machines

Article

A Novel Planning and Tracking Approach for Mobile Robotic Arm in Obstacle Environment

Jiabin Yu [1,2,3,*], Jiguang Wu [1], Jiping Xu [1,2,3], Xiaoyi Wang [4], Xiaoyu Cui [1,2,3], Bingyi Wang [1] and Zhiyao Zhao [1,2,3]

[1] School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China; 2130062071@st.btbu.edu.cn (J.W.); xujp@th.btbu.edu.cn (J.X.); xiaoyucui@btbu.edu.cn (X.C.); 2330602074@st.btbu.edu.cn (B.W.); zhaozy@btbu.edu.cn (Z.Z.)
[2] Beijing Laboratory for Intelligent Environmental Protection, Beijing Technology and Business University, Beijing 100048, China
[3] China Food Flavor and Nutrition Health Innovation Center, Beijing Technology and Business University, Beijing 100048, China
[4] School of Arts and Sciences, Beijing Institute of Fashion Technology, Beijing 100029, China; wangxy@btbu.edu.cn
* Correspondence: yujiabin@th.btbu.edu.cn

Abstract: In this study, a novel planning and tracking approach is proposed for a mobile robotic arm to grab objects in an obstacle environment. First, we developed an improved APF-RRT* algorithm for the motion planning of a mobile robotic arm. This algorithm optimizes the selection of random tree nodes and smoothing the path. The invalid branch and the planning time are decreased by the artificial potential field, which is determined by the specific characteristics of obstacles. Second, a Fuzzy-DDPG-PID controller is established for the mobile robotic arm to track the planned path. The parameters of the PID controller are set using the new DDPG algorithm, which integrated FNN. The response speed and control accuracy of the controller are enhanced. The error and time of tracking of the mobile robotic arm are decreased. The experiment results verify that the proposed approach has good planning and tracking results, high speed and accuracy, and strong robustness. To avoid the occasionality of the experiments and fully illustrate the effectiveness and generality of the proposed approach, the experiments are repeated multiple times. The experiment results demonstrate the effectiveness of the proposed approach. It outperforms existing planning and tracking approaches.

Keywords: mobile robotic arm; planning; tracking; APF-RRT*; PID; DDPG; FNN

check for updates

Citation: Yu, J.; Wu, J.; Xu, J.; Wang, X.; Cui, X.; Wang, B.; Zhao, Z. A Novel Planning and Tracking Approach for Mobile Robotic Arm in Obstacle Environment. *Machines* **2024**, *12*, 19. https://doi.org/10.3390/machines12010019

Academic Editors: Miguel Gabriel Villarreal-Cervantes, Alejandro Rodríguez-Molina and Efrén Mezura-Montes

Received: 24 October 2023
Revised: 24 December 2023
Accepted: 27 December 2023
Published: 29 December 2023

1. Introduction

With the development of science and technology, mobile robotic arms have become a hot topic in the robotic research community. According to the information on the target object and environment acquired by sensors, the mobile robotic arm can autonomously grab the object, so they are widely used in many domains [1,2]. The grabbing task of a mobile robotic arm depends on two functions. First, the robotic arm plans the motion path for grabbing the object [3]; second, the robotic arm utilizes the controller to track the planned path [4]. However, there are many obstacles and environmental disturbances around the mobile robotic arm in the real-world environment. These factors have a certain impact on the performance of planning and tracking [5]. Therefore, it is very important to propose a novel planning and tracking approach for mobile robotic arms to grab objects in obstacle environments.

Many algorithms have been presented for the motion planning of robotic arms. These algorithms mainly include Dijkstra [6], A* [7], Artificial Potential Field (APF) [8], PRM [9], and RRT [10]. The RRT has simple and flexible structures, and it has been popularly used for path planning in robotic fields. It also has many variants [11,12]. Based on the RRT, the RRT* introduces the notion of cost. As the sampled waypoints approach infinity, their path converges to the optimal solution with the least cost [13]. However, slow planning speed and many invalid branches are also realized. To solve these problems, Gammell et al. [14] developed the informed-RRT*. It limits the search area in an ellipsoidal subset of the state space, and it can accelerate the search for the optimal path. Chi et al. [15] provide the Risk-RRT*. The quality of paths is improved by utilizing the RRT* in a time-based framework. The velocity of the two algorithms above is accelerated, and the length of paths is decreased. However, there are many turning points in paths when working in areas of obstacles. Yuan et al. [16] propose an APF-RRT algorithm for the motion planning of robotic arms. It introduces a heuristic method based on the number of adjacent obstacles to escape from local minima. It adopts an adaptive step, generates a virtual new node strategy to explore the path, and removes redundant path nodes. Jiang et al. [17] combine the RRT with the APF for path cutting and optimization. The planned path is short and smooth, and the success rate of planning is improved. The two APF-RRT algorithms above decrease path distance and planning time. Nevertheless, the shape and position of different obstacles are not considered; thus, many invalid branches are generated in paths.

Mobile robotic arms generally utilize proportional integral derivative (PID) controllers [18] for tracking the planned path. However, especially in obstacle environments, the presence of environmental disturbance can cause a low response speed and large errors in tracking [19–23]. To enhance the adaptability of tracking, the fuzzy neural network (FNN) [24] is also widely used for tracking robotic arms. In [25], a sliding mode control (SMC) method is associated with the FNN to constitute a robust control scheme to cope with the tracking error caused by environmental disturbance. Du et al. [26] use the self-feedback incremental FNN to predict and update the compensation in real-time. The deviation distance caused by environmental disturbance is reduced. Mai et al. [27] developed an intelligence PID controller. In this controller, an adaptive FNN approximator and an adaptive robust compensator are utilized to reduce the uncertainties of the system. The three FNN controllers above reduce tracking errors, but the response speed is not accelerated effectively. With the deep reinforcement learning (DRL) [28] technology developing, the DRL algorithm, such as deep Q-network (DQN) [29] and deep deterministic policy gradient (DDPG) [30], is utilized for real-time tracking of the robotic arm. Wang et al. [31] designed a DQN-based PID (DQN-PID) controller. This controller is employed for image-based visual servo (IBVS) control. It also solves the problem of feature loss and large steady-state error and improves the servo accuracy of mobile robotic arms. To further enhance the performance of grabbing robotic arms, Geng et al. [32] propose an improved DDPG algorithm based on fractional-order control. It enhances the accuracy and adaptability of tracking robotic arms using environmental dynamics modeling and grab posture optimization. Afzali et al. [33] present a new modified convergence DDPG (MCDDPG) controller to control robotic arms with high DOF. It shows a significant enhancement in training time and stability of the control. The three controllers above improve accuracy and speed of tracking, whereas poor robustness is also achieved.

To solve the problems mentioned above, we propose a novel planning and tracking approach for mobile robotic arms to grab objects in obstacle environments. First, we develop an improved APF-RRT* algorithm for motion planning of the mobile robotic arm. It optimizes the process of expanding random trees, and the path is simplified via APF. Second, a Fuzzy-DDPG-PID controller is established for tracking control of the mobile robotic arm. We develop a new DDPG algorithm in which the update function and loss function are improved, and the FNN with the optimized fuzzy rules is integrated. The Fuzzy-DDPG is utilized for tuning the parameters of the PID controller.

The detailed contributions of this approach are as follows:

(1) An improved APF-RRT* algorithm is developed. First, for RRT*, the node selection process is optimized, and the Dubins curve is used to smooth the path. The total turning angle is reduced while decreasing the path length. Second, an APF is used to generate artificial potential fields according to the shape and position of obstacles and

keep the random tree away from the obstacles. Therefore, the invalid path branches are reduced, and the speed of planning is accelerated;

(2) A Fuzzy-DDPG-PID controller is established. To promote the adaptiveness of path tracking of the robotic arm in an environment with disturbance, a Fuzzy-DDPG algorithm is integrated with the PID controller. First, to solve the problem of slow disturbance capture in DDPG, the update function and loss function are improved, and the policy-making speed is accelerated. Second, a FNN is proposed to solve the problem of the strong subjectivity of the DDPG online network. The membership function and fuzzy rules of FNN are optimized, and the online network of DDPG can be updated in real-time. Therefore, the response speed and policy-making accuracy are enhanced. By combining the Fuzzy-DDPG with the PID controller, the tracking time, time delay, and error of tracking are reduced. The robustness is enhanced.

