Article

Analysis of Obstacle Avoidance Strategy for Dual-Arm Robot Based on Speed Field with Improved Artificial Potential Field Algorithm

Hui Zhang [1,2]**, Yongfei Zhu** [1]**, Xuefei Liu** [1] **and Xiangrong Xu** [1,*]

[1] School of Mechanical Engineering, Anhui University of Technology, Ma Anshan 243032, China; 3181@ahut.edu.cn (H.Z.); zyf19950410@163.com (Y.Z.); liuxf618@163.com (X.L.)
[2] School of Metallurgical Engineering, Anhui University of Technology, Ma Anshan 243032, China
[*] Correspondence: xuxr@ahut.edu.cn

Abstract: In recent years, dual-arm robots have been favored in various industries due to their excellent coordinated operability. One of the focused areas of study on dual-arm robots is obstacle avoidance, namely path planning. Among the existing path planning methods, the artificial potential field (APF) algorithm is widely applied in obstacle avoidance for its simplicity, practicability, and good real-time performance over other planning methods. However, APF is firstly proposed to solve the obstacle avoidance problem of mobile robot in plane, and thus has some limitations such as being prone to fall into local minimum, not being applicable when dynamic obstacles are encountered. Therefore, an obstacle avoidance strategy for a dual-arm robot based on speed field with improved artificial potential field algorithm is proposed. In our method, the APF algorithm is used to establish the attraction and repulsion functions of the robotic manipulator, and then the concepts of attraction and repulsion speed are introduced. The attraction and repulsion functions are converted into the attraction and repulsion speed functions, which mapped to the joint space. By using the Jacobian matrix and its inverse to establish the differential velocity function of joint motion, as well as comparing it with the set collision distance threshold between two robotic manipulators of robot, the collision avoidance can be solved. Meanwhile, after introducing a new repulsion function and adding virtual constraint points to eliminate existing limitations, APF is also improved. The correctness and effectiveness of the proposed method in the self-collision avoidance problem of a dual-arm robot are validated in MATLAB and Adams simulation environment.

Keywords: dual-arm robot; improved artificial potential field algorithm; velocity field; MATLAB and Adams

1. Introduction

In recent years, dual-arm robots have gradually emerged in the manufacturing and service industries and in other fields due to their strong coordination, operability, and high work efficiency [1–3]. Compared with single robotic manipulator, the movement of dual-arm collaborative robot control is more difficult. Moreover, in terms of kinematics, it is necessary to consider the collision avoidance path planning between the two robotic manipulators, while in view of dynamics, it is necessary to coordinate motion control among multiple joints. These cause the complication of control system obviously [4,5]. For most dual-arm robots, there are overlapping parts in the working areas of the two arms, so while one robotic manipulator is moving, this manipulator is being treated as a dynamic obstacle by the other robotic manipulator. Therefore, either one of the robotic manipulators ought to avoid dynamic obstacle in order to accomplish path planning [6,7].

According to the application of robot in production and in life, it can be known that the obstacles include two types, namely static obstacles and dynamic obstacles, and the

collision avoidance problem is essentially a robot path planning problem. Lots of researchers have spent a lot of time and energy on the above issue for deep study and have put forward many classic algorithms, such as artificial potential field method (APF) [8–10], rapidly exploring random tree (RRT) [11–13], C-space [14,15], grid-based algorithms [16,17], and a new novel approach for the application of dynamic safety zones based on the requirements of safety standards for collaborative robotics [18]. Among the above methods, the artificial potential field (APF) algorithm is widely applied in obstacle avoidance for its simplicity, practicability, and good real-time performance over other planning methods. However, owing to the fact that APF was firstly proposed to solve the obstacle avoidance problem of mobile robot in plane, some limitations have been produced. Typically, it is prone to fall into local minimum [19]. When encountering dynamic obstacles, it will not be applicable. For instance, when the object is far from the goal point, the attraction force becomes very large. Compared with the smaller repulsion force, the object may encounter obstacles in the path. Furthermore, at a certain position, the attraction force and the repulsion force on the object are equal, but the directions of those two forces are opposite, leading the object to fall into a local optimal solution. Consequently, all these limitations described above cannot make the robot arrive at the goal point.

To overcome the above disadvantages and deficiencies of the traditional APF (T-APF), a lot of improvements of APF have been proposed in recent years. For the problems that the APF often converges to local minima and hardly reaches the ending and oscillatory movement, the concept of gravity chain based on APF was proposed by Lei et al [20]. In their method, effective information of obstacle avoidance was put into the potential field through the gravity chain to generate a steering angle tangent to the rubber band. By using that angle to connect with the beginning and ending in the space of obstacle potential field, instead of the angle of artificial potential field, the local minima problem could be solved. To reduce oscillations and reach the goal in less time [21], a mixed algorithm of APF and RRT was proposed, in which APF provided a simple and effective path planning method while RRT provided a random disturbance. In addition, the NP-APF (new point-APF), which created the new point of the attractive force to improve performance of the APF, was also presented [22] to avoid those drawbacks of T-APF. To search for the goal point in unknown 2D environments, the proposed-APF (P-APF) algorithm, which avoids the deadlock and non-reachability problems of mobile robot navigation, was employed. However, that method is limited to the calculation of the effective front-face obstacle information associated with the velocity direction, such as size and shape of obstacle [23].

Inspired by the background described above and using previous research conducted by the authors for the obstacle avoidance case based on APF, this study presents an extension of the proposed algorithms based on potential field velocity (VP-APF) to solve the collision avoidance problem of a dual-arm robot. In our method, T-APF algorithm is used to establish the attraction and repulsion functions of the robotic manipulator, and then the concepts of attraction and repulsion speed are introduced. In addition, the attraction and repulsion functions are converted into the attraction and repulsion speed functions, which are mapped to the joint space. The Jacobian matrix and its inverse is introduced to establish the differential velocity function of the joint motion. Finally, by comparing with the set collision distance threshold between two robotic manipulators of the robot, the problem of collision avoidance is solved. Meanwhile, a new repulsion function and adding virtual constraint points is introduced to improve the APF to eliminate existing limitations. The correctness and effectiveness of the proposed method in the self-collision avoidance problem of the dual-arm robot are validated in MATLAB and Adams simulation environment.

2. Kinematics Model of 6-DOF Dual-Arm Robot

To study the collision avoidance problem of the robotic arm, it is firstly necessary to analyze its kinematics, so as to establish the posture relationship between each link of the robotic arm. A six-degree-of-freedom dual-arm robot was used as the research object. Both

the left arm and the right arm are composed of six rotating joints with a symmetrical structure, and the joints are numbered from 1 to 6, with 7 links in total. Link 0 is the base, which is a fixed joint. Every two adjacent joints are connected by a rotating joint.

Figure 1 shows the structure of the 6-DOF robotic manipulator, where 1–6 represents the rotating joints of the robotic manipulator, Z_1–Z_6, and X_1–X_6 are the coordinate axes corresponding to the joint numbers. In this paper, the improved D-H parameter method is used to establish the kinematics model of the manipulator. Any two adjacent links satisfy the following relationship (1):

$$^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Here, $S\theta_i = \sin\theta_i$, $C\theta_i = \cos\theta_i$, where Ti, $i = 0, 1, 2, .i - 1$ is a homogeneous matrix calculated from the D-H parameters.
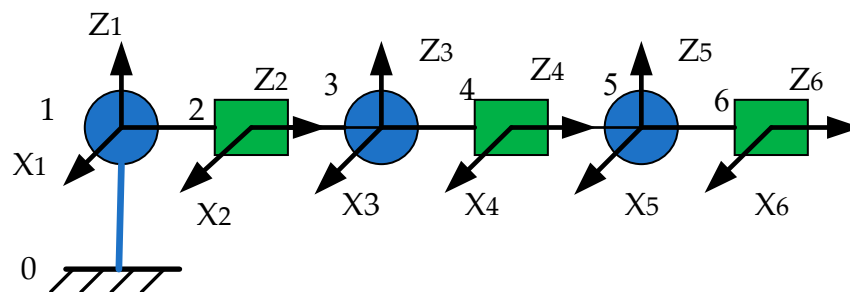

Figure 1. The structure of 6-DOF robotic manipulator.

