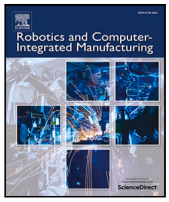
Robotics and Computer–Integrated Manufacturing 71 (2021) 102114

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcm



Full length article

An obstacle avoidance algorithm for robot manipulators based on decision-making force

Wei Zhang, Hongtai Cheng *, Lina Hao, Xingchen Li, Mingfang Liu, Xifeng Gao

Department of Mechanical Engineering and Automation, Northeastern University, Shenyang, Liaoning Province, 110819, China

ARTICLE INFO

Keywords:
Robot manipulator
Obstacle avoidance
Dynamic repulsion field
Decision-making force

ABSTRACT

Obstacle avoidance is a significant skill not only for mobile robots but also for robot manipulators working in unstructured environments. Various algorithms have been proposed to solve off-line planning and on-line adaption problems. However, it is still not able to ensure safety and flexibility in complex scenarios. In this paper, a novel obstacle avoidance algorithm is proposed to improve the robustness and flexibility. The method contains three components: A closed-loop control system is used to filter the preplanned trajectory and ensure the smoothness and stability of the robot motion; the dynamic repulsion field is adopted to fulfill the robot with primitive obstacle avoidance capability; to mimic human's complex obstacle avoidance behavior and instant decision-making mechanism, a parametrized decision-making force is introduced to optimize all the feasible motions. The algorithms were implemented in planar and spatial robot manipulators. The comparative results show the robot can not only track the task trajectory smoothly but also avoid obstacles in different configurations.

1. Introduction

In the industrial environment, the manipulators traditionally work inside the safety fences, and they follow the off-line planned, determined path repetitively and constantly [1]. In recent years, robots have gradually stepped out fences, such as domestic service robots, soft robots, collaborate robots [2,3] . It is necessary to ensure those robots can avoid unexpected obstacles and complete assigned tasks even in unstructured, open and dynamic environments.

Obstacle avoidance algorithms can usually be classified into two categories: off-line planning algorithms and on-line adjustment algorithms. Off-line obstacle avoidance algorithms solve obstacle avoidance problem by path planning [4,5], such as rapidly-exploring random tree (RRT), probabilistic road map (PRM). When encountering obstacles, the robot is required to redesign a viable global path [6–8], which is usually time-consuming. Although these algorithms have been improved in recent years, they are still unable to evade the unexpected obstacles and provide a responsive path planning operation [9–11]. Therefore, this paper mainly focuses on the on-line obstacle avoidance algorithm.

On-line obstacle avoidance algorithms are usually completed by combining closed-loop control with Artificial Potential Field (APF) methods [12,13]. The closed-loop control system is used to track the task trajectory of the manipulator. When the manipulator meets an obstacle, a potential field is defined around the obstacle [14,15]. Along the direction of the negative gradient of the potential field, a repulsive force is virtually imposed on the robot, which enables the manipulator to achieve real-time dynamic obstacle avoidance behavior. There are also some on-line obstacle avoidance methods that are using deformation, such as the Elastic Band approach which aims at real-time trajectory adaptation in dynamic environments [16–18]. In this method, the initial shape of the elastic band is a free path generated by a classical planner. In the presence of obstacles, this band is deformed by applying repulsive forces. However, if the path being executed becomes infeasible due to obstacles coming into its way, the reshaping algorithm cannot be applied any more [19].

Dynamic Motion Primitive (DMP) is a method that can encode robot motion in a form of a second-order dynamic equation [20]. It can be combined with APF methods to provide obstacle avoidance capability. In [21–23], static and dynamic potential fields are fused in the DMP framework. However, the limitations of APF, the possibility of being trapped in local minima still exists. They also lack the ability to represent non-primitive robot motions and return to original motion after surpassing the obstacle. To solve the local minimum problem, a rotational force field based method is proposed in [24] to enable the robot to escape from the trap point. However, because it is specifically designed for autonomous robot, only planar clockwise and counterclockwise cases are introduced.

* Corresponding author.
E-mail addresses: 1510086@stu.neu.edu.cn (W. Zhang), chenght@me.neu.edu.cn (H. Cheng), haolina@me.neu.edu.cn (L. Hao), 1810092@stu.neu.edu.cn (X. Li), 1710089@stu.neu.edu.cn (M. Liu), 1610072@stu.neu.edu.cn (X. Gao).

https://doi.org/10.1016/j.rcim.2020.102114
Received 12 November 2019; Received in revised form 25 December 2020; Accepted 26 December 2020
Available online 19 March 2021

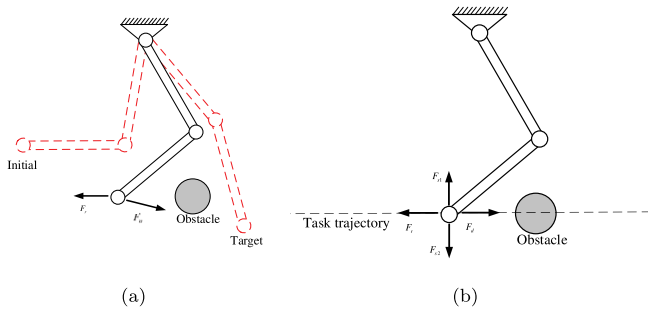


Fig. 1. Task configuration for the two DOF robot manipulator. (a) point to point steering problem with an obstacle; (b) Tracking a specified task trajectory with an obstacle on it.

Although the above algorithms are effective in many configurations, it also has some limitations. In case that the obstacle is on the task trajectory of the robot, the repulsion force and driving force are in a straight line. The repulsion force will decrease when the robot leaves the obstacle under the repulsion force, but the driving force will make the robot approach the obstacle, and the repulsion force will increase again. It will cause the robot to oscillate near the obstacle. The manipulator cannot avoid the obstacle under this configuration in a determined way [25], and the traditional method can only produce an obstacle avoidance path for the same configuration.

In order to solve the above problems, This paper introduces the concept of the decision-making force, which is inspired by Decision field theory (DFT). DFT is a dynamic-cognitive approach of human decision-making based on psychological [26] and is initially introduced as a deterministic-dynamic model of approach-avoidance conflict behavior [27]. Not long after that, the theory is recast into a stochastic dynamic model of decision behavior [28]. When the robot encounters an obstacle, the decision-making force will decide which direction is more suitable for obstacle avoidance.

In this paper, an obstacle avoidance algorithm combining the closed-loop control system, dynamic repulsion field, and decision force is proposed. It realizes the following goals: (1) the robot tracks a continuous motion trajectory instead of a particular target point. (2) the robot can recover the original motion trajectory after avoiding the obstacles. (3) the decision-making force provides a parametric solution instead of a unique avoidance option.

The rest of this paper is organized as follows. In Section 2, the architecture of the on-line obstacle avoidance system is described. In Section 3, the closed-loop control system, the repulsive field, the formulation and optimization mechanism of the decision-making force are discussed in details. In Section 4, a set of experiments are carried out under different configurations by varying the robot's and obstacle's locations. The paper is concluded and future research is briefly introduced in Section 5.

2. Architecture of the obstacle avoidance control system

2.1. Problem formulation

In robot applications, there are two main movements: Point-to-Point steering problem (P2P) and Trajectory Tracking Problem (TTP). Fig. 1 gives two examples. The main differences are that P2P focuses on the final target, while TTP cares the tracking accuracy.

APF method is proposed to plan a path from the initial position to the target and avoid the surrounding obstacles. There is no predefined path/trajectory for the robot. As shown in Fig. 1(a). The only provided information is the target $\mathbf{P}_d$ and obstacle locations and two kinds of APF functions: attractive force field F_a and repulsive force field F_r. In each position, those two forces can be derived according to the potential

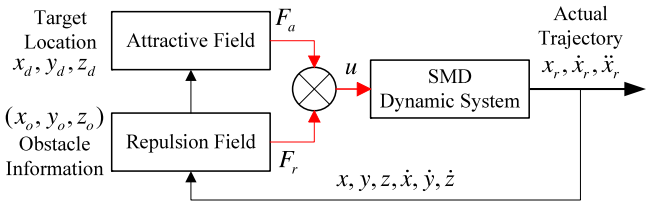


Fig. 2. Architecture of classic APF system. The composition of repulsive force and attraction force drive the robot toward the target position in an open-loop way.

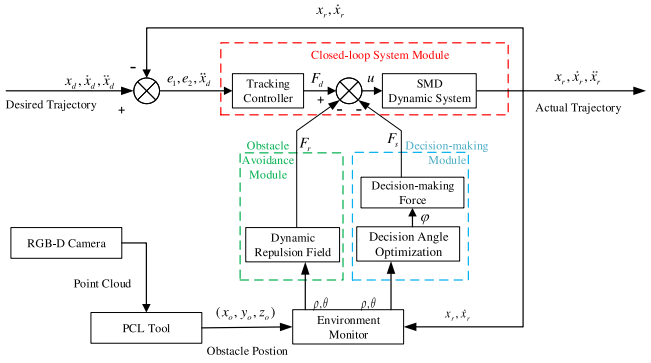


Fig. 3. Obstacle avoidance control system architecture.

field functions. Those two forces are imposed on the robot together and affect the robot motion. The repulsive force drives the robot away from obstacle while the attraction force pulls the robot toward the goal. This is illustrated in Fig. 2.

TTP is widely existed in industrial robot applications. Robot manipulators usually move along predefined trajectories, described in (x_d, $\dot{x}_d$, $\ddot{x}_d$). However, in unstructured environment, they also need to alter the predefined path to avoid the dynamic obstacles. Fig. 1(b) shows a typical obstacle avoidance problem for a robot manipulator.

The target trajectory can be seen as a moving/dynamic target point and existing APF can be embedded and construct a dynamic potential field. A commonly used method is to introduce a virtual repulsive force to change the robot velocity in both value and direction. However, because it is difficult to synchronize the target and actual robot motion, this dynamic attractive potential field may have phase shift and stability issues. Therefore, simple attractive force field is no longer applicable in trajectory tracking control problems.

And also, the resulting motion is sensitive to robot's original velocity and relative position between robot and the obstacle. It cannot guarantee the effectiveness in any configurations and can only produce one kind of obstacle avoidance behavior for the same configuration.

2.2. Overview of the proposed solution

This paper focuses on the obstacle avoidance problem for TTP of a robot manipulator. Fig. 3 shows the diagram of the proposed obstacle avoidance algorithm. It is mainly composed of a closed-loop system module, a repulsive potential field-based obstacle avoidance module, and a decision-making module. The input and output of the system are the desired trajectory (x_d, $\dot{x}_d$, $\ddot{x}_d$) and the actual trajectory (x_r, $\dot{x}_r$, $\ddot{x}_r$) of the end effector of the manipulator, respectively.

