT increases with the motion of the obstacle. Figure 5i shows the changes in the initial and slack triangular collision plane and the intersection points.

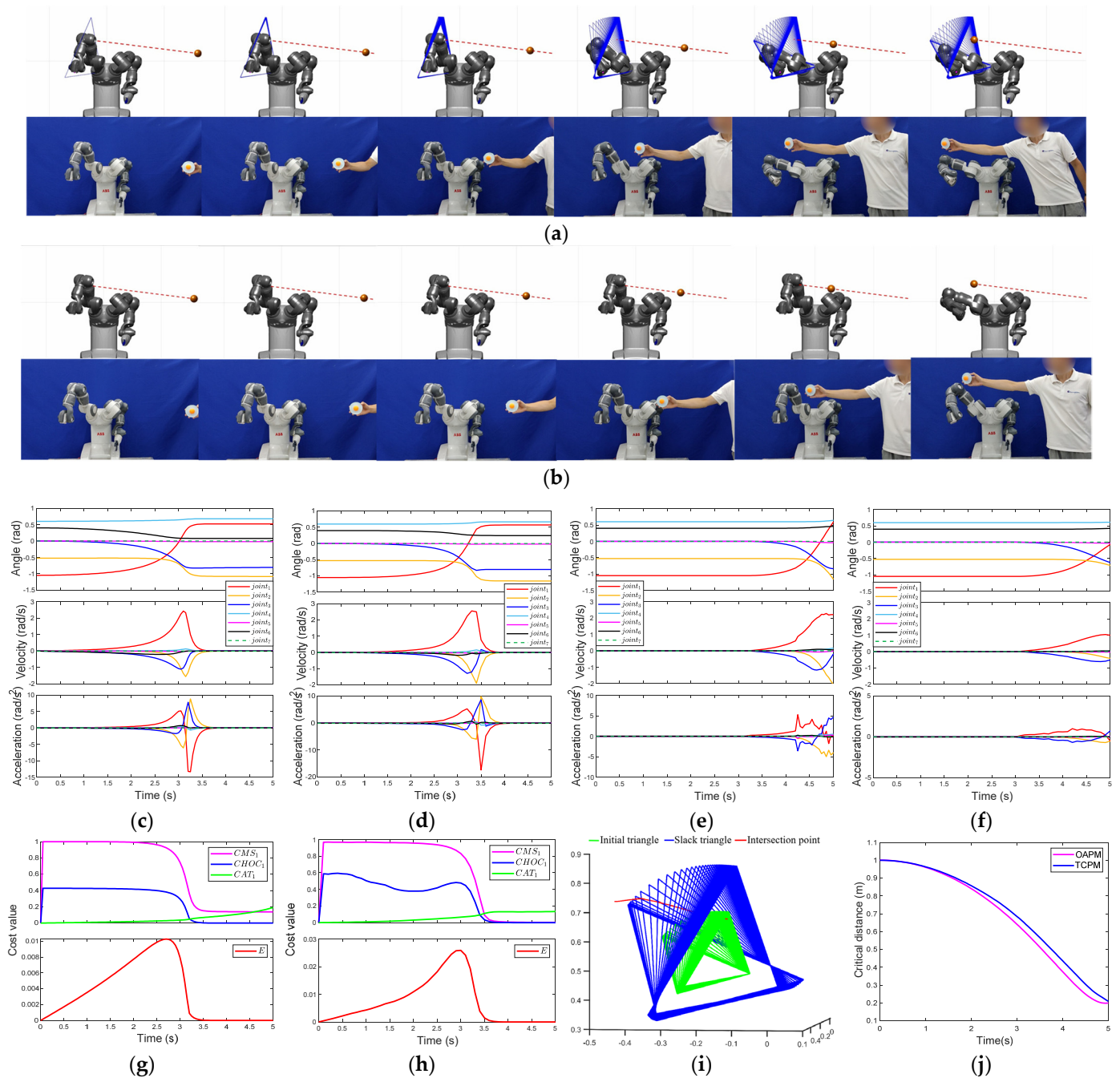


Figure 5. The comparison of TCPM and OAPM on ABB Yumi in the one-triangle case. (a) The simulation and experiment screenshots per second by TCPM; (b) the simulation and experiment screenshots per second by OAPM; (c) the joint angles, velocities, and accelerations in the simulation by TCPM; (d) the joint angles, velocities, and accelerations in the experiment by TCPM; (e) the joint angles, velocities, and accelerations in the simulation by OAPM; (f) the joint angles, velocities, and accelerations in the experiment by OAPM; (g) sub and total costs in the simulation by TCPM; (h) sub and total costs in the experiment by TCPM; (i) the change process of the triangular collision plane; (j) the comparison of the critical distance between TCPM and OAPM.

