

Article

Active Collision Avoidance for Robotic Arm Based on Artificial Potential Field and Deep Reinforcement Learning

Qiaoyu Xu *, Tianle Zhang , Kunpeng Zhou, Yansong Lin and Wenhao Ju

School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471023, China; ztl@stu.haust.edu.cn (T.Z.); kunpeng_1998@foxmaill.com (K.Z.); lys1390586510@163.com (Y.L.); juwenhao@stu.haust.edu.cn (W.J.)

* Correspondence: xiaoyu0622@163.com

Abstract: To address the local minimum issue commonly encountered in active collision avoidance using artificial potential field (APF), this paper presents a novel algorithm that integrates APF with deep reinforcement learning (DRL) for robotic arms. Firstly, to improve the training efficiency of DRL for the collision avoidance problem, Hindsight Experience Replay (HER) was enhanced by adjusting the positions of obstacles, resulting in Hindsight Experience Replay for Collision Avoidance (HER-CA). Subsequently, A robotic arm collision avoidance action network model was trained based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) and HER-CA methods. Further, a full-body collision avoidance potential field model of the robotic arm was established based on the artificial potential field. Lastly, the trained action network model was used to guide APF in real-time collision avoidance planning. Comparative experiments between HER and HER-CA were conducted. The model trained with HER-CA improves the average success rate of the collision avoidance task by about 10% compared to the model trained with HER. And a collision avoidance simulation was conducted on the rock drilling robotic arm, confirming the effectiveness of the guided APF method.

Keywords: robot collision avoidance; artificial potential field; deep reinforcement learning; hindsight experience replay



Citation: Xu, Q.; Zhang, T.; Zhou, K.; Lin, Y.; Ju, W. Active Collision Avoidance for Robotic Arm Based on Artificial Potential Field and Deep Reinforcement Learning. *Appl. Sci.* **2024**, *14*, 4936. <https://doi.org/10.3390/app14114936>

Academic Editor: Marco Troncossi

Received: 6 May 2024

Revised: 30 May 2024

Accepted: 4 June 2024

Published: 6 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the process of tunnel construction, workers often need to face an extreme construction environment, which brings a great challenge for the traditional tunnel construction methods, but also provides a broad practice space for intelligent construction and construction in tunnel engineering [1]. Meeting the complex requirements of desired tasks involves addressing the critical issue of collision avoidance for rock drilling robotic arms in intelligent construction processes. The issue of real-time collision avoidance path planning for a rock drilling robotic arm to evade two dynamic obstacles needs to be solved. How to ensure the real-time aspect of collision avoidance planning while also making sure that the planned actions allow the robotic arm to reach the target position, not just to avoid obstacles, is the main problem to be resolved.

The drilling robotic arm possesses seven degrees of freedom (DOF), and is computationally very expensive with higher DOFs and obstacles [2]. Existing algorithms mainly focus on planning a new collision-free path applications [3], which is not appropriate for real-time collision avoidance. However, the artificial potential field method and deep learning approaches are considered to solve this problem. The artificial potential field (APF) method, a local algorithm, can effectively generate full-body collision avoidance actions. Deep reinforcement learning (DRL) emerges as a promising approach, with its trained action models capable of global optima identification. But, each of these methods has a problem with real-time collision avoidance planning:

1. The artificial potential field has the problem of falling into local minima [4].

2. DRL training is inefficient because of the need to avoid multiple obstacles.

To actualize collision avoidance, a myriad of studies have been conducted, among which the research on artificial potential field (APF) emerges as a significant direction [5–9]. To solve the collision avoidance problem of the end-effector, it converts the distance between the end-effector of the robotic arm and the obstacle into a repulsive force [10,11]. Because in more scenarios it is necessary to ensure that the whole body of the robotic arm will not collide with the obstacle, the whole-body collision avoidance of the robotic arm has been considered in some studies. Li et al. [12] estimate the distance between the obstacle and the robot skeleton in real time using deep visual perception, thereby establishing a multi-joint repulsive force model. Additionally, there is also some research on robots without visual recognition. Control points are set on the robotic arm, and full-body collision avoidance is performed by calculating the repulsive force from the control point to the obstacle, and local minima are avoided by virtual obstacles [13,14]. However, the method of avoiding local minima through virtual obstacles requires entering the local minima position first, and after determining the position as local minima, setting the virtual obstacle to make the robotic arm leave the current position. This method is very inefficient for real-time motion planning. Generally speaking, there is almost no artificial potential field-based method that can provide real-time path planning to avoid local minimum positions.

To address the above problem reinforcement learning provides a new solution to this problem due to its global planning capability. In the study conducted by Cao et al. [15], a Deep Deterministic Policy Gradient (DDPG) algorithm is used for collision avoidance training for a three-degree-of-freedom robotic arm to solve the problem of collision of the robotic arm under static obstacles. Li et al. [16] used a combination of APF and DDPG to process the obstacle avoidance trajectory planning process in phases, designed the robotic arm artificial potential field method to approach the target, and then the obstacle avoidance phase set the DDPG dominant training, which improves the mobile robotic arms in the case of narrow passages and obstacle restraints Obstacle avoidance ability. For the collision avoidance motion planning of robotic arms, the exploration space is large, rewards are sparse, and the learning efficiency of the agent is relatively low. Andrychowicz et al. [17] were the first to propose Hindsight Experience Replay (HER), which enables an agent to learn efficiently from samples of sparse and binary rewards by setting up virtual targets. HER is widely used in path planning research, and scholars add HER to their methods to increase the training effect [18,19]. Further research has been conducted on the basis of HER, and improved algorithms have been proposed with some success. HER is considered to produce target bias in multi-target learning because of changing the likelihood of the sampled transitions and trajectories used in training. So, bias-corrected HER (BHER) was proposed to correct the bias [20]. The Hindsight Goal Ranking (HGR) method has been proposed to further improve the learning efficiency [21]. Although HER works well for the reach-target problem with binary rewards, it is not optimized for the collision avoidance problem.

To grapple effectively with these challenges, this paper presents an innovative action-based obstacle avoidance method that accomplishes active robotic arm collision prevention, guided by the potential field forces in tandem with DRL. The main contributions of this paper are as follows.

1. Propose a DRL-guided method for collision avoidance in a simplified robotic arm model. The local minima problem of APF is solved by DRL guidance, while the use of a simplified robotic arm reduces the difficulty of DRL training.
2. Proposed Hindsight Experience Replay for Collision Avoidance (HER-CA) algorithm to improve the training effect. The algorithm allows the agent to learn about collision avoidance from collision experience.
3. Full-body collision avoidance model for rock drilling robotic arm based on artificial potential field.

The rest of the paper is structured as follows. In Section 2, a simplified method for the robotic arm is shown first, followed by a Twin Delayed Deep Deterministic Policy Gradient

(TD3) and HER-CA based method. In Section 3, a full-body collision avoidance model based on APF is shown. In Section 4, we conducted experiments comparing HER-CA and Hindsight Experience Replay (HER), and simulation of robotic arm collision avoidance. Section 5 summarizes and concludes.

2. Deep Reinforcement Learning Method

2.1. Simplified Robotic Arm

In this study, the robotic arm is a 7-DOF rock-drilling robotic arm that has five slewing joints and two telescoping joints, developed by Gengli Machinery in Luoyang, China. The first three joints and the last three joints are in a spherical coordinate configuration in the robotic arm, connected by the fourth slewing joint. Figure 1 shows the coordinate structure of the robotic arm and Table 1 shows the D-H parameters of the robotic arm.