The rest of the contributions of this study are organized as follows: we introduce the experimental platform in Section 2. Section 3 focuses on the technical details of the proposed approach. We compare and analyze the proposed approach with different approaches in Section 4. Finally, the works of this study are summarized in Section 5.

2. The Experimental Platform

The experimental platform called Turtlebot3-ARM is a 5-degree-of-freedom (DOF) mobile robotic arm. It is generally composed of a mobile platform, a robotic arm, a LiDAR, RGB-D cameras, and stepping motors. The robotic arm has five joints, and each joint is equipped with a motor. For the robotic arm, there is a motor between the mobile platform and the robotic arm, which is employed to control the horizontal rotation of the robotic arm. There is a motor at the bottom, middle, and top joint of the robotic arm, respectively. The three motors are utilized for stretching and retrieving the robotic arm. There is a motor at the gripper, and it is leveraged for gripping and loosening the target object. The processor frequency of Turtlebot3-ARM is 1.43 GHz, the RAM is 4 GB, and the ROM is 64 GB. Its operating system is called the robot operating system (ROS), and it is built on Ubuntu 16.04. The parameters of Turtlebot3-ARM are shown in Table 1. The robotic arm part of Turtlebot3-ARM is shown in Figure 1a, where yellow fonts denote the positions of the joints. The coordinate system of the robotic arm joint is presented in Figure 1b.

Table 1. The parameters of Turtlebot3-ARM.

Parameters	Definition	Numerical Value
w (rad/s)	Maximum angular velocity of motor	0.5
a_1 (rad)	Maximum angle of Joint 1	2π
a_2 (rad)	Maximum angle of Joint 2	π
a_3 (rad)	Maximum angle of Joint 3	π
a_4 (rad)	Maximum angle of Joint 4	π
a_5 (rad)	Maximum angle of Joint 5	0.5π
x (m)	Minimum x-axis length of obstacle	0.05
y (m)	Minimum y-axis length of obstacle	0.05
z (m)	Minimum z-axis length of obstacle	0.05
R_s (m)	Safety range of mobile robotic arm	0.05

According to the kinematics analysis description of the robotic arm in [34], the kinematic equation of the 5-DOF robotic arm we applied is as follows:

$$M(q)q'' + C(q, q')q' + g(q) = F, \tag{1}$$

where $q = [q_1, q_2, q_3, q_4, q_5]$ is the 5×1 vector of joint displacement, $q_i = (x_i, y_i, z_i) \in R_3$, $i \in [1, 5]$. R_3 is the coordinate set of feasible area in the environment. q' is the 5×1 vector of joint velocity, q'' is the 5×1 vector of joint acceleration, F is the 5×1 vector of the motor torque, $M(q)$ is the 5×5 symmetric positive definite inertia matrix of robotic arm,

$C(q,q')$ is the 5 × 5 matrix of centripetal and Coriolis torques, and $g(q)$ is the 5 × 1 vector of gravitational torques. The kinematics of the robotic arm mainly has three properties:

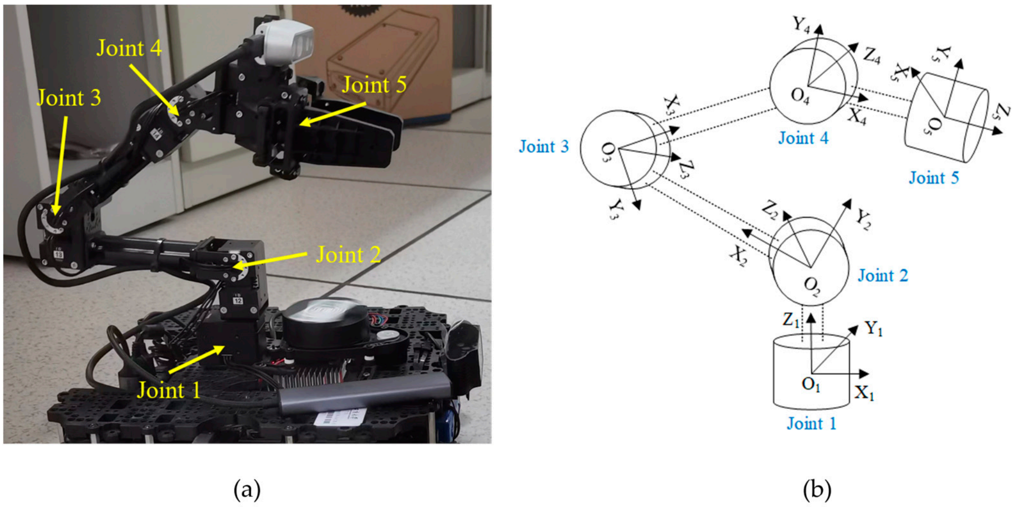


Figure 1. The robotic arm and its joint coordinate system (**a**) The robotic arm part of Turtlebot3-ARM and (**b**) The coordinate system of the robotic arm joint.

Property 1, the $C(q,q')$ and the $M(q)$ satisfy the following relationship:

$$(q')^T\left[\frac{1}{2}M'(q) - C(q,q')\right]q' = 0, \tag{2}$$

$$M'(q) = C(q,q') + C(q,q')^T. \tag{3}$$

Property 2, there exists a positive constant k_g, such that:

$$\|\frac{\partial g(q)}{\partial q}\| \leq k_g. \tag{4}$$

Property 3, there exists a positive constant k_c, such that:

$$\|C(x_i,y_i)z_i\| \leq k_c\|y_i\|\|z_i\|, i \in [1,5]. \tag{5}$$

After obtaining the dynamic matrix through the DDPG, k_g and k_c can be obtained using the following formula:

$$k_g = 5\left(\max_i\left|\frac{\partial g(q_i)}{\partial q_i}\right|\right), \tag{6}$$

$$k_c = 25\left(\max_{i,k}|C_k(q_i)|\right), \tag{7}$$

where $C_k(q_i)$ is the matrix of joint displacement torque.

3. The Proposed Approach

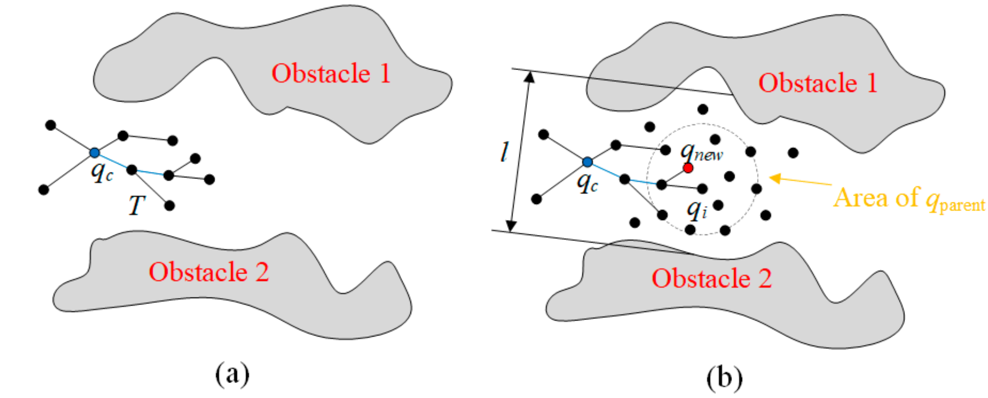
3.1. Planning Based on the Improved APF-RRT*

We develop an improved APF-RRT* algorithm as shown in Algorithm. First, in RRT*, the node-selecting process is optimized, and the Dubins curve [35] is applied to smooth the path. The path length and the total turning angle are reduced. Second, APF is combined with RRT*. It can generate artificial potential fields according to the shape and position of obstacles and keep the random tree away from the obstacles. The invalid and redundant path branches are cut, and the speed of planning is accelerated as well. The entire grabbing task of the robotic arm includes a stretching and retrieving process; we consider the two processes as one model to analyze the planning.

3.1.1. The Improved RRT*

For RRT*, our works are as follows:

As shown in Figure 2a, The random tree does not expand until the new nodes q_{new} are firmed. The current state of the random tree is T, and $q_c = (X, Y, Z)$ represents the center coordinate of the current state of the robotic arm joints.