It is necessary to expand d_i and d_m to adapt to the dynamic obstacle in OAPM. However, the expansion of the distance range reduces the standard of collision. It can lead to failure in accurately determining whether a collision will occur and affect the task execution of the manipulator. We equidistantly select 105 discrete points on the manipulator

links to calculate the minimum distance between them and the obstacle. The number of discrete points should be set in an appropriate range to trade off the computation and the smoothness of obstacle avoidance velocity. Figure 5j shows the change in the critical distance between the two methods. Due to the prediction made in advance for obstacle avoidance, the TCPM makes the manipulator keep a larger critical distance in the process of obstacle approaching than the OAPM.

4.2. Two-Triangle Case

We apply our method to the two triangular collision planes case. We use the left arm of ABB Yumi with a two-finger hand and add the center of the hand as the fourth vertex. The shoulder, elbow, wrist, and hand are connected in turn to form two triangles. The first triangular collision plane is composed of the shoulder, elbow, and wrist, shown in blue lines in Figure 6a, and the second one is composed of the elbow, wrist, and hand, shown in magenta lines.

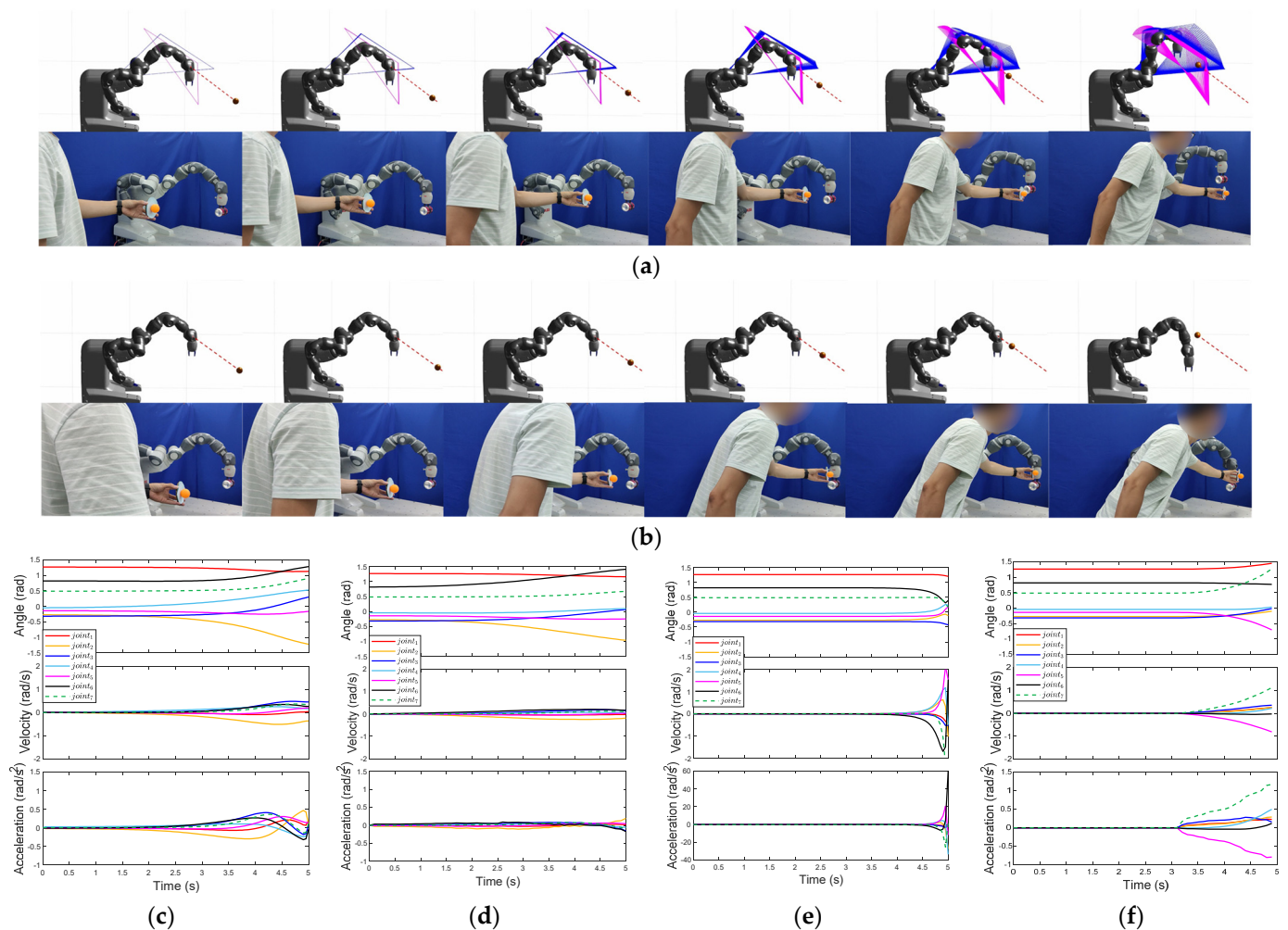


Figure 6. Cont.

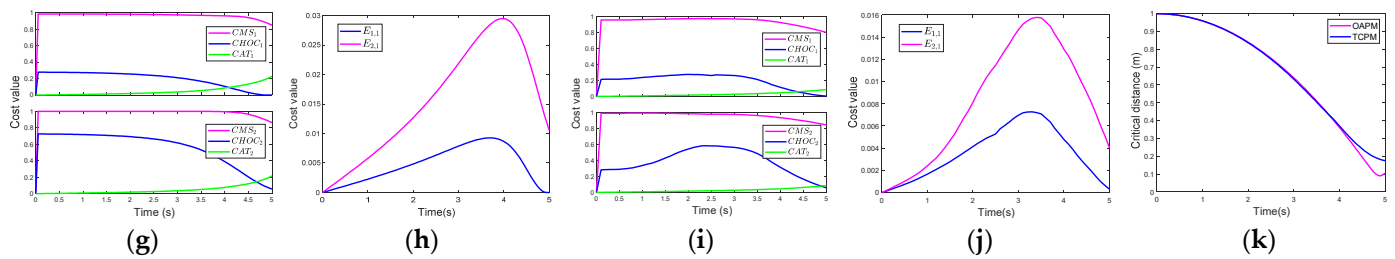