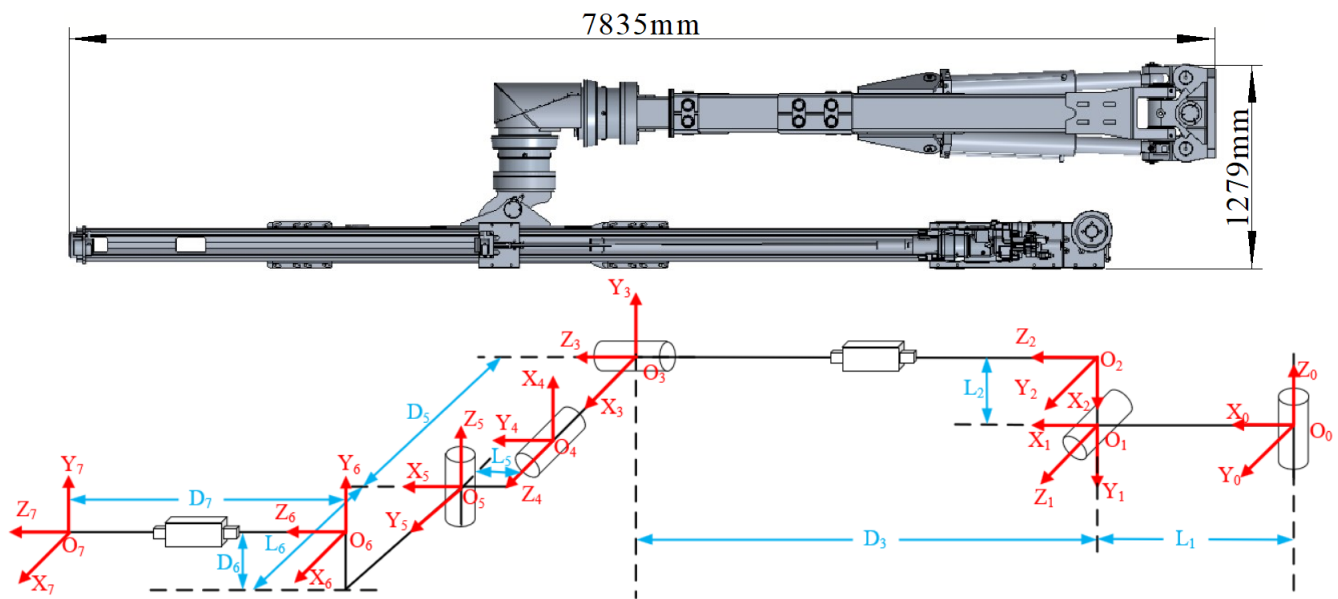


Figure 1. The structure and coordinate system of the robot.

Table 1. D-H parameters of 7-DOF rock-drilling robotic arm, q_1 – q_7 are the robotic arm joint values.

Joint	θ_i	D_i (mm)	α_i	L_i (mm)
1	q_1	0	$-\pi/2$	200
2	$\pi/2 + q_2$	0	$\pi/2$	−35
3	$\pi/2$	$4297 + q_3$	0	0
4	$\pi/2 + q_4$	0	$\pi/2$	0
5	$\pi/2 + q_5$	650	$\pi/2$	80
6	$\pi/2 + q_6$	679.5	$\pi/2$	241
7	0	$3058 + q_7$	0	0

The primary function of the deep reinforcement learning (DRL) method is to train an action model, which aids the robotic arm in successfully navigating around obstacles. In this scenario, the need is for a general directional obstacle avoidance, rather than a meticulous precision avoidance. When analyzing the structure of the rock drilling arm, the main body can be simplified as the upper arm paired with two segments of the propelling beam, represented as a cylinder Φ with the red line serving as the axis. The offset angle from Φ to the direction of the beam is denoted as θ_m , as illustrated in Figure 2.

Determine the obstacle coordinates for cylinder Φ in the joint space by leveraging the coordinates of the obstacle point in its operational space. Let the coordinates of the obstacle

in the operation space be x, y, z , and then the coordinates of the obstacle point in the joint space of the big arm are

$$q_{1obs} = \tan^{-1}\left(\frac{y}{x}\right) \quad (1)$$

$$q_{2obs} = \tan^{-1}\left(\frac{z}{x}\right) \quad (2)$$

where q_{1obs}, q_{2obs} are the positions of the obstacle points for the first two joints. The final simplified joint space obstacle points are obtained by performing one slewing of Φ according to the slewing joint angle q_4

$$x_{obs} = q_{1obs} + \theta_m \cos(q_4) \quad (3)$$

$$y_{obs} = q_{2obs} + \theta_m \sin(q_4) \quad (4)$$

where x_{obs}, y_{obs} are the coordinate value of the joint obstacle points after the simplification of the robotic arm.

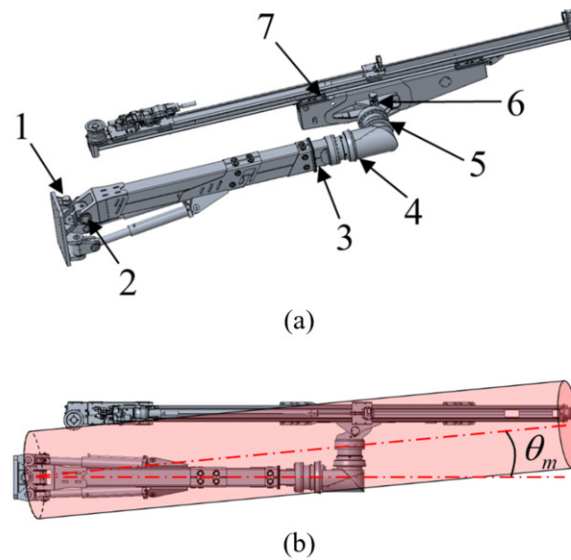


Figure 2. (a) Rock drill arm mod: 1—arm swing joint; 2—arm pitch joint; 3—arm telescoping joint; 4—beam slewing joint; 5—beam pitch joint; 6—beam swing joint; 7—beam telescoping joint. (b) Arm simplified.

2.2. Describe Environment

TD3 was first proposed by Fujimoto et al. [22], and it is a type of actor–critic method for deep reinforcement learning. This algorithm mainly solves the problems in the DDPG algorithm. TD3 learns two Q-functions instead of one, and then takes a more conservative (i.e., smaller) estimate of the Q-value as the target value. TD3's policy update is delayed compared to the Q-value update, in short, all of the above help to reduce overestimation bias in Deep Q-Learning [23]. The model is trained using the TD3 algorithm, and the environment is described below.

Under the context of the simplified operation of the robotic arm mentioned above, this obstacle avoidance planning issue can be recognized as a problem of planning a two-dimensional spatial path. In this research, two points of obstruction exist. The state begins by randomly locating the starting point, target point, and the two obstruction points. Following this, the state space s_t is designed as

$$s_t = \{x, y, x_{goal}, y_{goal}, x_{obs1}, y_{obs1}, x_{obs2}, y_{obs2}\} \in S \quad (5)$$

where S represents the entire state space and the other symbolism is shown in Table 2.

Define the action space as a_t by using Δx as the x-coordinate variation and Δy as the y-coordinate variation, and A represents the entire action space:

$$a_t = \{\Delta x, \Delta y\} \in A \quad (6)$$

Table 2. Description of state space elements.