2.2.1. The closed-loop control

The attractive force used in traditional APF has open-loop structure. The force drives the robot toward the target without considering the feedback. It works fine for static target in uncomplex environment. However, for dynamic target, such as a trajectory, this open-loop feature cannot guarantee the tracking stability and accuracy.

In order to balance the tradeoff between tracking performance and obstacle avoidance safety, in this paper, the attractive force is replaced by the driving force generated from a closed-loop control system. The closed-loop control system module makes the end of the manipulator track the desired trajectory before avoiding the obstacle. In the process of avoiding the obstacle, it makes the manipulator not far away from the desired trajectory, and can quickly resume the desired trajectory tracking after avoiding the obstacle.

As shown in Fig. 3, the closed-loop control system module can realize the dynamic tracking of a given desired trajectory, which consists of a trajectory tracking controller and a virtual spring mass damper (SMD) dynamic system. The inputs of the tracking controller include displacement error e_1, velocity error e_2, and desired acceleration $\ddot{x}_d$, and the output is the driving force F_d. The input of the SMD dynamic system is the generalized force u, and the output is the actual trajectory $(x_r, \dot{x}_r, \ddot{x}_r)$.

2.2.2. The obstacle avoidance module

The obstacle avoidance module can make the robot generate obstacle avoidance motion when comes across the obstacle. In order to increase flexibility, the function of the collision avoidance module is realized by the dynamic repulsion field. If the safety distance ρ and the direction angle θ are less than the set thresholds, the dynamic repulsive field will generate a repulsive force F_r to keep the robot away from the obstacle. The introduction of θ makes the repulsive not only related to robot position but also amplitude and direction of its velocity, which more comprehensively models the obstacle avoidance behavior.

2.2.3. The decision force

As shown in Fig. 1(b), the two degrees of freedom (DOF) robot manipulator is tracking a specified task trajectory with an obstacle on it. Because the speed vector, repulsive force and driving force are in a line, the end effector of the manipulator will oscillate under such circumstances. With the increase of the driving force, the manipulator may collide with obstacles. In addition to the repulsive force F_r, the perpendicular force F_{s_1} and F_{s_2} can also change the velocity. They provide alternative options for the obstacle avoiding problem. In this case, the decision force F_{s1} or F_{s2} can be exerted in the perpendicular direction of the repulsion force. Since the decision-making force F_{s1} can not only make the end effector of the manipulator away from the obstacle but also increase the closest distance between the link and the obstacle, F_{s1} is better than F_{s2}.

This process is similar to the human's obstacle avoiding decision mechanism. People can choose the avoiding behavior according to multiple issues in the real environment and one's preference. Therefore, as shown in Fig. 3 in this paper, the decision force is introduced to mimic such mechanism. It is perpendicular to the repulsive force and can deflect the robot motion to avoid obstacles. The decision-making module helps the robot choose a better direction to avoid the obstacle.

When no obstacle is detected, the robot's end-effector tracks the desired trajectory, only the closed-loop control system module works, and the obstacle avoidance module and decision-making force module have no output; when the obstacle is detected to affect the normal movement of the robot, The obstacle avoidance module and decision-making force module will output the repulsive force and decision-making force to the closed-loop control system, so that the robot can choose a suitable direction to stay away from the obstacle; when the robotic arm transpasses the obstacle, the robot will resume tracking the desired trajectory due to the effect of the closed-loop control system module. The obstacle's position can be obtained by processing the point cloud data from the RGB-D camera with the Point Cloud Library (PCL) tool.

3. Implementation of obstacle avoidance control system

3.1. Closed-loop control

A single DOF manipulator can be regarded as an SMD system. The dynamic model is:

$$m\ddot{x} + c\dot{x} + kx = u \tag{1}$$

where m, c and k are equivalent mass or inertia, damping and stiffness, respectively. u denotes the input force or torque.

For a given trajectory $x_d(t)$, the corresponding speed and acceleration are $\dot{x}_d(t)$, $\ddot{x}_d(t)$. In order to achieve tracking control of the trajectory, it needs to define two error terms e_1, e_2 as:

$$\begin{cases} e_1 = x - x_d \\ e_2 = \dot{x} - \dot{x}_d \end{cases} \tag{2}$$

Define Lyapunov function as:

$$V(t) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \tag{3}$$

Taking the derivative of Eq. (3) with respect to the time t, one gets

$$\begin{aligned} \dot{V}(t) &= e_1 e_2 + e_2 \dot{e}_2 \\ &= e_2\left(e_1 + \dot{e}_2\right) \\ &= e_2\left(e_1 + \ddot{x} - \ddot{x}_d\right) \\ &= e_2(e_1 - \tfrac{1}{m}[c\dot{x} + kx - u] - \ddot{x}_d) \end{aligned} \tag{4}$$

Obviously, $V = 0$ means that the system state converges to the target trajectory. In order to make the system converge, it is necessary to make $\dot{V} < 0$. Therefore, the control law can be constructed as follows:

$$u = -s \times \mathrm{fun}\left(e_2\right) + c\dot{x} + kx + m\ddot{x}_d - me_1 \tag{5}$$

where s represents the coefficient that controls the rate of convergence; fun refers to a type of first or third quadrant functions. There are many candidate functions for $\mathrm{fun}()$, such as $\arctan(x)$, $\mathrm{sign}(x)$,etc. In this paper, $\arctan(x)$ is chosen because its value is relatively flat around 0 and becomes steeper as it moves away from 0 and the function value is limited between $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

Remark 1. The above controller is not the only option for the closed control system. Any control algorithm that can achieve a fast and stable performance can be adopted. These methods include linear quadratic regulator (LQR) , PID, etc.

Remark 2. The dynamic model of a robot is usually a coupled multivariate non-linear differential equation. The closed controller is typically implemented inside the robot and changes according to different structures and physical parameters.

In this paper, the robot's task space/joint space dynamic model is treated as a combination of such a single DOF system, which can be easily transferred among different robots.

3.2. Repulsion field

In the traditional artificial potential field method, the potential field is defined around the obstacle. The repulsion force, which is derived from the gradient of the field will act on the robot. If the obstacle is equivalent to a point mass, the repulsion field can be defined as:

$$U_{re}(\mathbf{X}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{x})} - \frac{1}{\rho_0}\right)^2 & \rho(\mathbf{x}) \le \rho_0 \\ 0 & \rho(\mathbf{x}) > \rho_0 \end{cases} \tag{6}$$

where $\mathbf{x}$ is the robot position, η denotes a scaling coefficient, ρ_0 represents the maximum radius of the repulsion field and $\rho(\mathbf{x})$ is the distance between the robot and the obstacle. By taking the negative gradient

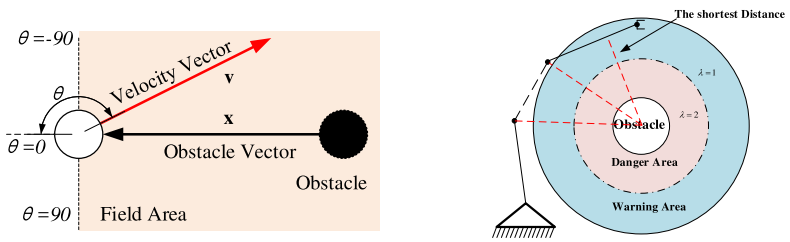


Fig. 4. Dynamic potential field parameters (a) The definition of theta, (b) The definition of λ.

of the potential field, the repulsive force acting on the robot can be obtained:

$$F_{re}(\mathbf{X}) = -\text{grad}\left[U_{re}(X)\right] = \begin{cases} \eta\left(\frac{1}{\rho(\mathbf{x})} - \frac{1}{\rho_0}\right)\frac{1}{\rho(\mathbf{x})^2}\frac{\partial\rho(\mathbf{x})}{\partial(\mathbf{x})} & \rho(\mathbf{x}) \le \rho_0 \\ 0 & \rho(\mathbf{x}) > \rho_0 \end{cases} \tag{7}$$

Since the above potential field only considers the distance between the end-effector and the obstacle but ignores the influence of the end-effector's speed and the direction of motion, which cannot thoroughly model the collision risk. Even in the same position, different speed vector means different collision risk, hence, it is not enough to consider the position distance only. In [29], the traditional potential field method is applied to relative velocity instead of position, which makes drone/UAV well controlled. However, for the manipulator, the velocity, position and the direction all will affect the final movement during obstacle avoidance [21]. To address these issues, the static potential field function is improved as:

$$U_{dyn} = \begin{cases} \lambda(-\cos\theta)^{\beta}\frac{\|(\mathbf{v})\|}{\rho(\mathbf{x})} & \frac{\pi}{2} < \theta \le \pi \\ 0 & 0 \le \theta \le \frac{\pi}{2} \end{cases} \tag{8}$$

where β is a constant, which can adjust the effect of the velocity vector angle θ. As shown in Fig. 4(a), θ represents the angle between the current velocity vector $\mathbf{v}$ and the end-effector position $\mathbf{x}$ (relative to the position of the obstacle), and can be calculated using Eq. (9). When θ is between $(\frac{\pi}{2}, \pi]$, it indicates that the robot is approaching the obstacle, and the dynamic repulsion field starts to work; when the value of θ is within $[0, \frac{\pi}{2}]$, the robot is away from the obstacle, and the dynamic repulsion field should not work.

$$\theta = \arccos\left(\frac{\mathbf{v}\cdot\mathbf{x}}{\|\mathbf{v}\|\,\|\mathbf{x}\|}\right) \tag{9}$$

λ denotes a scaling coefficient, as shown in Fig. 4(b), the space around the obstacle is divided into the warning area and the danger area. Which area the manipulator belongs to is determined by the shortest distance to the obstacle. When robot is in the warning area, λ is relatively small and the repulsion force of the robotic arm is relatively weak and λ is chosen as 1. In the danger area, λ is relatively large and the mechanical arm is strongly repulsed by the potential field and λ is set as 2.

The dynamic repulsive force is derived from a negative gradient of the potential function (Eq. (8)) as:

$$F_{dyn}(\mathbf{x},\mathbf{v}) = -\nabla_x U_{\text{dyn}}(\mathbf{x},\mathbf{v}) = \begin{cases} \lambda(-\cos\theta)^{\beta-1}\frac{\mathbf{v}}{\rho(\mathbf{x})}\left(\beta\nabla_x\cos\theta - \frac{\cos\theta}{\rho(\mathbf{x})}\nabla_x\rho(\mathbf{x})\right) & \frac{\pi}{2} < \theta \le \pi \\ 0 & 0 \le \theta \le \frac{\pi}{2} \end{cases} \tag{10}$$

The repulsive force generated from dynamic potential field considers all the above three factors and can improve the robot obstacle avoidance behavior. The repulsive force is inversely proportional to the distance ρ to the obstacle and positively proportional to the speed $\mathbf{v}$, and is zero when the speed $\mathbf{v}$ is zero or ρ is larger than the threshold; the repulsive force is also related to the angle between the velocity vector $\mathbf{v}$ and the direction vector to the obstacle; the repulse force is zero, if the angle θ is less than 90° (i.e., end-effector moves away from the obstacle).

