IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 July 2023, accepted 17 August 2023, date of publication 7 September 2023, date of current version 19 September 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3312763*

RESEARCH ARTICLE

# Obstacle Avoidance Path Planning for Apple Picking Robotic Arm Incorporating Artificial Potential Field and A* Algorithm

**MIN ZHUANG[1,2], GE LI[3], AND KEXIN DING[1]**

[1]School of Intelligent Manufacturing, Hangzhou Polytechnic, Hangzhou 311402, China
[2]College of Advanced Manufacturing Innovation (AMI), King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok 10520, Thailand
[3]School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China

Corresponding author: Min Zhuang (minzhuang009@163.com)

This work was supported in part by the Zhejiang University Visit Scholar Project—Research and Development of Intelligent Production Line "Digital Twin" Technology under Grant FX2018140, and in part by the National Natural Science Foundation of China—A Study on the Design Method of Rotary Transplanting Mechanism Facing the Requirements of Transplanting Track and Posture under Grant 51375459.

**ABSTRACT** With the development and maturity of the automated robotics industry, more and more apple plantations are introducing automated picking robotic arms for fruit picking. However, the complex environment in which apple fruit is picked has made it an urgent problem to optimise the robot's picking performance through obstacle avoidance path planning. The experiment selects the Six degrees of freedom manipulator as the research object, and on the basis of its Kinematics analysis, introduces the introduction of artificial potential field (APF) to carry out the path planning of the manipulator. At the same time, it integrates it with A* algorithm to jointly achieve the optimization of the parameters of the obstacle avoidance path of the manipulator. In addition, in order to avoid parameter optimization falling into local extremum during the path planning process, the IRRT algorithm is incorporated to re plan the path, improve the smoothness of the path, and finally verify its obstacle avoidance effect through simulation experiments. The results showed that in the convergence comparison, the research method had the minimum loss function value and the stable fitness value as soon as the iteration proceeded to the 50th and 20th generation, respectively. A On the dataset, the research method had the minimum MAPE value when the iteration proceeded to the 45th generation, with a value close to 0. At the same moment, the MAPE values of the IAPF algorithm, the IRRT algorithm and the literature were 0.052%, 0.108% and 0.218%, respectively. In the practical application analysis, when the robot arm starts running in three different starting positions a, b and c, the IRRT algorithm's obstacle avoidance path has a larger arc and tends to reach the target location through a longer path, while the research method tends to find a relatively closer obstacle avoidance path that can be passed smoothly. The above results show that the research method is highly adaptable to robotic arm path avoidance planning and can complete obstacle avoidance path planning faster and more reasonably, providing new technical support for optimising the path planning system of apple picking robots.

**INDEX TERMS** Robotic arm, obstacle avoidance path planning, APF, A* algorithm.

## I. INTRODUCTION

In recent years, many of the world's largest companies have relocated their production sites to China and automation technology is being rolled out into industrial production. However, traditional manufacturing processes no longer meet the requirements of the current industrial environment and the manufacturing industry must be further transformed with the introduction of new, highly efficient production methods and equipment. In addition, with the increased intelligence of

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

modern equipment, many of the world's leading laboratories are developing robots that can be used in different fields, which is driving the development of the discipline of robotics. Robotics is a combination of several disciplines such as computing, mechanics and control, and is one of the most popular disciplines at the moment. At present, a country's manufacturing capacity is increasingly dependent on its level of industrial automation, which cannot be improved without the development of robotics, all of which effectively promotes the integration between automated production technology and established production equipment. In the manufacturing industry, robotic arms play a very crucial role in helping workers to complete many of the more difficult tasks, such as the processing and assembly of mechanical parts and production, food packaging and processing in factories, cleaning operations for items and so on. Through the correct use of robotic arms, the efficiency and speed of industrial production is effectively and significantly increased, thus saving Dalang's workforce. In addition to the use of robotic arms in industrial production, many researchers are currently investigating how robots can be brought into human life to provide a more humane service to humans. The robotic arms of robots are just like human arms and are capable of performing a number of high-precision tasks. For example, some restaurants use mobile robot waiters, which are able to deliver the corresponding plates with meals exactly to the customers; some robots can also take on the role of librarians, correctly sorting and arranging the books in the library collection. Even in the field of firefighting, firefighters can remotely control robots to approach places that are inaccessible to humans and perform tasks that cannot be done by humans. All of the above points to the relevance of research into methods for the automatic control of robotic arms.

Currently, most of the robotic arms used in industrial production are operated according to a pre-set fixed trajectory or may be operated remotely by professionals. This type of robotic arm control has the obvious disadvantage of not being able to adapt to more complex working environments and work effectively. However, as the global population continues to grow and the economy continues to develop, so does the agricultural industry and the use of automation technology has become an important trend in the agricultural sector. Especially in the field of apple picking, the traditional manual picking method requires a lot of human resources and time, is less efficient and hardly meets the needs of modern agriculture. Automated picking robotic arms are an efficient and reliable alternative and have great potential for development [1]. During apple picking, the robot arm needs to navigate and avoid various obstacles such as branches, bushes and stones to find the best picking route to avoid damaging the fruit and the robot arm itself [2]. Therefore, how to achieve automatic path planning for robotic robotic arms and complete autonomous obstacle avoidance has become an important problem for robotic arm picking [3]. In the past few decades, a number of algorithms have been proposed for obstacle avoidance path planning, such as graph-based methods, Artificial Potential Field (APF), A* algorithms, genetic algorithms, etc.

Two of the more commonly used and effective algorithms are the APF and A* algorithms [4]. APF is based on the mechanics of viewing a robotic arm as a moving particle, abstracting its surroundings as a region consisting of potential fields, and calculating the forces on the particle in each direction based on the position and velocity of the moving particle. The A* algorithm avoids obstacles by finding the shortest path in order to achieve autonomous navigation [5]. Although there is currently a wealth of research in path planning, universal algorithms cannot be well applied in certain specific fields, and most of the current research on APF and A* algorithms is independent. There is still relatively little research on obstacle avoidance path planning that integrates the two. In this context, this study attempts to utilize the advantages of APF and A* algorithms to design an obstacle avoidance path planning method for apple picking robotic arms based on the APF-A* algorithm. At the same time, in order to avoid the parameter optimization falling into the local extreme value in the path planning process, the experiment also introduces the IRRT method to improve the Rate of convergence of the algorithm, and jointly constructs a new obstacle avoidance path planning method. Looking forward to providing a realistic theoretical basis and practical significance for subsequent related research.

## II. RELATED WORKS

With the rapid development of the intelligent robotics industry today, robot motion control technology has gradually become the focus of the industry, and experts are aware of the value of robotic robotic arm obstacle avoidance path planning systems. Many scholars have conducted research related to robot obstacle avoidance path planning. Wang's team has proposed a method for evaluating obstacle avoidance path planning metrics when tackling the problem of robot obstacle avoidance path planning performance affecting human-robot collaboration. A cost function was used to calculate the efficiency of each planned path. Experimental results showed that the method can improve the robot's path planning ability in different scenarios [6]. Ma et al. proposed a Bezier curve-based obstacle avoidance path planning method to address the problem of redundant nodes and peak inflection points in traditional algorithms for path planning. The process first obtains the Bezier curve control points and then adds an adaptive penalty factor to the function values. A comparison with existing methods is also made. The results indicated that such method can effectively produce smoother paths [7]. Gao et al. and other researchers have proposed a new QEA-based algorithm to address the lack of speed of traditional genetic algorithms in planning robot paths. The process QEA algorithm works in a discretized environment and calculates the path validity. Simulation experimental results demonstrated that the running time of the method was significantly

faster than that of the traditional genetic algorithm [8]. Han and Yu and other researchers have proposed a centralised decoupling algorithm for the robot obstacle avoidance path planning problem in grid graphs. The algorithm uses dynamic settings to enable the robot to re-route frequently to target locations. Experimental results revealed that the method is highly scalable and can provide better path solutions [9]. Wang and Zhou proposes a neural network-based robot path planning scheme to address the current problem of insufficient accuracy of genetic algorithms in robot path planning. The method performs dynamic path planning of the robot through fuzzy neural networks and compares the effect of other traditional particle swarm algorithms. The results suggested that the neural network-based obstacle avoidance path planning scheme has better control accuracy [10].

The APF and A* algorithms have also been studied by some scholars in other path planning areas. xu X and his team have proposed a hierarchical APF algorithm to address the shortcomings of traditional APF in collision avoidance. The new algorithm judges the risk of collision by measuring relative distance and relative speed. The experimental results show that this method is better at collision avoidance in dynamic environments [11]. Imrane et al. propose a scheme that combines multiple navigation methods to address the problem of insufficient planning comprehensiveness of a single navigation method. The scheme combines APF, neural network and interval type II fuzzy logic to process the signal and output the data. Experimental results show that the proposed scheme is more efficient in performing navigation planning [12]. researchers such as Du et al. proposed an obstacle avoidance scheme using APF considering UAVs flying in a multi-obstacle environment. The scheme uses APF for formation reconfiguration during UAV formation flight and for controlling the safe UAV spacing. Experimental results show that the proposed scheme has a high adaptability to different scenarios [13]. Long et al. and other researchers have proposed an A*-based optimisation algorithm to address the limitations of traditional BFO algorithms for path planning in the context of USVs. The process used sensitivity analysis to obtain the maximum impact parameters and the performance was evaluated with different size raster maps. The results show that AS-BFO is more efficient than conventional methods for path planning in USVs [14]. Gu et al. proposes an improved Label-A* algorithm based on the A* algorithm for the manoeuvring of unmanned boats in restricted waters. The process establishes a library of unmanned boat trajectory units and plans the unmanned boat trajectory in terms of trajectories. The experimental results show that the improved Label-A* algorithm can take into account the multifaceted needs of unmanned boat trajectories [15].