The specific parameters of the robot arm are shown in Table 1. By substituting the parameters in the table into Equation (1), the homogeneous transformation matrix of each link can be obtained. Therefore, the coordinate system transformation of the base and the end of the robot can be expressed as (2):

$$^RT_6 = {^RT_1}\,{^1T_2}\,{^2T_3}\,{^3T_4}\,{^4T_5}\,{^5T_6} \tag{2}$$

Table 1. D-H parameters of the 6-DOF robot.

| Link i.	Joint angle θ	Twist angle α	Length of Linkages (m)	Offset of Linkages (m)
1	θ_1	0	0	0
2	θ_2	-pi/2	d2	0
3	θ_3	-pi/2	0	0
4	θ_4	-pi/2	d4	0
5	θ_5	-pi/2	0	0
6	θ_6	-pi/2	d6	0

3. Improved Artificial Potential Field Algorithm Based on Velocity Field

3.1. Calculation of The Distance between Two Robotic Manipulator of Dual-Arm Robot

Studying the collision problem of a dual-arm robot is the prerequisite for path planning. Generally speaking, it is necessary to know the closest distance between the robotic manipulator and the obstacle before a collision occurs. The obstacles faced by a dual-arm robot when working include two parts: one is the object in the working environment, and the other is the robot itself. This article analyzes and studies the self-collision problem of

the dual-arm robot, that is, when robotic manipulator is moving, the manipulator arm is regarded as an obstacle, in the robot's attempt to avoid collision.

Before the distance is calculated, the model of the robotic manipulator is simplified into a line segment with radius, as shown in Figure 2.
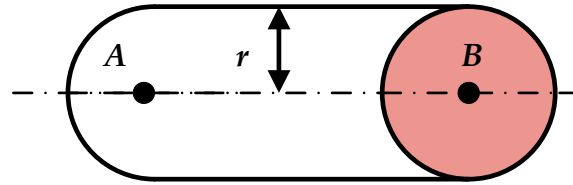
Figure 2. Simplified link structure.

To solve the distance between two lines in the space, line AB and CD are supposed, whose three-dimensional coordinates are A (x_1, y_1, z_1), B (x_2, y_2, z_2), C (x_3, y_3, z_3), D (x_4, y_4, z_4).

If there is a point M (X_1, Y_1, Z_1) in the line AB, its coordinates can be obtained by Equation (3):

$$\begin{cases} X_1 = x_1 + k_1(x_2 - x_1) \\ Y_1 = y_1 + k_1(y_2 - y_1) \\ Z_1 = z_1 + k_1(z_2 - z_1) \end{cases} \tag{3}$$

Provided there is a point N (X_2, Y_2, Z_2) in the line CD, its coordinates can be obtained by Equation (4):

$$\begin{cases} X_2 = x_3 + k_2(x_4 - x_3) \\ Y_2 = y_3 + k_2(y_4 - y_3) \\ Z_2 = z_3 + k_2(z_4 - z_3) \end{cases} \tag{4}$$

where k_1 and k_2 are natural numbers. When $0 \le k_1 \le 1$ and $0 \le k_2 \le 1$, the point M and N are on their respective line segments; otherwise, they are not. The distance between MN can be expressed as Equation (5):

$$\|MN\| = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \tag{5}$$

Next, the shortest distance between MN is calculated, and the parameter equation for k_1 and k_2 is established.

$$f(k_1, k_2) = MN^2 = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2 \tag{6}$$

Substitute Equations (3) and (4) into Equation (6),

$$f(k_1, k_2) = [(x_1 - x_3) + k_1(x_2 - x_1) - k_2(x_4 - x_3)]^2 + [(y_1 - y_3) + k_1(y_2 - y_1) - k_2(y_4 - y_3)]^2 + [(z_1 - z_3) + k_1(z_2 - z_1) - k_2(z_4 - z_3)]^2 \tag{7}$$

Then find the partial derivatives of k_1 and k_2 in Equation (7) and make them equal to zero:

$$\begin{cases} \dfrac{\partial f(k_1, k_2)}{\partial k_1} = 0 \\ \dfrac{\partial f(k_1, k_2)}{\partial k_2} = 0 \end{cases} \tag{8}$$

By solving the above formula, k_1 and k_2 can be obtained. The calculated results of k_1 and k_2 are the optimal solutions corresponding to the minimum value of Equation (7). There are three situations for the values of k_1 and k_2:

1. If the values of k_1 and k_2 were satisfied ($0 \le k_1 \le 1$ and $0 \le k_2 \le 1$), then the distance between MN is the shortest distance between line AB and CD, and the point M and N are the corresponding vertical feet.

2. If the values of k_1 and k_2 were not satisfied ($0 \le k_1 \le 1$ and $0 \le k_2 \le 1$), the distances from the point A and B to the line CD and the distance from the point C and D to the line AB are obtained separately. Thus, four vertical foot points, i.e., A_1, B_1, C_1, and D_1, can be obtained. Then, the minimum of the corresponding distances AA_1, BB_1, CC_1, and DD_1 is the shortest distance between line AB and CD.

3. If the values of k_1 and k_2 are not satisfied ($0 \le k_1 \le 1$ and $0 \le k_2 \le 1$), there could be no vertical foot point on line AB and line CD. In this case, only the distances of line AC, AD, BC, and BD are required, and the minimum value is the distance between line AB and CD. Therefore, the distance between the two robot manipulators can be solved by the distance between the two lines described above. By comparing with the set collision threshold of collision, whether a collision has occurred can then be determined.

3.2. Improved Artificial Potential Field Algorithm

In 1986, Khatib [24] firstly proposed the artificial potential field (APF) algorithm. The core content of the APF is that the robot is affected by two forces when moving towards the goal point at a certain speed. One is the attraction of the target position to the robot, the other is the repulsive force formed by obstacles on the robot. The resultant force of these two forces is applied to the robot, and thus the robot can move towards the position of the goal point and finally reach the position. Among the existing path planning methods, the artificial potential field (APF) algorithm is widely applied in obstacle avoidance for its simplicity and practicability and for its good real-time performance over other planning methods.

The motion and force of robot in the artificial potential field can be represented and analyzed in Figure 3.
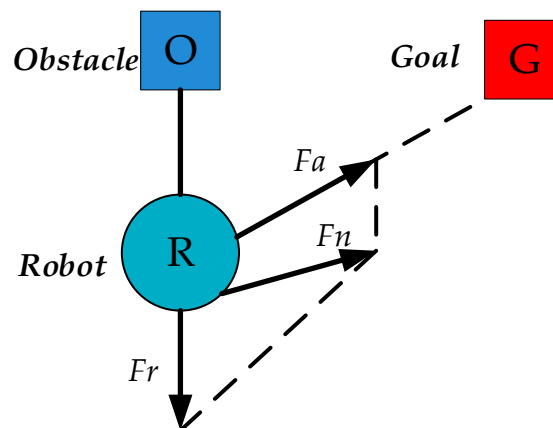
Figure 3. The force of the robot in the APF.

Fr is used to represent the repulsive force; *Fa* is used to represent the attractive force; and *Fn* is the resultant force under the interaction of the two forces, which makes the robot move towards the target.

