International Journal for Housing Science and Its Applications

Publish September 8, 2025. Volume 47, Issue 1 Pages 83-97

https://doi.org/10.70517/ijhsa47107

Intelligent path planning technology for dual-machine collaborative forestry manipulators based on high-precision tree segmentation

Hao Sun[1], Yanhu Zhu[1], Buyong Ren[1], Sifan Chen[1] and Ziqiang Guo[1,*]

[1] Linxia Power Supply Company, State Grid Corporation of China, Linxia, Gansu, 731800, China

Corresponding authors: (e-mail: zzaazz112@163.com).

Abstract To achieve path planning for dual-robot collaboration in forestry robotics, this paper uses visual imaging principles and image processing methods to locate and segment tree images. Then, a kinematic model of the forestry robot is constructed and solved. The Shoal Optimization Algorithm (SHO) and Informed-RRT* algorithm are combined to propose the SHO-Informed-RRT* algorithm for path planning in dual-robot collaboration in forestry robotics. Through simulation experiments, it is found that compared with other methods, the proposed SHO-Informed-RRT* algorithm performs best in terms of path planning time, planning speed, path length, average number of sampling points, and number of path nodes. After adopting the SHO-Informed-RRT* algorithm, the mechanical arm of the forestry robot achieves the optimal set target, and the average movement time of the mechanical arm is the shortest.

Index Terms image processing, path planning, SHO-Informed-RRT* algorithm, joint motion

I. Introduction

Forestry operations refer to the use of tools and methods to interact with forests in order to obtain forest products, serving as a crucial means for humans to access forest resources [1]. Given China's vast forest areas, the forestry economy has become an important component of the national economy. However, with societal development and changes in economic structure, the labor force has significantly decreased, resulting in a scarcity of workers engaged in forestry [2], [3]. Additionally, forest work environments are harsh and labor-intensive, making it difficult for manual labor to efficiently complete related tasks [4], [5]. Robots can effectively address these issues by replacing time-consuming and labor-intensive manual labor with mechanized and automated robots, thereby reducing labor costs and ensuring the safety of forestry workers [6]-[8]. Furthermore, the development of intelligent forestry robots holds even greater significance for promoting forestry modernization [9].

Traditional forestry machinery faces performance limitations in complex forest environments and can no longer meet operational requirements. Furthermore, rising labor costs have led to increasing forestry operation costs in recent years [10]-[12]. Therefore, research on dual-robot collaboration forestry manipulators is a key means to enhance forestry operation quality and efficiency. As a highly integrated mechatronic system, the operation planning of forestry robotic arms is a critical step in the operations of fruit-picking robots and industrial robots, enabling improvements in operational efficiency and safety [13]-[16]. Considering the many uncertainties in the system, intelligent planning of the robotic arm's movement trajectories is necessary for different environments and conditions [17], [18].

Currently, with the development of robotics technology, forestry robots have been widely researched and applied in almost all forestry operation fields. However, due to the complexity of forestry operation scenarios and the diverse working forms in forestry environments, researchers have also conducted studies on forestry robots tailored to different application scenarios. Reference [19] designed a multi-arm collaborative control strategy for apple harvesting, utilizing a Markov game framework to optimize the target harvesting sequence of the working arms, thereby enhancing the harvesting efficiency of forestry robots. Literature [20] developed a telescopic arm intelligent optimization system for large-scale fruit tree harvesting operations. It utilizes a hybrid neural-fuzzy application interface to optimize the kinematic and dynamic behavior knowledge of the robotic arm in the workspace, thereby obtaining the optimal action path for the robotic arm. Literature [21] proposes a dynamic planning method for multi-arm picking robots that integrates long short-term memory (LSTM) networks and proximal policy optimization (PPO), enabling the robotic arms to exhibit better adaptability, fault tolerance, and collaborative execution efficiency in complex fruit tree environments. Literature [22] introduces a system that uses a legged harvester to autonomously perform precise harvesting tasks. It relies on path planning algorithms and sensor data to plan the harvester's movement and gripping actions, enabling it to maintain good tree-gripping performance in rugged terrain and complex environments.