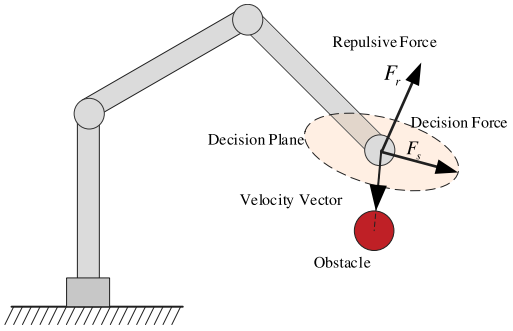


Fig. 5. A robotic manipulator that works in a three-dimensional space and is subjected to a decision-making force.

3.3. Decision-making force

There are two ways to avoid obstacles. One is to use the dynamic repulsion field to make the robot passively away from the obstacle, and the other is to use the decision mechanism to choose an obstacle avoidance direction actively. The direction of the decision-making force is perpendicular to the repulsive force and is determined by the repulsive force and the decision angle. Its value depends on factors such as obstacle distance and manipulator speed.

As shown in Fig. 1(b), the planar robot arm only needs to decide the decision-making force in two directions. However, for a robot manipulator operating in three-dimensional space, as shown in Fig. 5, the option of decision-making force is no longer in two directions, but is a plane perpendicular to the repulsive force. $\boldsymbol{F_r}$ denotes the repulsive force vector; $\boldsymbol{F_s}$ represents the decision force; the plane where $\boldsymbol{F_s}$ locates in is the decision plane. The decision force can follow any direction within the plane. Compared to the single repulsive force, this introduces great flexibility.

3.3.1. Decision-making force model

Direction of the decision-making force is closely related to the repulsive force, which is changing with space and time. In order to simplify the decision making process, this paper establishes the following decision-making force model as shown in Fig. 6.

The repulsive force vector is firstly rotated to coincide with the Z-axis of the reference frame. Consequently, the decision plane is transformed into the XY plane and $\boldsymbol{F_s}$ is converted to $\boldsymbol{F_s'}$ (after normalization). $\mathbf{r_z}$ is a vector perpendicular to the xoy plane and can be expressed by (11).

$$\boldsymbol{r}_z = \begin{bmatrix} 0 \\ 0 \\ \|\boldsymbol{F_r}\| \end{bmatrix} \tag{11}$$

Generally, the vector on the unit circle can be represented by one parameters: the vector angle ϕ. The normalized decision force is

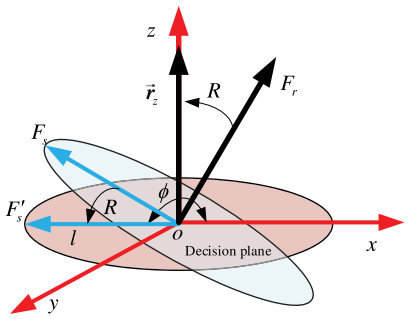


Fig. 6. Decision-making force model.

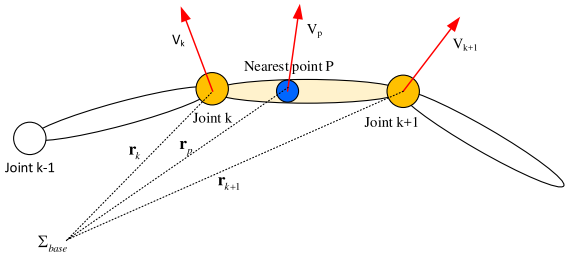


Fig. 7. The relationship between the joint velocity and the closest point velocity.

represented as

$$\boldsymbol{F}'_s = [cos\phi \quad \sin\phi \quad 0]^T \tag{12}$$

Since $\boldsymbol{F}_r$ and $\boldsymbol{r}_z$ are the normal vectors of decision plane and xoy plane, respectively, vector $\boldsymbol{F}_r$ to $\boldsymbol{r}_z$ and $\boldsymbol{F}_s$ to $\boldsymbol{F}'_s$ have the same transformation matrix $\mathbf{R}$. The transformation matrix $\mathbf{R}$ can be obtained by the following steps:

(1) Calculate the rotation angle α

The angle between the $\mathbf{F_r}$ and the vector $\mathbf{r_z}$ can be obtained by the inner product between the two vectors.

$$\alpha = \arccos\left(\frac{(\boldsymbol{F}_r, \boldsymbol{r}_z)}{\|\boldsymbol{F}_r\| \|\boldsymbol{r}_z\|}\right) \tag{13}$$

(2) Calculate the rotation axis $\boldsymbol{r}_\alpha$

The plane of rotation angle α is composed of $\mathbf{F}_r$ and $\boldsymbol{r}_z$, hence the axis of rotation must be perpendicular to the plane. Therefore, the vector can be calculated by cross-product of these vectors.

$$\boldsymbol{r}_\alpha = \frac{\boldsymbol{F}_r \times \boldsymbol{r}_z}{\|\boldsymbol{F}_r\| \|\boldsymbol{r}_z\|} \tag{14}$$

(3) Calculate the transformation matrix $\mathbf{R}$

According to the Rodrigues rotation formula [30]:

$$\mathbf{R} = 1 + \boldsymbol{r}_\alpha \sin(\alpha) + \boldsymbol{r}_\alpha^2(1 - \cos(\alpha)) \tag{15}$$

Then the decision force can be obtained using

$$\boldsymbol{F}_s = \lambda \boldsymbol{R}^{-1} \boldsymbol{F}'_s \tag{16}$$

From the above analysis, it can be seen that for each control step, except for variable λ and ϕ, the others are all constants. Similar to the amplitude of the repulsive force, value of the decision force is also related to robot speed and obstacle distance. λ is proportional to the speed $\boldsymbol{v}$ of the robot, and is inversely proportional to the distance ρ between the robot and the obstacle. Therefore λ can be defined as:

$$\lambda = \gamma \frac{|\boldsymbol{v}|}{\rho} \tag{17}$$

where γ is scaling coefficient. Using (12), (17) and (16), one can obtain a decision force for given conditions. The only undetermined variable is ϕ. How to choose the right decision force angle is the problem that needs to be solved.

3.3.2. Decision angle optimization

The introduce of ϕ provides a tool to deal with changing issues in different environmental configurations. By optimizing the decision angle, a smooth and effective obstacle avoidance motion can be achieved. The goal of optimization is to guarantee the robot to move away from the obstacle. Therefore, the optimization can be done by evaluating the change of robot velocity. In a control cycle, the velocity increment Δv of the end-effector can be expressed as

$$\Delta v = \frac{\boldsymbol{F_d} + \boldsymbol{F_{re}} + \boldsymbol{F_s}(\lambda, \phi)}{m} \Delta t \tag{18}$$

where Δt is the cycle time. And there are three forces (i.e., the actuator force $\boldsymbol{F_d}$, the repulse force $\boldsymbol{F_{re}}$, and the decision force $\boldsymbol{F_s}$) which are imposed on the end of the manipulator. The new speed v of the end-effector is

$$\boldsymbol{v} = \boldsymbol{v_{old}} + \Delta v \tag{19}$$

where $\boldsymbol{v_{old}}$ is current end-effector speed.

However, simply evaluating the end-effector speed cannot drive the robot links away from the obstacle. Therefore, it is necessary to consider the whole robot body and the nearest point p (relative to the obstacle) on the manipulator. This is illustrated in Fig. 7.

According to the geometric relationship, it can be obtained:

$$\boldsymbol{r_p} - \boldsymbol{r_k} = w \times (\boldsymbol{r_{k+1}} - \boldsymbol{r_k}) \tag{20}$$

where $0 \le w \le 1$ is the scaling coefficient which can be calculated by

$$w = \frac{\left\|\boldsymbol{r_p} - \boldsymbol{r_k}\right\|}{\left\|\boldsymbol{r_{k+1}} - \boldsymbol{r_k}\right\|}$$

Therefore the velocity for p point v_p can be got by

$$\boldsymbol{v_p} = \dot{\boldsymbol{r}}_{\boldsymbol{p}} = \dot{\boldsymbol{r}}_{\boldsymbol{k}} + s(\dot{\boldsymbol{r}}_{\boldsymbol{k+1}} - \dot{\boldsymbol{r}}_{\boldsymbol{k}}) = \boldsymbol{v_k} + s(\boldsymbol{v_{k+1}} - \boldsymbol{v_k}) \tag{21}$$

where v_k and v_{k+1} are the velocity of the joint k and joint $k+1$. In order to obtain the speed of the nearest point p, It is necessary to get the speed of the two ends of the link where the closest point p locates. For the end-effector speed v, there is

$$v = \boldsymbol{J}_n \dot{\theta}$$

where J_n is the Jacobian matrix and $\dot{\theta}$ is the joint speed. For each joint i, there is

$$v_i = \boldsymbol{J}_i \dot{\theta}$$

where $J_i, i = 1 \cdots n$ is the Jacobian matrix for each joint. Therefore, one gets

$$v_i = \boldsymbol{J}_i \boldsymbol{J}_n^{\dagger} v \tag{22}$$

where $\boldsymbol{J}_n^{\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_n$.

By substituting (22) into (21), it is able to obtain the velocity of the nearest point according to the relationship shown in Fig. 7.

Velocity toward the obstacle is denoted as v_p^-, and the velocity away from the obstacle is denoted as v_p^+. In order to evaluate the velocity of nearest point, the following rules should be considered.

- v_p^+ is better than v_p^- considering the robot safety issues and the obstacle avoidance task.
- If the manipulator cannot move away from the obstacle, it should approach the obstacle at a minimum speed.
- If the obstacle is unavoidable and the distance is small than threshold, the robot should stop immediately.

Therefore, the following index is proposed to measure the effect of $\boldsymbol{v_p}$.

$$\boldsymbol{v_m}(\phi) = dot(\boldsymbol{v_p}(\phi), p - p_0) \tag{23}$$

If direction of $\boldsymbol{v}_p$ is close to $p - p_0$, the robot will be away from the obstacle and $\boldsymbol{v}_m(\phi)$ is positive. Otherwise, $\boldsymbol{v}_m(\phi)$ is negative. A smaller $\boldsymbol{v}_m$ means the robot moves away from or approaches the obstacle slower than a bigger one.

In summary, for a given decision making angle ϕ, the resulting speed and the performance can be obtained by (18), (19), (21), (22) and (23).