In summary, although APF and A* algorithms have been studied in other obstacle avoidance path planning fields, research that uses both APF and A* algorithms to improve robot arm obstacle avoidance path planning is rare. In view of this, the study proposes a robotic arm obstacle avoidance path planning system based on both APF and A* algorithms, which is urgently needed to provide more reference solutions for the development of the apple picking robotic arm obstacle avoidance path planning industry.

**TABLE 1. Standard D-H parameters of six degrees of freedom manipulator.**

| Connecting rod serial number ($i$) | $a_{i-1}$ | $\alpha_{i-1}$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0° | 0 | $l_{1a}$ | $\theta_1$ (0°) |
| 2 | 0° | $-\pi/2$ | $l_{1b}$ | $\theta_2$ (0°) |
| 3 | $l_2$ | 0 | 0 | $\theta_3$ (0°) |
| 4 | 0° | $-\pi/2$ | $l_3$ | $\theta_4$ (0°) |
| 5 | 0° | $\pi/2$ | $l_4$ | $\theta_5$ (0°) |
| 6 | 0° | $-\pi/2$ | $l_5$ | $\theta_6$ (0°) |

## III. OBSTACLE AVOIDANCE PATH PLANNING METHOD FOR ROBOTIC ARMS FOR PICKING OPERATIONS

### A. KINEMATIC ANALYSIS OF A SIX DEGREE OF FREEDOM PICKING ROBOT ARM

As the robotic arm contains many joints, the different joints are constrained and linked to each other, making the robotic arm only able to move in a regular space. This makes the movement of the robot arm less flexible and variable than that of a robot, which in turn leads to traditional path planning methods not being able to fully meet the requirements of robot arm obstacle avoidance path planning. The object of study is a six-degree-of-freedom robot arm, which is a modular arm consisting of six rotating joints of different lengths acting in series with each other. In this arm, only the chi-square transformation matrix between two links is determined, and then multiple matrix operations are performed between the different links to establish the positional relationship from the base to the end link. Then the coordinate systems on all the connecting rods are determined using D-H modelling. In order to facilitate the construction of the corresponding matrices for subsequent studies, all the parameters on the robot arm are first determined and added to the D-H parameter table. The specific D-H parameters are shown in Table 1.

As shown in Table 1, $i$ represents the linkage serial number; $a_i$ represents the shortest distance between the two joint axes (i.e. the length of the rod); $\alpha_i$ represents the angle between the two axial joint links; the study determines the relationship between the positions of the two rods through the distance $d_i$ and the angle $\theta_i$ between the two rods. After determining the rod parameters, when the rod serial number is 1, the rod parameters between $O_0$ and $O_1$ exist with offset $d_1 = l_{1a}$ and have $\alpha_0$ and $a_0$ as $U_{att}$, and so on for all parameters. The standard D-H coordinate system parameters are used to establish the coordinate system of the six degree of freedom robotic arm, and the final D-H matrix obtained needs to

satisfy equation (1).

$$\begin{cases} {}_{i}^{i+1}A \\ = {}_{i}^{i+1}R\left(X_i, \alpha_i\right) {}_{i}^{i+1}T\left(X_i, \alpha_i\right) {}_{i}^{i+1}R\left(Z_i, \theta_{i+1}\right) {}_{i}^{i+1}T\left(Z_i, d_{i+1}\right) \\ {}_{i}^{i+1}A \\ = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 & a_i \\ \sin\theta_{i+1}\cos\alpha_i & \cos\theta_{i+1}\cos\alpha_i & -\sin\alpha_i & -\sin\alpha_i d_{i+1} \\ \sin\theta_{i+1}\sin\alpha_i & \cos\theta_{i+1}\sin\alpha_i & \cos\alpha_i & \cos\alpha_i d_{i+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{cases} \tag{1}$$

In equation (1), ${}_{i}^{i+1}A$ denotes the flush transformation matrix from $L_i$ to $L_{i+1}$; ${}_{i}^{i+1}R$ denotes the rotation matrix; and V denotes the translation matrix. The corresponding parameters in the D-H parameters are substituted into equation (1) to obtain the corresponding transformation matrix, which in turn is calculated to obtain $\left\{{}_{0}^{1}A, {}_{1}^{2}A, {}_{2}^{3}A, {}_{3}^{4}A, {}_{4}^{5}A, {}_{5}^{6}A\right\}$, as shown in equation (2).

$$\begin{cases} {}_{0}^{1}A = \begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & l_{1a} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}_{1}^{2}A = \begin{bmatrix} C2 & -S2 & 0 & 0 \\ 0 & C2 & 1 & l_{1b} \\ -S2 & -C2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}_{2}^{3}A = \begin{bmatrix} C3 & -C3 & 0 & l_2 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}_{3}^{4}A = \begin{bmatrix} C4 & -S4 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ -S4 & -C4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}_{4}^{5}A = \begin{bmatrix} C5 & -S5 & 0 & 0 \\ 0 & 0 & -1 & -l_4 \\ S5 & C5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}_{5}^{6}A = \begin{bmatrix} C6 & -S6 & 0 & 0 \\ 0 & 0 & 1 & l_5 \\ -S6 & -C6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{cases} \tag{2}$$

In equation (2), ${}_{0}^{6}A$ represents the overall transformation matrix from the base to the end of the robot arm, which is the relative position of the linkage to the end of the base in the coordinate system $O_0$. And ${}_{0}^{6}A$ is the matrix form obtained by multiplying all matrices $\{{}_{0}^{1}A, {}_{1}^{2}A, {}_{2}^{3}A, {}_{3}^{4}A, {}_{4}^{5}A, {}_{5}^{6}A\}$ together; $C1$ denotes $\cos\theta_i$; $S1$ denotes $\sin\theta_i$. The above D-H modelling allows the linkage parameters of the robot arm used in the study to be obtained, which in turn allows the equations of forward motion of the robot arm to be obtained, which in turn allows the position of the end of the robot arm in relation to the base and the co-ordinate system posture to be obtained if the rotation angle of the axis joint is known. A kinematic model of the six degree of freedom apple picking robot arm joints involved in the study was developed using D-H modelling and is shown in Figure 1.

By deriving the forward kinematics of the robotic arm, all joint angles can be found. In the study, so the values of the variables of the joints are determined, all variables are substituted to obtain the attitude of the end of the robotic arm. In the operation of a tandem robot arm, the solution to the

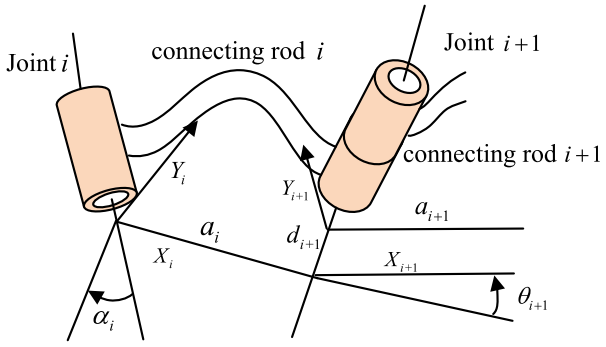


**FIGURE 1.** Coordinate system of manipulator linkage.

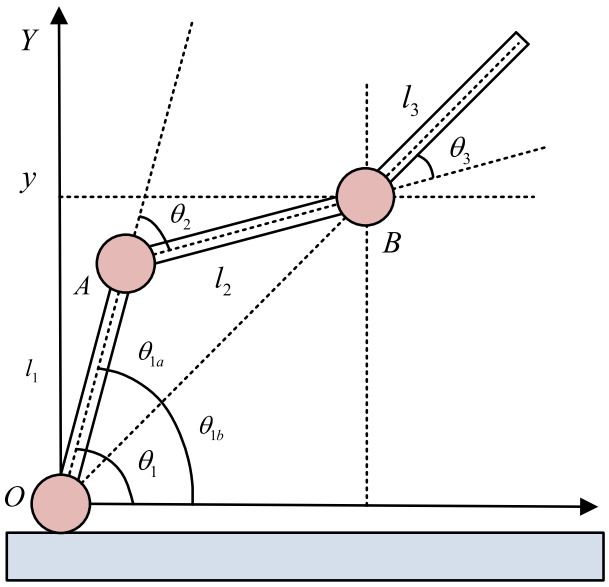


**FIGURE 2.** Simplified for the robotic arm model.

forward kinematics of the arm can be obtained only if the size of the arm is determined, but the solution to the inverse kinematics is not unique, so the geometric and algebraic methods are generally used to solve for the inverse kinematics of the joint angles of the arm. As an example, a three-degree-of-freedom robotic arm model is set up in a two-dimensional plane, see Figure 2.