Literature [23] indicates that collision-free collaboration, task sequence optimization, and dynamic re-planning are fundamental requirements for multi-arm collaborative work in forestry robots. To this end, multi-arm task planning is modeled as a Markov game, and deep reinforcement learning is introduced to enhance the accuracy of robotic arm action decisions. Reference [24] proposes a rubber intelligent tapping device based on a mobile robot platform. It uses the Gauss-Newton method to fit tree point cloud data to plan the robot's action path and employs the extended Kalman filter (EKF) algorithm for accurate robot localization to meet the requirements of the robotic arm's rubber collection task. Reference [25] improves forestry operation robots using an IoT-based smart forest and navigation system, enabling them to perform various tasks in forest environments more efficiently and sustainably. It is evident that existing intelligent planning methods for forestry robotic arms heavily rely on visual sensors to scan and analyze the working environment of forestry operations. Therefore, researching and designing more optimized and intelligent tree segmentation algorithms to obtain the complete three-dimensional structure of the forest is of great significance for multi-robotic arm collaborative path planning.

This paper first uses visual imaging and image processing methods to achieve high-precision localization and image segmentation of trees. Then, a kinematic model is established for the dual forestry robots, and forward and inverse kinematics are solved. After preliminary path planning for the forestry robots, a collision detection model is established. By integrating the Stable Hippocampus Optimization (SHO) algorithm and the Informed-RRT* algorithm, the SHO-Informed-RRT* algorithm is constructed to further enhance the path planning efficiency of dual-robot collaboration in forestry operations. To test the effectiveness of the proposed SHO-Informed-RRT* algorithm, simulation experiments are conducted. The proposed algorithm is compared with other planning methods in terms of path and joint aspects through simulation tests, evaluating the advantages and disadvantages of each method based on planning time, path length, and node count.

II. Visual imaging and image processing

II. A. Principles of Visual Imaging

Coordinate transformation can effectively convert observed three-dimensional object information into more precise three-dimensional coordinates, enabling the visual system to identify objects with greater accuracy. Assuming a point P is located in a specific space, its coordinates can be represented as P(X, Y, Z). A camera coordinate system can be established with the camera's center point as the origin, where the X and Y coordinate axes are parallel to the sides of the image, and the Z axis is perpendicular to the image, pointing directly toward the front of the lens. The spatial point P is represented as $P(X_c, Y_c, Z_c)$

By specifying a reference coordinate system, the position of an object can be transformed from the world coordinate system to the camera coordinate system, including rotations and translations of the coordinate system. The transformed coordinates are shown in Equation (1):

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{1}$$

The above equation can be simplified to the following equation (2):

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

Among them, $R = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$, $T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$, where R is the rotation matrix and T is the translation matrix, both of which are external parameters of the camera and are unrelated to each other.

Based on the coordinates of the image center, establish an image coordinate system xoy, in which the x and y coordinate axes are parallel to the length and width directions of the image. A perspective projection image is obtained using the pinhole imaging principle. In this image coordinate system, the projection of point $P(X_c, Y_c, Z_c)$ is $P(x, y)$.

According to the similarity triangle theorem, equation (3) can be derived:

$$\frac{X_c}{Z_c} = \frac{x}{f}, \frac{Y_c}{Z_c} = \frac{y}{f} \tag{3}$$

Write the above equation in matrix form as shown in (4):

$$Z_c\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{4}$$

Images captured by a camera exist in the form of pixels, and their pixel coordinate system uov needs to be converted to the image coordinate system xoy. The origin coordinates can be represented by (u_0, v_0) .

The conversion matrix between the pixel coordinate system and the image coordinate system is shown in formula (5):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

In this context, d_x, d_y represent the physical dimensions of the pixel, and u_0 and v_0 are the pixel coordinates.