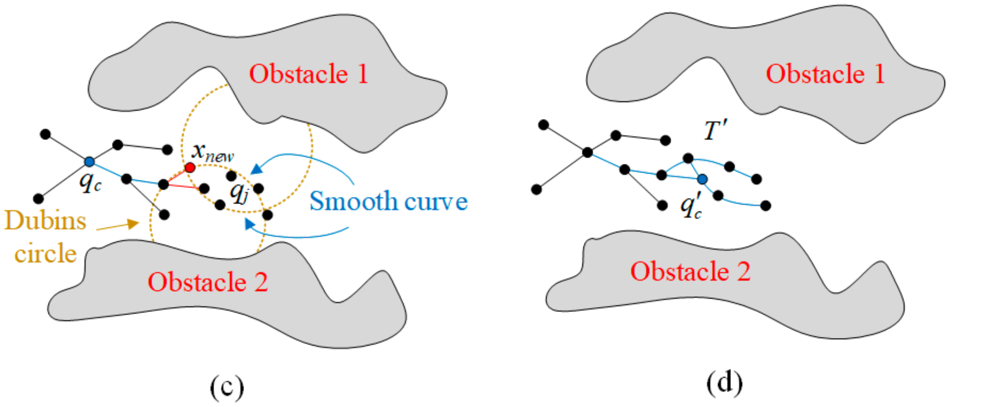


Figure 2. The principle of the improved RRT* (**a**) Current State of random tree, (**b**) Selection of q_{parent}, (**c**) Dubins smooth process and (**d**) New state of random tree.

When q_{new} has been firmed, the selection circle of q_{parent} is formulated based on the boundary point coordinates of the obstacle. The parent node q_{parent} and its nearest nodes are searched in this circle, and the radius of this circle is R. The process is shown in Figure 2b; the node set D for the candidate q_{parent} are as follows:

$$D = \{q_i || q_i - q_m| = R < l - 0.03\},\ i = 1, 2, \ldots, N, \tag{8}$$

where $q_i (i = 1, 2, \ldots, N)$ is the candidate parent node in D, q_m is the center node of D. To ensure the effect of obstacle avoidance, l is the distance between the two edge points of obstacles.

According to the difference in cost and steering angles of nodes in D, the Dubins curves are used to separate high-cost nodes in the forms of circles with different radii. The low-cost nodes are connected and smoothed simultaneously. This step is shown in

Figure 2c. After this step, the coordinate of the new random tree state for five robotic arm joints is confirmed. The center point C_k and radius r_k of separating circle is as follows:

$$C_k = \begin{cases} \frac{1}{5}\sum\limits_{j=1}^{5} q_j, 0^\circ \leq \angle q_{j_1}\overrightarrow{q}_{j_2} < 30^\circ \\ \frac{1}{5}\sum\limits_{j=1}^{5} q_j + \frac{q_{new}\left|q_{j_1} - q_{j_2}\right|_{\min}}{\sin\left(\frac{\theta_{\min}}{2}\right)}, \angle q_{j_1}\overrightarrow{q}_{j_2} \geq 30^\circ \end{cases}, \tag{9}$$

$$r_k = \left|q_{j_1} - q_{j_2}\right|_{\max} + 0.02, \tag{10}$$

where $q_j(j \in [1,5])$ is the candidate node for joints of robotic arm, $\theta(q_{j_1}, q_{j_2})$ is inclination angle between two nodes of q_j. If $0^\circ \leq \angle q_{j_1}\overrightarrow{q}_{j_2} < 30^\circ$, the nodes unseparated on circle C_k can be connected without smoothing, else they need to be smoothed. $\left|q_{j_1} - q_{j_2}\right|_{\min}$ is the minimum distance between two nodes of q_j, $\theta_{\min}$ is the minimum inclination angle between two nodes of q_j, and $\left|q_{j_1} - q_{j_2}\right|_{\max}$ is the maximum distance between two nodes of q_j.

After separation, according to the principle of path cost calculation of the D*Lite algorithm [35], a path cost function is utilized to calculate the minimum cost of the remaining nodes. Thereby, the nodes that meet the minimum cost are connected as a new random tree state T'. This step is shown in Figure 2d, and the path cost function is as follows:

$$c = \sqrt{2} \times \min(|X' - X|, |Y' - Y|, |Z' - Z|) + \frac{1}{5}|X - X|, \tag{11}$$

where $q'_c = (X', Y', Z')$ represent the center coordinate of a new state of the robotic arm joints.

3.1.2. Addition of APF

Based on the improved RRT*, we merged the APF to make the path as if entering an expanded obstacle area. The specific works of APF are as follows:

As shown in Figure 3a, there is no artificial potential field in the environment until the robotic arm starts stretching or retrieving. When the state of the random tree has changed, a repulsive potential field $U_{ri}(i = 1, 2, \ldots)$ is generated by the zone of obstacle i behind q_{new}, as shown in Figure 3b. The gravitational potential field U_a is generated by the target object or robotic arm base. The potential field equations are as follows:

$$U_a = d_{tar} k_1 \rho(q_{new}, q_{tar}) - \frac{1}{2} k_0 d_{tar}^2, \tag{12}$$

$$U_{ri} = \frac{1}{2}\mu_i \left(\frac{1}{\rho(q_{new}, q_{ci})} - \frac{1}{\rho(q_{ci})} \right), \tag{13}$$

where d_{tar} is the threshold of the gravitational potential field, q_{tar} is the coordinate of the target object or robotic arm base. k_0 is the invariable gravitational gain constant, k_1 is the variable gravitational gain constant, μ_i is the repulsive gain constant, q_{ci} is the center point of the passed zone in the obstacle i, $\rho(q_{new}, q_{ci})$ represents the distance vector from q_{new} to q_{ci}, and $\rho(q_{ci})$ represents the influence range of the obstacle i. As the random tree expands, U_a and U_{ri} will also change, and they are decided by k_1 and μ_i respectively. k_1 and μ_i are represented as follows:

$$k_1 = \frac{k_0 \sum\limits_{i=1}^{n} S_i}{\sum\limits_{i=1}^{n} \left|q_{ci} - q_{fi}\right|}, \tag{14}$$

$$\mu_i = \frac{S_i}{\|q_{ci} - q_{ni}\|}, \tag{15}$$

where S_i, $i \in [1,2]$ is the area of the zone that the random tree has passed through in obstacle i, and S_i is observed by the RGB-D camera on the robotic arm. q_{fi} is the farthest point from q_{ci} in S_i, and q_{ni} is the closest point from q_{ci} in S_i.

Algorithm: Improved APF-RRT*

```
Input: Process (stretch, retrieve), obstacle area $S_i$, current state $T$
Output: Path($T$, $q_{new}$, $q_{min}$)
Step 1:    $V \leftarrow q_i$;
Step 2:    $i \leftarrow 0$;
Step 3:    Motion ← Process(stretch, retrieve)
Step 4:    For $i$ = 1:N do
Step 5:        $q_{rand} \leftarrow$ Sample($q_{init}$, $T$);
Step 6:        $i \leftarrow i + 1$;
Step 7:        $q_{nearest} \leftarrow$ Nearest($q_{rand}$);
Step 8:        $q_{new} \leftarrow$ Steer($q_{nearest}$, $q_{tar}$);
Step 9:        if CollisionFree($q_{nearest}$, $q_{new}$, $S_i$) then
Step 10:           $q_{nearest} \leftarrow$ Nearest($q_{new}$, $D$);
Step 11:           $q_{parent} \leftarrow$ Parent($q_{nearest}$, $R_k$);
Step 12:       For $j$ = 1:M do
Step 13:           if ExcludeDubinsCircle($q_{new}$) then
Step 14:               $q_{new} \leftarrow$ PlanNode($q_{parent}$, $U_a$, $U_r$);
Step 15:               Cost($q_{min}$) = Cost($q_{nearest}$) + Cost($q_{new}$, $d$);
Step 16:           $T' \leftarrow$ Path($T$, $q_{new}$, $q_{min}$);
Step 17:       Return $T'$;
```