Figure 6. The comparison of TCPM and OAPM on ABB Yumi in the two-triangle case. (a) The simulation and experiment screenshots per second by TCPM; (b) the simulation and experiment screenshots per second by OAPM; (c) the joint angles, velocities, and accelerations in the simulation by TCPM; (d) the joint angles, velocities, and accelerations in the experiment by TCPM; (e) the joint angles, velocities, and accelerations in the simulation by OAPM; (f) the joint angles, velocities, and accelerations in the experiment by OAPM; (g) sub costs in the simulation by TCPM; (h) total costs in the simulation by TCPM; (i) sub costs in the experiment by TCPM; (j) total costs in the experiment by TCPM; (k) the comparison of the critical distance between TCPM and OAPM.

Considering the limit due to the number of DOF and the show effect, we make the manipulator only maintain the orientation of the end effector in this simulation and experiment. It can be seen from Figure 6 that the arm grasping a cup can avoid the obstacle and maintain the orientation of the end effector at the same time. At the beginning of the simulation and experiment, due to the approach of the obstacle, the CATs in the two triangular collision planes increase. It leads to an increase in total costs for both planes. However, with the gradually enhancing effect of our method, the total costs decrease in the end due to the CMSes and CHOCs, shown in Figure 6g–j. From the comparison between Figure 6c,e, it can be seen that our method can make earlier obstacle avoidance motion and hold a larger critical distance in the end, as shown in Figure 6k. The results show that our method also works in this case, and the obstacle avoidance model can be extended into the multiple triangles case with the increase in the links.