In summary, the conversion relationship between the world coordinate system and the image coordinate system is shown in Equation (6):

$$z_c\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} R & T \end{pmatrix}\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \tag{6}$$

It can be simplified as follows: $z_c\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = M_1 M_2 \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$, where the key variables are shown in equation (7):

$$M_1 = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}; \quad M_2 = \begin{pmatrix} R & T \end{pmatrix} \tag{7}$$

In this context, u, v represent arbitrary points in the pixel coordinate system, while u_0, v_0 indicate the pixel coordinates of the center point of the image, z_c represents the distance from the camera to the object, and M_1 is the camera's internal parameter matrix, where f denotes the focal length of the lens, which is the vertical distance from the lens to the image plane, dx, dy respectively indicate the length and width of a single pixel point in the image coordinate system; M_2 is the external parameter matrix, which assists the system in more conveniently measuring the distance between objects. Generally, during the calculation process, the world coordinate system and the camera coordinate system are aligned to calculate the distance between objects. If they are the same, the result shown in Equation (8) can be obtained:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

II. B. Image processing methods

II. B. 1) Simple image processing

(1) Image grayscaling

Using grayscale images instead of traditional color images can significantly speed up the image processing process, thereby better completing recognition and positioning tasks.

(2) Image enhancement

Through image enhancement technology, the grayscale values of images can be changed to improve image contrast. This technology can effectively enhance image clarity, making the target in the image more prominent and easier to extract features from.

II. B. 2) Image Filter Operator Analysis

(1) Mean filter

The mean filter has good linear characteristics [26] and can effectively extract information from data using the neighborhood averaging method. When a pixel is affected by noise, its original gray value will deviate significantly from the surrounding values. By using their values as substitute values, a "smoothed" effect can be obtained.

The pixel (x, y) to be processed is mapped to a specific template. By calculating the template, the average value of each pixel in the entire image can be obtained. The calculated value replaces the grayscale value of each pixel. The grayscale value of pixel (x, y) can be calculated using the method in equation (9):

$$\begin{aligned} f(x,y)=\frac{1}{8}\big[&f(x-1,y-1)+f(x-1,y)+f(x-1,y-1)+f(x,y-1)\\ &+f(x,y+1)+f(x+1,y-1)+f(x+1,y)+f(x+1,y+1)\big] \end{aligned} \tag{9}$$

(2) Gaussian filter

The design concept of the Gaussian filter is to construct an accurate and reliable mathematical model by applying the Gaussian function [27]. After processing with this model, image data can be effectively converted into low-energy information, thereby effectively suppressing noise and ultimately achieving effective denoising of image data. During the filtering process, features such as image edges and corners become blurred, which leads to energy loss.

The Gaussian function is shown in Equation (10):

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{10}$$

As the variance σ increases, the Gaussian kernel window of the discrete filter also becomes larger, covering a wider range. By using a MATLAB program to create a 3×3 discrete Gaussian kernel, efficient calculations can be achieved. The function is shown in equation (11) below:

$$filter = fespecial(gaussian',3,1) \tag{11}$$

The Gaussian matrix is shown in equation (12) below:

$$filter=\begin{bmatrix} 0.0751 & 0.1238 & 0.0751\\ 0.1238 & 0.2042 & 0.1238\\ 0.0751 & 0.1238 & 0.0751 \end{bmatrix} \tag{12}$$

(3) Median filtering

Median filters aim to obtain optimal image quality by comparing the gray values of pixels surrounding each pixel point to determine the pixel closest to the center value. By using median filtering, noise can be effectively suppressed, thereby avoiding the problem of blurred edge contours. The basic process is as follows:

The two-dimensional median filter output is given by formula (13):

$$g(x,y)=med\{f(x-k,y-1),k,1\in W\} \tag{13}$$

$f(x,y)$ represents the original image, $g(x,y)$ represents the processed image, and l and k are used to indicate the sampling window size.

II. B. 3) Image segmentation technology

(1) Edge detection method—Canny edge detection method [28]

The specific steps of the classic Canny operator edge detection are as follows:

Step 1: Use a two-dimensional Gaussian plane to achieve surface flatness. Here, assume that $f(x,y)$ is the input image, $G(x,y)$ represents the Gaussian function, and the smoothed image is represented by $f_s(x,y)$:

$$\begin{cases} G(x,y,\sigma)=\dfrac{1}{2\pi\sigma^2}\exp(-\dfrac{x^2+y^2}{2\sigma^2}) \\ f_s(x,y)=f(x,y)*G(x,y) \end{cases} \tag{14}$$