The attraction field produced by F_a can be obtained by Equation (9):

$$U_{att}(q) = \frac{1}{2}\zeta d^2\left(q, q_{goal}\right) \tag{9}$$

where $U_{att}(q)$ is attraction field, ζ is the correction coefficient of the attractive potential field, and $d\left(q,q_{goal}\right)$ is the distance between the robot and the target position. By taking the derivative of Equation (9), the expression of attraction F_a is acquired:

$$F_a = \zeta\left(q_{goal} - q\right) \tag{10}$$

The repulsion field produced by F_r can be obtained by Equation (11):

$$U_{rep}(q) = \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{d(q,q_{obs})} - \dfrac{1}{d_0}\right)^2, d(q,q_{obs}) \leq d_0 \\ 0 \qquad\qquad\qquad d(q,q_{obs}) > d_0 \end{cases} \tag{11}$$

Where η is the correction coefficient of the repulsion potential field; $d(q,q_{obs})$ is the real-time distance between the robot and the obstacle; d_0 represents the influence radius of obstacle. When $d(q,q_{obs}) > d_0$, the repulsion field will not affect the movement of the robot. Taking the derivative of Equation (11), the expression of repulsive force F_r can thus be obtained:

$$F_r = -\nabla U_{rep}(q) = \begin{cases} \eta\left(\dfrac{1}{d(q,q_{obs})} - \dfrac{1}{d_0}\right)\dfrac{1}{d^2(q,q_{obs})}\nabla d(q,q_{obs}), d(q,q_{obs}) \leq d_0 \\ 0 \qquad\qquad\qquad\qquad\qquad d(q,q_{obs}) > d_0 \end{cases} \tag{12}$$

Thus, the potential energy function and the resultant force received by the robot is:

$$U = U_{att}(q) + U_{rep}(q) \tag{13}$$

$$F_n = F_a + F_r \tag{14}$$

The T-APF still has limitations. For example, when the combined force of the attractive and repulsive forces is zero, the robot eventually does not reach the goal point, that is, the robot falls into the local minimum position. If there is an obstacle at the goal point, the repulsive force of the robot arm in the potential field is very large. The attraction force is very small, which makes it difficult for the robot arm to reach the goal point. In order to overcome that issue, the relative position of the robot and the goal point can be considered and analyzed comprehensively. By taking into account the repulsive force generated by the obstacle boundary, the improved repulsive force field function can be obtained by Equation (15):

$$U_{rep}(q) = \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{d(q,q_{obs})} - \dfrac{1}{d_0}\right)^2 d^n\left(q,q_{goal}\right), d(q,q_{obs}) \leq d_0 \\ 0 \qquad\qquad\qquad\qquad\qquad d(q,q_{obs}) > d_0 \end{cases} \tag{15}$$

Taking the derivative of Equation (13), the expression of repulsive force *Fr* can be obtained:

$$F_r = -\nabla U_{rep}(q) = \begin{cases} F_{r1} + F_{r2}, d(q,q_{obs}) \leq d_0 \\ 0 \qquad\quad d(q,q_{obs}) > d_0 \end{cases} \tag{16}$$

where:

$$\begin{cases} F_{r1} = \eta \left(\dfrac{1}{d(q,q_{obs})} - \dfrac{1}{d_0} \right) \dfrac{d^n\left(q,q_{goal}\right)}{d^2\left(q,q_{obs}\right)} \dfrac{\partial d(q,q_{obs})}{\partial q} \\[4mm] F_{r2} = \dfrac{n}{2} \eta \left(\dfrac{1}{d(q,q_{obs})} - \dfrac{1}{d_0} \right)^2 d^{n-1}\left(q,q_{goal}\right) \dfrac{\partial d\left(q,q_{goal}\right)}{\partial q} \end{cases} \tag{17}$$

The direction of F_{r1} is determined by the direction in which the obstacle points to the robot, and the direction of F_{r2} is determined by the direction in which the robot points to the goal point. When the surrounding environment of robotic manipulator is more complicated, the APF based on the improved repulsion function may still have a local minimum, and the manipulator can stop falling into it by adding random disturbances.

3.3. Definition of Attraction Speed and Repulsion Speed

The structure of a dual-arm robot is complex, and the manipulator cannot be used as a mass point to calculate the force and movement of the robot in the artificial potential field. When the goal point is a dynamic obstacle, the traditional artificial potential field algorithm cannot be used. Therefore, speed field [25] with improved artificial potential field algorithm is proposed, In this method, artificial potential field can be established in the Cartesian space at the joints of the manipulator, and the potential field force received by each joint is then converted into a differential velocity for calculation, as shown in Figure 4.
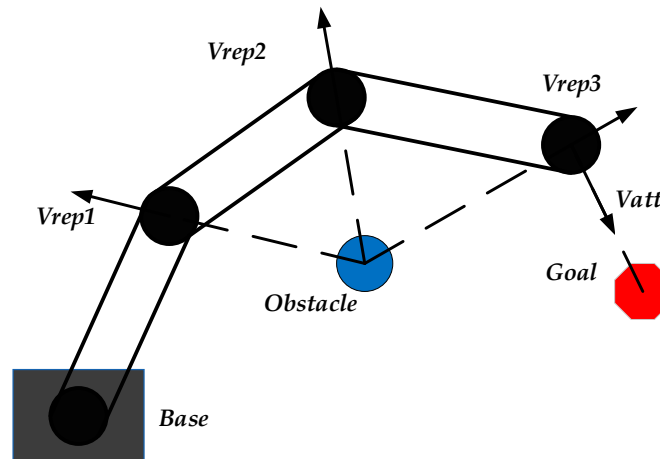
Figure 4. The speed of the robot in the artificial potential field.

In Figure 4, v_{att} is defined as the attraction speed of the goal position to the end of the robot, and v_{rep} is defined as the repulsion velocity of the obstacles to the joint positions of the robot. The advantage of converting the potential field force received by the joints into differential speeds for calculation is that the force received in the artificial potential field is transformed into a speed problem. Thus, the connection between the joint angle and the target position can be established by combining it with the knowledge of Jacobi. These can simplify the process and reduce the variables, which is required for calculation.

According to the potential field force that the robot receives in the artificial potential field, the attraction speed (Equation 18) of the robot can be obtained:

$$v_{\text{att}}=\begin{cases} \dfrac{p_{goal}\text{-}p_{end}}{\left\| p_{goal}\text{-}p_{end} \right\|}, & \left\| p_{goal}\text{-}p_{end} \right\|<p_0 \\[4mm] p_0\dfrac{p_{goal}\text{-}p_{end}}{\left\| p_{goal}\text{-}p_{end} \right\|}, & \left\| p_{goal}\text{-}p_{end} \right\|\geq p_0 \end{cases} \tag{18}$$

where p_{end} and p_{goal} are the positions of the end effector of the manipulator and the target point in Cartesian space, respectively; and p_0 is the distance threshold between p_{end} and p_{goal}.

Then, the repulsion speed is defined according to the improved repulsion formula as follows.

$$v_{\text{rep}}=\begin{cases} (v_{\text{rep1}}+v_{\text{rep2}})\dfrac{p_{obs}\text{-}p_j}{\left\| p_{obs}\text{-}p_j \right\|}, & \left\| p_{obs}\text{-}p_j \right\|\leq p_1 \\[4mm] 0 & \left\| p_{obs}\text{-}p_j \right\|>p_1 \end{cases} \tag{19}$$

where p_{obs} is the position of the obstacle in Cartesian space, p_j is the point on the link closest to the obstacle, η is the speed coefficient of the repulsion force, and p_1 is the influence distance of the obstacle repulsion. Accordingly, v_{rep1} and v_{rep2} can be obtained by Equations (11), (12), (15), and (17):

$$\begin{cases} v_{\text{rep1}}=\eta\left(\dfrac{1}{\left\| p_{obs}\text{-}p_j \right\|}-\dfrac{1}{p_1}\right)\dfrac{\left\| p_{goal}\text{-}p_{end} \right\|^2}{\left\| p_{obs}\text{-}p_j \right\|^2} \\[6mm] v_{\text{rep2}}=\dfrac{n}{2}\eta\left(\dfrac{1}{\left\| p_{obs}\text{-}p_j \right\|}-\dfrac{1}{p_1}\right)^2\left\| p_{goal}\text{-}p_{end} \right\|^{n\text{-}1} \end{cases} \tag{20}$$

Next, the defined attraction speed and repulsion speed to the joint space are mapped, and the knowledge of the Jacobian matrix and its inverse is used to obtain the following formula:

$$\mathrm{d}q_{att}=J^{\text{-}1}(q_{\text{end}})v_{att}dt \tag{21}$$

$$\mathrm{d}q_{rep}=J^{\text{-}1}(q_m)v_{rep}dt \tag{22}$$

where $J^{-1}(q_{end})$ is the inverse of the Jacobian matrix of the end effector of the robot, $J^{-1}(q_m)$ is the inverse of the Jacobian matrix of the point m that satisfies the shortest distance between the two links of the dual-arm robot, and dt is the unit time. The multiplication of v_{att} and dt is the angular displacement of the robot arm in dt time, and so dq_{att} and dq_{rep} are the two differential velocities of the joint. Equation (23) can then be obtained by adding Equations (21) and (22):

$$\mathrm{d}q=\mathrm{d}q_{att}+\mathrm{d}q_{rep} \tag{23}$$

Since the motion speed of the robot arm can be further changed by adjusting the control coefficient, when the two robotic manipulators are in motion, the self-avoidance problem between the two manipulators can plan the path with the other robotic manipulator as a dynamic obstacle.