Symbol	Representation
x	Current position x-coordinate
y	Current position y-coordinate
x_{goal}	Target position x-coordinate
y_{goal}	Target position y-coordinate
x_{obs1}	Obstacle 1 location x-coordinate
y_{obs1}	Obstacle 1 location y-coordinate
x_{obs2}	Obstacle 2 location x-coordinate
y_{obs2}	Obstacle 2 location x-coordinate

A binary reward function for arrivals and collisions is designed, while the step length is added as a penalty to the reward function in order to make the agent try to take the shortest path, and the reward function is $r_t(a_t, s_t)$:

$$r_{step} = -(|\Delta x| + |\Delta y|)k_{step} \quad (7)$$

$$r_{collision} = \begin{cases} -C_{collision}, & \text{if collision} \\ 0, & \text{if no collision} \end{cases} \quad (8)$$

$$r_{reach} = \begin{cases} C_{reach}, & x < C_{thr} \wedge y < C_{thr} \\ 0, & x > C_{thr} \vee y > C_{thr} \end{cases} \quad (9)$$

$$r_t(a_t, s_t) = r_{step} + r_{collision} + r_{reach} \quad (10)$$

where r_{step} is the step reward function, and k_{step} is the reward coefficient. Where $r_{collision}$ is the collision reward function, and $C_{collision}$ represents the reward value for which a collision occurs. r_{reach} is the arrival reward function, C_{reach} is the arrival reward value, and C_{thr} is the range of arrival determination.

2.3. Hindsight Experience Replay for Collision Avoidance

Hindsight Experience Replay (HER) is an experience pooling mechanism for improving the training efficiency of agents used in sparse and binary reward situations. The basic idea is that every experience of an agent still contains valuable information even if it fails to reach the goal. By treating a hindsight goal from failed experiences as the original goal, HER enables the agent to receive rewards frequently.

To improve HER for the collision problem, the path points are considered as target points in the HER algorithm while the obstacle points in the collision experience are moved. The location of the new obstacle point is at a random location, thus there is a high probability that the experience will become a collision-free experience. In this way, agent learns the collision avoidance ability from the experience of failed collision avoidance, and achieves the purpose of accelerating deep reinforcement learning training and improving the training effect, as shown in Figure 3.

There are two ways to keep obstacles away when using HER-CA, obstacle reset and obstacle move away. Obstacle reset is when an obstacle is randomly placed into the space as if resetting the environment, and obstacle move away is to move the obstacle away by x_{random} , y_{random} distance, where x_{random} , y_{random} are random numbers. We will perform a comparison of the two later in the course of the experiment.

In HER-CA we save the data recorded in each game as OP, save the data of one of the steps as OS, and use the future strategy [17] for goal conversion, i.e., converting a random step after the current position to the goal position, and using obstacle resets to generate possible collision-free experiences. See Algorithm 1 for a more formal description of the algorithm.

Algorithm 1 Hindsight Experience Replay for Collision Avoidance

```

1: Initialize: TD3
2: for  $epoch \leftarrow 1$  to  $MAX_e$  do
3:   Play a number of games and add the data from each game to the data pool PL.
4:   for  $trun \leftarrow 1$  to  $MAX_t$  do
5:     Randomly select a game OL from PL.
6:     Randomly select a step OS from OL.
7:     Get the future step coordinates  $x_{future}, y_{future}$ .
8:      $x_{goal} \leftarrow x_{future}, y_{goal} \leftarrow y_{future}$ 
9:     if OS collision then
10:       $x_{obsi} \leftarrow x_{obsi} + x_{random}, x_{random} \in [-40, 40]$ 
11:       $y_{obsi} \leftarrow y_{obsi} + y_{random}, y_{random} \in [-40, 40]$ 
12:      Limit  $x_{obsi}$  and  $y_{obsi}$  to 0 to 90.
13:     end if
14:     Recalculating OS's reward value.
15:     One network training using OS's data.
16:   end for
17:   Soft update of the network model.
18:   Save Action Models.
19: end for
  
```

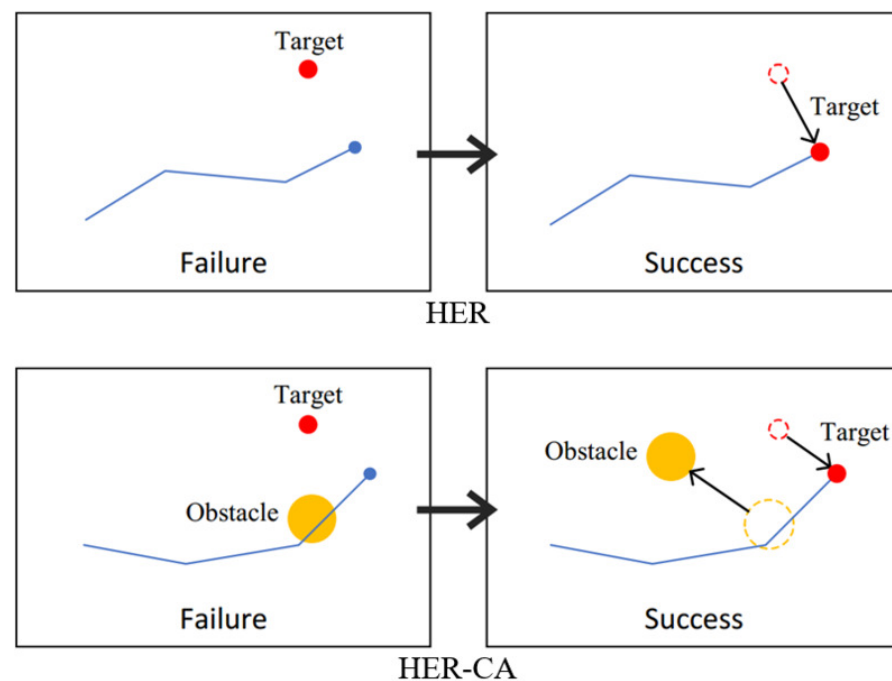


Figure 3. The figure on the left shows the original experience and the figure on the right shows the transformed experience.

3. Guided Artificial Potential Field

3.1. Attractive Force

The attractive force is utilized for guiding the rock drilling robotic arm to achieve the desired target position. The target orientation of the arm is determined by both the position

and attitude of its endpoint, which can be represented as a rotation matrix T_{goal} . In a bid to simplify the computation of the APF, we calculate the target angle vector Q_{goal} of the robotic arm's joints. This is achieved by resolving the inverse kinematics using an iterative method. Consequently, we compute the vector for the joint attractive force (or moment) by using the difference between the current joint angle vector of the robotic arm, that is, Q_{cur} and the target angle as showcased in (11):

$$M_{att} = K_{att} \odot \tan^{-1}((Q_{goal} - Q_{cur}) \odot K_{att}^r) \quad (11)$$

where M_{att} is the vector of attractive scaling factors, and the elements in K_{att}^r are the inverse of the corresponding elements of K_{att} . Equation (11) uses the inverse cotangent function $\tan^{-1}$ to limit the maximum value of the attractive force (or moment), preventing excessive speed of motion when the robotic arm is moving at a distance from the target position.