Step 2: Gradient solution. By using the 2×2 neighborhood finite difference mean to calculate the gradient, effective edge detection can be performed on the smoothed image $f_s(x,y)$, and the partial derivatives of x and y can be calculated using this method, as shown in formula (15):

$$\begin{cases} g_x[i,j]=(f_s[i+1,j]-f[i,j]_s+f_s[i+1,j+1]-f_s[i,j+1])/2 \\ g_y[i,j]=(f_s[i,j+1]-f_s[i,j]+f_s[i+1,j+1]-f_s[i+1,j])/2 \end{cases} \tag{15}$$

By applying the calculation technique of the second norm, it can be seen that there is a close relationship between the magnitude M of the gradient and its direction, as shown in formula (16):

$$\begin{gathered} M[i,j]=\sqrt{g_x[i,j]^2+g_y[i,j]^2} \\ \theta[i,j]=\arctan(g_x/g_y) \end{gathered} \tag{16}$$

Step 3: Suppress non-maximum values. After suppression, candidate edge points can be recorded.

Step 4: Adjust the high and low thresholds. By adjusting the thresholds, the discontinuous regions in the image can be stitched together to ensure edge continuity.

(2) Threshold segmentation method - maximum interclass variance method

By setting the threshold of the grayscale histogram, the image can be easily segmented. When there is a significant difference in grayscale between the foreground and background in the image, threshold segmentation technology can effectively achieve the desired results.

Through statistical analysis, it can be observed that variance is an important metric for measuring the differences between two categories of data. Using the maximum interclass variance method, an image can be divided into “background” and “object” regions. The segmentation threshold T is determined by calculating the grayscale value t_k of each pixel, thereby achieving effective image segmentation. If an image has dimensions of $M\times N$, its grayscale values can be divided into L levels, namely $\{0,1,2,\ldots,L-1\}$ respectively. n_i denotes the number of pixels with grayscale level i, where $i\in[0,L-1]$. It follows that $MN=n_1+n_2+n_3+\cdots+n_{L-1}$. The process is as follows:

First: Histogram normalization as in equation (17):

$$\begin{gathered} p_i=n_i/MN \\ \sum_{i=0}^{L-1}p_i=1, p_i\geq 0 \end{gathered} \tag{17}$$

Second: Set a threshold of $t(i)=i$, and divide the image into two categories, A and B, based on $t(i)$. The probabilities of categories A and B appearing are shown in equation (18) below:

$$P_A=\sum_{i=0}^{T}P_i, P_B=\sum_{i=T+1}^{L-1}P_i=1-P_A \tag{18}$$

If A is a background with a grayscale level between 0 and N, then the value of P_B can be obtained by analyzing the elements in the histogram. This value reflects the probability of the background appearing. If B is a target, then the value of P_B is the probability of it appearing.

$$\begin{cases} W_A=\sum_{i=0}^{T}i\cdot\dfrac{P_i}{P_A} \\ W_B=\sum_{i=T+1}^{L-1}i\cdot\dfrac{P_i}{P_B} \end{cases} \tag{19}$$

$$W_0=P_A\cdot W_A+P_B\cdot W_B=\sum_{i=0}^{L-1}i\cdot P_i \tag{20}$$

$$\sigma^2=P_A\cdot(W_A-W_0)^2+P_B\cdot(W_B-W_0)^2 \tag{21}$$

W_A and W_B represent the grayscale values of regions A and B, respectively, while w_0 is the average value of the entire grayscale image. From the various formulas in (17) to (21), it is found that the size of the inter-class variance σ^2 is closely related to the segmentation threshold T, whose value range is L-1. Therefore, the grayscale value that maximizes the inter-class variance can be obtained through enumeration.