The average computation time of the TCPM is approximately 50 ms (one-triangle case) and 60 ms (two-triangle case) per iteration on a single CPU. The average computation time of the OAPM is approximately 1 ms on the same computing equipment. The computation time of most of the obstacle avoidance planning methods approximately lies between 1 ms and 2 s [13,15]. There is a trade-off between computational efficiency and dynamic obstacle avoidance performance. We sacrifice a certain level of computational efficiency to achieve predictable obstacle avoidance by considering the velocity of obstacles. The above simulations and experiments can verify our method works well in dynamic obstacle avoidance.

The value of the parameters in the simulation and experiment are presented in Table 1.

Table 1. Parameters used in the simulations and experiments.

Parameters	k	α	β	ρ	ϵ	v_o	K_p	d_i	d_m	ΔT_{sim}	ΔT_{exp}	λ
Value	10	1	1	1	2.0	0.03 m/s	1.0	0.8 m	0.6 m	0.05 s	0.1 s	10

5. Conclusions

In this paper, we bring the triangle element and velocity information into the dynamic obstacle avoidance model based on the geometric configuration of the redundant manipulator. We propose a predictive obstacle avoidance model, the triangular collision plane, build three kinds of cost functions and apply this model to the gradient projection method for the coordination between obstacle avoidance and task execution. The collision cost neural network is constructed by the collision cost neuron to obtain the collision cost. When the CMS, CHOC, and CAT are not equal to 0 at the same time, the collision will occur, and

obstacle avoidance motion is generated. This model has the ability to accurately predict the collision based on the motion of the obstacle, realizes the active obstacle avoidance of the manipulator, and improves the safety of the system. The other advantage is that the geometric continuity of our model ensures the smoothness of the obstacle avoidance motion without considering the sampling number of the discrete model. Both the simulations and the experiments in this paper show the characteristics of our methods. In future work, we will try to apply our method to multi-task coordination to further improve the system safety, and we will also try to shorten computation time from the aspect of further optimizing models, improving the code, and using the computing equipment with higher computation performance.

Author Contributions: Conceptualization, F.J. and H.J.; methodology, F.J.; software, F.J.; validation, F.J., H.J. and B.W.; formal analysis, F.J.; investigation, F.J. and B.W.; resources, F.J., H.J. and J.Z.; data curation, F.J. and B.W.; writing—original draft preparation, F.J.; writing—review and editing, F.J. and H.J.; visualization, F.J.; supervision, F.J. and H.J.; project administration, H.J.; funding acquisition, H.J. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by STI 2030—Major Projects 2021ZD0201400 and in part by the National Natural Science Foundation of China under Grant 92048301.

Data Availability Statement: The data is unavailable due to privacy.

Conflicts of Interest: No conflicts of interest exist in the submission of this manuscript, and the manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously and was not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