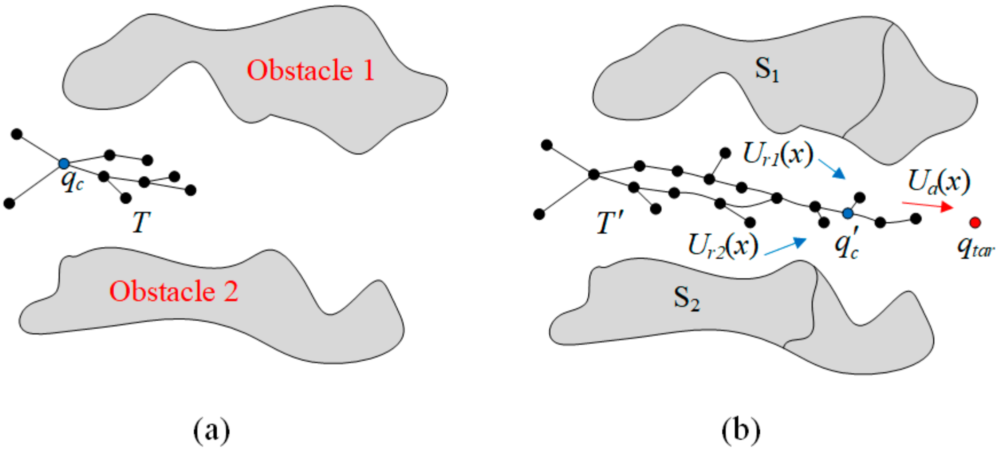


Figure 3. The principle of the improved APF-RRT* (**a**) The state without APF and (**b**) The state with APF.

3.2. Tracking Based on the Fuzzy-DDPG-PID

A Fuzzy-DDPG-PID controller is established in this study. It can tune the parameters of PID and reduce tracking errors. As shown in Figure 4, where ω is the angular velocity vector of the motor, $\psi(\omega)$ is the input variable, $\psi\prime(\omega)$ is the output variable, $c(\omega)$ is the control instruction output by the PID controller, $d(\omega)$ is the disturbance function, K_p is the proportional parameter, K_i is the integral parameter, and K_d is the differential parameter.

For the Fuzzy-DDPG algorithm, our work mainly includes two parts. First, for the DDPG, the update function and loss function are improved to accelerate the policy-making speed of tracking control. Second, the membership function and fuzzy rules of FNN are optimized. The FNN is utilized to update the DDPG's online network in real-time. Therefore, the response speed and control accuracy of the controller are enhanced.

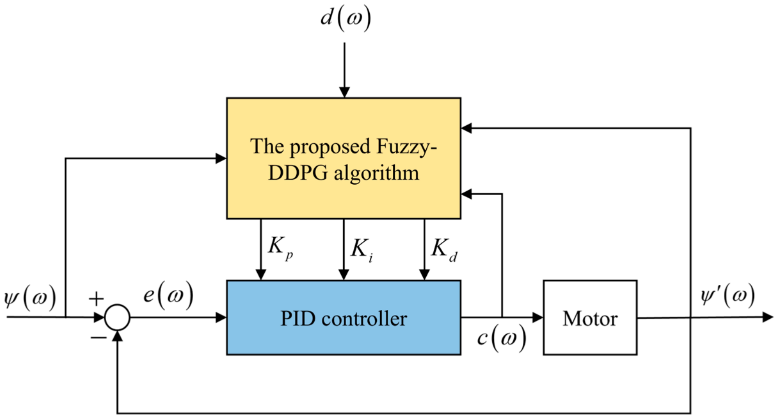


Figure 4. The principle of the proposed Fuzzy-DDPG-PID controller.

3.2.1. DDPG Algorithm

The DDPG can analyze the states of the system via the Actor–Critic construction and then make policies to determine the parameters of PID. For the DDPG, our specific works are as follows:

In DDPG, the robotic arm states that T and T' are important factors. They are utilized for determining the action of the mobile robotic arm, and they are functions of the robotic arm joint coordinates $q_i,\ i \in [1,5]$. The DDPG we proposed is shown in Figure 5.

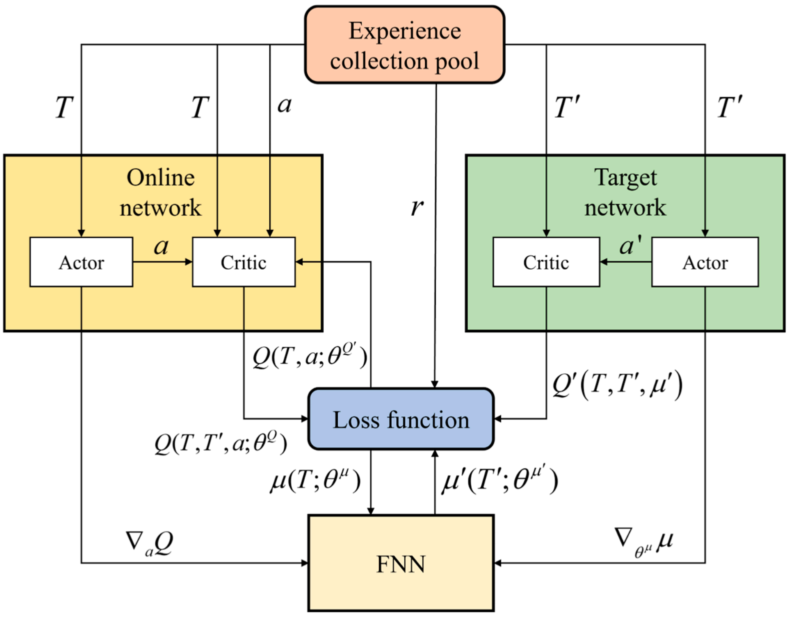


Figure 5. The principle of the Fuzzy-DDPG.

In the online Actor network, we have considered the rewards r from T to T'. The r is executed and acquired by action a. The a is the function of $M(q)$, $C(q,q')$ and $g(q)$ in Equation (1), and the r is the function of q in Equation (1). The update function $\mu(T;\theta^{\mu})$ of the environmental state, which is based on the gradient-rise method, is used as follows:

$$\nabla_{\theta} J = E_s[\nabla_a Q(T,r,a;\theta^{Q}) \times \nabla_{\theta}\mu(T;\theta^{\mu})], \tag{16}$$

where E_s is the parameter selection function to minimize network error. θ^{μ} and θ^{Q} are the parameters of the online Actor network and the Critic network, respectively, and Q is the value function acquired in the online Critic network.

In the online Critic network, the evaluation of the T' is fully considered, and the gradient of the online Critic network is as follows:

$$\nabla_{\theta Q} = E[(r + \nabla_{\theta}\gamma Q'(T, T', \mu'(T; \theta^{\mu'})) - Q(T, a; \theta^{Q'})) \times \nabla_{\theta} Q(T, T', a; \theta^{Q})], \quad (17)$$

where μ' is the new environmental state from the target Actor network. Q' is the value function acquired in the target Critic network, and it is utilized to determine the parameters of the PID controller. $\theta^{\mu'}$ and $\theta^{Q'}$ are the parameters of the target Actor network and Critic network, respectively, and $\gamma \in (0, 1)$ is the parameter of the target Critic network.

We improve the loss function of DDPG based on the updated results of FNN; the loss function $L(\theta^f)$ is as follows:

$$L(\theta^f) = \frac{1}{N}\sum_i [\alpha F(T, T'; \theta^c) - (1-\alpha)\mu(T, r; \theta^{\mu})]^2, \quad (18)$$

where θ^f is the Critic network parameter selected by fuzzy rules, F is the value function of the Critic network calculated using DDPG and FNN, and $\alpha \in (0, 1)$ is the loss parameter.

3.2.2. The Addition of FNN

DDPG needs to determine the next state of the mobile robotic arm according to expert experience. However, due to the state transformation of the environment not being considered, the response speed is slow, and environmental parameters are difficult to obtain in DDPG. Therefore, we added FNN to provide updated parameters for the online network of DDPG in real time.