Figure 5 depicts the complete flowchart of the proposed method. This flowchart shows the initialization information of the proposed algorithm; comparing the calculated distance with the safety distance determines whether the algorithm continues to execute, and the algorithm stops when the iteration times is equal to the set times. The relevant parameter calculation processes are explained in Algorithms 1–2.
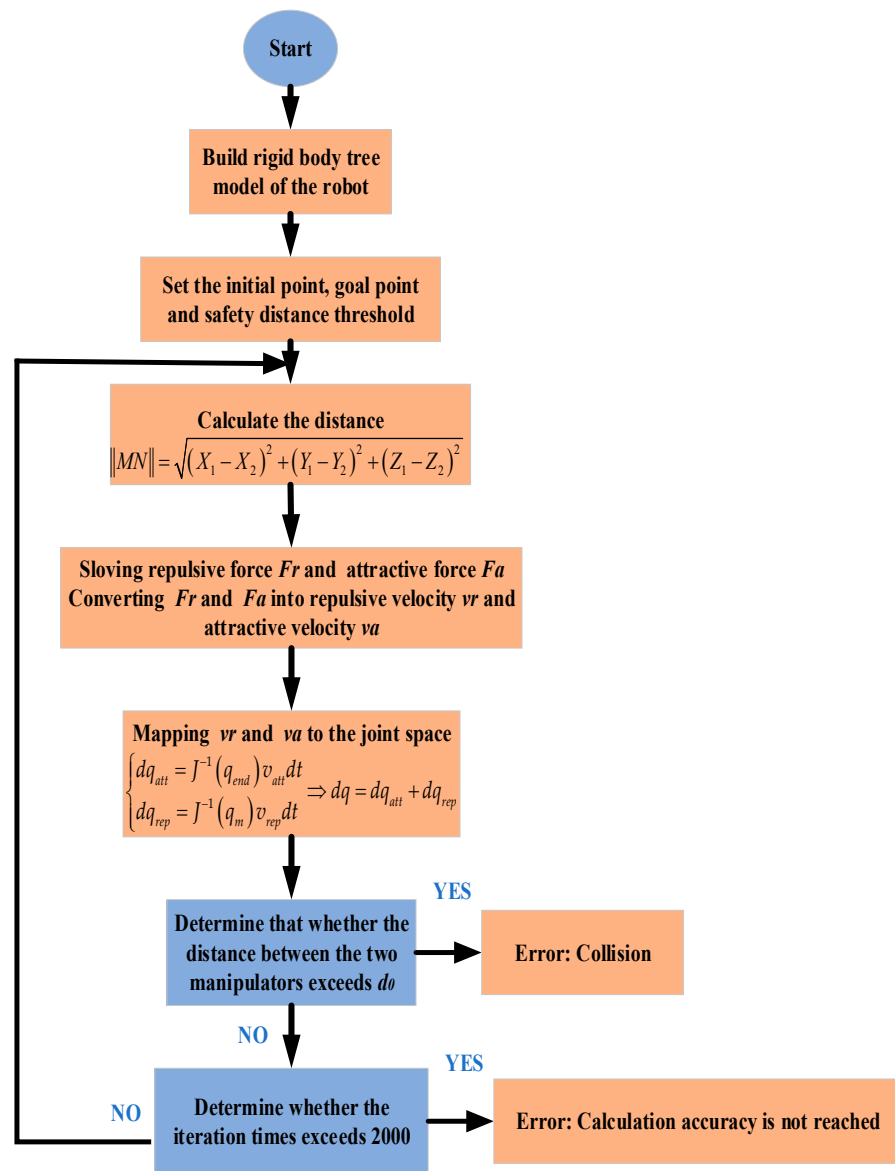
Figure 5. Flowchart of speed field with improved artificial potential field algorithm.

First, Algorithm 1 generates a distance calculation formula and loop termination condition. This is a process of *i* for-loops nested, where *i* is the iteration times; at the end of each iteration, a new *dq* is updated to calculate the distance between the manipulators and the goal point. Lastly, whether the distance norm is less than 0.0001 is used as the condition for the termination of the entire loop.

Algorithm 1 Calculate distance between links.

1: Build robot model in rigid body tree. %Specifications of the Rigid body tree are denoted by D-H parameters.

2:

3: for *i*=1:2000 % i is iteration times

　...

4: L7 = getTransform (robot, q_l, 'body7'); R7 = getTransform (robot1, q_r, 'body7') %Get the end pose of left and right manipulator.

5: a=norm (L_7-lf), b= norm (R_7-rf), % lf and rf are goal points, a and b are distance calculation formulas.

6: if norm (dq) < 0.002&&a %where initial dq= [0 0 0 0 0 0].

dq1= [0.001*ones (3,1); zeros (3,1)]. %Add a random disturbance
7:   end
8: dq=potential field force (d, d0, lf rf, robot, dq1) %Update *dq*
9: q₁=double(ql+dq)
10: repeat step 6 to 9 to obtain right *qr*
11: if norm(L7-lf) <0.00001
12:     break;
13:   end
14: end

Second, Algorithm 2 is the process to obtain the joint force. For this procedure, the core of the process is the value of *dj*, which is a vector, and its direction has a perceptible impact on *Fr*. Furthermore, the initial value of η also affects the calculation of *Fr*. Finally, the new *dq* is obtained through the inverse solution of the robot, which is brought into Algorithm 1 to loop.

Algorithm 2 Calculation of joint potential field force.

1: dq=potential field force (d, d₀, lf rf, robot, dq1)
2: z=0
3:
4: if norm (lf-l(end,:)) >0.05
5:     lf₀=0.05;
6:     lf₀=lf₀/ norm (lf-l(end,:)) * (lf-l(end,:));% Calculate attractive force, *l(end,:)* is the pose of left manipulator .
7:   else
8:    lf₀=lf-l(end,:);
9: end
10: Fr1=1/600*(1/(norm(dj)-d)-1/d₀) * (lf₀) ^3/(norm(dj)-d) ^2
11: Fr2=1/200*(1/(norm(dj)-d)-1/d₀) * (lf₀) ^2
12: Fr=Fr1+Fr2 %*Fr* is repulsive force and *dj* is the distance between goal point and motion particle.
13: dq=ik (lf₀, q1, robot, dq1) %get a new *dq*

## 4. Simulation and Experiment

4.1. Simulation in a Static Environment Based on MATLAB

The experiment is made by the case of two manipulators of the dual-arm robot colliding at a goal point, where the space coordinates of the left and right manipulators are l_L=[0.618 -0.0427 -0.354], l_R=[0.618 0.0427 -0.354], and the initial angles are both set to q_L=[0 0 0 0 0 0], q_R=[0 0 0 0 0 0].