3.2. Repulsive Force

3.2.1. Control Point Repulsion

The role of the repulsive force is to move the arm away from the obstacle. In order to achieve full-body collision avoidance of the arm, the main structure of the arm is regarded as a collection of multiple control points (as shown in Figure 4), and the positions of the control points are derived by means of the positive kinematics of the robot. When an obstacle enters the radius of the repulsive field of a control point (as shown in Figure 5), the obstacle generates a repulsive force on the control point:

$$F_{rep} = \dot{\alpha}_f \left(\frac{1}{r} - \frac{1}{R_{field}} \right) \frac{1}{r^2} K_f \tan^{-1}(K_g \times g) \quad (12)$$

where F_{rep} refers to the vector of the repulsive force. The term r symbolizes the distance between the obstacle and the control point. R_{field} demarcates the radius within which the repulsive force field influences, while $\dot{\alpha}_f$ represents the direction vector that emanates from the obstacle and leads to the control point. Where g is the weighted average value representing the distance of the joint position from the target position, and K_f represents the proportionality coefficient of the repulsive force in the operation space. When obstacles are near the target position, the repulsive force exceeds the attractive force, which results in a target inaccessibility problem. Where $\tan^{-1}(K_g \times g)$ makes the repulsive force decrease when the distance from the target is closer, and solves the target inaccessibility problem.

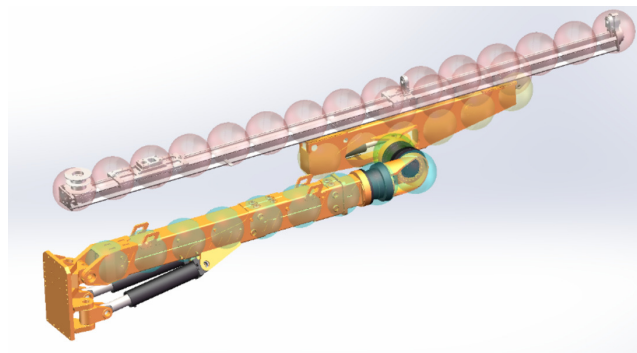


Figure 4. Schematic diagram of the control points of the rock drilling robotic arm.

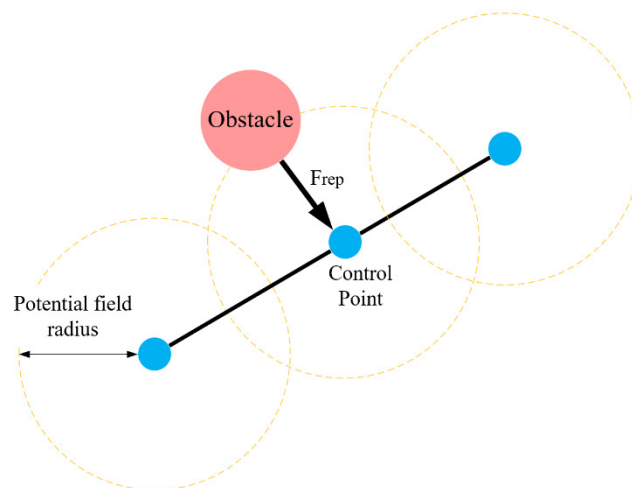


Figure 5. Schematic diagram of repulsive forces generated at control points.

3.2.2. Joint Space Repulsion

To operate a robotic arm, the production of joint force (or moment) is crucial, the repulsive force vector is converted into the robotic arm joint force (or moment) by the Jacobian method.

The end-effector positional state of the robotic arm (containing position and angle) can be regarded as a vector $\dot{p}$ and the joint value of the robotic arm can be regarded as a vector $\dot{q}$:

$$\dot{p} = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}, \dot{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} \quad (13)$$

where x, y, z represent the end-effector spatial position coordinates, and α, β, γ are the spatial attitude angles. Where $q_1 \sim q_7$ are the robotic arm joint values. Theoretically, any element in $\dot{p}$ can be calculated by $\dot{q}$, so there is (14)

$$\begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} f_1(\dot{q}) \\ f_2(\dot{q}) \\ f_3(\dot{q}) \\ f_4(\dot{q}) \\ f_5(\dot{q}) \\ f_6(\dot{q}) \end{bmatrix} \quad (14)$$

Each term on the left side of (15) is fully differentiated, and the right side is obtained by taking partial derivatives for q_1 to q_7 :

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \alpha \\ \Delta \beta \\ \Delta \gamma \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \dots & \frac{\partial f_1}{\partial q_7} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \dots & \frac{\partial f_2}{\partial q_7} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_6}{\partial q_1} & \frac{\partial f_6}{\partial q_2} & \dots & \frac{\partial f_6}{\partial q_7} \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \Delta q_6 \\ \Delta q_7 \end{bmatrix} \quad (15)$$

The vector on the left side of the equation represents the amount of change in the position of the end-effector denoted as ΔP , the matrix with 7 rows and 6 columns on the right side is the Jacobi matrix which we will denote as J , and the vector on the far right side of the equation represents the amount of change in the joints denoted as ΔQ , and (15) can be abbreviated as (16)

$$\Delta P = J \Delta Q \quad (16)$$

In the differential scenario, this equation can be seen as a mapping relationship that converts joint velocities into end-effector velocities. By numbering the control points, with n carrying the number of joints and m representing the ordering of the control points, the Jacobian matrix J_{nm} of the control points can be derived. In the differential case, where time Δt is consistently one, velocity is represented by force in this document. The three components of the vector $\vec{F}_{rep}$ are utilized as the initial three components of ΔP . Since the repulsive force does not produce torque on the control point so the last three elements of ΔP representing the angular velocity vector are θ , which gives ΔP . From this the joint space repulsive force (or moment) vector can be calculated.

$$M_{rep} = \text{pinv}(J_{nm}) \Delta P \odot K_{rep} \quad (17)$$

where M_{rep} signifies the joint repulsive force (or moment) vector. The function pinv stands for the pseudo-inverse of the matrix. Meanwhile, K_{rep} denotes the vector of repulsion scale factors.

3.3. Guidance

The issue of target unreachability is addressed during the computation of the repulsive force. The pre-established Action Model is employed to resolve the local minima problem and direct the movement of the robotic arm.

The target values for the first two joints, simplified points for joint space obstacles, and the current joint values of the robotic arm are fed into the Action Model to arrive at the subsequent position Q_{mod} for the two joints. This Q_{mod} is then utilized as the target values for the initial two joints to ascertain a temporary target position Q_{target} , which guides the robotic arm by generating an attractive force at the joints.

Since Action Model does not incorporate precise obstacle avoidance and only provides directional guidance, fewer training steps are necessary. The guidance can take place at short intervals. However, the proximity to the temporary target position may result in slower guidance. Thus, to expedite the process, the maximum attractive force is implemented when the arm is quite distant from the final target.

$$M_{att1} = C_{mattm} \frac{M_{att1}}{|M_{att1}|} \quad (18)$$

$$M_{att2} = C_{mattm} \frac{M_{att2}}{|M_{att2}|} \quad (19)$$

where M_{att1} represents the attractive moment value of the first joint, M_{att2} represents the attractive moment value of the second joint, and C_{mattm} represents the attractive moment maximum.

The rapid alteration of the temporary target position during the guidance process may precipitate a sudden change in the attractive moment, potentially resulting in a shock.