As shown in Figure 6, FNN consists of four layers. The first layer is the input layer, whose nodes correspond to input variables $x_i, i \in [1, N]$. After this layer, $x_i, i \in [1, N]$ are converted to fuzzy variables. The second layer is the membership layer, and its corresponding function A_{ij} is employed to quantify the dimension of membership for input variables. The third layer is the fuzzy rule layer. φ_j, $j = 1, 2, \ldots, L$ represents the truth value of rules. w_{jk} is the weight of the consequent part. The last layer is the output layer. Its nodes correspond to output variables y_i, $i \in [1, M]$. In this study, the input variable x is $\mu(T; \theta^{\mu})$, and the output variable y is the update function $\mu'(T; \theta^{\mu'})$ integrated into the kinematical equation of Equation (1).

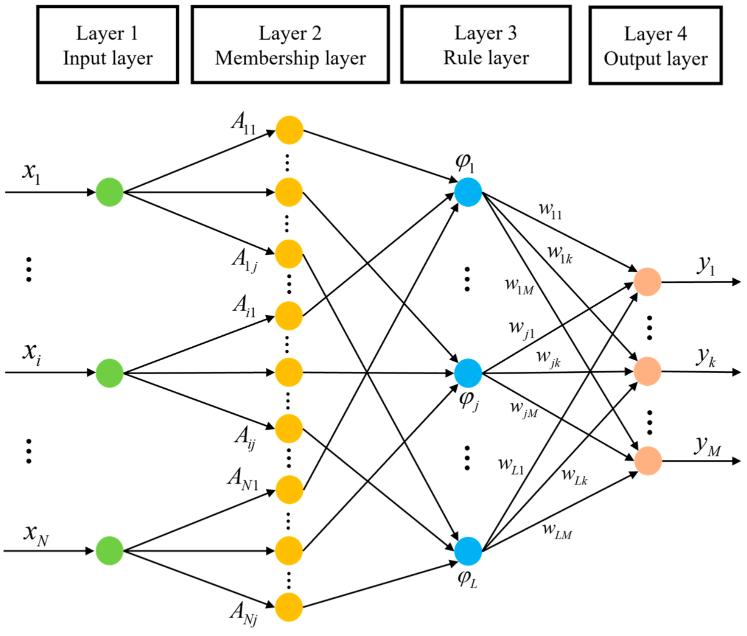


Figure 6. The structure of FNN.

For FNN in this study, we consider the state transformation of the environment, the membership function, and the fuzzy rule, which are optimized to make the update parameters of the DDPG online network provided by FNN more accurate. Thereby, the response speed and control accuracy are improved. The equations of the improved membership function $A_{ij}(x_i)$ and fuzzy rule $\varphi_j(x_i)$ are as follows:

$$A_{ij}(x_i) = \exp\left[-\frac{(x_i - c_{ij} - \overline{c})^2}{\sigma_{ij}^2}\right], \tag{19}$$

$$\varphi_j(x_i) = A_{ij}(x_i) \times \frac{j}{2} \times k, \tag{20}$$

where c_{ij} and $\overline{c}$ are the center and average values of $A_{ij}(x_i)$ respectively.

4. Experiments and Analysis

To demonstrate the effectiveness of the proposed approach, we conduct real-world object-grabbing experiments of the mobile robotic arm in an obstacle environment. We place some bottles as obstacles in a real-world environment, and one of these bottles is used as the target object to be grabbed. As shown in Figure 7, the information on the obstacle environment and target object are acquired via the RGB-D camera of the mobile robotic arm, and then the grabbing task of the mobile robotic arm is performed. The entire grabbing task can be divided into the stretching and retrieving process. The algorithm and control module of this robotic arm are realized based on the MoveIt development platform, in which the real-time states of the robotic arm are also acquired, as shown in Figure 8. We present the experimental results of motion planning and tracking control for the entire grabbing task, and the performance of the proposed approach is compared with other approaches, as shown in Sections 4.1 and 4.2.

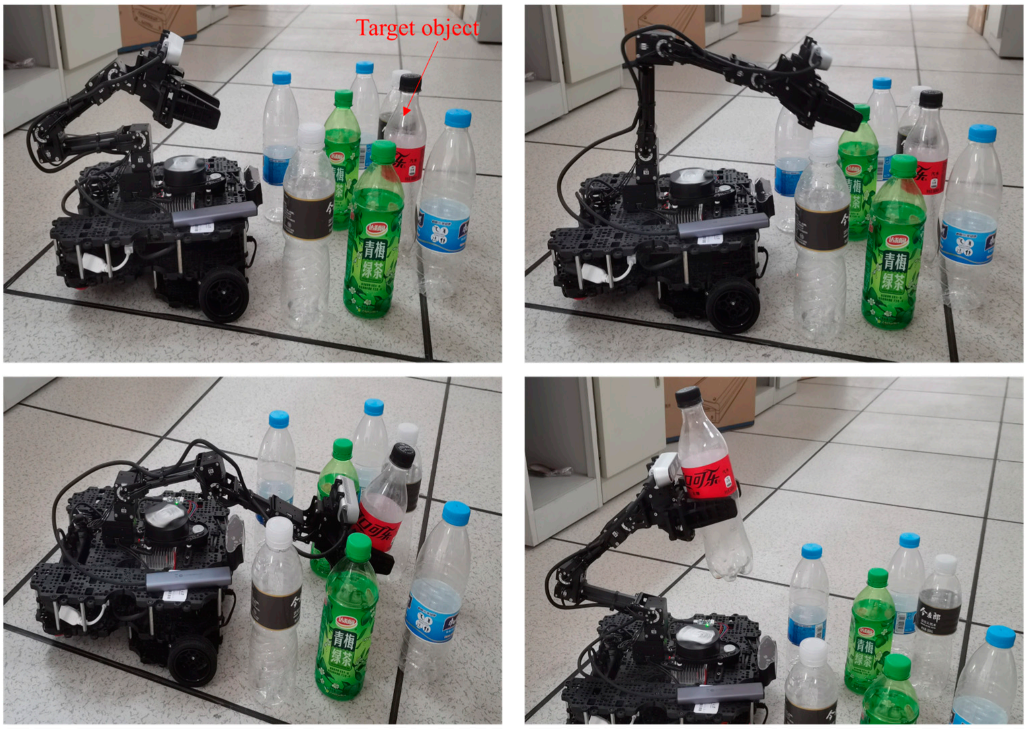


Figure 7. The entire grabbing task in real-world environment.

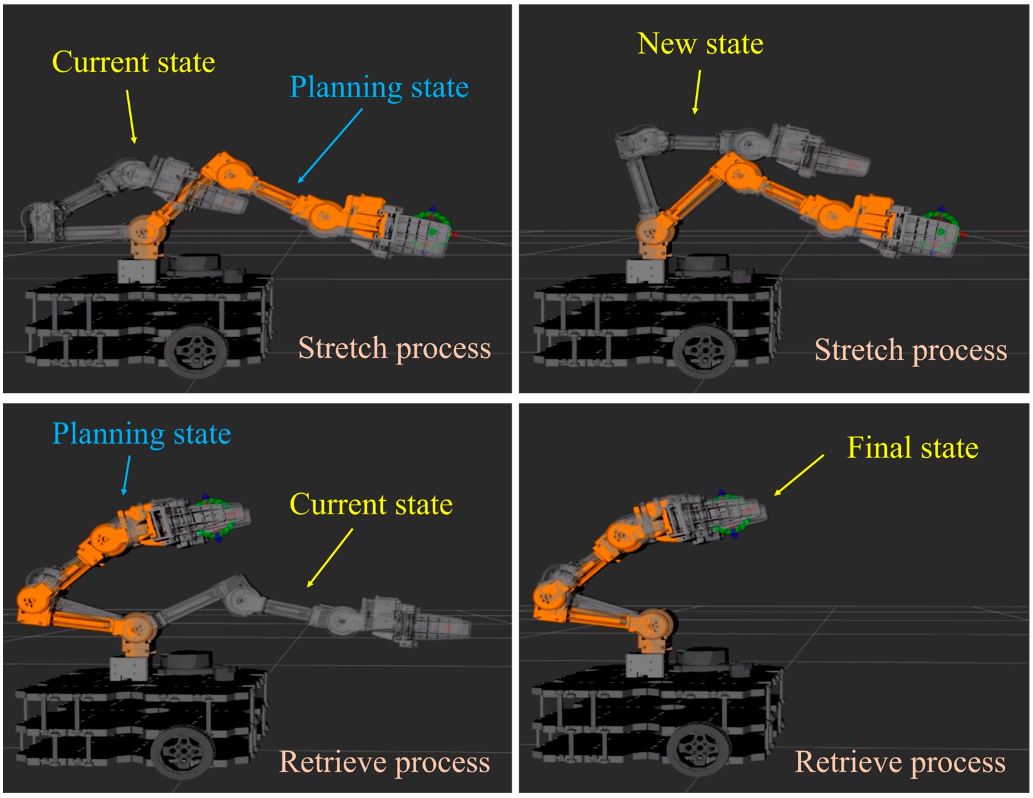


Figure 8. The real-time state of robotic arm in MoveIt.