The collision of the robotic manipulators at the goal point and the movement trajectory of the two robotic manipulators during the collision, which was planned based on the fifth degree polynomial, are shown in Figure 6. It can be also seen that the seventh link of the two robot arms collided at the goal point. Figure 7 shows the variations of each joint angle of the left manipulator during the process of the collision. From it we can see that the variation of each joint angle changes smoothly.
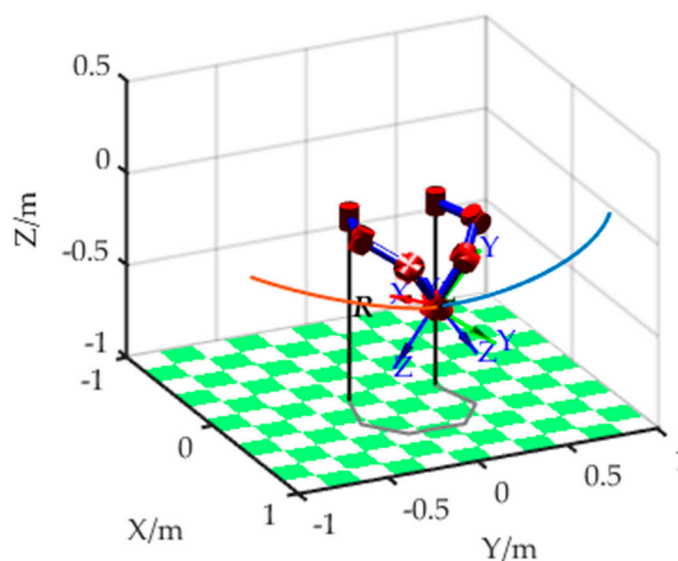
**Figure 6.** Manipulators of the dual-arm robot collide at the goal point.
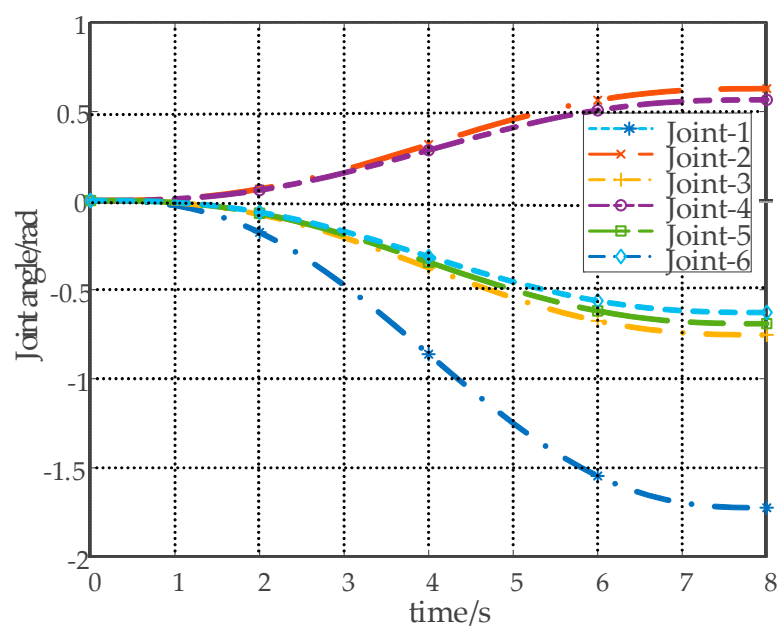
Figure 7. Variation of each joint angle of the left manipulator of the robot.

After the collision experiment was carried out, the goal position was not changed. In order to make the collision avoidance experiment more obvious, the minimum safety distance (*d*) was set to be *0.4* m. The rigid body tree model was then used to carry out the collision experiment. In order to prove the validity and correctness of the proposed algorithm, the experimental results were compared with T-APF, and the results are shown in Figures 8–11.
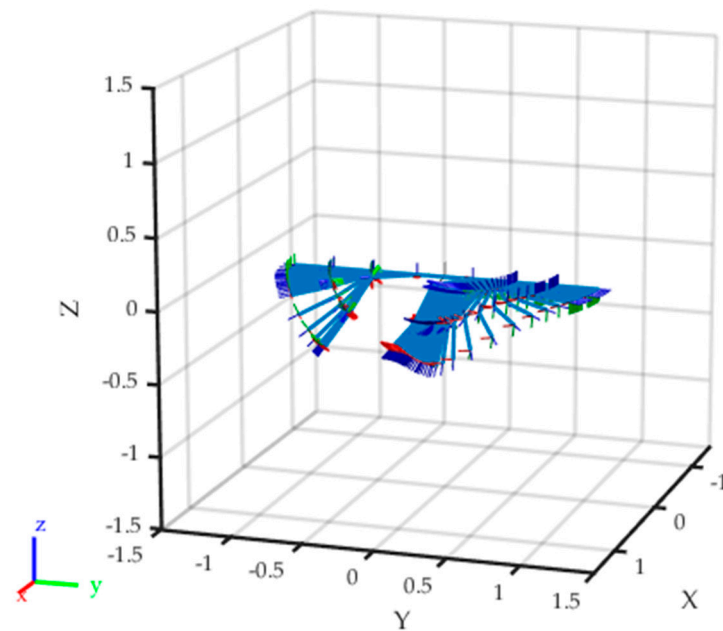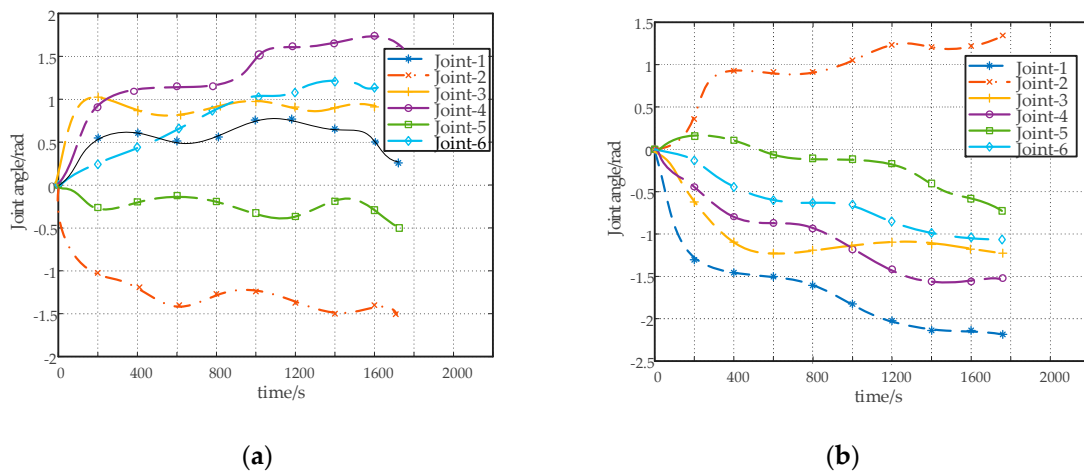
**Figure 8.** Collision avoidance experiment of the rigid body tree model of the robot.
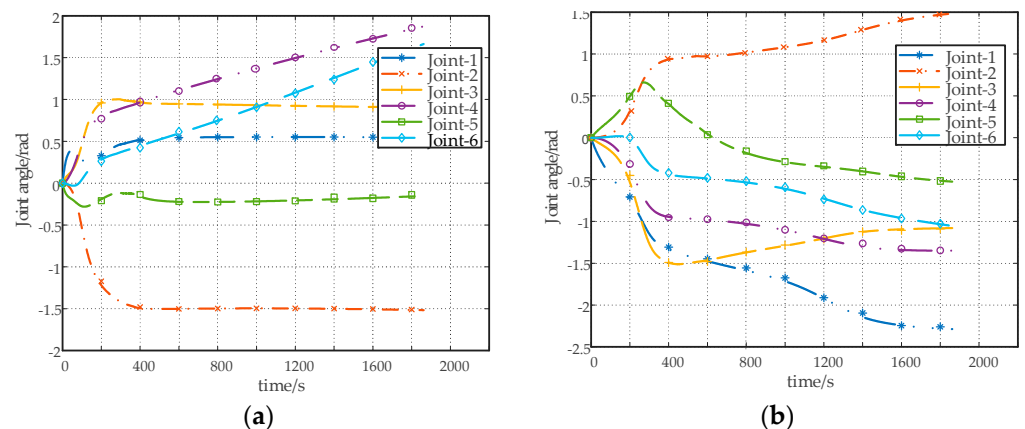
Figure 9. Results of obstacle avoidance with VP-APF. (**a**) Variation of each joint angle of the left arm of the robot during collision avoidance; (**b**) variation of each joint angle of the right arm of the robot during collision avoidance.