To circumvent such a situation, a limit is imposed on the rate of change of the attractive moment, as follows:

$$M_{att1} = M_{attold1} + C_{ali} \frac{C_{alr}(M_{att1} - M_{attold1})}{1 + C_{alr}|M_{att1} - M_{attold1}|} \quad (20)$$

$$M_{att2} = M_{attold2} + C_{ali} \frac{C_{alr}(M_{att2} - M_{attold2})}{1 + C_{alr}|M_{att2} - M_{attold2}|} \quad (21)$$

where M_{attold_1} , M_{attold_2} symbolize the attractive moment value at the previous point in time, and C_{ali} and C_{alr} are constants used to adjust the function.

At the end of Algorithm 2, L_{step} is utilized as a step factor. It multiplies the combined force (or moment) M to ascertain the joint action quantity designated as ΔQ .

Algorithm 2 Guided APF

```

1: while Run do
2:   if  $T_{goal}$  changed then
3:     Calculate the joint angle  $Q_{goal}$ .
4:      $Q_{target} \leftarrow Q_{goal}$ 
5:   end if
6:   Get the position of the obstacle points  $OP$ .
7:   Get the current joint value  $Q_{cur}$ .
8:   if  $\text{abs}(Q_{goal} - Q_{cur}) > Thr$  then
9:     Generate  $s_t$  as input to Action Moled, and take a step to get  $Q_{mod}$ .
10:     $Q_{target_1} \leftarrow Q_{mod_1}$ ,  $Q_{target_2} \leftarrow Q_{mod_2}$ 
11:    Use  $Q_{target}$  to calculate  $M_{att}$ .
12:    Amplify  $M_{att_1}$  and  $M_{att_2}$  to maximum.
13:   else
14:     Use  $Q_{goal}$  to calculate  $M_{att}$ .
15:   end if
16:   Attractive limits are imposed on  $M_{att_1}$  and  $M_{att_2}$ .
17:   Calculate the coordinates of the control points  $CP$ .
18:   for  $n \in CP$  do
19:     for  $m \in OP$  do
20:       if  $|nm| < R_{field}$  then
21:         Calculate the repulsive moment at these two points  $M_{rep\_tmp}$ .
22:          $M_{rep} \leftarrow M_{rep} + M_{rep\_tmp}$ 
23:       end if
24:     end for
25:   end for
26:    $M \leftarrow M_{att} + M_{rep}$ 
27:    $\Delta Q \leftarrow M \odot L_{step}$ 
28: end while

```

4. Experiments

In this segment, we implement three distinct experiments to substantiate the validity of the methodology delineated in this paper. Initially, a comparative analysis is performed between HER and HER-CA. This experiment serves to assess the efficacy of each algorithm when subjected to diverse parameters. To define the environment, we tapped into OpenAI's Gym library whereas network training was facilitated through the Pytorch library. The upcoming experiment is designed for the simulation of robotic arm collision avoidance. The purpose of these trials was to observe and evaluate the arm's ability to evade static as well as dynamic obstacles. MATLAB and CoppeliaSim served as the platforms on which these robotic arm collision avoidance simulations were executed.

4.1. Comparison of HER and HER-CA

Firstly, we configured the environment. We set the joint limits for the initial two joints of the robotic arm to $\pm 45^\circ$, and applied a 45° offset to them, creating a two-dimensional joint space where both dimensions ranged from 0 to 90° . During the environment's initialization, the starting point, obstacle point, and target point were indiscriminately positioned within the joint space. The path is judged to be a collision when it is within 3° of an obstacle, and a reach when it is within 2° of the target point. The actions given by the action model plus a Gaussian noise were explored for each game, and the maximum number of steps in each game was five. Both the action model and the value model use a fully connected neural

network with 3 hidden layers, each having 256 neurons. The action model uses the ReLU activation function and the network output is processed using the Tanh function with the learning rate set to 3×10^{-5} . The value model uses the ReLU activation function and the learning rate is set to 3×10^{-4} . All networks use the Adam optimizer.

Each epoch plays 200 games and stores the data from each game in the data pool, and then performs 20 network trainings. Each network training is randomly collected from the data pool for 50 steps, where each step has an 80% probability of being processed by HER or HER-CA. In addition a soft update of the network model is performed at the end of each epoch.

Data acquisition was achieved by testing preserved action network models. Specifically, each action model was saved at intervals of every 10 epochs, and the success rate of each model was ascertained by running 5000 games with every individual action model.

The initial experiment sought to evaluate the effectiveness between obstacle resetting and obstacle moving away in the context of using the HER-CA method. As depicted in Figure 6, using the move away strategy slightly outperforms the resetting method, exhibiting a marginally higher average and peak success rate.

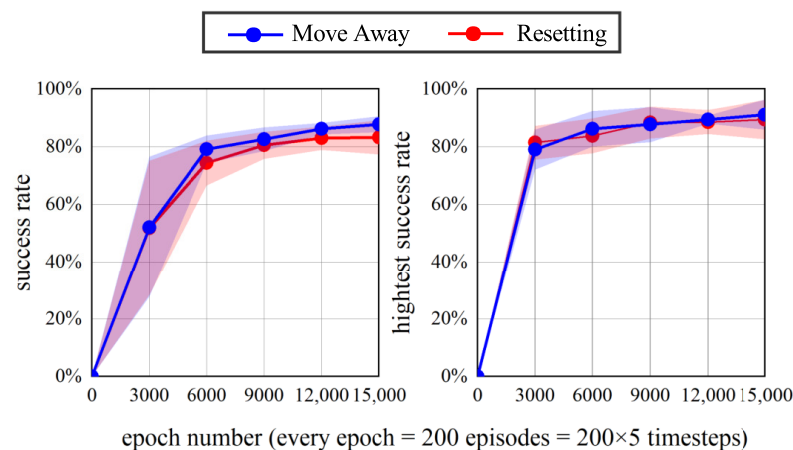


Figure 6. Comparison plot of training success rates for different collision-free experience generation methods. Blue represents obstacle reset red represents obstacle removal, and the shaded area represent one standard deviation.

Subsequent experiments compared the effects of HER and HER-CA, with setups involving one obstacle and two obstacles for model training scenarios. The experimental results are shown in Figure 7, which indicates that the model trained by HER-CA exhibits an average success rate improvement of about 10% compared to the model trained by HER.

We used the results of the success test for the full training process to calculate the average success rate, and the success rate for the 5000 epochs at the end of the training was used for interval estimation to calculate the confidence level. The average success rate improvement of HER-CA in the case of one obstacle was 9.4%, and the confidence level for an improvement greater than 5% was 71.1%. The average success rate of HER-CA in the case of two obstacles was improved by 10.1%, and the confidence level for an improvement greater than 5% was 88.5%.

The one-obstacle case HER-CA carried less effect enhancement than the two-obstacle case, and the reason for this phenomenon may be that one obstacle is inherently less difficult to avoid, leading to the fact that HER can also be trained slightly more efficiently.

In the third experiment we tested the performance of HER-CA in a 3D path finding task. The space is a cube space with a side length of 45, in which Obstacles and target points were randomly generated in a cubic space with coordinates from 3 to 45. The performance of the HER-CA and HER was tested for the 2 obstacles and 3 obstacles cases, respectively. A decaying learning rate is used, with an initial learning rate of 5×10^{-6} for the action model

and 5×10^{-5} for the value model, decaying by half every 20,000 epochs. The remaining parameters were the same as in the previous experiment.