III. Forestry robot path planning

III. A. Establishment of Kinematic Models

The prototype platform system is realized through the collaborative operation of two Eft-ER20 six-degree-of-freedom forestry robots with identical model parameters for hub grinding. One robot holds the hub, while the other holds the grinding tool and follows the grinding trajectory points in real time. To describe the motion relationship between adjacent linkages, this paper adopts the improved D-H method to establish the coordinate systems of each linkage, as shown in Figure 1. Description as follows: The coordinate origin O_i is located at the intersection of the common perpendicular line a_i and the axis of joint J_i; the z_i axis coincides with the axis of joint J_i, and its direction can be arbitrarily chosen; the x_i axis coincides with a_i and points toward the J_{i+1} axis; the y_i axis is determined by the right-hand rule. (x_0, y_0, z_0) is the world coordinate system, $(x_{10}, y_{10}, z_{10}), (x_{R0}, y_{R0}, z_{R0})$ are the base coordinate systems of the left and right robots, respectively, and the distance between the two robots is 2.5 m.

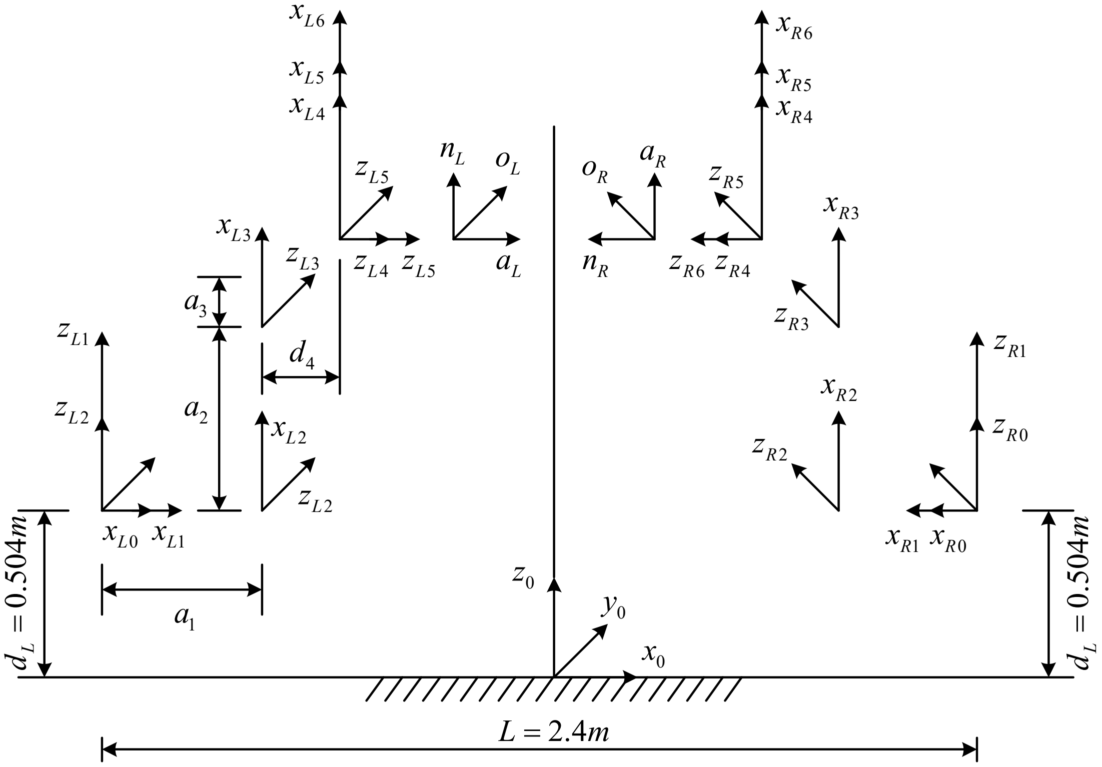


Figure 1: Double robot coordinate system

III. B. Kinematic solution for dual robots

III. B. 1) Solving for positive kinematics

The forward kinematics problem involves solving for the pose of the end effector in the reference coordinate system, given the known motion angles of each motion axis and the link parameters. Let ${}^{i-1}_{i}T$ be the transformation matrix of the link coordinate system $\{i\}$ relative to $\{i-1\}$, then we have:

$$\begin{gathered} {}^{i-1}_{i}T = Rot(x, \alpha_{i-1}) Trans(x, \alpha_{i-1}) Rot(z, \theta_i) \\ Trans(z, d_i)(i = 1,2,3,4,5,6) \end{gathered} \tag{22}$$

From equation (22), we obtain the general formula:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23} $$

Among them, $c = \cos, s = \sin$.