In Figure 2, the coordinates of the joint point B are assumed to be $(x, y)$ and the orientation angle of the robot arm is $\theta$. $\theta_2$ can be obtained according to the principle of the cosine theorem (but the value of $\theta_2$ is uncertain and needs to be analysed on a case-by-case basis), as calculated in equation (3).

$$\begin{cases} \cos\theta_2 = \left(x^2 + y^2 - l_1^2 - l_2^2\right)/2l_1l_2 \\ \theta_2 = a\cos\left[\left(x^2 + y^2 - l_1^2 - l_2^2\right)/2l_1l_2\right] \end{cases} \tag{3}$$

In equation (3), $l_1$ and $l_2$ denote the distance between the joints of the different two axes of the robot arm respectively. To facilitate the calculation of $\theta_1$, the angle is divided into two joint angles of $\theta_{1a}$, $\theta_{1b}$. The value of $\theta_1$ is determined so that if $\theta_2$ is greater than 0, $\theta_1 = \theta_{1a} - \theta_{1b}$ is obtained; when $\theta_2$ is

less than 0, $\theta_1 = \theta_{1a} + \theta_{1b}$ is obtained. since the sum of the joint angles is equal to the directional angle $\theta$ of the robotic arm, $\theta_3 = \theta - \theta_1 - \theta_2$ can be obtained. the above process is calculated in equation (4).

$$\begin{cases} \cos\theta_{1a} = \left(x^2 + y^2 + l_1^2 - l_2^2\right) / 2l_1\sqrt{x^2 + y^2} \\ \theta_{1b} = a\tan 2\,(y, x) \end{cases} \tag{4}$$

The inverse solution of the kinematics using the algebraic method is achieved by using the inverse of the sub-matrices and multiplying them sequentially left by the known positional matrix. The first assumption is made about the overall attitude matrix of the robot arm, where the values of the matrix are determined in the right half of the matrix and the left half is the matrix parameterised by the joint angles. This is shown in equation (5).

$${}^1_0A\,{}^2_1A\,{}^3_2A\,{}^4_3A\,{}^5_4A\,{}^6_5A = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_x \\ n_z & o_z & a_z & p_x \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

In equation (5), ${}^1_0A$, ${}^2_1A$, ${}^3_2A$, ${}^4_3A$, ${}^5_4A$, ${}^6_5A$ represents the overall attitude matrix of the robot arm. With the above equation, the forward and reverse motion joint angles of the robotic arm can be calculated more clearly. This provides a more efficient motion parameter for the arm's obstacle avoidance control system, which in turn enables the rapid operation of the arm.

### B. OBSTACLE AVOIDANCE PATH PLANNING DESIGN FOR APPLE PICKING ROBOTIC ARMS BASED ON APF-A* ALGORITHM

Apple picking robotic arms are inevitably affected by obstacles when moving in the workspace, which in turn hinders the rational path planning of the robotic arm. Therefore, based on the kinematic analysis of the robot arm, the study proposes to use the APF method for path planning and to integrate it with the A* algorithm to finally avoid the obstacles perfectly. The robot arm obstacle avoidance model of the APF method is shown in Figure 3.

As shown in Figure 3, the APF method involves constructing a virtual force field in the workspace of the robot arm and solving for it. Where the target point creates an overall gravitational potential field on the end of the robot arm, and the obstacle creates a local repulsive potential field on the arm. When the obstacle is outside the repulsive potential field, the end of the arm is only influenced by the gravitational potential field; when the obstacle is within the repulsive potential field, the arm is influenced by both the repulsive potential field and the gravitational potential field, thus enabling the arm to plan its obstacle avoidance path.

In the APF algorithm space model, when the target point is designated as the lowest ''valley'' of the potential field, the arm can generally reach the target state according to the guidance of the combined potential field, through the calculation of the potential function of the gravitational field

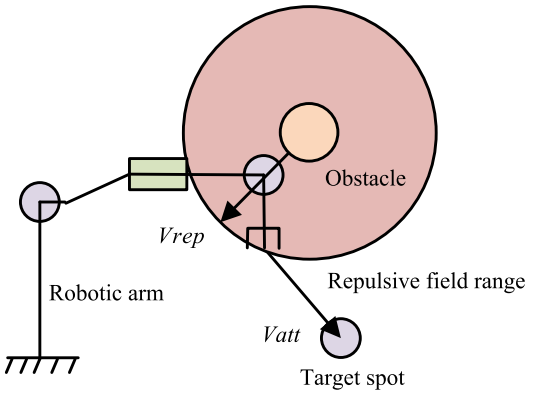


**FIGURE 3.** Mechanical arm obstacle avoidance model of the APF method.

is shown in equation (6).

$$U_{att}\,(C_i) = 1/2k_a d^2\left(C_i, C_{goal(i)}\right) \tag{6}$$

In equation (6), $U_{att}\,(C_i)$ denotes the potential function of the gravitational field; $k_a$ denotes the gravitational scale parameter; $C_{i(i=1,2,\cdots,6)}$ denotes the joint position under the current configuration of the robotic arm; $C_{goal(i)}$ denotes the joint position under the target configuration. The distance between the robotic arm and the target object can be obtained from the $d\left(C_i, C_{goal(i)}\right)$ function. Then the distance is derived to obtain the negative gradient $F_{att}$ of the $U_{att}$ and the potential function of the repulsive force, which is calculated in equation (7).

$$\begin{cases} F_{att}\,(C_i) = -\nabla U_{\text{att}}\,(C_i) = -k_a d\left(C_i, C_{goal(i)}\right) \\ \qquad\qquad \nabla d\left(C_i, C_{goal(i)}\right) \\ U_{rep}\,(C_i) = \begin{cases} 0 \\ \qquad d\left(C_i, C_{obs(i)}\right) > d_0 \\ \dfrac{1}{2}k_r\left(\dfrac{1}{d\left(C_i, C_{obs(i)}\right)} - \dfrac{1}{d_0}\right)^2 \\ \qquad 0 \le d\left(C_i, C_{obs(i)}\right) \le d_0 \end{cases} \end{cases} \tag{7}$$

In equation (7), $k_r$ represents the repulsive force proportional parameter; $C_{obs(i)}$ represents the spatial position corresponding to the collision between the robot arm and the obstacle. $d\left(C_i, C_{obsl(i)}\right)$ represents the distance between the joint and the obstacle; $d_0$ represents the repulsive force radius of the obstacle. In fact the effect of the repulsive field is extremely small and the strength of the repulsive field is 0 when the distance exceeds $d_0$. The expression of the repulsive force $F_{rep}$ can be deduced in equation (8).

$$\begin{aligned} F_{rep}\,(C_i) &= -\nabla U_{rep}\,(C_i) \\ &= \begin{cases} 0 \\ \qquad d\left(C_i, C_{obsl(i)}\right) > d_0 \\ k_r\left(\dfrac{1}{d\left(C_i, C_{obs(i)}\right)} - \dfrac{1}{d_0}\right)\dfrac{\nabla d\left(C_i, C_{obs(i)}\right)}{d^2\left(C_i, C_{obs(i)}\right)} \\ \qquad 0 \le d\left(C_i, C_{obsl(i)}\right) \le d_0 \end{cases} \end{aligned} \tag{8}$$

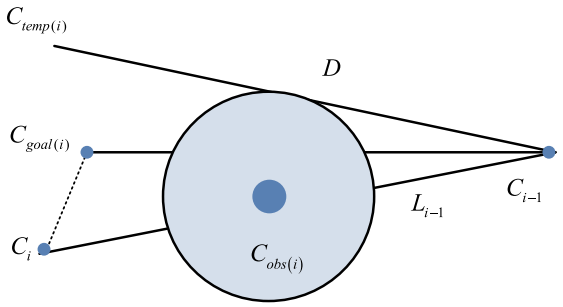


**FIGURE 4.** Temporary target point location.

Guided by the virtual potential field, the robot arm moves in the direction of the fastest reduction of the combined potential field $U_{tot}$, where the final desired target is usually located. When $U_{tot} = U_{att} + U_{rep}$ is satisfied, the combined potential field $U_{tot}$ is represented as a superimposed potential field of two forces. In order to improve the robotic arm's ability to perceive environmental information outside the adjacent joint space, the study improves the APF method. Assuming that the $i$th step position of the end joint is $C_i$, the previous $n$ step joint position of the current configuration is $C_{i-n}$. If the angle of each joint in the recent $n$ step search motion of the robotic arm is small, the calculated value keeps oscillating repeatedly within the range of the difference $\eta$ and conforms to equation (9).

$$\begin{cases} d\left(C_i, C_{i-n}\right) \leq \eta \\ d\left(C_i, C_{goal(i)}\right) \leq d_1 \end{cases} \tag{9}$$

In equation (9), $C_i$ is the current posture; $C_{goal(i)}$ is the desired posture. When the robotic arm is in a picking operation, the spatial obstacle situation is more complex and if the potential field needs to be re-established for that point, the correspondence will be more difficult and with little success. Therefore, the study uses an adaptive approach to calculate a temporary target position to help control the arm's movement to that point and escape the effects of the extremes. Then the temporary target point during the operation of the robot arm was determined, where the temporary target point location is shown in Figure 4.