4.1. Experiments and Analysis of Planning

To verify the availability of the improved APF-RRT* in motion planning, we make planning comparisons of four different algorithms. Algorithm 1 represents the traditional RRT*, Algorithm 2 denotes the improved RRT* in Section 3.1.1, Algorithm 3 combines the traditional APF with Algorithm 2, and Algorithm 4 is the improved APF-RRT* proposed in this study. The coordinates of the robotic arm base are set as (−0.135, −0.147), and the coordinates of the target object are set as (0.152, 0.139). We leverage some bottles to construct the obstacle environment, and the structure of the environment can be referred to in Figure 7. The mobile robotic arm plans the path according to the environmental information and simultaneously tracks the planned path to execute the grabbing task. The results of planning are shown in Table 2.

Table 2. The planning results for target object grabbing.

Algorithm Name	Path Length (m)	Total Steering Angle (°)	Branch Account	Planning Time (s)
Algorithm 1	0.4312	95.32	34	12.64
Algorithm 2	0.3264	75.21	30	11.75
Algorithm 3	0.2771	61.54	23	10.44
Algorithm 4	0.2355	54.85	15	9.31

First, through the planning results of Algorithms 1 and 2, the effect of the improved RRT* is analyzed. Compared to Algorithm 1, since the parent node selection process is optimized and the path is smoothed, the short path and small total steering angle are realized. The results illustrate that the improved RRT* is superior to the traditional RRT*. Second, through the planning results of Algorithms 3 and 4, we analyze the motion planning effect of the APF developed in this study. Compared to Algorithm 3, the APF integrated into Algorithm 4 has been optimized by incorporating obstacle information. It significantly reduces invalid branches and planning time. The results demonstrate that the APF developed in this study is superior to traditional APF. By synthetically analyzing Algorithms 1 to 4, the improved APF-RRT* can reduce path length and total steering angle, and the invalid branches and planning time can also be decreased. It has good motion planning performance.

To avoid the occasionality of the experiment result above, we repeat the planning experiment three times. In the repetitive experiments, the coordinates of the robotic arm base, the target object, and the environmental layouts are changed. The three time planning results for target object grabbing are shown in Table 3.

Table 3. The three time planning results for target object grabbing.

Coordinates	Algorithm Name	Path Length (m)	Total Steering Angle (°)	Branch Account	Planning Time (s)
(0.142, 0.137) (−0.145, −0.134)	Algorithm 1	0.4134	90.43	31	13.08
	Algorithm 2	0.3329	74.55	26	12.59
	Algorithm 3	0.2726	62.13	21	11.97
	Algorithm 4	0.2173	50.77	13	11.05
(−0.138, 0.143) (0.155, −0.149)	Algorithm 1	0.4527	97.66	40	14.62
	Algorithm 2	0.3591	78.73	35	13.29
	Algorithm 3	0.2864	64.35	27	12.47
	Algorithm 4	0.2485	56.31	16	10.84
(0.133, −0.156) (−0.148, 0.143)	Algorithm 1	0.4297	94.65	32	12.47
	Algorithm 2	0.3362	76.31	24	11.77
	Algorithm 3	0.2538	60.87	17	10.95
	Algorithm 4	0.2253	51.94	10	9.48

4.2. Experiments and Analysis of Tracking

The mobile robotic arm also tracks the planned path in real-time when performing motion planning. To verify the tracking performance of the proposed Fuzzy-DDPG-PID controller, we make comparisons on four different controllers. Controller 1 is the traditional PID controller. Controller 2 combines the PID controller with the DDPG in [33]. Controller 3 integrates the conventional FNN with Controller 2. Controller 4 is the proposed Fuzzy-DDPG-PID controller. We let these controllers uniformly track the paths planned via the improved APF-RRT*. To fully prove the different effects of these controllers, we provide a comparison of the step response performance of each joint motor, and the entire tracking process performance of the four controllers is presented.

As described in Section 2, the robotic arm used in this study is a 5 DOF robotic arm, and each joint is equipped with a motor. Because the impact caused by the controller's different types is negligible for joint 5, we provide a comparison of the step response performance of joints 1 to 4. As shown in Figures 9 and 10, we cut out the step response curves from the stretching and retrieving process of the robotic arm, respectively. From Figures 9 and 10, the step response trends of the four motors corresponding to the four controllers are not the same in the environment with disturbance. For example, some exhibit that Controller 1 achieves no overshoots, while others acquire. However, without the participation of DDPG, the parameters of PID are difficult to be tuned in Controller 1. Thereby, the delayed response time and large overshoots are acquired. The considerable deviation of steady-state values from the optimal steady-state values is also realized. Since the parameters of PID are tuned via DDPG, the response time of Controller 2 is advanced, and the steady-state values are closer to the optimal steady-state values. However, the online network cannot obtain the relation between environmental transformation and robotic arm motions, and it leads to low policy-making accuracy of DDPG. Therefore, the overshoots of Controller 2 cannot be significantly reduced. Because the improved DDPG utilizes FNN to update the environmental state of the online network, Controller 3 reaches early response time, and overshoots are reduced, but the effect is not remarkable. With the new DDPG in Controller 4 integrated with optimized FNN, the relation of the environmental transformation and robotic arm motions can be fully acquired, so the response time is earlier than that of others, and the overshoots are significantly reduced. Its steady-state values are close to the optimal steady-state values. The experimental results illustrate that the proposed Fuzzy-DDPG-PID controller can accelerate response speed and significantly reduce overshoots. It can make steady-state values close to the optimal steady-state values. It has better tracking control effects.

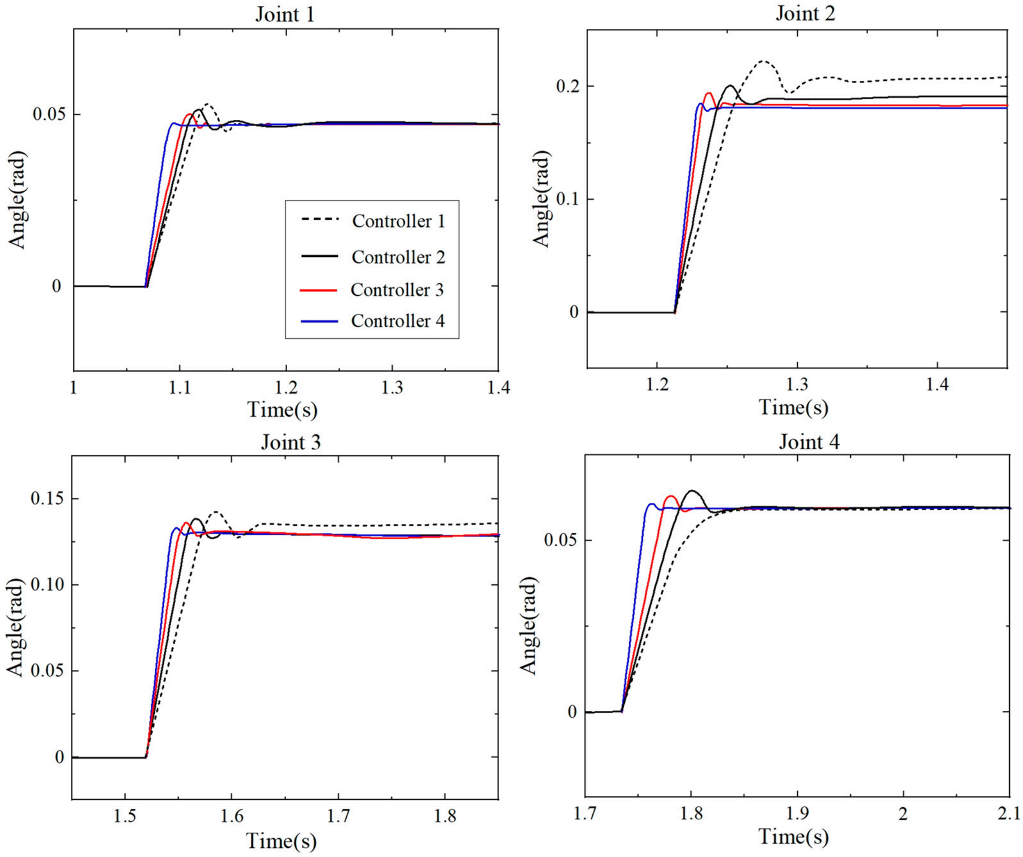


Figure 9. The step response performance of stretching.