References

1. Haddadin, S.; Luca, A.D.; Albu-Schaffer, A. Robot Collisions: A Survey on Detection, Isolation, and Identification. *IEEE Trans. Robot.* **2017**, *33*, 21. [[CrossRef](#)]
2. Cortés, J.; Siméon, T. Sampling-Based Motion Planning under Kinematic Loop-Closure Constraints. In *Algorithmic Foundations of Robotics VI*; Erdmann, M., Overmars, M., Hsu, D., van der Stappen, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 17, pp. 75–90.
3. Ghajari, M.F.; Mayorga, R.V. Specialized PRM Trajectory Planning For Hyper-Redundant Robot Manipulators. *WSEAS Trans. Syst.* **2017**, *16*, 7.
4. Stilman, M. Task constrained motion planning in robot joint space. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 3074–3081.
5. Hua, X.; Wang, G.; Xu, J.; Chen, K. Reinforcement learning-based collision-free path planner for redundant robot in narrow duct. *J. Intell. Manuf.* **2020**, *32*, 471–482. [[CrossRef](#)]
6. Maciejewski, A.A.; Klein, C.A. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments. *Int. J. Robot. Res.* **1985**, *4*, 109–117. [[CrossRef](#)]
7. Petrič, T.; Gams, A.; Likar, N.; Žlajpah, L. Obstacle Avoidance with Industrial Robots. In *Motion and Operation Planning of Robotic Systems*; Carbone, G., Gomez-Bravo, F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; Volume 29, pp. 113–145.
8. Zhang, H.; Jin, H.; Liu, Z.; Liu, Y.; Zhu, Y.; Zhao, J. Real-Time Kinematic Control for Redundant Manipulators in a Time-Varying Environment: Multiple-Dynamic Obstacle Avoidance and Fast Tracking of a Moving Object. *IEEE Trans. Ind. Inf.* **2019**, *16*, 28–41. [[CrossRef](#)]
9. Ericson, C. Real-time collision detection. In *The Morgan Kaufmann Series in Interactive 3D Technology*; Eberly, D.H., Ed.; Morgan Kaufmann Publishers: Burlington, VT, USA, 2005.
10. Ratliff, N.; Zucker, M.; Bagnell, J.A.; Srinivasa, S. CHOMP: Gradient optimization techniques for efficient motion planning. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 489–494.
11. Kalakrishnan, M.; Chitta, S.; Theodorou, E.; Pastor, P.; Schaal, S. STOMP: Stochastic trajectory optimization for motion planning. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 4569–4574.
12. Park, C.; Pan, J.; Manocha, D. ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments. In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), Sao Paulo, Brazil, 25–29 June 2012; pp. 207–215.

13. Schulman, J.; Ho, J.; Lee, A.; Awwal, I.; Bradlow, H.; Abbeel, P. Finding locally optimal, collision-free trajectories with sequential convex optimization. In Proceedings of the Robotics: Science and systems (RSS), Berlin, Germany, 24–28 June 2013; Volume 9, pp. 1–10.
14. Escande, A.; Miossec, S.; Kheddar, A. Continuous gradient proximity distance for humanoids free-collision optimized-postures. In Proceedings of the 2007 7th IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA, USA, 29 November–1 December 2007; pp. 188–195.
15. Salehian, S.S.M.; Figueroa, N.; Billard, A. A unified framework for coordinated multi-arm motion planning. *Int. J. Robot. Res.* **2018**, *37*, 1205–1232. [[CrossRef](#)]
16. Nie, J.; Wang, Y.; Mo, Y.; Miao, Z.; Jiang, Y.; Zhong, H.; Lin, J. An HQP-Based Obstacle Avoidance Control Scheme for Redundant Mobile Manipulators Under Multiple Constraints. *IEEE Trans. Electron.* **2023**, *70*, 6004–6016. [[CrossRef](#)]
17. Khan, A.H.; Li, S.; Luo, X. Obstacle Avoidance and Tracking Control of Redundant Robotic Manipulator: An RNN-Based Metaheuristic Approach. *IEEE Trans. Ind. Inf.* **2019**, *16*, 4670–4680. [[CrossRef](#)]
18. van den Berg, J.; Guy, S.J.; Lin, M.; Manocha, D. Reciprocal n-Body Collision Avoidance. In *Robotics Research*; Pradalier, C., Siegwart, R., Hirzinger, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 70, pp. 3–19.
19. Choi, M.; Rubenecia, A.; Shon, T.; Choi, H.H. Velocity Obstacle Based 3D Collision Avoidance Scheme for Low-Cost Micro UAVs. *Sustainability* **2017**, *9*, 1174. [[CrossRef](#)]
20. Han, R.; Chen, S.; Wang, S.; Zhang, Z.; Gao, R.; Hao, Q.; Pan, J. Reinforcement Learned Distributed Multi-Robot Navigation With Reciprocal Velocity Obstacle Shaped Rewards. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5896–5903. [[CrossRef](#)]
21. Zhao, L.; Zhao, J.; Liu, H. Solving the Inverse Kinematics Problem of Multiple Redundant Manipulators with Collision Avoidance in Dynamic Environments. *J. Intell. Robot. Syst.* **2021**, *101*, 30. [[CrossRef](#)]
22. Hoffmann, H.; Pastor, P.; Park, D.-H.; Schaal, S. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 2587–2592.
23. Wampler, C.W. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 93–101. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.