As shown in Figure 4, $C_i$ represents the current pose; $C_{goal(i)}$ represents the desired pose; $C_{temp(i)}$ represents the temporary target point; $C_{obs(i)}$ represents spatial obstacles; $L_{i-1}$ represents the effective length of the repulsive field; $C_{i-1}$ represents the end joint of the robotic arm. Assume that a normal to the plane $C_{i-1}C_iC_{goal(i)}$ is made through the centre of the ellipsoid $C_{obs(i)}$. This normal will intersect the envelope model at the point AAA. By making a ray through $C_{i-1}$ and $D$, the position of the temporary target point $C_{temp(i)}$ can then be chosen to be on that ray, where the distance between $C_{i-1}C_{temp(i)}$ represents the length of $m$ times $L_{i-1}$ and satisfies equation (10).

$$d\left(C_{i-1}, C_{temp(i)}\right) = m \cdot d(C_{i-1}, C_i) \tag{10}$$

In equation (10), when the temporary target point $C_{temp(i)}$ position time parameter is set to 1, it means that $C_{temp(i)}$ is set on the $C_{i-1}D$ ray and the length of distance from $C_{i-1}$ is $d\left(C_{i-1}, C_i\right)$. Through the above method guideline calculation, the picking robot arm can timely and accurately find the existence of extreme values during the actual operation. When the corresponding extremes are generated, the position of the temporary target point can be obtained by adaptive calculation, and finally the linkage robot arm can be perfectly controlled to move to that target point and avoid the obstacle accurately. The lack of a constraint on the target distance $d\left(C_i, C_{goal(i)}\right)$ in the conventional algorithm for the repulsive potential function $U_{rep}$ usually results in the field strength of the repulsive field being determined only by the magnitude of $d\left(C_i, C_{goal(i)}\right)$. The study therefore optimises the repulsive potential function and incorporates a distance influence into the original equation, i.e. the distance $d\left(C_i, C_{goal(i)}\right)$ between the $C_i$ joint of the robot arm and $C_{goal(i)}$. After adjustment, the new repulsive potential function is finally obtained in equation (11).

$$Urep\left(Ci\right) = \begin{cases} 0 \\ \qquad d\left(C_i, C_{obs(i)}\right) > d_0 \\ \frac{1}{2}kr\left(\frac{d\left(C_i, C_{goal(i)}\right)}{d\left(C_i, C_{obs(i)}\right)} - \frac{1}{d_0}\right)^2 \\ \qquad \frac{d_0}{2} < d\left(C_i, C_{obs(i)}\right) \leq d_0 \\ \frac{1}{2}kr\left(\frac{d\left(C_i, C_{goal(i)}\right)}{d\left(C_i, C_{obs(i)}\right)} - \frac{d\left(C_i, C_{goal(i)}\right)}{d_0}\right)^2 \\ \qquad 0 \leq d\left(C_i, C_{obs(i)}\right) \leq \frac{d_0}{2} \end{cases} \tag{11}$$

In equation (12), in the improved $U_{rep}$, when $d\left(C_i, C_{obs(i)}\right) \in (d_0/2, d_0)$ is present, it means that $C_i$ is just entering the action range of $C_{obs(i)}$. Even though the $U_{rep}$ is slowly increasing at this point, the decreasing $d\left(C_i, C_{goal(i)}\right)$ is able to constrain the growth of $U_{rep}$ all the way through. The gravitational potential energy $U_{att}$ to which the apple picking robot arm is subjected in the workspace is in fact the gravitational potential energy $U_{att}\left(C_i\right)$ of the six joints. where the repulsive potential energy $U_{rep}$ to which it is subjected is expressed as the sum of the repulsive potential energy $U_{rep}\left(C_i\right)$ of the six joints, then the combined potential energy $U_{tot} = U_{att} + U_{rep}$ to which the arm is subjected. See equation (13) for a detailed expression.

$$\begin{cases} U_{att} = \sum_{i=1}^{6} U_{att}(C_i) \\ U_{rep} = \sum_{i=1}^{6} U_{rep}(C_i) \end{cases} \tag{12}$$

At this point, the combined potential energy of 36 configurations in the front joint space of the robotic arm is calculated, and the minimum collision free joint angle combination is

obtained through comparison. The obtained combination is then set to the next moving configuration. According to this method, the robotic arm will continuously move in the direction of the fastest decrease in the combined potential field until it enters the effective range of the target. If it is determined that the robotic arm is affected by local extremum, stop controlling it using the artificial potential field method, start using adaptive methods to calculate the position of the temporary target, and use the A* algorithm to control the robotic arm to move to that point to escape the local extremum. When it is detected that it has successfully escaped, the artificial potential field method is used to continue the search. If the end reaches the effective range of the target, the planning is considered successful. The overall control process of the robotic arm integrated with the artificial potential field-A* algorithm is shown in Figure 5.

The robot arm then goes cycle after cycle, as quickly as it can, in the direction with the lowest total potential energy until it is within an effective range of the target, following the flow shown in Figure 5. When the local extremes have an impact on the arm, the APF control of the arm is stopped, and an adaptive approach is used to determine the position of the temporary target. The A* algorithm is then used to control the arm away from the local extremes and begin searching for the global extremes. When the detection of the robotic arm successfully moves away from the local extremes, it starts to change to APF for another search. If the target is within the effective distance, the search for the target is considered successful. The actual and estimated cost of the A* algorithm in practice is shown in equation (13).

$$\begin{cases} g(n) = \sum_{m=1}^{n} d(P_{m-1}, P_m) \\ h(n) = \max\left[d\left(P_n, P_{goal}\right)\right] \end{cases} \tag{13}$$

In equation (13), $g(n)$ denotes the actual spend; $h(n)$ denotes the estimated spend, and the whole experiment is calculated using the Euclidean metric. In addition, in order to store the search path information, the study requires the creation of Open and Close tables. Firstly, create a search graph G containing only the starting node S, and place S in an unexpanded node table called Open to create an Open table. Next, establish an extended node table called Close, with an initial empty table to create the Close table. The Open table stores the spatial configurations that are waiting to be checked, while the Close table stores the spatial configurations that are already qualified and do not need to be checked again. The resulting flow of the extreme value escape method based on the improved A* algorithm is shown in Figure 6.

Calculations are carried out using the A* algorithm to enable the evaluation of the picking robotic arm to successfully reach the temporary target point location $P_{goal}$. Once there, a judgement is made as to whether the robotic arm escapes from the effects of the extremes. If it has not escaped from the extremes, the process is repeated according to the above process, a temporary target point location is specified and the robotic arm is guided to move to a new location until it escapes the effects of the extremes.

### C. FUSION AND IMPROVED RRT-APF ROBOT OBSTACLE AVOIDANCE PATH SMOOTHNESS RE-PLANNING METHOD DESIGN

After combining the APF and A* algorithm, it is applied to the obstacle avoidance path planning of the manipulator, which can effectively prevent the manipulator from falling into the local optimal solution when looking for the optimal obstacle avoidance path. However, the smoothness of the manipulator in finding the optimal path needs to be improved. In order to improve the stability of the manipulator during operation, the research introduces the Rapidly Exploring Random Tree (RRT) algorithm here. The RRT algorithm can not only effectively improve the convergence speed of the hybrid algorithm, but also effectively improve the stability of the manipulator during operation to a certain extent, making it have better applicability. The sampling principle of the improved RRT-APF is shown in Figure 7.

Among them, the red $p_{next}$ node is the next node to be generated by the APF algorithm. After judging that the node collides or falls into a local minimum, the improved APF-RRT algorithm is switched for collaborative planning. During collaborative planning, black $p_{next}$ nodes are generated by the improved RRT algorithm, and $O_i$ are initialized. Compared with the traditional APF-RRT algorithm, the node $p_{nearest}$, $r_1$, $r_3$ generated by the traditional RRT. Although it can help escape from local minima and improve the smoothness of the path planning system of the manipulator, the escape speed is slow, and due to the strong randomness of the RRT algorithm, the sampling points stay near obstacles, and a better path cannot be formed. In the experiment, the improved RRT algorithm is used for sampling, and the generated $p_{new}$ nodes are better in path length and direction, so a smoothest obstacle avoidance path can be generated. The algorithm flow is shown in Figure 8.