Figures 8 shows that the robot arm did not collide during the movement, and Figure 9a,b shows the variations of each joint angle of the left manipulator and the right manipulator during the process of the obstacle avoidance. The variation of the joint angle in Figure 7 was obtained based on the fifth-order polynomial trajectory planning without considering collision, while the variation of the joint angle during the process of collision avoidance in Figure 9a,b was affected by the number of iterations. When the 200th iteration was performed, there were slight fluctuations, and the overall variation of each joint angle was smooth and without obvious shock. Through the above experimental analysis, it is shown that the proposed method can help to avoid a collision of the manipulator of the dual-arm robot.

Figure 10. Results of obstacle avoidance with T-APF. (**a**) Variation of each joint angle of the left arm of the robot during collision avoidance; (**b**) variation of each joint angle of the right arm of the robot during collision avoidance.

Figure 10a,b shows the variations of each joint angle of the left manipulator and right manipulator during the process of the obstacle avoidance with T-APF. Compared with Figure 9, when the 200th, 800th, and 1600th iterations were performed, it was observed that the corresponding angular velocity and the angular acceleration change significantly in a short period of time, which brings in a shake of the manipulator when moving. The joint angles with obvious fluctuations are shown in Figures 10.
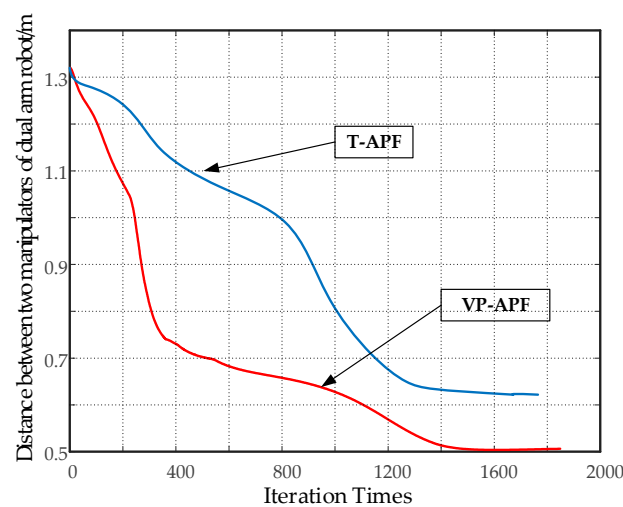
Figure 11. Variation of distance between two manipulators with VP-APF and T-APF.

From the related data in Figure 11 and Table 2 of the simulation experiment, the conclusion can be drawn that the convergence speed and simulation efficiency with VP-APF are improved compared to T-APF. Furthermore, the variations of each joint angle of the manipulators are smoother which they meet the requirement of being smooth and stable in motion without shock for the manipulator in the process of obstacle avoidance.

Table 2. Simulation comparison result of different algorithms.

Algorithms	Simulation time/s	Iterations times	Distance
T-APF	124.375	1785	0.62
VP-APF	103.421	1856	0.51

4.2. Simulation in a Static Environment Based on MATLAB & Adams

In order to further verify the effectiveness of the proposed method, the coordinated simulation of MATLAB and Adams were used for verification. Firstly, the angle data of each joint obtained by MATLAB was imported into Adams, and the Spline curve function was generated. The Spline function was then set as the change function of the joint driving force to drive the robot model in Adams. The specific parameters of the model in Adams are shown in Table 3, and the specific process is shown in Figure 12.
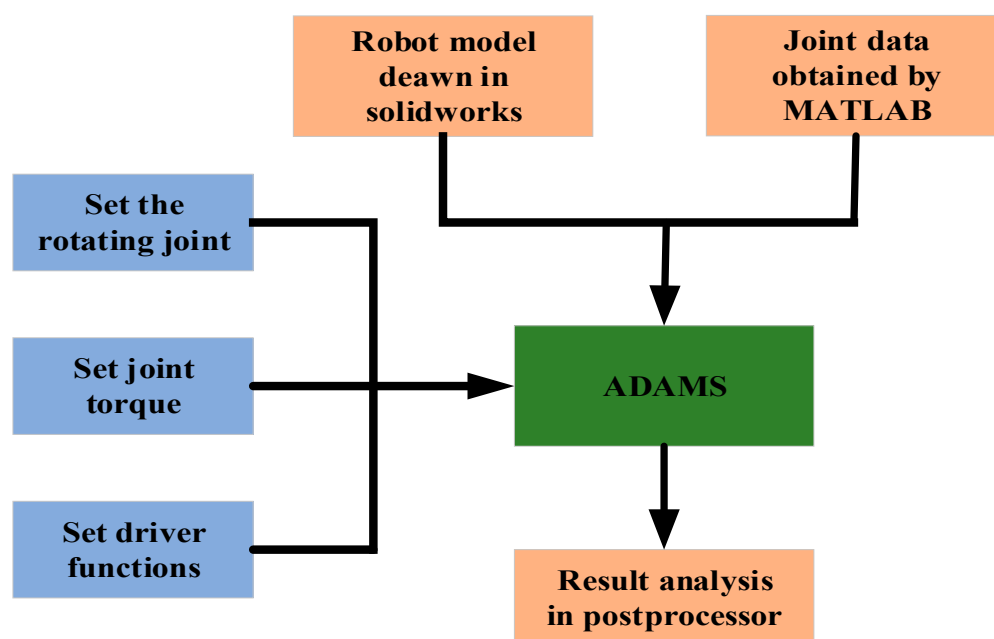

Figure 12. Coordinated simulation process of MATLAB and Adams.

After the above process was set up, physical parameters such as the mass and the moment of inertia of each joint can be obtained, as recorded in Table 3. After the simulation button is clicked, the dual-arm robot moves under the action of the driving function, and the collision avoidance results of the robot at the end of the simulation are shown in Figure 13.

Table 3. Dynamic parameters of the dual-arm robot model in Adams.

| Link i | Mass | Centroid Coordinates | Moment of Inertia | | |
			I_{xx}	I_{yy}	I_{zz}
1	64.2	9.72×10^{-2}; -10.5; 38.9	8.52	6.00	3.04
2	4.18	0.496; 595; 233	0.0463	0.0406	0.0303
3	3.29	0.496; 768; 233	0.0392	0.0333	0.0247
4	7.67	0.496; 988; 233	0.0654	0.0634	0.0352
5	3.29	0.496; 1208; 233	0.0392	0.0333	0.0247
6	7.67	0.496; 1408; 233	0.654	0.0634	0.0352

(**a**)

(**b**)

(**c**)

(**d**)