To verify the entire tracking process performance of the proposed Fuzzy-DDPG-PID controller, we also compare the tracking time, planning-tracking time difference (time-delay), maximum tracking error, and minimum tracking error of four controllers in Table 4. The planning algorithm and controllers utilized are the same as above. From Table 4, since the parameters of PID are difficult to tune without the participation of DDPG, the large values of four indexes are achieved by Controller 1, and the difference between the maximum and minimum tracking errors is large. Because the parameters of PID are tuned via DDPG, the values of four indexes are reduced by Controller 2 to 4. Due to the improvements of the update function and loss function in DDPG, the values of four indexes accomplished by Controller 4 are smaller, and the difference between maximum and minimum tracking error is also smaller. Therefore, the proposed Fuzzy-DDPG-PID controller can reach high tracking speed, small tracking error, and strong robustness.

Table 4. The performance comparison of the entire tracking process.

Controller	Tracking Time (s)	Time-Delay (s)	Maximum Tracking Error (m)	Minimum Tracking Error (m)
Controller 1	12.09	1.25	0.0485	0.0309
Controller 2	11.46	0.97	0.0433	0.0277
Controller 3	9.73	0.64	0.0359	0.0251
Controller 4	7.33	0.49	0.0283	0.0195

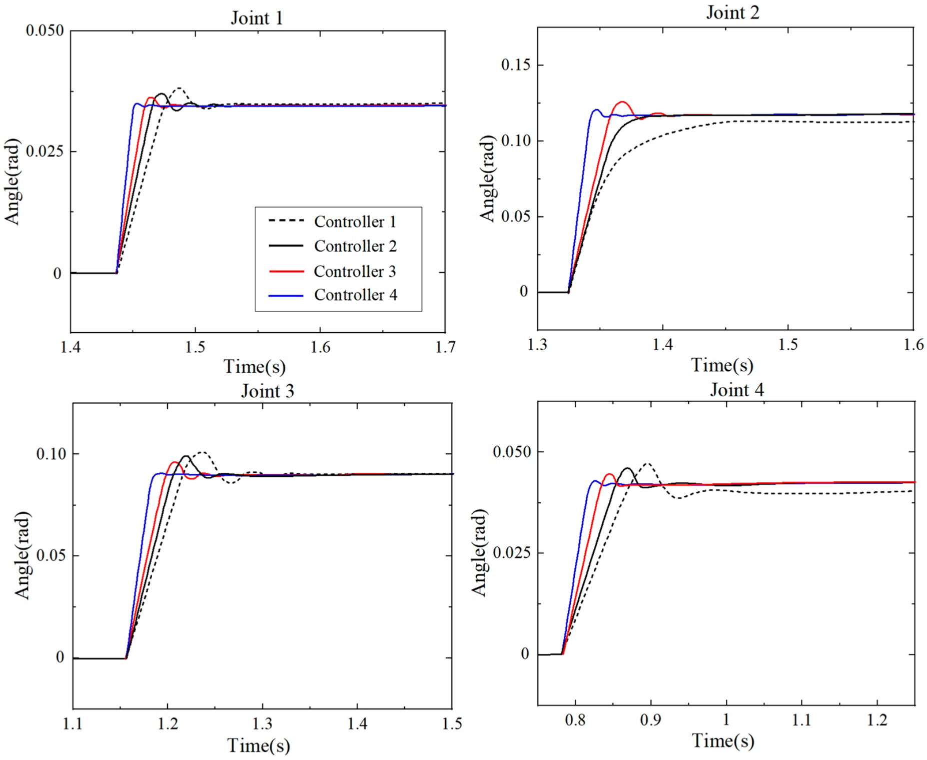


Figure 10. The step response performance of retrieving.

To avoid the occasionality of the entire tracking process result, we repeat the experiment of the entire tracking process five times. In the repetitive experiments, the layouts and disturbances of the environment are changed. The five times experimental results of the entire tracking process are shown in Figure 11.

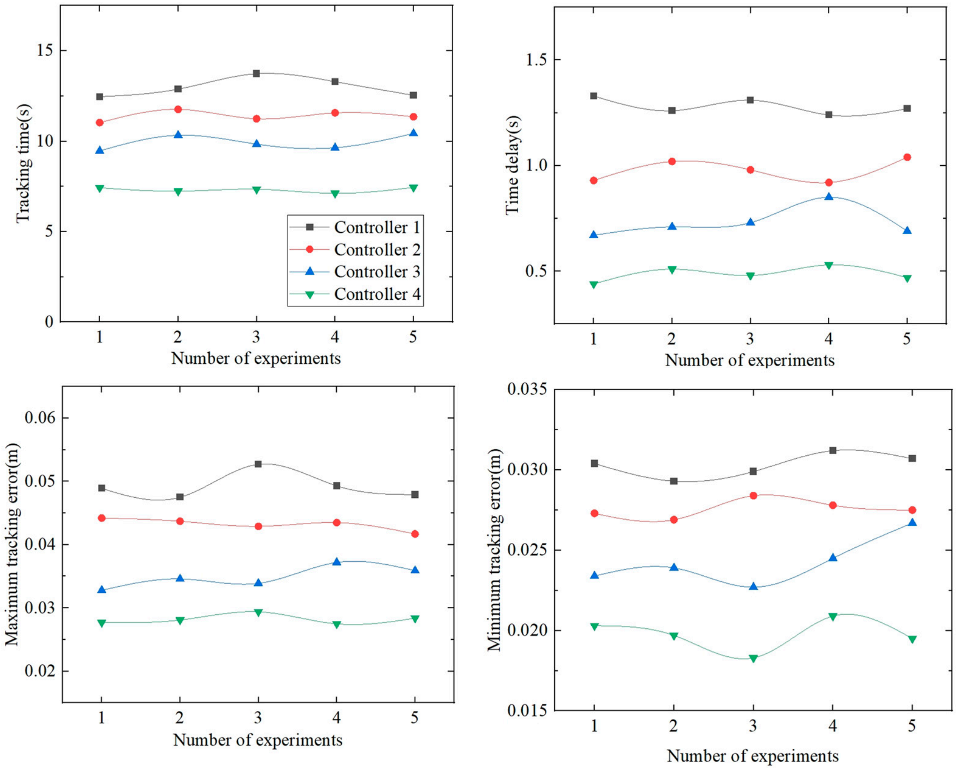


Figure 11. The five times experimental results of the entire tracking process.

5. Conclusions

A novel planning and tracking approach for a mobile robotic arm to grab objects in an obstacle environment is proposed in this study. To verify the effectiveness of the proposed approach, the real-world object-grabbing experiments of a mobile robotic arm are conducted in obstacle environments, and the proposed approach is compared and analyzed. We compared the planning results of two different RRT* and proved that the improved RRT* planned shorter paths with smaller total turning angles. Based on the improved RRT*, we compared the planning results of two different APF-RRT*. The results illustrate that the improved APF-RRT* acquires fewer invalid branches and planning time. It has better planning performance compared to other algorithms. To avoid the occasionality of the one experiment result, we repeated the experiment three times. The results demonstrate that the improved APF-RRT* is superior to existing algorithms. Based on the improved APF-RRT*, we compare and analyze the tracking performance of the proposed Fuzzy-DDPG-PID controller. The step response performances of the joint motors illustrate that the proposed Fuzzy-DDPG-PID controller reaches an earlier response time and smaller overshoot. The steady-state values are close to the optimal steady-state values. Therefore, it has high response speed and control accuracy. The results of all the tracking processes of the four controllers demonstrate that the proposed Fuzzy-DDPG-PID controller can realize short tracking time and time delay. Its maximum and minimum tracking errors are small, and the difference values between the maximum and minimum tracking errors are also small. Therefore, the proposed Fuzzy-DDPG-PID controller can achieve high tracking speed, small tracking errors, and strong robustness. To avoid the occasionality of the one experiment result, we repeat the experiment of the entire tracking process five times. The results demonstrate that the proposed Fuzzy-DDPG-PID controller outperforms the existing controllers.