Therefore, the following optimization problem can be formulated.

$$\phi^* = \underset{\phi}{\operatorname{argmin}}\, v_m(\phi), \phi \in [0, 2\pi) \tag{24}$$

In this paper, we adopt the random search method [31] to optimize the decision angle. First, we generate N candidate values for the decision angle, then calculate their corresponding $\boldsymbol{v}_m$, and finally select the ϕ which has minimum $\boldsymbol{v}_m$ value as the decision angle.

Remark 3. The above is solved using the minimum $\boldsymbol{v}_m$ as the target function. It can also maximize the shortest distance between the obstacle. The direction which gets the largest distance(the robot is safest), will be selected as the right decision-making direction. In general application cases, the decision angles can be optimized in a specific range so as to assign the moving direction according to one's preference. This introduces great flexibility.

Workflow of the proposed algorithm is shown in Algorithm 1.

Algorithm 1 Decision-making force based obstacle avoidance algorithm.

Require: The obstacle information: p_o, r_o
The target trajectory: $(x_d, \dot{x}_d, \ddot{x}_d)$

for $k = 1$ to K_t **do**
  Obtain current state $x_k, \dot{x}_k$ and target state $x_d(k), \dot{x}_d(k)$
  Calculate control force F_d based on control law (5)
  Calculate repulsive force F_r according to (9) and (10)
  Randomly generate N candidate decision angles $\phi_j, j = 1 \cdots N$
  for $j = 1$ to N **do**
    Evaluate ϕ_j using (23) $\rightarrow \boldsymbol{v}_m(\phi_j)$
  end for
  Choose ϕ_* using (24)
  Calculate F_s using (12), (17) and (16)
  $u \leftarrow F_d + F_r + F_s$
  Update $x_k, \dot{x}_k, \ddot{x}_k$ using multiple SMD model (1)
  $(x_r, \dot{x}_r, \ddot{x}_r) \leftarrow (x_k, \dot{x}_k, \ddot{x}_k)$
  Stream trajectory to robot motion controller
end for

4. Experiments and discussions

In order to verify the effectiveness of the proposed algorithm, the simulation experiment of the two-link manipulator was carried out in V-rep, and the physical experiment was carried out on the experimental platform based on ABB IRB1200 robot.[1]

Three groups of experiments are performed:

- 2-DOF VREP simulations
- 6-DOF experiments with a round object
- 6-DOF experiments with a complex object

In these three obstacle avoidance experiments, to compare the difference in performance, the following methods are implemented: the static repulsive field (SRF), the dynamic repulsive field (DRF), the static repulsive and decision-making force (SRDF), the dynamic repulsive and decision-making force (DRDF) respectively. In order to show the advantages, DMP with static repulsion field(DMPSF) and dynamic repulsion field respectively(DMPDF) are also tested and compared [21–23].

[1] We put part codes and videos of simulations and experiments on https://github.com/pz10150127/ObstacleAvoidancebasedonDecisionmakingForce.

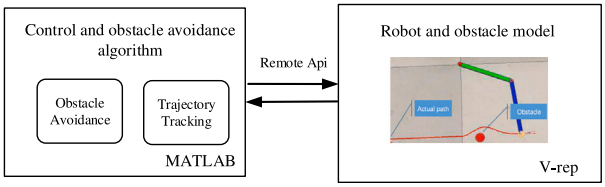


Fig. 8. Simulation environment.

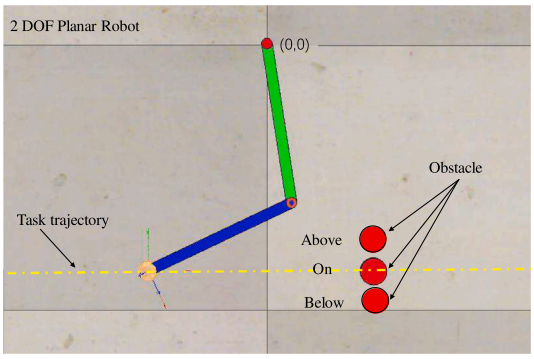


Fig. 9. The 2-DOF planar robot in V-rep.

4.1. Obstacle avoidance experiments with a 2-DOF planar robot in V-rep

This section compares the above mentioned methods in different configurations (Straight and curved trajectories; the obstacle is above, on or below the original trajectory).

4.1.1. Comparison of SRF/DRF/SRDF/DRDF

As shown in Fig. 8, the simulation environment is constructed in V-rep, including the 2-DOF planar robot and the obstacle model (i.e., a sphere). Trajectory tracking and obstacle avoidance algorithms are implemented in MATLAB. The remote APIs [32] are used to establish the communication between MATLAB and V-rep. The Matlab side sends the desired position, velocity, and acceleration commands to the V-rep side, and then the V-rep side feeds back the actual position, velocity, and acceleration of the robot.

As shown in Fig. 9, a 2-DOF planar robot is built in V-rep. The robot rotates around the origin (0 m,0 m), and the length of each link is 0.3 m. The robot's desired trajectory is a horizontal straight line from (−0.4 m, −0.4 m) to (0.4 m, −0.4 m); The obstacle is placed at three different places: above the trajectory (0.1 m, −0.39 m), on the path (0.1 m, −0.40 m) and below the trajectory (0.1 m, −0.41 m).

In Fig. 10, the obstacle is above the task trajectory. Figs. 10(a) and 10(b) show that when the manipulator encounters an obstacle, the repulsion force keeps the robot end effector away from the obstacle. The DRF generates a smoother motion than SRF and the robot return back to normal trajectory quickly. However, it does not follow a correct obstacle avoidance direction. The repulsive force will push the end effector downward. However, the robot base is located above. The robot links have already collided with the obstacle.

On the contrary, with the help of decision force, SRDF and DRDF methods both avoid the obstacle with proper detection. Figs. 10(c) and 10(d) show the resulting robot motion. Because static repulsive field does not consider robot speed value and angle, the forces changes significantly and affect the robot even the robot has passed the obstacle. Comparisons among the results show that the DRDF method provides the best result (a smoother, safer, and efficient obstacle avoidance movement).

In Fig. 11, the obstacle is on the task trajectory. In Figs. 11(a) and 11(b), when the robot encounters an obstacle, because the robot path

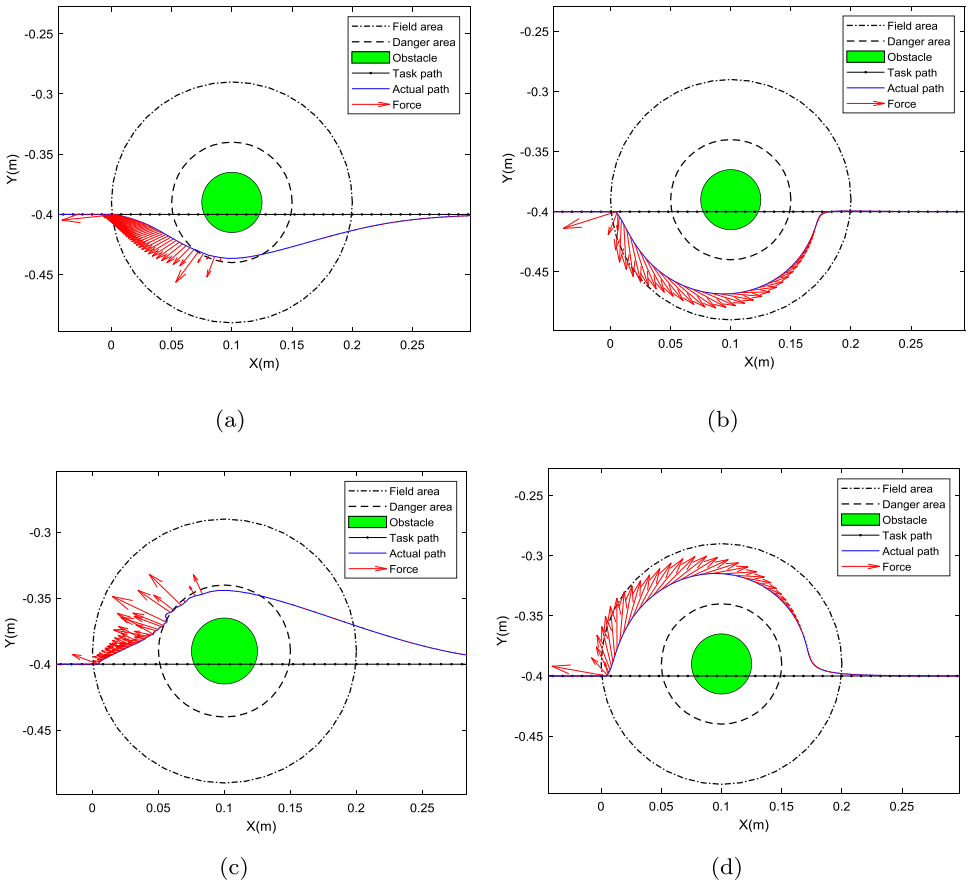


Fig. 10. The obstacle is above the task path and its position is (0.1 m, −0.39 m) (a) with the static repulsive field, (b) under the dynamic repulsive field, (c) under the static repulsive and decision-making force, (d) under the dynamic repulsive and decision-making force.

Table 1
The comparisons between SRF, DRF, SRDF and DRDF.

	Obstacle position			Local minima	Smoothness	Convergence speed
	Above	On	Below			
SRF	Failed	Failed	Success	Yes	No	Slow
DRF	Failed	Failed	Success	Yes	Yes	Fast
SRDF	Success	Success	Success	No	No	Slow
DRDF	Success	Success	Success	No	Yes	Fast

and obstacle are in a line, the driving force and repulsive force are pushing away from each other. Consequently, the robot oscillates under their joint action. In some circumstances, the noise may bring the robot away from local minima, however the moving direction is random and may still lead to a collision.

In Figs. 11(c) and 11(d), the decision-making force provides a suitable obstacle avoidance direction. The robotic manipulator successfully avoids the obstacle without any oscillation. Comparisons among the results show that the DRDF can overcome the local minimum problem, ensure a safer obstacle avoidance distance, and quickly recover the tracking task's track after obstacle avoidance.

In Fig. 12, the obstacle is below the task trajectory. The repulsive force is upward and can drive the robot toward its base. Therefore, these four methods all successfully complete the obstacle avoidance task. For a quality evaluation, the dynamic repulsive field can make the robot have a more considerable safety distance, while the decision field can make the robot have a better and robust obstacle avoidance direction.

Table 1 summarizes the feature differences between those four methods. It can be concluded that the DRDF method is robust to different configurations. The advantage of dynamic repulsive field is adopted. The dynamic control system ensures the robot can return to normal motion after avoiding the obstacle. The decision force can guarantee the robot follow a proper direction to ensure safety of the robot links.

4.1.2. Comparison of DRDF with DMPSF and DMPDF

The robot's task trajectories are a horizontal straight line from (−0.4 m, −0.4 m) to (0.4 m, −0.4 m) and a sine wave trajectory($y = -0.4 + 0.03\sin(10x)$, $x \in [-0.4\text{ m}, 0.4\text{ m}]$). The results are plotted in Fig. 13.