(**e**)
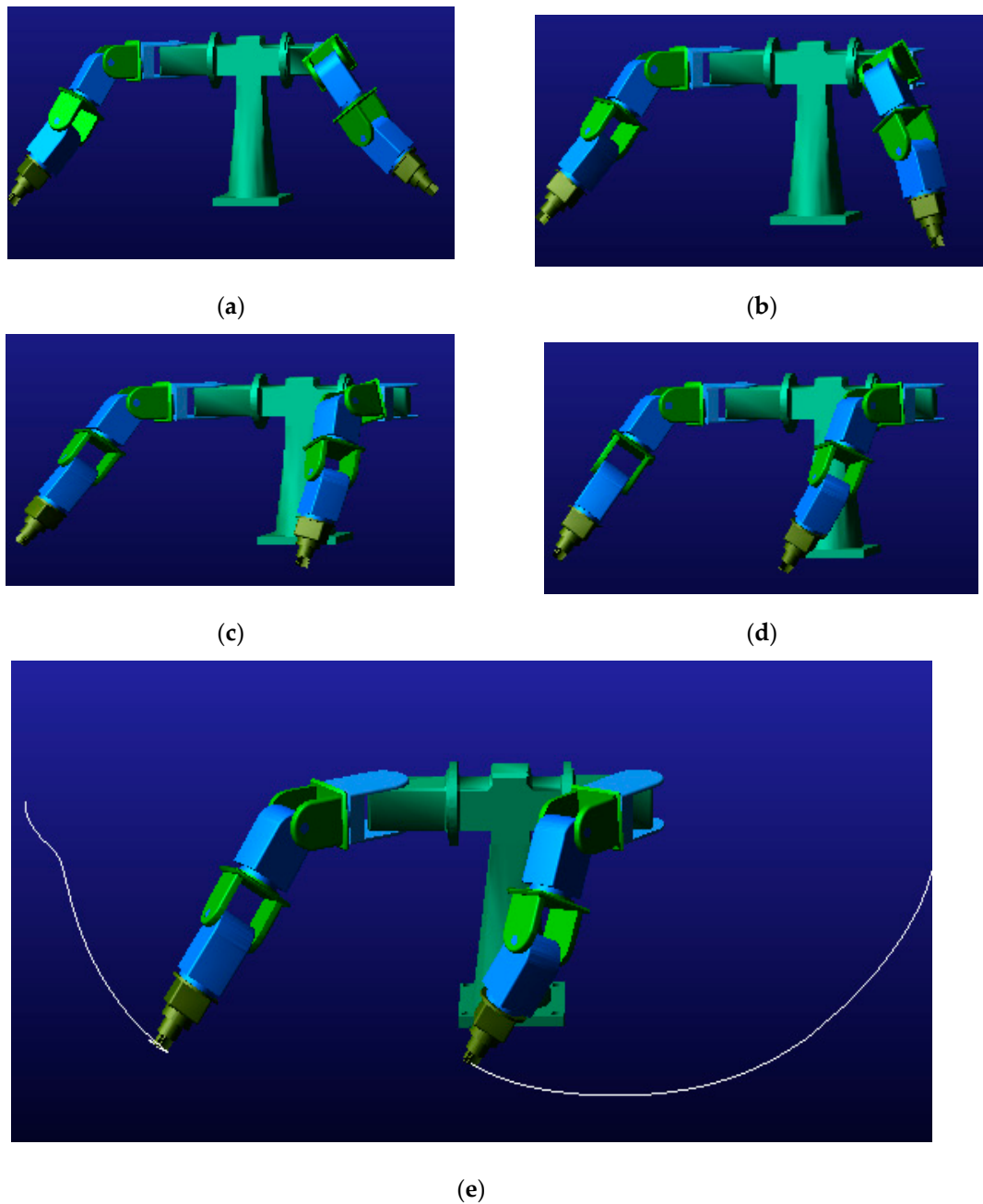
Figure 13. The movement process of dual-arm robot in Adams. (**a-d**) Images representing different states of the dual-arm robot at each moment. (**e**) Image representing the trajectory of each manipulator during movement.

Figure 13 shows the obstacle avoidance movement of the dual-arm robot in Adams; Figures 13a–d are the different poses of the robot at a certain time, and Figure 13e is the pose at the end of the dual-arm robot obstacle avoidance movement and the trajectory during the process of obstacle avoidance. Compared with the collision avoidance effect of the rigid body tree model of the robot in MATLAB, it can be seen that the dual-arm robot in Adams can achieve collision avoidance, which verifies the correctness of the modeling. It can also be found that the final distance between the arms of robot is a little larger than the distance of robot model in MATLAB because the joint diameter may affect the experimental results when the model is established. In order to obtain the specific situation of the robotic manipulator during obstacle avoidance movement in Adams, the changes of the joint angle, the joint angular velocity, and torque during the movement are analyzed.

By clicking on the postprocessor module, the corresponding parameter change curve can be obtained.

Figure 14a,b show the variation of joint angle of left and right arm, from which we can see that the two manipulators changed their path at the 2 s and 4 s points, and the process of obstacle avoidance is smooth. Since the joint angle of the two manipulators are always changing, and is impacted by iterations and is used to calculate the distance between the two robotic manipulators, the corresponding joint angular velocity changes rapidly and frequently to prevent the system from exceeding the set threshold of distance. The variations of each joint angular velocity of the replanned path during the process of dynamic obstacle avoidance are shown in Figure 14c,d. Figure 14e,f shows the variation of each joint torque, from which we can see that the left manipulator, as the active manipulator, has a smooth variation of torque without obvious shaking. On the contrary, the right arm is the passive manipulator and needs to change position according to the movement of the left manipulator, resulting in frequent changes in the torque in a short period of time.
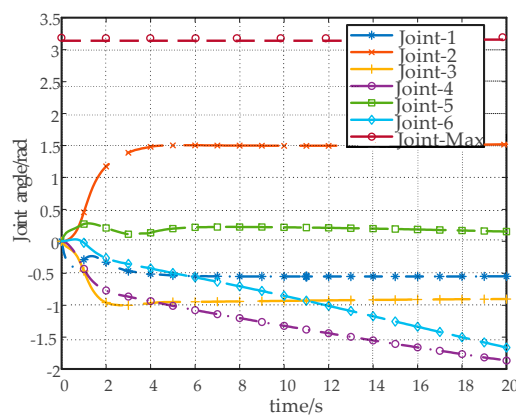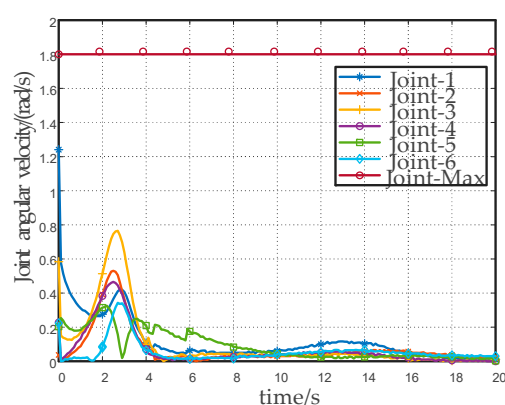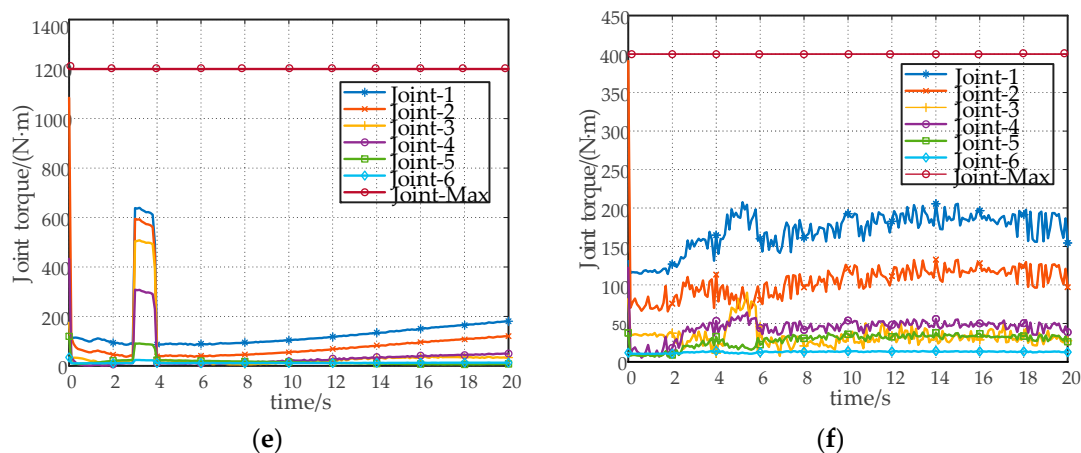

(**a**)
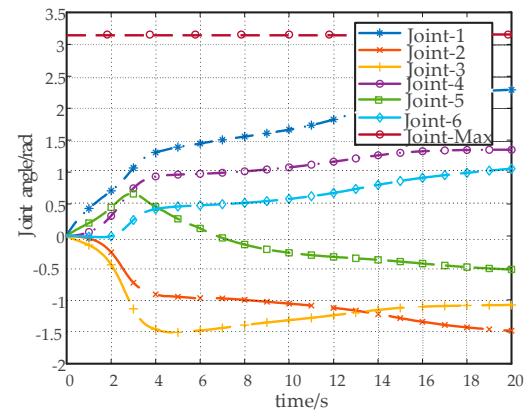
(**b**)
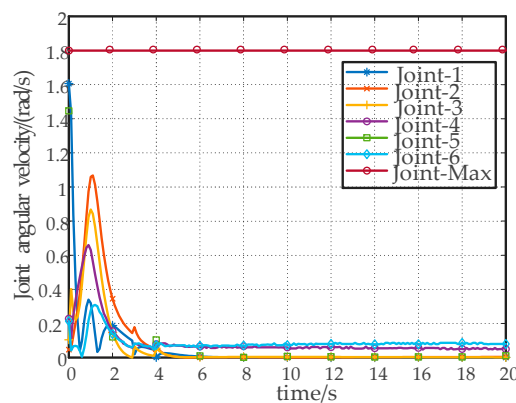
(**c**)

(**d**)

Figure 14. The parameters changes of the robot during obstacle avoidance. Graphs in (**a,c,e**) represent the parameter changes (joint angle, joint angular velocity, and torque) of the right robotic manipulator during obstacle avoidance. Graphs in (**b,d,f**) represent the parameter changes (joint angle, joint angular velocity, and torque) of the left robotic manipulator during obstacle avoidance.