The experimental results are shown in Figure 8. In the case of 2 obstacles, the average success rate of HER-CA improves by 20.1% compared to HER, and the confidence level of improvement greater than 10% is 82.7%. In the case of 3 obstacles, HER-CA has a maximum success rate of 90.9%, while HER could not be successfully trained, with a maximum success rate of only 29.7% at an epoch number of 9000.

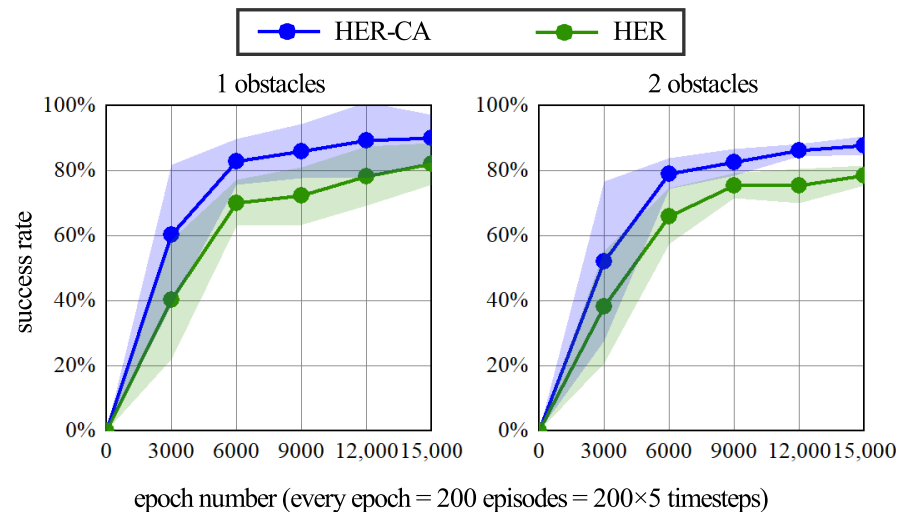


Figure 7. Comparison plot of training success rates for different numbers of obstacles, where blue and green correspond to HER-CA and HER, respectively.

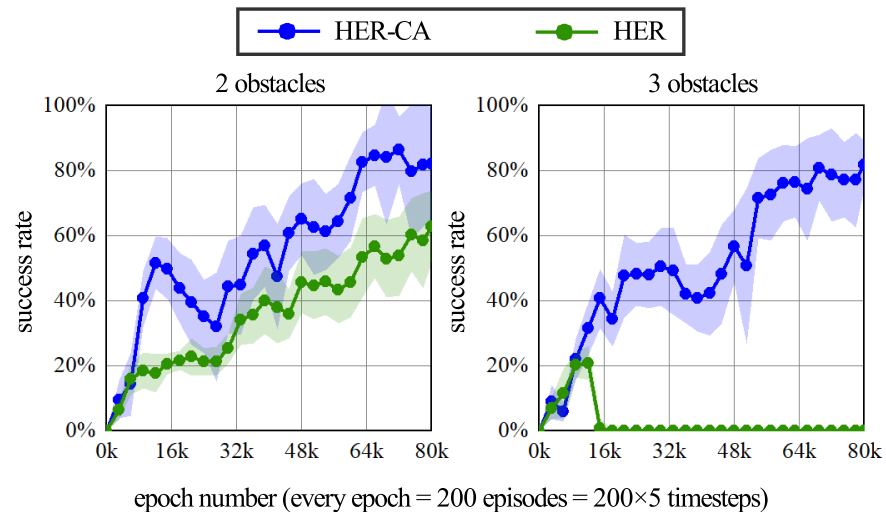


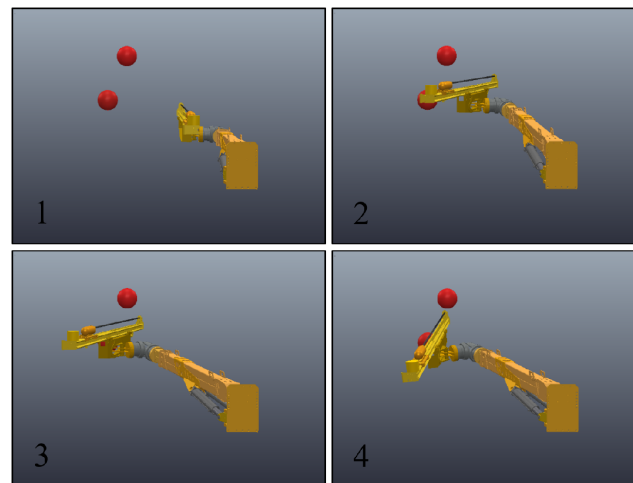
Figure 8. The comparison between HER and HER-CA in three-dimensional pathfinding tasks.

4.2. Static Obstacle Avoidance

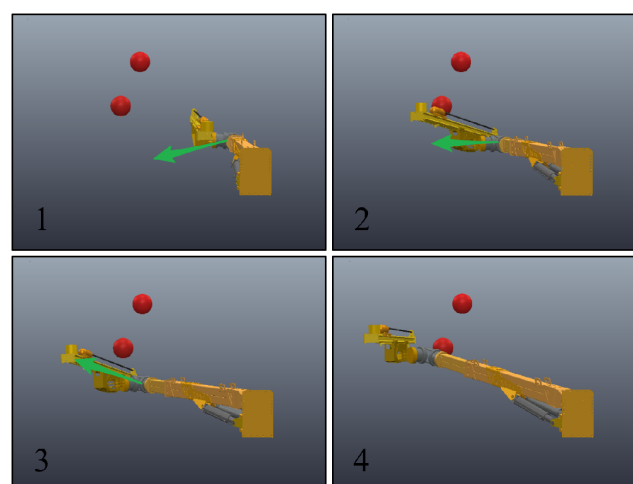
In the experimental paradigm, the robotic arm is tasked to navigate from its initial position to the target position while avoiding two stationary obstacles. The task information of the robotic arm and the definition of obstacles are shown in Table 3. Comparative experiments utilizing the APF method were also conducted. As shown in Figure 9, when using the APF (artificial potential field) method for collision avoidance, the robotic arm directly tries to move towards the target position. This leads to it getting stuck in a local minimum position, unable to reach the target point.

Table 3. Simulation information for robotic arms and obstacles.

Spherical Obstacle and Robot	Information
Start joint value of the robot arm	$[0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$
Goal position and attitude of the end-effector	$[8680 \text{ mm}, 4506 \text{ mm}, 1700 \text{ mm}, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$
Goal joint value of the robot arm	$[38.51^\circ, -9.71^\circ, 1515 \text{ mm}, 0^\circ, 9.71^\circ, -38.51^\circ, 1300 \text{ mm}]$
Coordinates of obstacle 1	$[6002 \text{ mm}, 3106 \text{ mm}, 1089 \text{ mm}]$
Coordinates of obstacle 2	$[6292 \text{ mm}, 2556 \text{ mm}, 2389 \text{ mm}]$
Obstacle radius	300 mm

**Figure 9.** The process of real-time collision avoidance movement using only the APF method. From 1 to 4 represents the process of the robotic arm from the beginning to the end of the simulation.

However, our guided APF method enables the arm to circumvent the two obstacles achieving its destined position, as shown in Figure 10. The closest proximity of the entire robotic arm from the obstacles is represented in Figure 11. A collision could potentially occur when this minimum distance reduces to less than 150 mm. However, in this experiment, the shortest distance is consistently greater than this collision threshold. The most minimal distance observed was 161 mm, occurring at 4.6 s.