In future research, we will build more complex experimental environments, and more disturbances will be considered. We will also extend the research of mobile robotic arms from single-target object grabbing to multi-target object grabbing, with a single mobile robotic arm to multi-mobile robotic arms collaboration. Moreover, the planning and tracking research of the robotic arm will be extended to the target object grabbing performed via the unmanned surface vessel (USV) [35] and other unmanned platforms.

Author Contributions: Conceptualization, J.Y., J.W. and J.X.; methodology, X.W. and Z.Z.; formal analysis, J.Y. and Z.Z.; validation, J.W. and J.Y.; writing—original draft, J.Y., J.W. and J.X.; writing—review and editing, J.W., B.W. and X.C.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 61903008), Project of Cultivation for Young Top-notch Talents of Beijing Municipal Institutions (No. BPHR202203043).

Data Availability Statement: Data are contained within the article.

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

References

1. Li, Y.; Yang, C.G.; Yan, W.S.; Cui, R.X.; Annamalai, A. Admittance-Based Adaptive Cooperative Control for Multiple Manipulators with Output Constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3621–3632. [CrossRef] [PubMed]
2. Sepehri, A.; Moghaddam, A.M. A Motion Planning Algorithm for Redundant Manipulators using Rapidly Exploring Randomized Trees and Artificial Potential Fields. *IEEE Access* **2021**, *9*, 26059–26070. [CrossRef]
3. Lu, X.H.; Jia, Y.M. Trajectory Planning of Free-Floating Space Manipulators with Spacecraft Attitude Stabilization and Manipulability Optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7346–7362. [CrossRef]
4. Wang, L.J.; Lai, X.Z.; Zhang, P.; Wu, M. A Control Strategy Based on Trajectory Planning and Optimization for Two-Link Underactuated Manipulators in Vertical Plane. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 3466–3475. [CrossRef]
5. Nie, J.M.; Wang, Y.A.; Mo, Y.; Miao, Z.Q.; Jiang, Y.M.; Zhong, H.; Lin, J. An HQP-Based Obstacle Avoidance Control Scheme for Redundant Mobile Manipulators under Multiple Constraints. *IEEE Trans. Ind. Electron.* **2023**, *70*, 6004–6016. [CrossRef]
6. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]

7. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
8. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; Volume 2, pp. 500–505.
9. Lavalle, S.M. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *Res. Rep.* **1999**, *31*, 293–308.
10. Valle, S.M.L.; Kuffner, J.J. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400.
11. Devaurs, D.; Siméon, T.; Cortés, J. Optimal Path Planning in Complex Cost Spaces with Sampling-Based Algorithms. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 415–424. [CrossRef]
12. Ju, T.; Liu, S.; Yang, J.; Sun, D. Rapidly exploring random tree algorithm-based path planning for robot-aided optical manipulation of biological cells. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 649–657. [CrossRef]
13. Iram, N.; Amna, K.; Zulfiqar, H. A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms. *Int. J. Comput. Sci. Netw. Secur.* **2016**, *16*, 20–27.
14. Gammell, J.D.; Srinivasa, S.S.; Barfoot, T.D. Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2997–3004.
15. Chi, W.Z.; Meng, M.Q.H. Risk-RRT: A robot motion planning algorithm for the human robot coexisting environment. In Proceedings of the 18th International Conference on Advanced Robotics (ICAR), Hong Kong, China, 10–12 July 2017; pp. 583–588.
16. Yuan, Q.N.; Yi, J.H.; Sun, R.T.; Bai, H. Path Planning of a Mechanical Arm Based on an Improved Artificial Potential Field and a Rapid Expansion Random Tree Hybrid Algorithm. *Algorithms* **2021**, *14*, 321. [CrossRef]
17. Jiang, Q.S.; Cai, K.; Xu, F.Y. Obstacle-avoidance path planning based on the improved artificial potential field for a 5 degrees of freedom bending robot. *Mech. Sci.* **2023**, *14*, 87–97. [CrossRef]
18. Jin, Y. Decentralized adaptive fuzzy control of robot manipulators. *IEEE Trans. Syst. Man Cybern.* **1998**, *28*, 47–57.
19. Mummadi, V. Design of robust digital PID controller for H-bridge soft switching boost converter. *IEEE Trans. Ind. Electron.* **2011**, *58*, 2883–2897. [CrossRef]
20. Wai, R.J.; Lee, J.D.; Chuang, K.L. Real-time PID control strategy for maglev transportation system via particle swarm optimization. *IEEE Trans. Ind. Electron.* **2011**, *58*, 629–646. [CrossRef]
21. Kim, K.; Rao, P.; Burnworth, J. Self-Tuning of the PID Controller for a Digital Excitation Control System. *IEEE Trans. Ind. Appl.* **2010**, *46*, 1518–1524.
22. Muszynski, R.; Deskur, J. Damping of torsional vibrations in high dynamic industrial drives. *IEEE Trans. Ind. Electron.* **2010**, *57*, 544–552. [CrossRef]
23. Khan, M.A.S.K.; Rahman, M.A. Implementation of a wavelet based MRPID controller for benchmark thermal system. *IEEE Trans. Ind. Electron.* **2010**, *57*, 4160–4169. [CrossRef]
24. Wang, J.; Wang, C.H.; Chen, C.L.P. The Bounded Capacity of Fuzzy Neural Networks (FNNs) via a New Fully Connected Neural Fuzzy Inference System (F-CONFIS) with Its Applications. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 1373–1386. [CrossRef]
25. Cheng, M.B.; Su, W.C.; Tsai, C.C. Robust tracking control of a unicycle-type wheeled mobile manipulator using a hybrid sliding mode fuzzy neural network. *Int. J. Syst. Sci.* **2010**, *43*, 408–425. [CrossRef]
26. Du, G.L.; Liang, Y.H.; Gao, B.Y.; Otaibi, S.A.; Li, D. A Cognitive Joint Angle Compensation System Based on Self-Feedback Fuzzy Neural Network with Incremental Learning. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2928–2937. [CrossRef]
27. Mai, T. Hybrid adaptive tracking control method for mobile manipulator robot based on Proportional–Integral–Derivative technique. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 6463–6480. [CrossRef]
28. Ander, I.; Elena, L.; Ander, A.; Andoni, R.; Iker, L.; Carlos, T. Learning positioning policies for mobile manipulation operations with deep reinforcement learning. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 3003–3023.
29. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
30. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
31. Wang, F.; Ren, B.M.; Liu, Y.; Cui, B. Tracking Moving Target for 6 degree-of-freedom Robot Manipulator with Adaptive Visual Servoing based on Deep Reinforcement Learning PID Controller. *Rev. Sci. Instrum.* **2022**, *93*, 045108. [CrossRef]
32. Geng, H.; Hu, Q.; Wang, Z. Optimization of Robotic Arm Grasping through Fractional-Order Deep Deterministic Policy Gradient Algorithm. *J. Phys. Conf. Ser.* **2023**, *2637*, 012006. [CrossRef]
33. Afzali, S.R.; Shoaran, M.; Karimian, G. A Modified Convergence DDPG Algorithm for Robotic Manipulation. *Neural Process. Lett.* **2023**, *55*, 11637–11652. [CrossRef]
34. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. Robot Modeling and Control. *Ind. Robot Int. J.* **2006**, *17*, 709–737.
35. Yu, J.B.; Yang, M.; Zhao, Z.Y.; Wang, X.Y.; Bai, Y.T.; Wu, J.G.; Xu, J.P. Path planning of unmanned surface vessel in an unknown 593 environment based on improved D*Lite algorithm. *Ocean Eng.* **2022**, *266*, 112873. [CrossRef]

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