Taking a gripping robot as an example, the kinematic solution is obtained as follows:

$$ \begin{gathered} {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{5}_{6}T = \begin{bmatrix} c\theta_5 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{24} $$

The kinematic equation is:

$$ {}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T \tag{25} $$

Substituting the variables of each joint into equation (24) yields the forward kinematics of the gripping robot.

Let the world coordinate system be denoted by b. Then, ${}^{b}_{0}T$ is the transformation matrix of the gripping robot relative to the world coordinate system, and ${}^{b}_{7}T$ is the transformation matrix of the grinding robot relative to the world coordinate system. By analyzing the relationship between the left and right robot base coordinate systems and the world coordinate system, the transformation matrix from the base coordinate system to the world coordinate system can be obtained, i.e.:

$$ {}^{b}_{0}T = \begin{bmatrix} 1 & 0 & 0 & -1.2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.504 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{b}_{7}T = \begin{bmatrix} 1 & 0 & 0 & 1.2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0.504 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{26} $$

When performing grinding operations, the robot needs to determine the pose of the wheel hub in its own base coordinate system, which requires further calculation of ${}^{b}_{0}T^{-1}$ and ${}^{b}_{7}T^{-1}$. This allows the robot to determine the coordinates of the end-effector in its own base coordinate system based on its pose in the world coordinate system, and then use inverse kinematics to calculate the angular values of each joint's motion.

III. B. 2) Solving inverse kinematics

The inverse kinematics problem involves determining the angle values of each joint when reaching the desired pose, given the known parameters of the linkages and the desired pose of the end effector relative to the reference coordinate system. This problem is primarily solved using geometric methods, inverse transformation methods, and analytical methods. The three axes of the Effort-ER20 intersect at a single point and have a vector closed solution, so this paper uses the inverse transformation method to solve the inverse kinematics problem:

$$ {}^{b}_{e}T = {}^{b}_{0}T({}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T)\,{}^{6}_{e}T \tag{27} $$

Let the position-orientation matrix of the gripping robot's end effector be:

$$P_1 = \begin{bmatrix} n_x & o_x & a_x & p_{0x} \\ n_y & o_y & a_y & p_{0y} \\ n_z & o_z & a_z & p_{0z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{28}$$

Then there are:

$${}_0^eT_e^{-1}{}^0T\,{}_0^eT^{-1} = {}_0^eT^{-1}P_1^6\,{}_eT^{-1} = {}_1^0T\,{}_2^1T\,{}_3^2T\,{}_4^3T\,{}_5^4T\,{}_6^5T = \begin{bmatrix} n_x & o_x & a_x & p_{0x} \\ n_y & o_y & a_y & p_{0y} \\ n_z & o_z & a_z & p_{0z} \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}_6^0T \tag{29}$$

III. C. Basic Principles of Path Planning

III. C. 1) Path Planning Definition

This study defines path planning as follows: Let $X \subseteq R$ be the state space for motion planning. Define the set $X_{obs} \subset X$ as the obstacle space, and $X_{free} = X \setminus X_{obs}$ as the obstacle-free space, i.e., the free space. In the free space, $X_{start} \in X_{free}$ is defined as the starting position, and $X_{goal} \in X_{free}$ is defined as the target position. The motion planning problem is to find a collision-free path in the space X that starts from the initial position $\sigma(0) = X_{\text{start}}$ and reaches the target position $\sigma(1) = X_{goal}$, defined as the set $\sigma[0,1] \to X_{free}$.

Define $c(\sigma)$ as a cost function that maps each collision-free path to a non-negative real number. The process of algorithmic gradual optimization is defined as: in an elliptical space with the major axis length as the initial path length, the process of iteratively updating the path with the lowest cost through the algorithm. Let the optimal path cost function be σ^*, i.e.:

$$\sigma^* = \arg\min_{\sigma \in \Sigma} \left\{ c(\sigma) \,\middle|\, \begin{array}{c} \sigma(0) = X_{start}, \sigma(1) = X_{goal}, \\ \forall s \in [0,1], \sigma(s) \in X_{free} \end{array} \right\} \tag{30}$$

III. C. 2) Collision Detection Model

By performing collision detection on each joint, determine whether the robotic arm has collided with an obstacle. Simplify the spatial model using geometric enveloping methods: use a cylindrical envelope for the robotic arm's linkages, a spherical envelope for spherical obstacles in space, and an axial bounding box for rectangular obstacles.