**Figure 10.** Robotic arm motion process using our guidance APF method. The green arrow represents the direction of the temporary target pose given by the action model. From 1 to 4 represents the process of the robotic arm from the beginning to the end of the simulation.

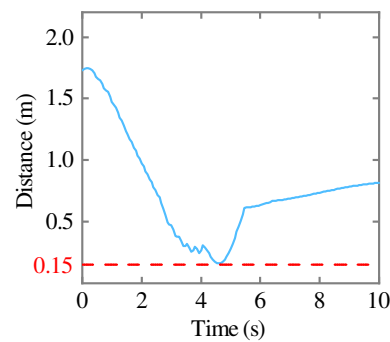


Figure 11. The closest distance between the robot and the obstacles. The red dotted line is the distance limit.

4.3. Dynamic Obstacle Avoidance

This experiment was conducted to corroborate the effectiveness of our novel active collision avoidance method in bypassing dynamic obstacles to reach the target position. During the experiment, we positioned the obstacle to move in the path of the robotic arm, while the arm was required to traverse from the starting position to the same final goal position as the previous experiment, avoiding any collision on its path. The initial positions of the two obstacles are in the upper left side of the beam, moving towards the lower right side at different speeds, and the obstacle information is shown in Table 4. As demonstrated in Figure 12, the robotic arm initially attempts to pass underneath the obstacles. When the obstacles move down and block the path, it changes its direction and moves upwards, passing through the gap between the two obstacles. In the end, the robotic arm successfully navigates around the dynamic obstacles and reaches its target position smoothly.

Table 4. Simulation information for dynamic obstacles.

Spherical Obstacle	Information
Coordinates of obstacle 1	[6002 mm, 3106 mm, 1089 mm]
Coordinates of obstacle 2	[6292 mm, 2556 mm, 2389 mm]
Velocity of obstacles 1	[0, −10, −30] (mm/iteration)
Velocity of obstacles 2	[0, −40, −10] (mm/iteration)
Obstacle radius	300 mm

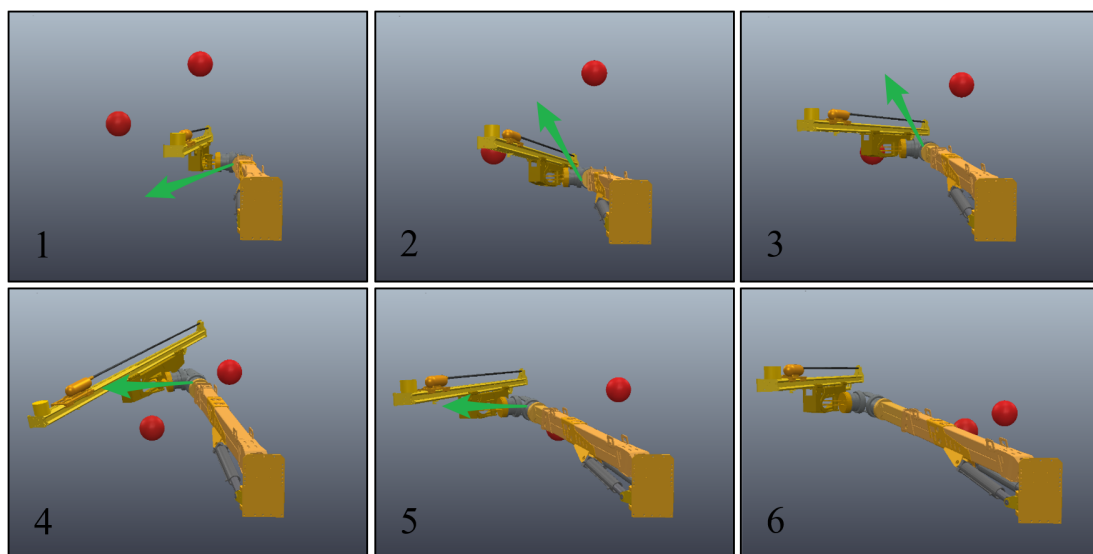


Figure 12. Robotic arm motion process using our guidance APF method. The green arrow represents the direction of the temporary target pose given by the action model. From 1 to 6 represents the process of the robotic arm from the beginning to the end of the simulation.

The change in each joint angle is shown in Figure 13. The closest proximity between the robotic arm and the moving obstacle during this collision avoidance process is 200.5 mm. This smallest measured distance occurred at the 3 s, which is graphically represented in Figure 14.

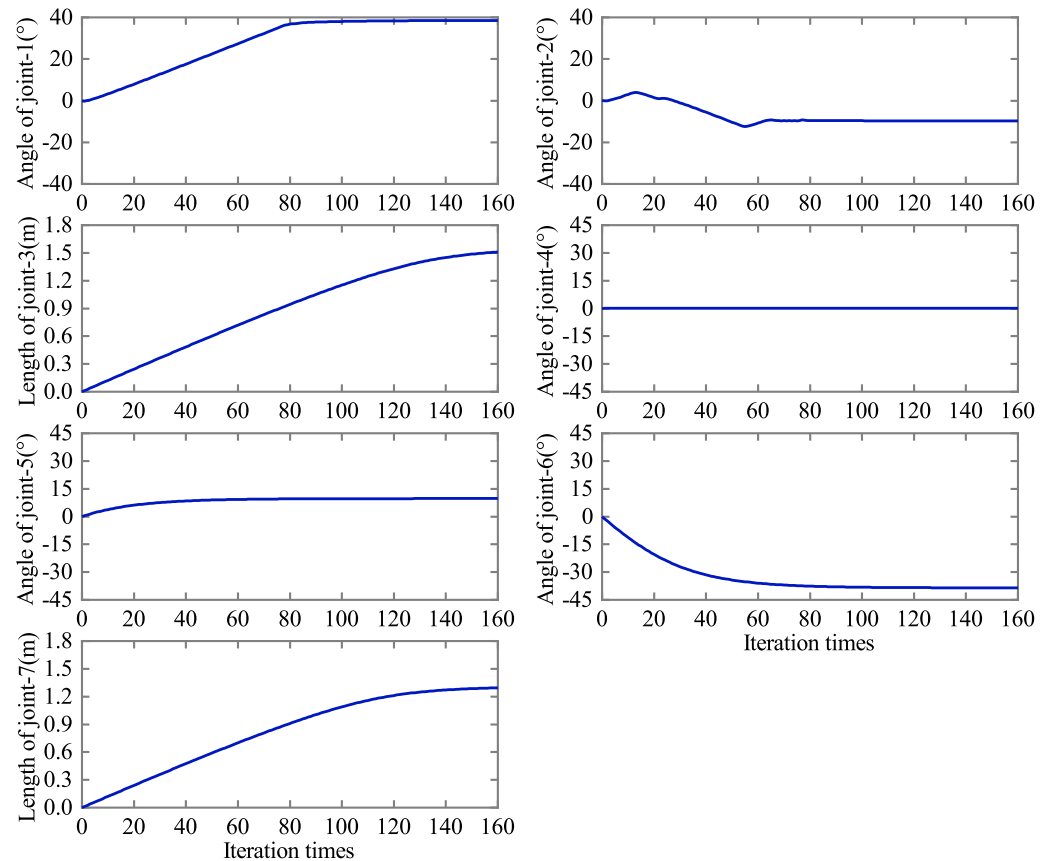


Figure 13. The change in the value of each joint.