In Figure 8, the preset parameters include a starting point, a target point, and a step size. Taking the starting point as the root node, the path search is carried out through the improved APF-RRT algorithm, and the nearest obstacle, random point effect area and random effect intensity are calculated for each sampling, so as to speed up the convergence speed of the algorithm and make the algorithm approach the optimal, and Ability to utilize space near obstacles. The path optimized by the hybrid algorithm is a multi-segment path, and the curvature of the path near obstacles is easily mutated, resulting in shaking of the robotic arm during movement. Therefore, the technology of improving the smoothness of the trajectory path by using the 4th degree B-spline curve is studied, so that the path tends to be smoother. The function expression of B-spline curve is shown in formula (14).

$$B_{curve} = \sum_{i=0}^{m} N_{i,p}(u) P_i \tag{14}$$

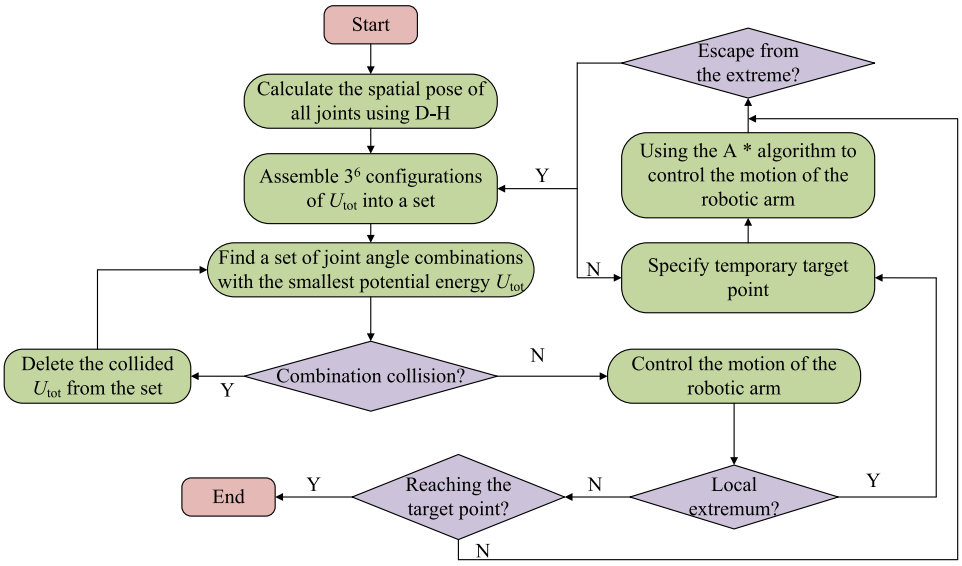


**FIGURE 5.** Integrated artificial potential field-A* algorithm for the overall process of robotic arm control.

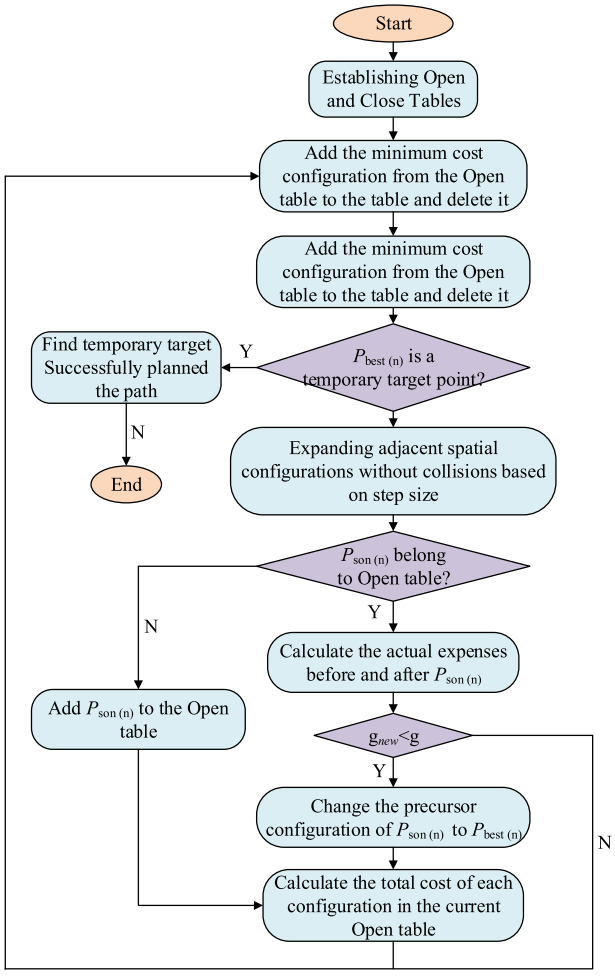


**FIGURE 6.** Flow chart of the extremal escape method based on the improved A* algorithm.

In equation (14), denotes the $N_{i,p}(u)$ $p$ B-sample basis function; $P_i$ denotes the $i$ control point and $u \in (0, 1)$. To effectively reduce significant changes in path length after

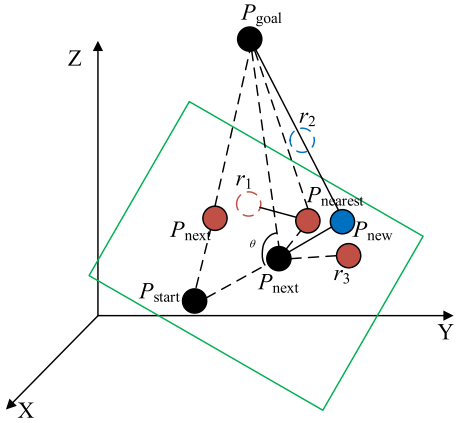


**FIGURE 7.** Sampling principle of the improved RRT-APF algorithm.

path optimization, the experiment smoothed the planned path. Firstly, perform noise reduction at the inflection point to make the path smoother at the inflection point, and then use the least squares method for smoothing. The specific calculation is shown in equation (15).

$$B(t) = A_0 t(1-t)^3 + A_1 t(1-t)^2 + A_2 t^2 + A_3 t^3 + b \quad t \in [0, 1] \tag{15}$$

In equation (15), $A_0, A_1, A_2, A_3$ represents four adjacent points on the path; $t$ represents the scale factor; $b$ represents the correction factor.

## IV. IMPLEMENTATION AND TESTING OF OBSTACLE AVOIDANCE PATH PLANNING FOR APPLE PICKING ROBOTIC ARM WITH APF-A* ALGORITHM

### A. PERFORMANCE ANALYSIS OF AN OBSTACLE AVOIDANCE PATH PLANNING SYSTEM FOR PICKING ROBOTIC ARMS

In an effort to validate the effectiveness of the APF-A* algorithm as applied to an apple picking robotic arm obstacle

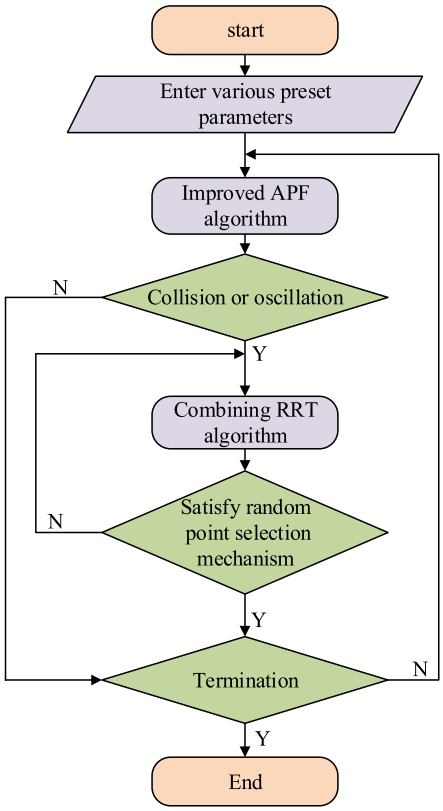


**FIGURE 8.** Hybrid algorithm flow.

**TABLE 2.** Basic software and hardware environment settings.

| Parameter variables | Parameter selection |
|---|---|
| CPU | Intel®CoreTM i7-7700HQ CPU |
| Internal storage | 16G |
| GPU | NVIDIA GTX 1650Ti |
| Operating environment | MATLAB 2016 |
| operating system | Wndows10 |
| Robot software system | Ubuntu ROS |
| Mobile chassis | Kobuki |
| Laser radar | Rplidar A1 |

avoidance path planning system, the study first tested the performance of the constructed algorithm. The basic hardware and software environment setup for the experiments is shown in Table 2.

To ensure that the experiments were conducted in a reasonable manner, all algorithm experiments were conducted in the same environment. The study selected the literature [21], the IAPF algorithm, the Improved Rapidly-Exploring Random Trees (IRRT) algorithm and the algorithm constructed by the Institute for performance comparison [22], [23]. The ROC curves obtained from the four algorithms run on the test set are shown in Figure 9.

In Figure 9, the area under the ROC curve is the magnitude of the AUC value [24]. The area under the ROC curve for the research method, the literature [21], the IAPF algorithm and the IRRT algorithm are 0.955, 0.897, 0.815 and 0.758 respectively. Comparative analysis shows that the AUC value of the

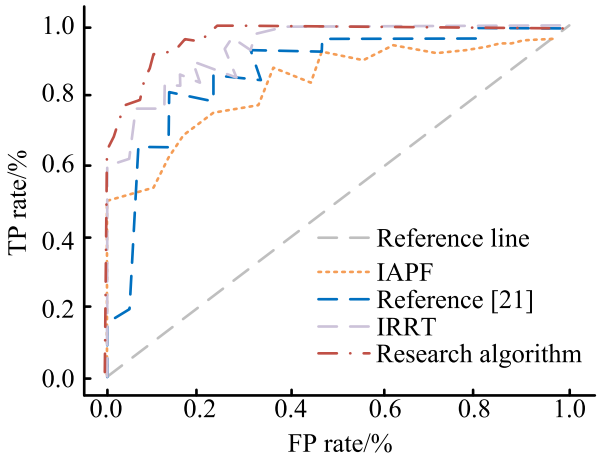


**FIGURE 9.** ROC curves of four models.

research method is significantly larger than the other algorithms. This indicates that the detection results of the robotic arm obstacle avoidance system obtained by the research method are more realistic, and the model has higher accuracy and generalization ability during the operation process, which is conducive to the robotic arm to achieve better obstacle avoidance path planning during the operation process. On this basis, the study analyses the convergence speed of the four algorithms, and the results are shown in Figure 10.