Collision detection between the robotic arm and spherical enveloped spatial obstacles can be simplified to calculating the distance between the centerline of the cylinder and the center of the obstacle sphere.

Define the three-dimensional spatial coordinates of the spherical obstacle M_o as:

$$M_o = (X_{Mo}, Y_{Mo}, Z_{Mo}) \tag{31}$$

Define the three-dimensional coordinates of the perpendicular foot from the center line of the cylinder to the center of the obstacle ball as M_v:

$$M_v = (X_{Mv}, Y_{Mv}, Z_{Mv}) \tag{32}$$

The distance between the robotic arm and the obstacle d is:

$$d = |M_v M_o| = \sqrt{(X_{Mv} - X_{Mo})^2 + (Y_{Mv} - Y_{Mo})^2 + (Z_{Mv} - Z_{Mo})^2} \tag{33}$$

When the sum of the radius of the spherical obstacle and the radius r of the cylinder is greater than or equal to d, a collision is considered to have occurred; otherwise, no collision is considered to have occurred, expressed as:

$$\begin{cases} d \le r_c + r_o, \text{Collision} \\ d > r_c + r_o, \text{No collision} \end{cases} \tag{34}$$

For axial bounding box collision detection, if segment A of the robotic arm is outside the axial bounding box of the obstacle, no collision is considered to have occurred; otherwise, a collision is considered to have occurred. Depending on the specific circumstances of the obstacle, collision detection for the robotic arm's path planning is performed using formulas (31) to (34).

III. D. Hippocampus Optimization Algorithm

The SHO algorithm is a heuristic optimization algorithm based on the behavior of seahorses in nature [29], which aims to solve various optimization problems. The SHO algorithm combines local search and global search strategies and has the advantages of stronger global search capabilities, high adaptability, high efficiency, and ease of adjustment.

III. D. 1) Movement Behavior

The SHO algorithm simulates two different movement patterns of the seahorse: one for global search and the other for local exploration, enabling flexible search of complex search spaces. In local exploration mode, the seahorse's movement toward the elite individual xtelite (the current population's optimal individual) is simulated using Levy flight, prompting the algorithm to cover different regions with early incremental probability during iteration, thereby helping to avoid local sparsity.

The SHO local exploration position update formula is:

$$x_i^l = x_i^t + Levy(\lambda)[(x_{elite}^t - x_i^t)xyz + x_{elite}^t] \tag{35}$$

In the equation, $x = \rho\cos\theta$, $y = \rho\sin\theta$, and $z = \rho\theta$ represent the three-dimensional components of the coordinates under spiral motion, which helps update the search position of the hippocampus; $\rho = u' \times e^{\theta v}$ represents the length of the radius defined by the logarithmic spiral constant u' and v; θ is a random value in the range [0, 2 π] rad; Levy(λ) is the Levy flight distribution function; λ is a random number in the range [0, 2]; $t \in [0, n]; l \in [1, n+1]$.

In global search mode, random movement helps to better explore the search space.

The SHO global search position update formula is:

$$x_i^l = x_i^t + \mathrm{rand}(0,1) l \beta_t (x_i^t - \beta_t x_{\mathrm{elite}}^t) \tag{36}$$

In the equation, l is a constant coefficient; β_l is the random walk coefficient of Brownian motion, which is essentially a random value that follows a standard normal distribution.