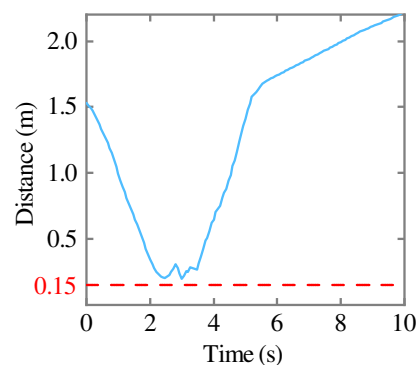


Figure 14. The closest distance between the robot and the obstacles. The red dotted line is the distance limit.

5. Conclusions

The HER-CA method is proposed, which has improved the success rate of the model trained with the HER method by about 10%. Then a development was proposed for robotic arm's collision avoidance, which combines deep reinforcement learning for guidance and APF for full-body collision avoidance. The algorithm enables the robotic arm to demonstrate higher intelligence during the dynamic collision avoidance process, as it can autonomously find a path to avoid local minima positions in real time, as shown in Figure 12. There

is uncertainty in the results of the movements generated by the neural network, which can lead to less smooth movements of the robotic arm. However, this does not detract from the possible application of the above methods to unmanned intelligent robotic arms. Applications under other robotic arms and avoidance of shaped obstacles have not been considered yet, and this will be our next research direction.

Author Contributions: Q.X.: conceptualization, writing—original draft, data curation, formal analysis, validation, visualization. T.Z.: writing—review and editing, conceptualization, data curation, formal analysis, validation. K.Z.: writing—review and editing, conceptualization, data curation. Y.L.: writing—review and editing. W.J.: writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, L.; Zou, H.; Liu, H.; Tu, W.; Chen, Y. Research Status and Development Trends on Intelligent Construction of Drill-and-BlastTunnels. *China J. Highw. Transp.* **2024**, 1–27. Available online: <https://kns.cnki.net/kcms/detail/61.1313.U.20231207.1324.002.html> (accessed on 3 June 2024).
2. Ananthanarayanan, H.S. Complete Path Planning of Higher DOF Manipulators in Human Like Environments. Ph.D. Thesis, University of Dayton, Dayton, OH, USA, 2015.
3. Zhang, H.; Jia, Y.; Guo, Y.; Qian, K.; Song, A.; Xi, N. Online Sensor Information and Redundancy Resolution Based Obstacle Avoidance for High DOF Mobile Manipulator Teleoperation Regular Paper. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 244. [CrossRef]
4. Song, D.; Qu, J. Particle Swarm Optimization Algorithm Based on Artificial Potential Field. In Proceedings of the 2nd International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018), Qingdao, China, 30–31 March 2018; pp. 591–594.
5. Yu, H.; Ning, L. Coordinated Obstacle Avoidance of Multi-AUV Based on Improved Artificial Potential Field Method and Consistency Protocol. *J. Mar. Sci. Eng.* **2023**, *11*, 1157. [CrossRef]
6. Wang, R.; Wang, J.; Wang, N. Research on path planning of mobile robot based on improved artificial potential field. In Proceedings of the 2015 Joint International Mechanical, Electronic and Information Technology Conference, Chongqing China, 18–20 December 2015; pp. 660–663. [CrossRef]
7. Song, J.; Zhao, M.; Liu, Y.; Liu, H.; Guo, X. Multi-Rotor UAVs Path Planning Method based on Improved Artificial Potential Field Method. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8242–8247. [CrossRef]
8. Liu, Z.; Wang, X.; Li, K. Research on path planning of multi-rotor UAV based on improved artificial potential field method. *MATEC Web Conf.* **2021**, *336*, 07006. [CrossRef]
9. Chen, Q.; Liu, Y.; Wang, P. An Autonomous Obstacle Avoidance Method for Dual-Arm Surgical Robot Based on the Improved Artificial Potential Field Method. In Proceedings of the 2022 International Conference on Intelligent Robotics and Applications, Harbin, China, 1–3 August 2022; pp. 496–508.
10. MIAO, T.t.; FANG, H.z.; AN, K.; SUN, Y.b.; FANG, Z.h.; GUAN, X.q. Obstacle avoidance planning algorithm of robot arm based on attraction factor of artificial potential field. *Comput. Eng. Des.* **2024**, *45*, 578–586. [CrossRef]
11. Liu, H.; Wu, J.; Fan, S.; Jin, M.; Fan, C. Integrated virtual impedance control based pose correction for a simultaneous three-fingered end-effector. *Ind. Robot* **2018**, *45*, 255–266. [CrossRef]
12. Li, H.; Wang, X.; Huang, X.; Ma, Y.; Jiang, Z. Multi-Joint Active Collision Avoidance for Robot Based on Depth Visual Perception. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 2186–2189. [CrossRef]
13. Chen, Y.; Chen, L.; Ding, J.; Liu, Y. Research on Real-Time Obstacle Avoidance Motion Planning of Industrial Robotic Arm Based on Artificial Potential Field Method in Joint Space. *Appl. Sci.* **2023**, *13*, 6973. [CrossRef]
14. Park, S.O.; Lee, M.C.; 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. [CrossRef]
15. Cao, Y.; Guo, Y.; Li, L.; Zhu, B.; Zhao, Z. Reinforcement Learning-based Trajectory Planning for Manipulator Obstacle Avoidance. *J. Mech. Transm.* **2023**, *47*, 40–46+96. [CrossRef]

16. Li, Y.; Zhang, C.; Chai, L. Collaborative obstacle avoidance trajectory planning for mobile robotic arms based on artificial potential field DDPG algorithm. *Comput. Integr. Manuf. Syst.* **2024**, 1–15. Available online: <https://link.cnki.net/urlid/11.5946.TP.20230920.0949.008> (accessed on 3 June 2024).
17. Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, P.; Zaremba, W. Hindsight Experience Replay. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
18. Tymoteusz, L.; Andrzej, M. Reinforcement Learning-Based Algorithm to Avoid Obstacles by the Anthropomorphic Robotic Arm. *Appl. Sci.* **2022**, *12*, 6629–6629. [[CrossRef](#)]
19. Evan, P.; MyeongSeop, K.; JaeHan, P.; JiHun, B.; JungSu, K. Path Planning for Multi-Arm Manipulators Using Deep Reinforcement Learning: Soft Actor-Critic with Hindsight Experience Replay. *Sensors* **2020**, *20*, 5911–5911. [[CrossRef](#)] [[PubMed](#)]
20. Bai, C.; Wang, L.; Wang, Y.; Wang, Z.; Zhao, R.; Bai, C.; Liu, P. Addressing Hindsight Bias in Multigoal Reinforcement Learning. *IEEE Trans. Cybern.* **2023**, *53*, 392–405. [[CrossRef](#)] [[PubMed](#)]
21. Luu, T.M.; Yoo, C.D. Hindsight Goal Ranking on Replay Buffer for Sparse Reward Environment. *IEEE Access* **2021**, *9*, 51996–52007. [[CrossRef](#)]
22. Fujimoto, S.; Van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv* 2018, arXiv:1802.09477.
23. Dankwa, S.; Zheng, W. Twin-Delayed DDPG: A Deep Reinforcement Learning Technique to Model a Continuous Movement of an Intelligent Robot Agent. In Proceedings of the ICVISIP 2019: 3rd International Conference on Vision, Image and Signal Processing, Vancouver, BC, Canada, 26–28 August 2019.

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