Through the simulation experiment of MATLAB and Adams, it can be known that the proposed method can solve the obstacle avoidance problem of the dual-arm robot, and the overall effect is better with a smooth process. However, it is noticeable that the proposed method still has shortcomings with a long running time, as can be seen from Table 2. This is because the distance and the inverse solution of the two manipulators need to be calculated continuously when the two manipulators are moving to the goal point to the loop. Therefore, we intend to conduct further studies, and more highlights of the algorithm should be improved.

5. Conclusions and Future Work

Aiming at solving the question about the possible collisions between the manipulators of the dual-arm robot during a collaborative task operation, an artificial potential field collision avoidance algorithm based on the potential field velocity was proposed. Firstly, the traditional artificial potential field method was used to establish the attraction and repulsion functions of the manipulator, and then the concepts of attraction and repulsion speed were introduced. The attraction and repulsion functions were converted into the attraction and repulsion speed functions, which were mapped to the joint space. By combining Jacobi matrix and its inverse, the differential velocity function of the joint motion was established. Finally, after comparing it with the set collision distance threshold between two manipulators of the dual-arm robot, the collision avoidance can be solved. The simulation experiment results verified the correctness and effectiveness of the proposed method in the self-collision avoidance problem of the dual-arm robot. Through experiments in MATLAB and Adams, it can be found that the proposed improved artificial potential field method based on the velocity potential field can solve the collision problem of the dual-arm robot. There are also limitations in this paper, as the proposed method has more iterations, and the simulation time is longer in the experiment. Therefore, parameter optimization is expected to be researched in the future.

References

1. He, Z.; Yuan, F.; Chen, D.; Wang, M. Dynamic Obstacle Avoidance Planning for Manipulators of Home. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Dali, China, 6–8 December 2019; pp. 2737–2742.
2. Glorieux, E.; Riazi, S.; Lennartson, B.E.V. Productivity/energy optimization of trajectories and coordination for cyclic multi-robot systems. *Robot. Comput. Integr. Manuf.* **2018**, *49*, 152–161.
3. Indri, M.; Trapani, S.; Lazzero, I. Development of a virtual collision sensor for industrial robots. *Sensors* **2017**, *17*, 1148.
4. Liang, J.; Gong, Z.; Wang, W.; Hou, Z.; Han, C.S. Dual quaternion based kinematic control for Yumi dual arm robot. In Proceedings of the IEEE International Conference on Ubiquitous Robots and Ambient Intelligence, Jeju, Korea, 28 June–1 July 2017; pp. 114–118.
5. Phukan, S.; Mahanta, C. A position synchronization controller for co-ordinated links (cool) dual robot arm based on integral sliding mode: Design and experimental validation. *Int. J. Autom. Comput.* **2021**, *18*, 110–123.
6. Liu, Q.; Chen, C.Y.; Wang, C.; Wen, W. Common workspace analysis for a dual-arm robot based on reachability. In Proceedings of the IEEE International Conference on CIS & RAM, Ningbo, China, 19–21 November 2017; pp. 797–802.
7. Kivelä, T.; Mattila, J.; Puura, J. Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance. In *Advances in Service and Industrial Robotics*; Springer: Cham, Switzerland, 2017; pp. 208–216.
8. Kim, J.O.; Khosla, P. Real-time obstacle avoidance using harmonic potential functions. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 9–11 April 1991; pp. 790–796.
9. Abhishek, T.S.; Schilberg, D.; Doss, A.S.A. Obstacle Avoidance Algorithms: A Review. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Chennai, India, 2–3 October 2020; Volume 1012, 012052.
10. Palmieri, G.; Scoccia, C.; Palpacelli, M.C.; Callegari, M. Motion planning and control of redundant manipulators for dynamical obstacle avoidance. *Machines* **2021**, *9*, 121.
11. Zhang, C.; Zhou, L.; Li, Y.; Fan, Y. A Dynamic Path Planning Method for Social Robots in the Home Environment. *Electronics* **2020**, *9*, 1173.
12. Han, B.; Luo, X.; Luo, Q.; Zhao, Y.; Lin, B. Research on Obstacle Avoidance Motion Planning Technology of 6-DOF Manipulator. In Proceedings of the International Conference on Geometry and Graphics, São Paulo, Brazil, 9–13 August 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 604–614.
13. Wang, J.; Liu, S.; Zhang, B.; Yu, C. Manipulation Planning with Soft Constraints by Randomized Exploration of the Composite Configuration Space. *Int. J. Control Autom. Syst.* **2021**, *19*, 1340–1351.
14. Lin, Z.; Fu, J.; Shen, H.; Gan, W. Non-singular tool path planning by translating tool orientations in c-space. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 1835–1848.
15. Gabriely, Y.; Rimon, E. C-space characterization of contact preserving paths with application to tactile-sensor based mobile robot navigation. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1792–1797.
16. Willms, A.R.; Yang, S.X. Real-time robot path planning via a distance-propagating dynamic system with obstacle clearance. *IEEE Trans. Syst. Man Cybern. Part B* **2008**, *38*, 884–893.
17. Jung, J.H.; Kim, D.H. Local Path Planning of a Mobile Robot Using a Novel Grid-Based Potential Method. *Int. J. Fuzzy Log. Intell. Syst.* **2020**, *20*, 26–34.
18. Scalera, L.; Giusti, A.; Cosmo, V.D.; Riedl, M.; Matt, D.T. Application of Dynamically Scaled Safety Zones Based on the ISO/TS 15066: 2016 for Collaborative Robotics. *Int. J. Mech. Control* **2020**, *21*, 41–50.
19. Park, S.O.; Min, C.L.; Kim, J. Trajectory Planning with Collision Avoidance for Redundant Robots Using Jacobian and Artificial Potential Field-based Real-time Inverse Kinematics. *Int. J. Control Autom. Syst.* **2020**, *18*, 2095–2107.
20. Lei, T.; Dian, S.; Gu, G.; Zhou, K.; Feng, X. A novel potential field method for obstacle avoidance and path planning of mobile robot. In Proceedings of the 2010 IEEE International Conference on Computer Science & Information Technology, Chengdu, China, 9–11 July 2010; pp. 633–637.
21. Xu, J.; Park, K.S. A real-time path planning algorithm for cable-driven parallel robots in dynamic environment based on artificial potential guided RRT. *Microsyst. Technol.* **2020**, *26*, 3533–3546.
22. Lee, D.; Jeong, J.; Kim, Y.H.; Park, J.B. An improved artificial potential field method with a new point of attractive force for a mobile robot. In Proceedings of the 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, China, 29–31 December 2017; pp. 63–67.
23. Weerakoon, T.; Ishii, K.; Nassiraei, A. An artificial potential field based mobile robot navigation method to prevent from deadlock. *J. Artif. Intell. Soft Comput. Res.* **2015**, *5*, 189–203.
24. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–99.
25. Flacco, F.; Kroger, T.; Luca, A.D.; Khatib, O. A depth space approach to human-robot collision avoidance. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 338–345.