From Fig. 13(a), it is able to find that all these three algorithms can avoid the obstacle finally. However, during the executing process, DMPSF and DMPDF oscillate significantly and have the risk to fall into local minima. Fig. 13(b) shows that both DMPSF and DMPDF cost much time to get out of local minima. After escaping from the "trap", although it can drive the robot toward the final goal, the actual trajectory is not identical to the planned ones. On the other hand, benefiting from the decision-making force and closed-loop control mechanism, DRDF does not fall into a local minimum and can quickly resume trajectory tracking after surpassing the obstacle.

The sine wave motion tracking capability is also tested and shown in Fig. 14. Still, because the obstacle is on the trajectory, the actual motion of DMPSF is not very smooth, and the convergence of DMPSF and DMPDF is slower than DRDF. The obstacle avoiding trajectory generated by DRDF is smooth, oscillate-free, and efficient.

Their differences are summarized in Table 2.

4.2. Obstacle avoidance experiment with a 6-DOF ABB IRB1200 robot

The above simulation results have already demonstrated the superiority of DRDF. The DRDF will be further verified by the 6-DOF robot

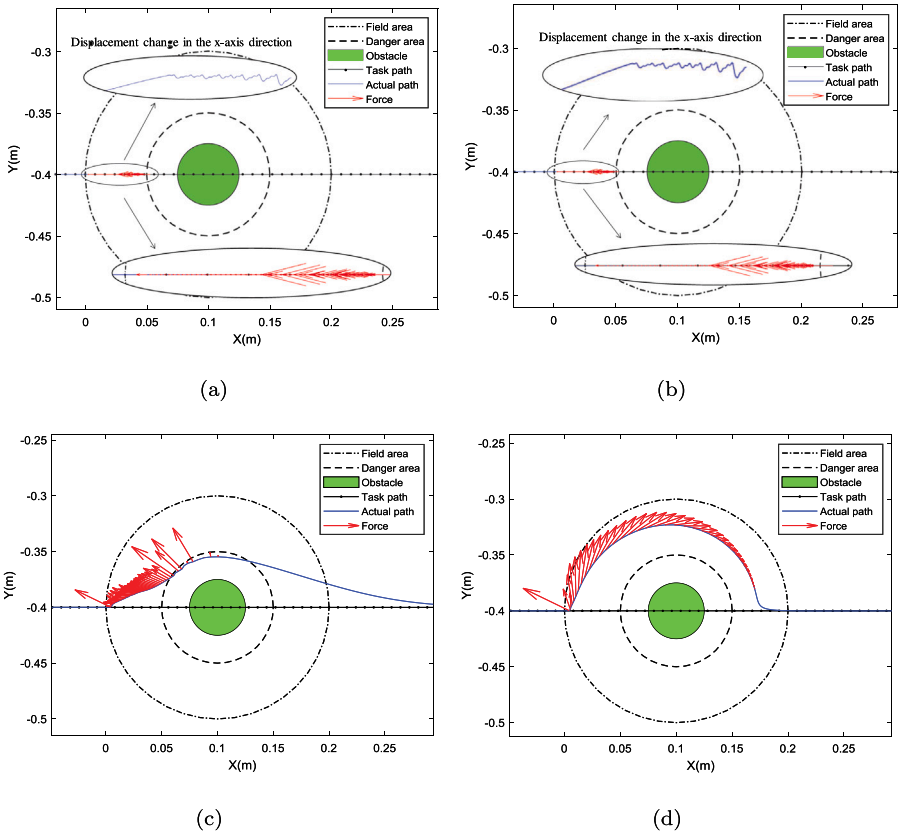


Fig. 11. The obstacle is on the task path and its position is (0.1 m, −0.40 m) (a) with the static repulsive field, (b) with the dynamic repulsive field, (c) with the static repulsive field and decision-making force, (d) under the dynamic repulsive field and decision-making force.

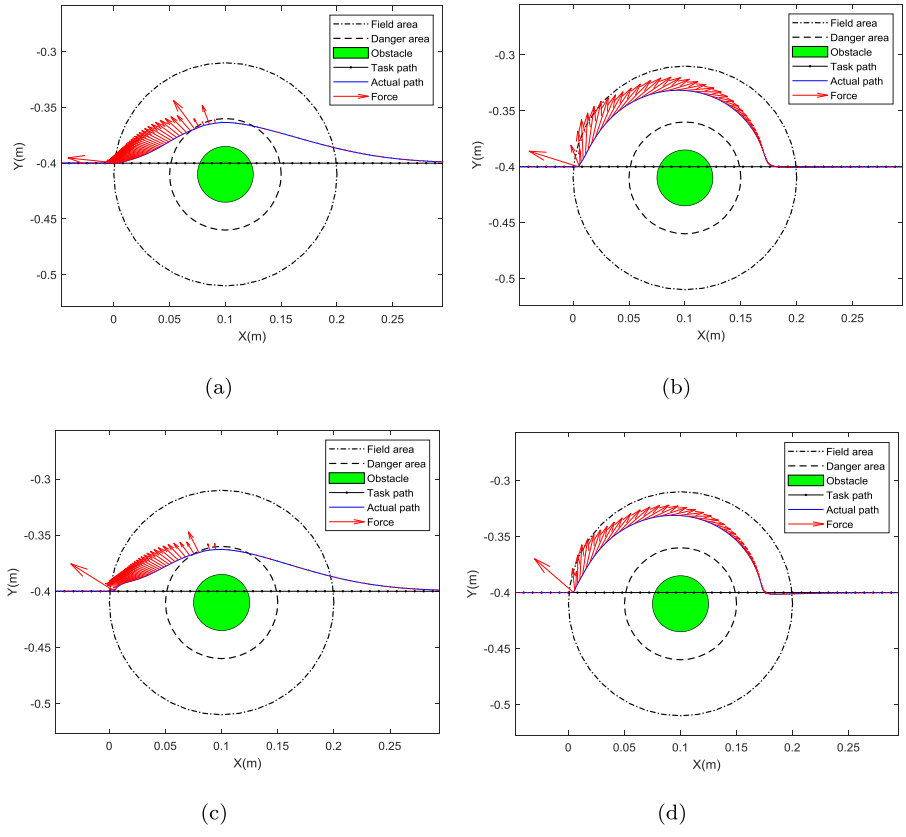


Fig. 12. The obstacle is below the task path and its position is (0.1 m, −0.41 m) (a) under the static repulsive field, (b) under the dynamic repulsive field, (c) under the static repulsive and decision-making force, (d) under the dynamic repulsive and decision-making force.

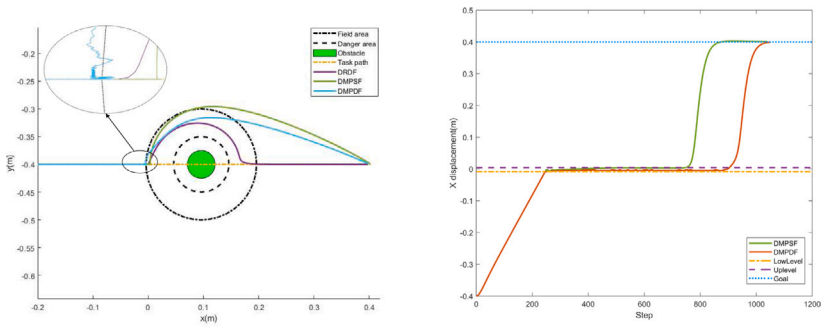


Fig. 13. Comparison between DRDF, DMPSF, and DMPDF when the task trajectory is a straight line. (a) the result of comparison (b) displacement change in the X axis of DMPSF and DMPDFinate Transformation between components.

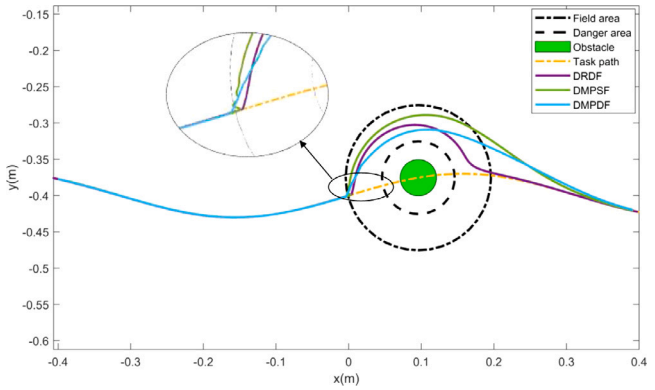


Fig. 14. The comparison between DRDF, DMPSF, and DMPDF when the task trajectory is a sine wave.

Table 2
The comparisons of DRDF with DMPSF and DMPDF.

	Local minima	Smoothness	Convergence speed
DMPSF	Yes	No	Slow
DMPDF	Yes	Yes	Slow
DRDF	No	Yes	Fast

experimental platform. As shown in Fig. 15(a), the experiment platform is based on an industrial robot manipulator: ABB IRB1200, which can carry a payload of up to 7 kg with a reach of 700 mm. The robot is mounted on the workbench, and an Orbbec Astra Pro RGB-D sensor is attached on the camera mount rack to detect the obstacle. An external computer is used to run the obstacle algorithm, and the computer is connected to the robot IRC5 controller by using a serial interface with bidirectional communication protocol: receiving the robot positions and sending the motion commands.

Since the collision between the manipulator and the obstacle can be divided into two types, manipulator links collision and end-effector collision. This section conducts two experiments for the above two cases, that is, the obstacle is on the task trajectory, and the obstacle is near the link.

4.2.1. Obstacle detection and coordinate transformation

Since the position of the obstacle in the environment is uncertain, it is necessary to obtain the obstacle location when performing a task path. In this section, the RGB-D sensor is used to obtain the point cloud of scenes. If an obstacle occurs in the viable area, its position/size can be get by processing the point cloud with the Point Cloud Library(PCL). By recording the original scene point cloud, one can segment the difference between current and model point clouds. The differences can be seen as obstacles. By using the Random Sample Consensus (RANSAC) algorithms, one can fit the obstacle using a sphere model. As shown in Fig. 16, The green dots are the segmented sphere obstacle, which can be used to find the center of the sphere and radius. Because of the voxel filter, the detection error is about 3 mm.

The obstacle is detected in the sensor coordinate frame. It has to be transformed into the robot base coordinate frame before re-planning the robot motion. The calibration process follows our published work [33]. As shown in Fig. 15(b), the base and camera can be calibrated using two approaches, Base-World-Object-Camera or Base-Tool-Object-Camera. By placing the object at different locations and establishing the correspondence between camera and robot, the hand-eye transformation matrix can be estimated by solving an AX = ZB problem. The detailed description can be found in [33].