Figure 10(a) shows a comparison of the convergence of different algorithms. the IRRT algorithm stabilises after the 60th iteration with a minimum loss function value of 0.0121; the algorithm in the literature [21] reaches convergence at the 80th iteration with a minimum loss function value of 0.0125; the IAPF algorithm starts to enter a stable state only after the 90th iteration, and thereafter remains stable at The IAPF algorithm did not start to stabilize until the 90th iteration, and then remained stable around the loss function value of 0.0128. Figure 10(b) shows the adaptation curves of the different algorithms, which show that the research method and IRRT algorithm reach a smooth adaptation value in the 20th and 25th iterations respectively. Taken together, the above results show that the research method is able to enter the convergence state earlier in the training process. In addition, the mean absolute percentage error (MAPE) of the different algorithms running on two different datasets, A and B, was compared and the results are shown in Figure 11.

In Figure 11, it can be found that the MAPE values of all four algorithms on different datasets show a significant decreasing trend as the iterative run progresses. Figure 11(a) shows the change curve of MAPE values for the four algorithms on data set A. It can be found that when the iteration proceeds to the 45th generation, the research method has the smallest MAPE value with a value close to 0. At the same moment, the MAPE values of the IAPF algorithm, IRRT algorithm and literature [21] are 0.052%, 0.108% and 0.218% respectively, which are all larger than the research method. Figure 11(b) shows the change curves of MAPE

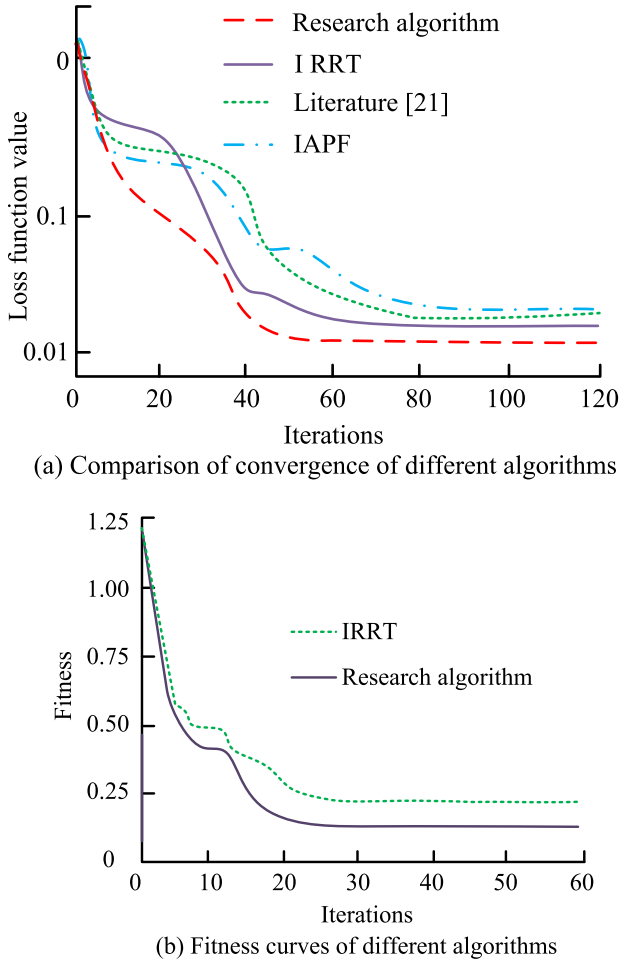


**FIGURE 10.** Curve of convergence speed and fitness of different algorithms.

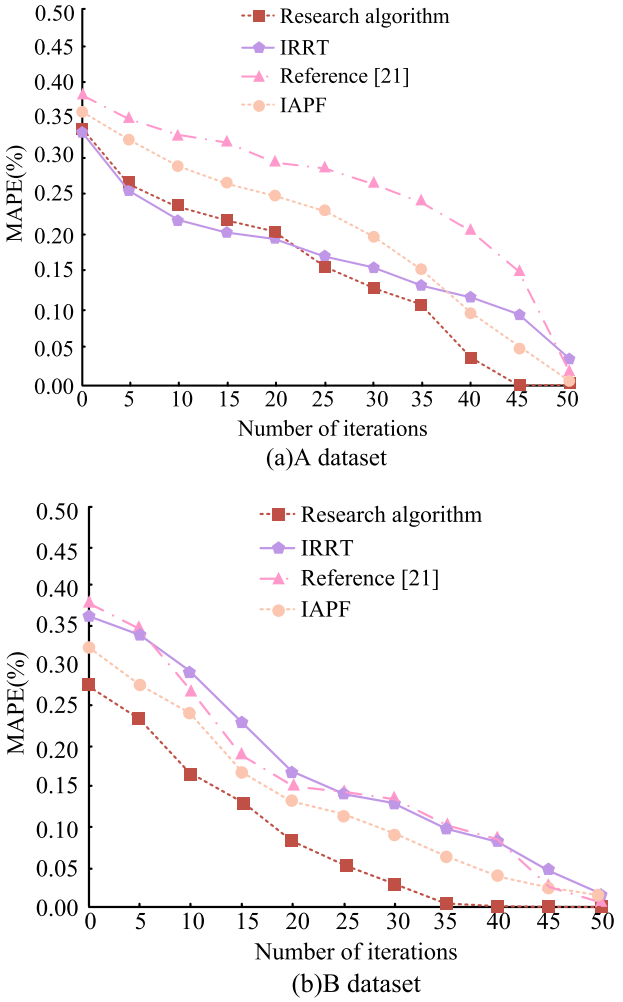


**FIGURE 11.** MAPE values for different datasets.

values for different algorithms on the B data set. It can be found that when the MAPE value of the research method reaches 0, the corresponding number of iterations is the 35th; at this time the MAPE values of the IAPF algorithm, the IRRT algorithm and the literature [21] are 0.078%, 0.112% and 0.123% respectively. Analysis of the obtained results shows that the differences between the research method and the other algorithms are more obvious, which also represents the better feasibility, less error and better performance of the research method in the direction of robotic arm obstacle avoidance path planning.

### *B. ANALYSIS OF THE PRACTICAL APPLICATION EFFECT OF ROBOTIC ARM OBSTACLE AVOIDANCE PATH PLANNING ALGORITHM*

To ensure the practical application of the improved algorithm, the study applied the constructed algorithm to apple picking robotic arm obstacle avoidance path planning for testing [25], [26]. The real experimental site of the robotic arm is an apple tree orchard planted by an actual fruit grower; the simulated experimental environment is a barrier-free environment for testing. Sensor equipment is equipped with cameras, radar, laser scanners, sonar and infrared sensors to collect environmental data. The control system in the obstacle avoidance path planning process includes multiple computers to process the data generated during the operation of the robotic arm. The data collection and analysis tool is SPSS20.0; at the same time, it is equipped with corresponding safety facilities to prevent operations that injure staff or damage equipment during the operation of the robotic arm. The experimental robot was first placed in an obstacle-free environment to test the path planning performance in the absence of obstacles using two methods to generate paths and perform automatic movements. The path planning is shown in Figure 12.

As seen in Figure 12, when the robot uses both methods for global path planning in an accessible environment, the route lengths generated using the IRRT algorithm and the research build algorithm are both shorter. The path generated by the IRRT algorithm tends to be a straight line, and the path generated by the research algorithm is a straight line connecting the start points. This indicates that the research method has superior obstacle avoidance path planning performance than the IRRT algorithm and has significant applicability in an obstacle-free environment.