4.2.2. Obstacle avoidance with the obstacle on the task path

The task path of the robot is a straight line from (0.589 m, −0.3 m, 1.157 m) to (0.589 m, 0.3 m, 1.157 m). The obstacle(a sphere with a diameter of 100 mm) is located at (0.589 m, 0 m, 1.157 m), just on the task trajectory. The desired robot motion is feeded into the proposed trajectory tracking and obstacle avoidance system. The actual robot motion is derived according to the algorithm given in Algorithm 1. In each step, decision angles are randomly sampled from $[0, 2\pi)$, which follows uniform distribution. Number of the random candidates $N = 18$.

Fig. 17 shows one of the movements of the ABB IRB1200 robot in the obstacle avoidance process with the DRDF. It can be seen that not only the end effector of the robot manipulator but also the robot body always maintains a safe distance from the obstacle. At t = 2 s, the manipulator was following the task path. At t = 5 s, the obstacle was detected. From t = 5 s to t = 10 s, the manipulator's end was avoiding the obstacle with the DRDF. At t = 10 s, the manipulator recovered to follow the task path.

As shown in Fig. 18, the robot can avoid the obstacle very smoothly. When the robot approaches the obstacle, the dynamic repulsive force and decision-making force will help the robot to choose a suitable direction to keep away from the obstacle. Finally, when the robot passes the obstacle, the repulsive force and decision forces disappear and the driving force generated by close loop system drives the robot back to the normal trajectory.

The candidate decision angles are randomly generated in each decision-making process as shown in Fig. 19(a). Each angle is converted into a decision force and evaluated by the safety distance(as mentioned in Remark3, the greater the distance, the safer it is). This randomness greatly increases the flexibility, as shown in Fig. 19(b), even for the same configuration, it is able to generate multiple different feasible obstacle avoidance paths. Those paths can all ensure that the manipulator will not collide with the obstacle. It is able to remove this uncertainty by finding exact solution for (24) in each step. However, it

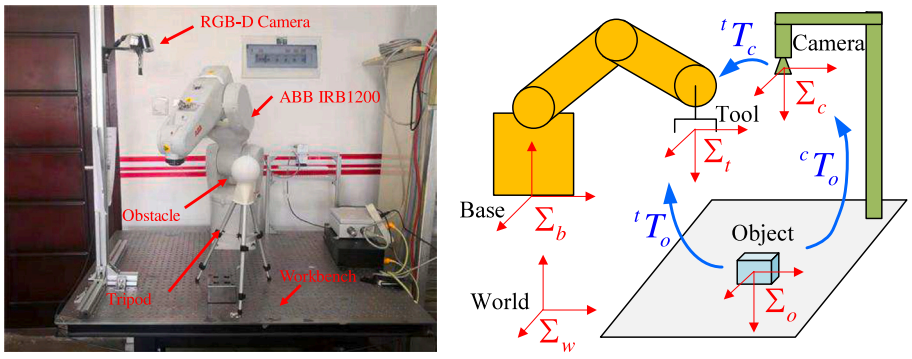


Fig. 15. Experiment environment based on ABB IRB1200. (a) photo of the platform (b) Coordinate Transformation between components.

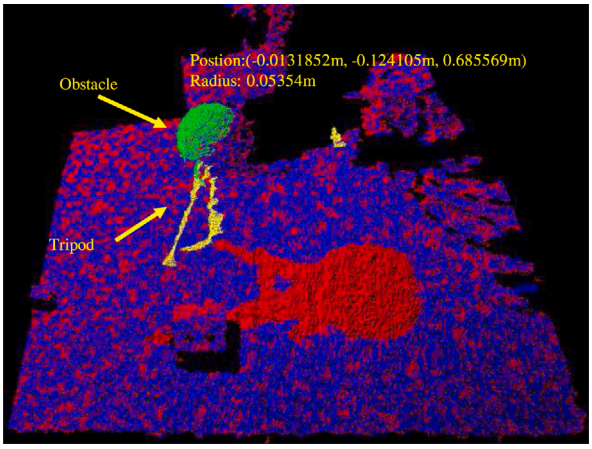


Fig. 16. Detecting the obstacle based on PCL.

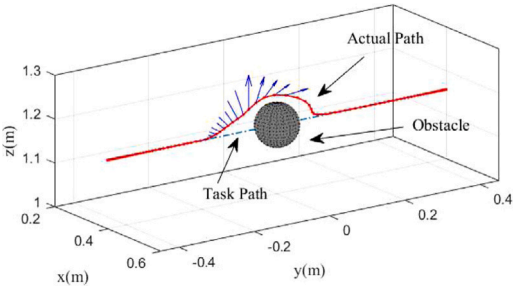


Fig. 18. ABB IRB1200 obstacle avoidance process with the obstacle on the task path.

will need a lot of function evaluations and computations. Considering that the collision risk a relative concept, it is not necessary to achieve absolute optimal for the decision selection problem. After all, efficiency is another issue for online path planning.

4.2.3. Obstacle avoidance with the obstacle near the links

The obstacle avoidance algorithm proposed in this paper can also be used to avoid link collisions. In this experiment, The task path of the robot is a straight line from (0.35 m, −0.35 m, 1.2 m) to (0.35 m,

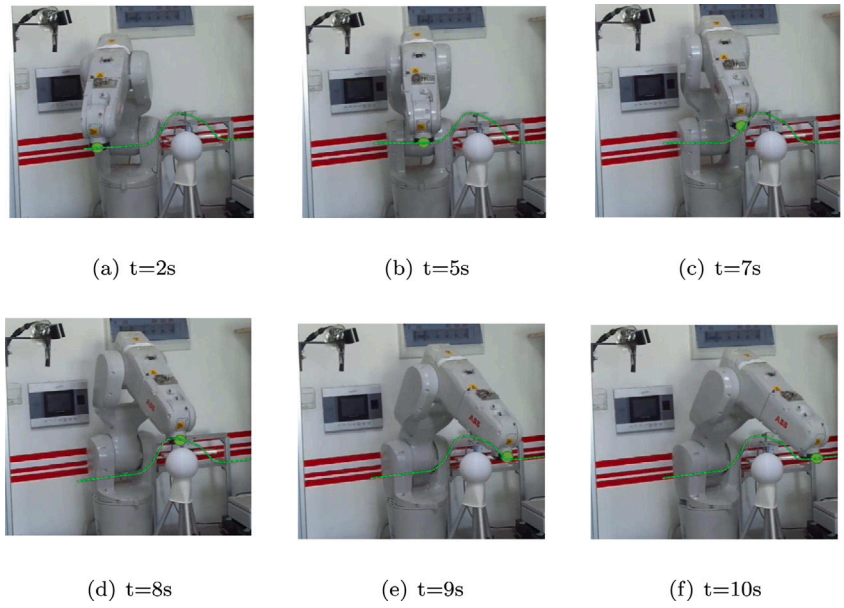


Fig. 17. The movements of the ABB IRB1200 Manipulator in the obstacle avoidance process under the action of DRDF. (a), (b): The robot is tracking task trajectory. (c), (d) The robot is avoiding the obstacle. (e) The robot has completed obstacle avoidance. (f) The robot recovers to track the task path.

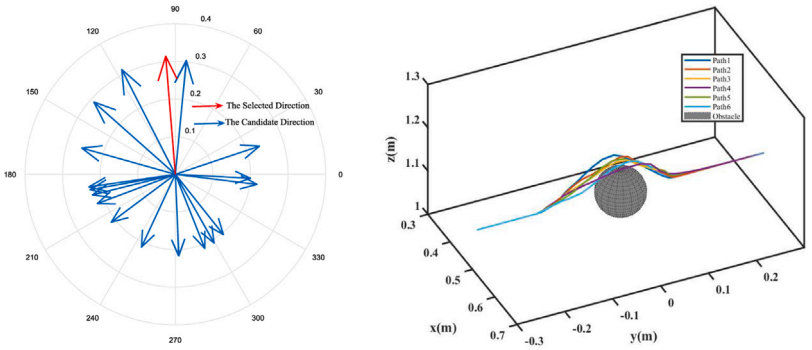


Fig. 19. Multiple obstacle avoidance paths and the selection process of decision-making force. (a) The process of choosing the direction of decision-making force, the greater the shortest distance between the robot and the obstacle, the safer the robot is. (b) For the same configuration, the decision force generate multiple paths.

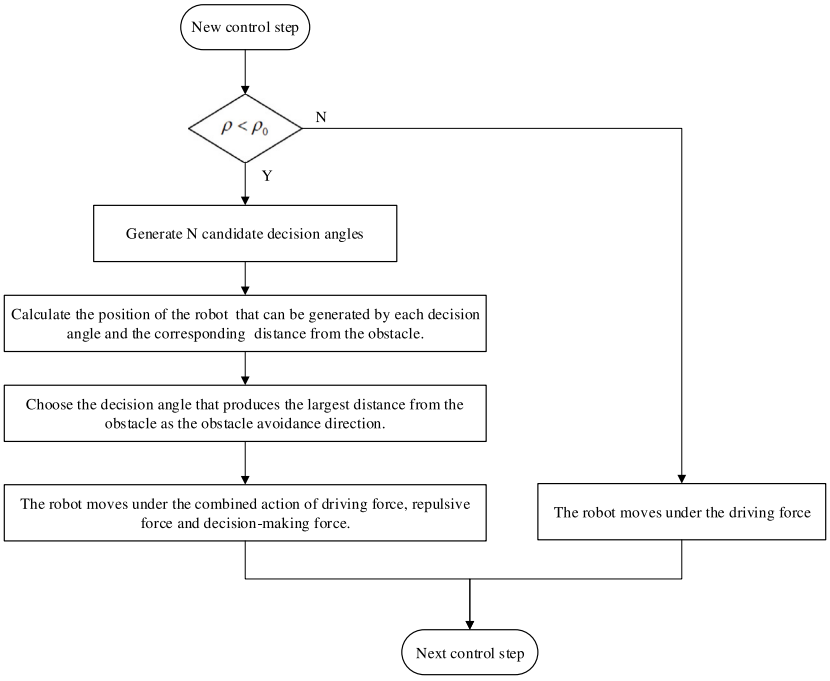


Fig. 20. The working process of DRDF at a specific point.