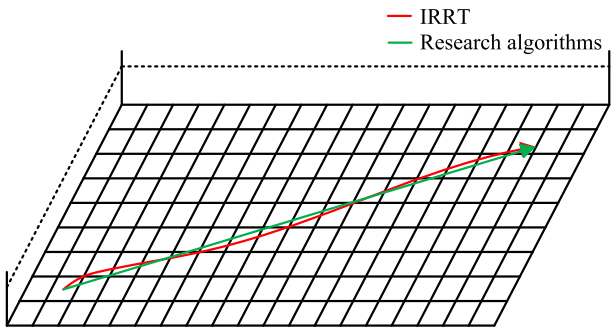


**FIGURE 12.** Barrier-free path planning.

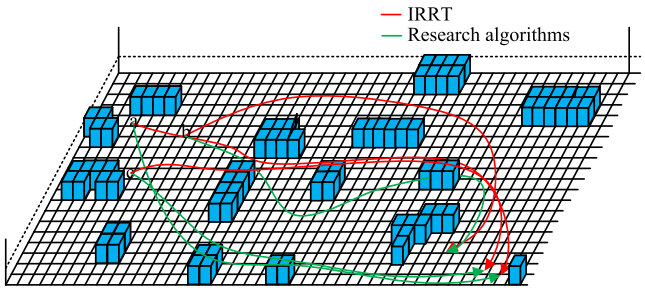


**FIGURE 13.** Schematic diagram of actual robot movement.

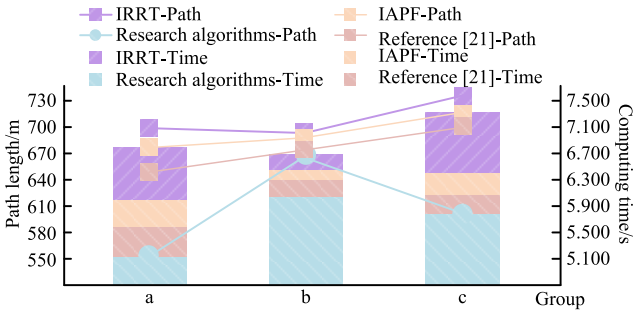


**FIGURE 14.** Comparison of the planned length and operation time of the obstacle avoidance path.

Next, an apple orchard planted by a certain fruit farmer was selected for the experiment, with a total of 8 apple trees in the overall experimental area. After marking the selected fruits, three different starting positions of a, b, and c were set. The apple picking robotic arm was used to explore the target object and the obstacle avoidance path of the robotic arm was recorded. The actual obstacle avoidance planning path is shown in Figure 13.

In Figure 13 it can be seen that there are significant differences between the obstacle avoidance paths generated by the research method and the IRRT algorithm during the actual apple picking process of the robotic arm. In the three different starting positions of a, b and c, the IRRT algorithm's avoidance path is more curved and tends to take a longer path to the target position. The research method tends to find a relatively closer path that can be successfully passed. The actual running path length and running time of the robotic arm was also observed along with the robotic arm obstacle avoidance path planning and the results are shown in Figure 14.

The comparison between IRRT and research methods is shown in Figure 12. It can be found that in group a experiment, the path length of the IRRT algorithm is 6.90955m and the running time is 6.985s; The path length of the research method is 5.63440m, and the running time is 5.162s. In group B experiments, the path length of the IRRT algorithm is 6.86140m and the running time is 6.928s; The path length of the research method is 6.58698m, and the running time is 6.018s; In group c experiments, the path length of the IRRT algorithm was 7.21772m and the running time was 7.382s; The path length of the research method is 6.02576m, and the running time is 5.836s. However, IAPF and literature [21] significantly exceeded the research methods in terms of both run length and run time. When the data from the three groups were compared, it was discovered that groups a and c had the most pronounced optimisation effects, with length optimisation amounts exceeding 0.15m and time optimisation rates exceeding 20%. Group b's optimisation effects were marginally less pronounced but had a time optimisation rate of 11.3%. The aforementioned findings demonstrate that the study technique can efficiently shorten the obstacle avoidance path diversion during the actual movement of the apple picking robot arm while decreasing the planning time for the obstacle avoidance path. The angular trends of the six joint angles of the six-degree-of-freedom robotic arm during the path planning process were tallied with the end coordinates, and the results are shown in Figure 15.

Figure 15 shows the change curve of joint angles versus end coordinates for the six degree of freedom robotic arm. In Figure 15(a), when the robot arm has reached step 97, its individual joint angles are {1.29, 0.82, 0, −2.76, −0.57, 0.08}. At this point the effective range of desired joint angles {1.3, 0.8, 0, −2.8-0.6, −0.1} has been reached, indicating successful path planning and the apple picking operation can begin. Figure 15(b) represents the trend of the component changes of the end linkage position in the X, Y and Z axis directions. All are within [−1.5,2.0]. Finally, the traditional algorithms and research methods are applied to pear and orange picking, and the search time and node number of obstacle avoidance path of the manipulator are comprehensively analyzed. The test results are shown in Figure 16.

As shown in Figure 16, without curve smoothing, the search times of the traditional A* algorithm at 0.15, 0.25, and 0.35 obstacle ratios are 3.275 seconds, 5.943 seconds, and 4.903 seconds, respectively; The search time of the optimization method at 0.15, 0.25, and 0.35 obstacle ratios is 0.739 seconds, 1.106 seconds, and 1.229 seconds, respectively. The time optimization effect in the three obstacle ratio environments is 77.43%, 81.39%, and 74.93%, respectively, with optimization rates reaching over 70%. The number of nodes generated by the traditional A* algorithm is 288, 464, and 368, respectively. The number of nodes generated by the research method is 32, 48, and 47, respectively, with an optimization rate of over 85%. This indicates that the obstacle avoidance path curve of the research method is smoother in actual operation, and it can reach the target

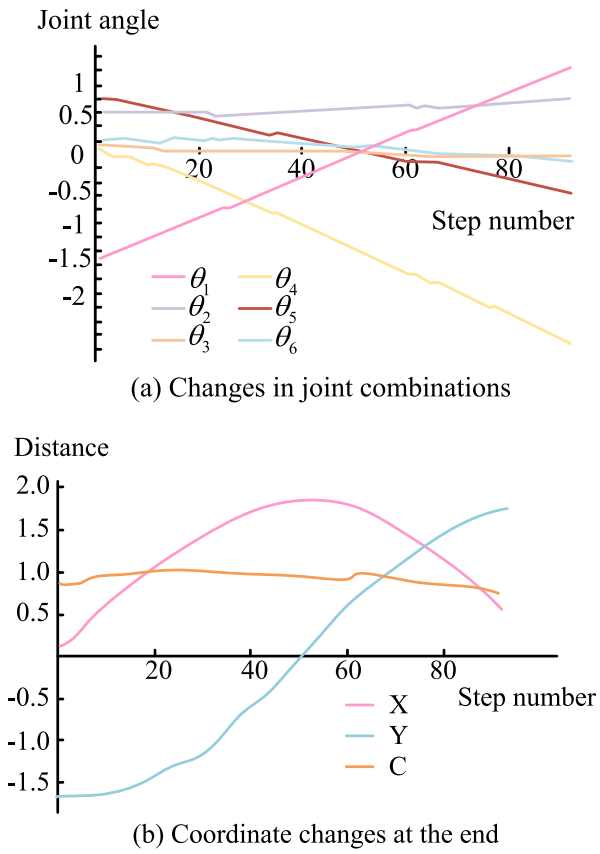


**FIGURE 15.** The angular trend of the different joint angles.

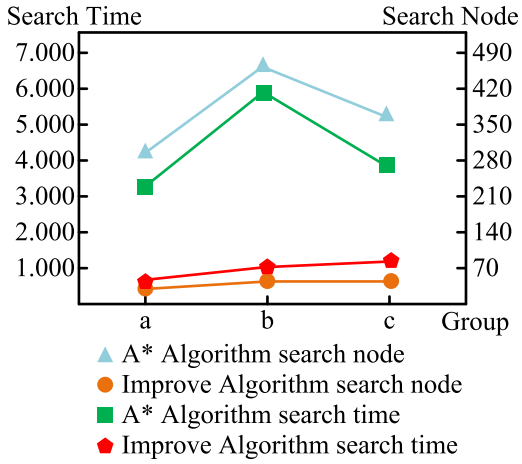


**FIGURE 16.** Path search time and node.

point more quickly. At the same time, the robotic arm system designed by the research institute was applied to several different fruit plantations to compare the accuracy of picking different fruits, as shown in Table 3.

In Table 3, it can be found that compared to other algorithms, the research method achieved picking accuracy of 96.0%, 98.0%, 94.0%, 98.0%, and 94.0% for peaches, bananas, apricots, apples, and lychees, respectively. This indicates that the research method can be applied to picking fruits of different types and sizes, while achieving high picking accuracy.

**TABLE 3.** Comparison of picking accuracy of different fruits using different methods.

| Parameter/% | IRRT | IAPF | Reference [21] | Research method |
|---|---|---|---|---|
| Accuracy of picking peaches | 86.0 | 74.0 | 92.0 | 96.0 |
| Accuracy of banana picking | 78.0. | 84.0 | 86.0 | 98.0 |
| Accuracy of apricot picking | 90.0 | 78.0 | 66.0 | 94.0 |
| Accuracy of apple picking | 62.0. | 94.0. | 82.0 | 98.0 |
| Accuracy of picking lychees | 90.0. | 68.0 | 90.0 | 94.0 |
| Average accuracy | 81.2 | 79.6 | 83.2 | 96.0 |

## V. CONCLUSION

To address the difficulty in obstacle avoidance path planning of apple picking robotic arm during task execution, the study proposes a robotic arm obstacle avoidance path planning method based on APF and A* algorithm. The research takes the six-degree-of-freedom apple picking robotic arm as the research object and uses the standard D-H parameter table to establish the corresponding coordinate system in order to obtain the forward and inverse kinematic equations. Then the APF method is used for path planning, and it is intergrated with the A* algorithm to finally avoid obstacles perfectly. The data showed that the area under the ROC curve for the research method, literature [21], IAPF algorithm and IRRT algorithm were 0.955, 0.897, 0.815 and 0.758 respectively for the comparison of AUC values. a and B data sets, the research method had the smallest MAPE value at the 45th and 35th generation of the iteration, with the values converging to 0. Comparing a, b and c groups Comparing the obstacle avoidance paths with the running times, it can be seen that the research method is more effective in groups a and c, with length optimisation of over 0.15m and time optimisation of over 20%. In the angular trend for the different joint angles, when the robot arm moves to step 97, its individual joint angles are {1.29, 0.82, 0, −2.76, −0.57, 0.08}, having reached the effective range of the desired joint angles. The research method achieved picking accuracy of 96.0%, 98.0%, 94.0%, 98.0%, and 94.0% for peaches, bananas, apricots, apples, and lychees, respectively. The above results show that the research method is highly feasible and effective in planning the actual obstacle avoidance path of the apple picking robot arm, and can effectively reduce the length of the obstacle avoidance path and reduce the planning operation time. However, the trajectory path planned by the research is not very smooth, and the robot arm may jitter and cannot be controlled in the actual application, so how to plan a smooth robot arm obstacle avoidance path through the secondary optimization algorithm is still a key research direction in the future.

## REFERENCES

[1] S. He, Y. Deng, C. Yan, Z. Gao, and C. H. Lee, "A tolerance constrained robot path circular interpolation method for industrial SCARA robots," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 235, nos. 6–7, pp. 1061–1073, 2021.

[2] A. J. Mohammed, K. I. Ghathwan, and Y. Yusof, "Optimal robot path planning using enhanced particle swarm optimization algorithm," *Iraqi J. Sci.*, vol. 61, no. 1, pp. 178–184, Jan. 2020.

[3] T. A. Badmos, P. O. Omolaye, J. Mebawondu, and H. A. Aliyu, "Robot path planning performance evaluation of a dynamic environment," *IOSR J. Electron. Commun. Eng.*, vol. 13, no. 6, pp. 19–26, 2020.

[4] F. Jiang, C. Jin, H. Liao, H. Li, Y. Wu, Y. Liu, J. Peng, and Z. Huang, "An artificial potential field-based lithium-ion battery SOC equilibrium method in electric vehicles," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 12682–12687, 2020.

[5] I. Kholishoh, M. Mardainis, S. Susandri, and K. Andesa, "Geolocation apps using A* algorithm for Android based traders," *IT J. Res. Develop.*, vol. 6, no. 1, pp. 17–29, Mar. 2021.

[6] W. Wang, Y. Chen, and Y. Jia, "Evaluation and optimization of dual-arm robot path planning for human–robot collaborative tasks in smart manufacturing contexts," *ASME Lett. Dyn. Syst. Control*, vol. 1, no. 1, pp. 1–7, Jan. 2021.

[7] J. Ma, Y. Liu, S. Zang, and L. Wang, "Robot path planning based on genetic algorithm fused with continuous Bezier optimization," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–10, Feb. 2020.

[8] L. Gao, R. Liu, F. Wang, W. Wu, B. Bai, S. Yang, and L. Yao, "An advanced quantum optimization algorithm for robot path planning," *J. Circ. Syst. Comput.*, vol. 29, no. 8, pp. 95–143, 2020.

[9] S. D. Han and J. Yu, "DDM: Fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1350–1357, Apr. 2020.

[10] G. Wang and J. Zhou, "Dynamic robot path planning system using neural network," *J. Intell. Fuzzy Syst.*, vol. 40, no. 2, pp. 3055–3063, Feb. 2021.

[11] X. Xu, W. Pan, Y. Huang, and W. Zhang, "Dynamic collision avoidance algorithm for unmanned surface vehicles via layered artificial potential field with collision cone," *J. Navigat.*, vol. 73, no. 6, pp. 1306–1325, Nov. 2020.

[12] M. L. Imrane, A. Melingui, J. J. B. M. Ahanda, F. B. Motto, and R. Merzouki, "Artificial potential field neuro-fuzzy controller for autonomous navigation of mobile robots," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 235, no. 7, pp. 1179–1192, Aug. 2021.

[13] Z. Du, D. Zhao, J. Shi, and J. Lu, "Formation flight in complex environments using an artificial potential field," *J. Aerosp. Inf. Syst.*, vol. 18, no. 3, pp. 1–12, Mar. 2021.

[14] Y. Long, Z. Zuo, Y. Su, J. Li, and H. Zhang, "An A*-based bacterial foraging optimisation algorithm for global path planning of unmanned surface vehicles," *J. Navigat.*, vol. 73, no. 6, pp. 1247–1262, Nov. 2020.

[15] S. Gu, C. Zhou, Y. Wen, X. Zhong, M. Zhu, C. Xiao, and Z. Du, "A motion planning method for unmanned surface vehicle in restricted waters," *Proc. Inst. Mech. Eng., M, J. Eng. Maritime Environ.*, vol. 234, no. 2, pp. 332–345, May 2020.

[16] S. Wu, Y. Du, and Y. Zhang, "Mobile robot path planning based on a generalized wavefront algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–12, Mar. 2020.

[17] A. Dutta, A. Bhattacharya, O. P. Kreidl, A. Ghosh, and P. Dasgupta, "Multi-robot informative path planning in unknown environments through continuous region partitioning," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 6, pp. 2204–2211, 2020.

[18] P. Wang, Z. Huang, F. Zhou, B. Chen, Y. Wu, Y. Liu, F. Li, and J. Peng, "Comfort-aware cooperative cruise control of multiple high-speed trains: An artificial potential field approach," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15223–15228, 2020.

[19] S. Laaroussi, A. Baataoui, A. Halli, and K. Satori, "Dynamic mosaicking: Combining A* algorithm with fractional Brownian motion for an optimal seamline detection," *IET Image Process.*, vol. 14, no. 13, pp. 3169–3180, Nov. 2020.

[20] X. Sun, W. Hu, X. Xue, and J. Dong, "Multi-objective optimization model for planning metro-based underground logistics system network: Nanjing case study," *J. Ind. Manag. Optim.*, vol. 19, no. 1, pp. 170–196, 2023.

[21] Z. Huang, J. Li, J. Huang, J. Ota, and Y. Zhang, "Motion planning for bandaging task with abnormal posture detection and avoidance," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 5, pp. 2364–2375, Oct. 2020.

[22] T. Zhao, G. Zhang, and H. Liang, "Optimal path selection algorithm for UAV formation guided by a UV virtual fence," *Appl. Opt.*, vol. 61, no. 11, pp. 3182–3189, 2022.

[23] K. Chang, D. Ma, X. Han, N. Liu, and P. Zhao, "Lyapunov vector-based formation tracking control for unmanned aerial vehicles with obstacle/collision avoidance," *Trans. Inst. Meas. Control*, vol. 42, no. 5, pp. 942–950, Mar. 2020.

[24] T. Xu, H. Zhou, S. Tan, Z. Li, X. Ju, and Y. Peng, "Mechanical arm obstacle avoidance path planning based on improved artificial potential field method," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 2, pp. 271–279, Feb. 2022.

[25] J. Hu, M. Wang, C. Zhao, Q. Pan, and C. Du, "Formation control and collision avoidance for multi-UAV systems based on Voronoi partition," *Sci. China Technol. Sci.*, vol. 63, no. 1, pp. 69–76, 2020.

[26] Y. Wang, X. Li, and J. Zhang, "Review of wheeled mobile robot collision avoidance under unknown environment," *Sci. Progr.*, vol. 104, no. 3, pp. 160–173.

**MIN ZHUANG** was born in Fujian, China, in 1972. He received the bachelor's degree in computer application technology from Zhejiang University, in 2000, and the master's degree from Asian City University, in 2018. From 2000 to 2016, he was an Engineer with Hangzhou Radio and Television University. Since 2016, he has been an Associate Professor with Hangzhou Polytechnic. Ten papers were published, of which two were SCI-authored. His research interests include intelligent manufacturing equipment technology, manipulator, and algorithm.

**GE LI** was born in Zhejiang, in 1958. He received the bachelor's and master's degrees from Xinjiang Agricultural University, in 1982 and 1986, respectively, and the Ph.D. degree in mechanical engineering from Zhejiang University, in 2007. From 1982 to 2000, he was a Lecturer and an Associate Professor with Xinjiang Agricultural University. Since 2000, he has been a Professor with the School of Mechanical and Automatic Control, Zhejiang Sci-Tech University. He has published 20 papers, five of which were included in Ei compendex. His research interests include mechanical optimization design and mechanism control.

**KEXIN DING** was born in Henan, China, in 1994. He received the bachelor's degree majored in electronic science and technology from Xinyang Normal University, in 2017, and the master's degree majored in control science and engineering from the Zhejiang University of Technology, in 2020. Currently, he is a Lecturer with Hangzhou Polytechnic. Six papers were published, of which three were SCI-authored. His research interests include intelligent control technology and algorithm.