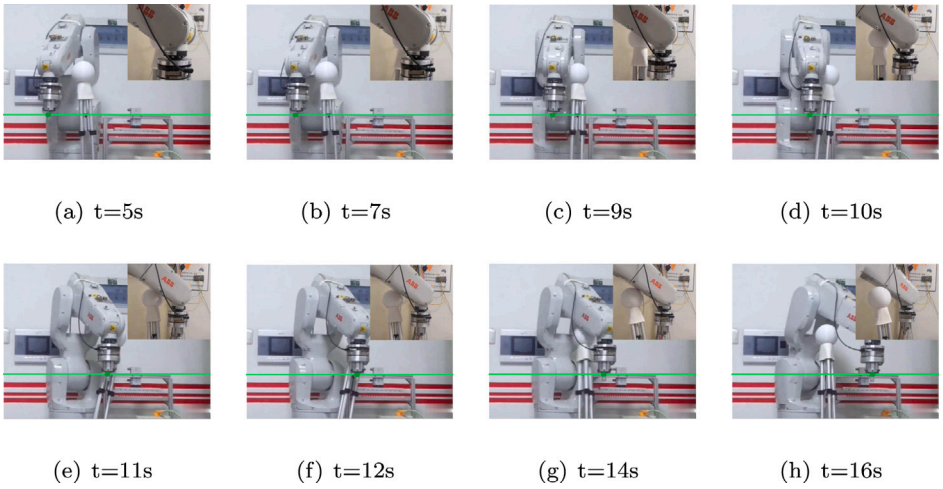


Fig. 21. ABB IRB1200 robot followed the desired trajectory without obstacle avoidance algorithm.

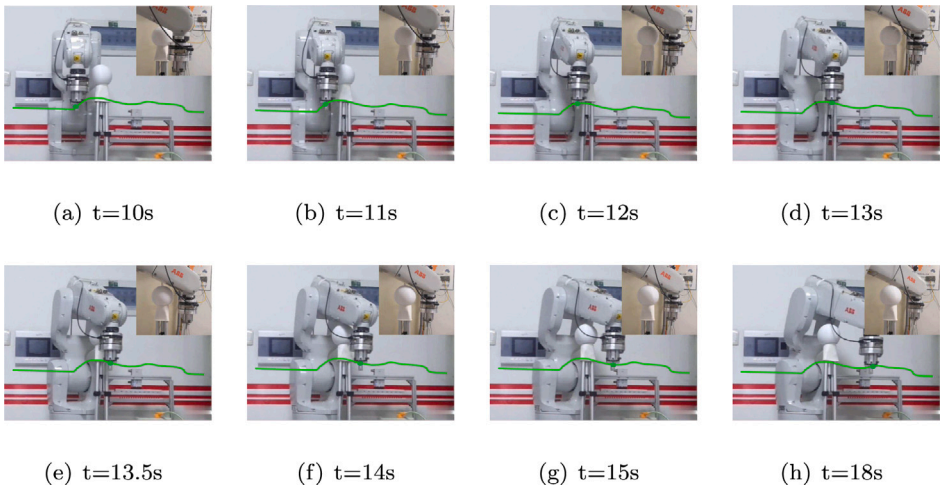


Fig. 22. A sequence of still photos shows the movement of the robot, while it successfully avoids the obstacle by using the DRDF.

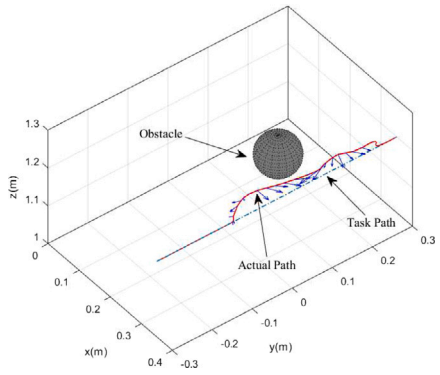


Fig. 23. The change of end-effector's trajectory of the manipulator during obstacle avoidance.

0.35 m, 1.2 m). The obstacle(a sphere with a diameter of 10 mm) is located at (0.28 m, −0.15 m, 1.2 m).

For the redundant manipulator, the obstacle avoidance can be achieved by moving in null space without changing the end trajectory. The literature [34] uses the method of dynamic neural network to optimize the manipulability of the manipulator, and it make the manipulator avoid obstacles more easier. However, IRB1200 is a non-redundant manipulator, avoiding the obstacle means part or original motion has to be affected. Fig. 20 shows the detail working process of DRDF at a specific point.

Fig. 21 shows that the manipulator was following the task path without using the obstacle avoidance algorithm, it will collide with the obstacle. At t = 7 s, the obstacle began to come into contact with the mechanical link. From t = 7 s to t = 14 s, the robot arm knocked the obstacle away from its original position. At t = 16 s, the arm left the obstacle.

Fig. 22 shows that the manipulator was following the task path with the DRDF algorithm. At t = 10 s, the obstacle was detected. From t = 11 s to t = 15 s, with the DRDF, the manipulator successfully avoided the link collisions by changing the end-effector's position.

As shown in Fig. 23, In the process of avoiding the obstacle, the end of the manipulator changes smoothly with the decision force and dynamic repulsion field. After avoiding the obstacle, the manipulator can quickly recover the trajectory tracking.

The above experimental results prove that the proposed DRDF obstacle avoidance algorithm is also effective for obstacle avoidance of spatial robots. DFDR can ensure safety of the robot end and the robot's links by changing the end position of the robot.

4.2.4. Complex-shaped obstacle avoiding experiments

In order to verify whether DRDF can be applied to avoid complex shaped obstacles, the following experiments are preformed. To simulate the human–robot interaction scenario, a bionic robot hand developed in our lab is used as the obstacle. The platform is shown in Fig. 24(a). The Microsoft Azure Kinect is used to sense the environment and detect the bionic hand. The captured point cloud is processed using PCL library as shown in Fig. 24(b). By performing the segmentation, detection and distance querying, the minimum robot to obstacle distance can be evaluated on the fly.

A straight line from point (0.35 m, −0.35 m, 1.05 m) to point (0.35 m, 0.35 m, 1.05 m) which coincides with the bionic hand is designed as the task trajectory. It can be found in Fig. 25 as a red dash line.

Because the bionic hand is located in front of the robot, the feasible surpassing region is limited. Therefore, in the experiments, the search range of the decision angles are also set according to preference. The initial values of decision angle θ_0 are set as $0\pi, 0.25\pi, 0.5\pi, 0.75\pi$, and their ranges are $\theta_0 \pm 0.25\pi$. The number of candidate decision angles is N = 18. The results are shown in Fig. 25(a). It is clearly to see that DRDF can generate different new collision-free trajectories according to different decision parameters.

Because the bionic hand's shape is irregular, the force modulated obstacle avoiding trajectory is not circular. However, all of them can avoid the obstacle and maintain a safe distance. The trajectory is smooth, and when the obstacle is surpassed, it returns to original motion right away. Fig. 26 is a sequence of snapshots when the robot avoids the complex shaped obstacle by using path 1 in Fig. 25(a). From t = 4 s to t = 10 s, the manipulator was following the task path. At t = 10 s, the obstacle was detected. From t = 10 s to t = 18 s, the manipulator was avoiding the obstacle with the DRDF. After t = 18 s, the manipulator began to recover the task path.

Fig. 25(b) shows the obstacle avoiding trajectory generating by DMPSF and DMPDF. It can be found that when the robot first approach the neighborhood of the bionic hand, because the robot is moving toward the obstacle, the repulsive force acts as a speed reducer instead of a direction tuner. The robot is trapped in a local minima. After a while, the robot moves away with the help of random noise repulsive force. Because the motion is dominated by the repulsive force (which is dominated by the object shape), hence only a unique solution is generated.

The above results are summarized in Table 3. Compared to DRDF, DMP based methods may be trapped at local minima and rely on the shape of the obstacle or effect of the random noise, which cannot guarantee the robustness when different objects are encountered. Meanwhile, only one unique solution may be generated for a configuration, which limits the capability when it is applied in a complex

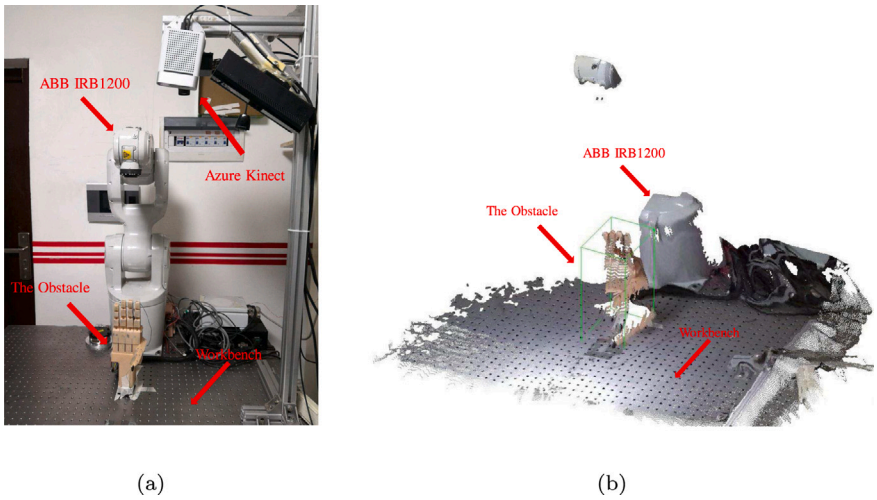


Fig. 24. Experiment environment with complex shaped obstacle. (a) Experiment platform (b) Detection the obstacle based on PCL.

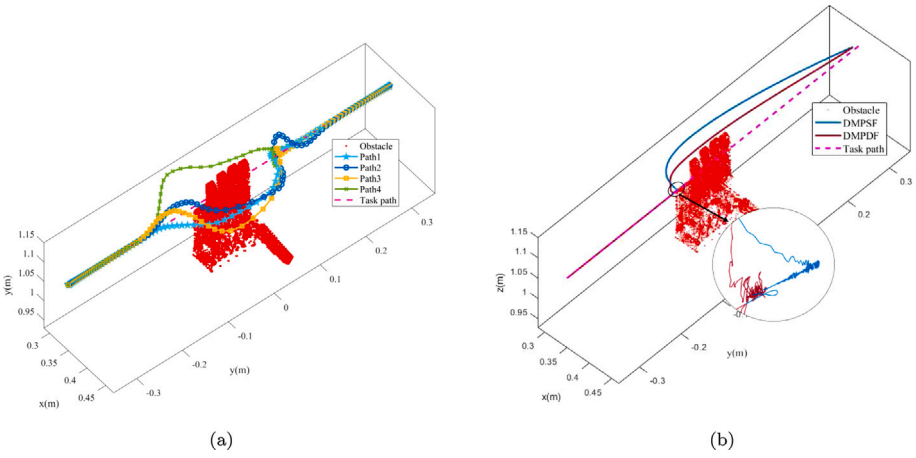


Fig. 25. Comparison between DRDF and DMP based algorithms with complex shaped obstacle.

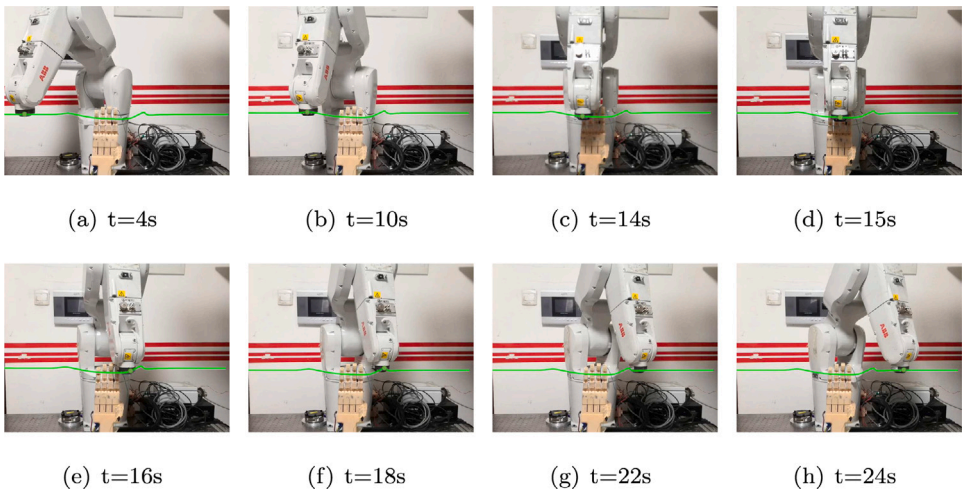


Fig. 26. A sequence of still photos shows the robot's movement, and the robot successfully avoids the complex shaped obstacle by using path 1 in Fig. 25(a).

Table 3
The comparisons of DRDF with DMPSF and DMPDF.

	Local minima	Trajectory num	Convergence speed
DMPSF	Yes	Single	Slow
DMPDF	Yes	Single	Slow
DRDF	No	Multiple	Fast

environment with complex objects; furthermore, after surpassing the obstacle, DMP based method can only guarantee the reachability of the target points instead of the target trajectory. Because those target trajectory is offline planned and may also encode some of the process and environment information. Maintaining the original task plan in some tasks is also important. Therefore, in the above aspects, DRDF shows significant advantages over DMP based methods.

5. Conclusions

This paper proposes an algorithm for obstacle avoidance of robot manipulator, which consists of a closed-loop control system, a dynamic repulsion field, and a decision-making force. This solution fulfills its advantages. The closed-loop control system can ensure the smoothness of the re-planed trajectory and ensure the robot returns to its original motion after passing the obstacle, which is vital for some tasks. The dynamic repulsion field can generate smooth force for a different speed, distance, and moving direction. Finally, introducing a decision-making force can bring great flexibility into the online adaption framework and guarantee the end effector and robot links' safety. The proposed solution shows effectiveness, robustness, and smoothness in the simulation experiments and physical experiments.

CRediT authorship contribution statement

Wei Zhang: Planning ,Execution, Formal analysis. **Hongtai Cheng:** Planning ,Execution, Formal analysis. **Lina Hao:** Planning ,Execution, Formal analysis. **Xingchen Li:** Planning ,Execution, Formal analysis. **Mingfang Liu:** Planning ,Execution, Formal analysis. **Xifeng Gao:** Planning ,Execution, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the financial support of the National Science Foundation of China (No. U1613205, 62073063), Fundamental Research Funds for the Central Universities, China (N18240007-2, N2003003).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.rcim.2020.102114.

References

[1] M. Safeea, P. Neto, R. Bearee, On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case, Robot. Auton. Syst. 119 (2019) 278–288, http://dx.doi.org/10.1016/j.robot.2019.07.013.

[2] M. Askarpour, D. Mandrioli, M. Rossi, F. Vicentini, Formal model of human erroneous behavior for safety analysis in collaborative robotics, Robot. Comput.-Integr. Manuf. 57 (JUN.) (2019) 465–476, http://dx.doi.org/10.1016/j.rcim.2019.01.001.

[3] X. Gao, X. Li, C. Zhao, L. Hao, C. Xiang, Variable stiffness structural design of a dual-segment continuum manipulator with independent stiffness and angular position, Robot. Comput.-Integr. Manuf. (ISSN: 0736-5845) 67 (2021) 102000, http://dx.doi.org/10.1016/j.rcim.2020.102000.

[4] J.J. Kuffner, S.M. LaValle, RRT-connect: An efficient approach to single-query path planning, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), 2, 2000, pp. 995–1001, http://dx.doi.org/10.1109/ROBOT.2000.844730.

[5] L.E. Kavraki, P. Svestka, J. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, in: International conference on robotics and automation, 12, (4) 1996, pp. 566–580, http://dx.doi.org/10.1109/70.508439.

[6] R. Diankov, J. Kuffner, Randomized statistical path planning, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2007, pp. 1–6, http://dx.doi.org/10.1109/IROS.2007.4399557.

[7] M. Toussaint, Robot trajectory optimization using approximate inference, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 1049–1056, http://dx.doi.org/10.1145/1553374.1553508.

[8] B. Burns, O. Brock, Toward optimal configuration space sampling, in: Robotics: Science and Systems, 2005, pp. 105–112.

[9] K. Wei, B. Ren, A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm, Sensors 18 (2) (2018) 571, http://dx.doi.org/10.3390/s18020571.

[10] H. Akbaripour, E. Masehian, Semi-lazy probabilistic roadmap: a parameter-tuned, resilient and robust path planning method for manipulator robots, Int. J. Adv. Manuf. Technol. 89 (2017) 1401–1430, http://dx.doi.org/10.1007/s00170-016-9074-6.

[11] S. Liu, D. Sun, C. Zhu, A dynamic priority based path planning for cooperation of multiple mobile robots in formation forming, Robot. Comput.-Integr. Manuf. (ISSN: 0736-5845) 30 (6) (2014) 589–596, http://dx.doi.org/10.1016/j.rcim.2014.04.002.

[12] L. Žlajpah, T. Petrič, Obstacle avoidance for redundant manipulators as control problem, in: Serial and Parallel Robot Manipulators-Kinematics, Dynamics, Control and Optimization, IntechOpen, 2012, pp. 203–230, http://dx.doi.org/10.5772/32651.

[13] B. Zi, J. Lin, S. Qian, Localization, obstacle avoidance planning and control of a cooperative cable parallel robot for multiple mobile cranes, Robot. Comput.-Integr. Manuf. (ISSN: 0736-5845) 34 (2015) 105–123, http://dx.doi.org/10.1016/j.rcim.2014.11.005.

[14] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: Proceedings. 1985 IEEE International Conference on Robotics and Automation, 2, 1985, pp. 500–505, http://dx.doi.org/10.1109/ROBOT.1985.1087247.

[15] M. Safeea, P. Neto, Minimum distance calculation using laser scanner and IMUs for safe human-robot interaction, Robot. Comput.-Integr. Manuf. 58 (2019) 33–42, http://dx.doi.org/10.1016/j.rcim.2019.01.008.

[16] S. Quinlan, O. Khatib, Elastic bands: Connecting path planning and control, in: 1993 Proceedings IEEE International Conference on Robotics and Automation, IEEE, 1993, pp. 802–807, http://dx.doi.org/10.1109/ROBOT.1993.291936.

[17] O. Brock, O. Khatib, Elastic strips: A framework for motion generation in human environments, Int. J. Robot. Res. 21 (12) (2002) 1031–1052, http://dx.doi.org/10.1177/027836490202101202.

[18] M. Shahabi, H. Gharibl, M. Beschi, Obstacle avoidance of redundant robotic manipulators using safety ring concept, Int. J. Comput. Integr. Manuf. (2019) 1–10, http://dx.doi.org/10.1080/0951192X.2019.1599438.

[19] E. Yoshida, F. Kanehiro, Reactive robot motion using path replanning and deformation, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 5456–5462, http://dx.doi.org/10.1109/ICRA.2011.5980361.

[20] D. Papageorgiou, T. Kastritsi, Z. Doulgeri, A passive robot controller aiding human coaching for kinematic behavior modifications, Robot. Comput.-Integr. Manuf. (ISSN: 0736-5845) 61 (2020) 101824, http://dx.doi.org/10.1016/j.rcim.2019.101824.

[21] D. Park, H. Hoffmann, P. Pastor, S. Schaal, Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields, in: Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, 2008, pp. 91–98.

[22] H. Hoffmann, P. Pastor, D. Park, S. Schaal, Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance, in: 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 2587–2592, http://dx.doi.org/10.1109/ROBOT.2009.5152423.

[23] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors, Neural Comput. 25 (2) (2013) 328–373, http://dx.doi.org/10.1162/NECO_a_00393.

[24] A.D. Dang, H.M. La, T. Nguyen, J. Horn, Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force–based approach, Int. J. Adv. Robot. Syst. 16 (3) (2019) http://dx.doi.org/10.1177/1729881419847897.

[25] T. Petrič, L. Žlajpah, Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem, Robot. Auton. Syst. 61 (9) (2013) 948–959, http://dx.doi.org/10.1016/j.robot.2013.04.019.

[26] J.R. Busemeyer, J.T. Townsend, Fundamental derivations from decision field theory, Math. Social Sci. (ISSN: 0165-4896) 23 (3) (1992) 255–282, http://dx.doi.org/10.1016/0165-4896(92)90043-5.

[27] J.R. Busemeyer, J.T. Townsend, Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment, Psychol. Rev. 100 (3) (1993) 432, http://dx.doi.org/10.1037/0033-295x.100.3.432.

[28] J.R. Busemeyer, A. Diederich, Survey of decision field theory, Math. Social Sci. 43 (3) (2002) 345–370, http://dx.doi.org/10.1016/S0165-4896(02)00016-1, Random Utility Theory and Probabilistic measurement theory.

[29] A.C. Woods, H.M. La, A novel potential field controller for use on aerial robots, IEEE Trans. Syst. Man Cybern. Syst. 49 (4) (2019) 665–676, http://dx.doi.org/10.1109/TSMC.2017.2702701.

[30] R.W. Brockett, Robotic manipulators and the product of exponentials formula, in: P.A. Fuhrmann (Ed.), Mathematical Theory of Networks and Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 1984, pp. 120–129.

[31] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (1) (2012) 281–305.

[32] X. Li, W. Chen, W. Zhang, X. Gao, L. Hao, Acceleration of the development for motion planning algorithms using V-REP, in: 2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA), 2019, pp. 7–12, http://dx.doi.org/10.1109/WRC-SARA.2019.8931909.

[33] X. Liang, H. Cheng, RGB-d camera based 3D object pose estimation and grasping, in: 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 2019, pp. 1279–1284, http://dx.doi.org/10.1109/CYBER46603.2019.9066550.

[34] L. Jin, S. Li, H.M. La, X. Luo, Manipulability optimization of redundant manipulators using dynamic neural networks, IEEE Trans. Ind. Electron. 64 (6) (2017) 4710–4720, http://dx.doi.org/10.1109/TIE.2017.2674624.