III. D. 2) Predatory behavior

Predatory behavior plays a key role in the SHO algorithm, adjusting the algorithm's behavior by simulating the success or failure of predation. The mathematical expression for predatory behavior is:

$$x_i^2 = \begin{cases} \alpha[x_{elite}^t - rand(0,1)x_i^1] + (1-\alpha)x_{elite}^t, r_2 > 0.1 \\ (1-\alpha)[x_i^1 - rand(0,1)(x_{elite}^t)] + \alpha x_i^1, \text{other} \end{cases} \tag{37}$$

In the equation, x_i^1 represents the new position of the seahorse after performing a movement behavior, i.e., first completing the movement described in Section 2-1, then using the newly obtained position to perform predation, resulting in the new position x_i^2; r_2 is a random number in the range [0, 1]; α decreases linearly with the number of iterations to adjust the movement step length of the seahorse when hunting prey, i.e.:

$$\alpha = (1 - \frac{t}{T})^{\frac{2t}{T}} \tag{38}$$

Among them, T is the maximum number of iterations.

III. D. 3) Proliferation Behavior

The proliferation behavior selected by SHO simulates natural processes. Different movement patterns approximately follow a normal distribution. Taking $r_i = 0$ as the boundary point, the individuals with the best fitness are selected as parents to generate new individuals, and the best individual x_{elite}^l is selected. This helps maintain the diversity of the search and improves the performance of the improvement.

III. E. Informed-RRT algorithm integrated with SHO*

By integrating the SHO algorithm and the Informed-RRT* algorithm [30], it is expected that a global optimal solution can be found more quickly and stably during the path planning process, thereby further improving path planning efficiency.

III. E. 1) Optimizing Sampling Strategies

Traditional path planning algorithms typically generate sample points randomly throughout the entire environment, but this often results in wasted time. The improved sampling strategy is more oriented toward the target direction. In the early stages of sampling, the direction between the target point and the initial point is given special consideration, which helps to generate sample points closer to the target point throughout the search process, thereby narrowing the search range. Additionally, a trade-off mechanism is employed to limit the maximum distance of sampling points in the direction toward the target, preventing abrupt jumps and balancing the breadth and depth of the search. Simultaneously, randomness is increased within a certain range to enhance path diversity, enabling better adaptation to complex environmental applications.

III. E. 2) Obstacle avoidance design

In path planning, obstacle avoidance design is critical to ensuring the safe, efficient, and reliable operation of the system. The algorithm in this paper optimizes and improves the strategy for handling obstacles encountered, thereby enhancing the efficiency of path planning.

First, node information is introduced to enable the algorithm to capture obstacles around the robot in real time. This allows nodes to continuously detect collision risks during movement and perform real-time collision detection during path planning. Although the sampling range is restricted in the initial stage to be closer to the target point, obstacles are still inevitably encountered during operation. The new node x_{new} is generated from the starting node x_{start} to the target node x_{goal}. First, find the node $x_{nearest}$ closest to x_{rand}, then extend a certain distance l_c along the direction from x_{rand} to $x_{nearest}$ to generate the new node, Finally, $x_{nearest}$ and x_{new} are connected to form a straight line L. To explore obstacles in the environment, mathematical formulas and geometric calculation methods can be used to handle encountered obstacle segments.

Second, this paper employs a method of randomly perturbing node positions for collision detection, as shown in Figure 2. If a collision is detected, the system first attempts to generate a new node position by randomly perturbing the current position to avoid the collision. Simultaneously, the system repeatedly attempts to generate new node positions until a position that does not collide with the obstacle is found or the predetermined iteration count is reached, at which point the search for nodes is restarted.

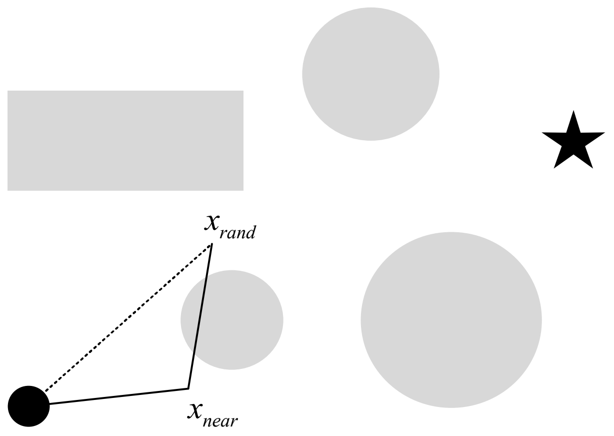


Figure 2: Collision detection

The probability density function of the normal distribution of the random disturbance term is:

$$f(x;\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{39}$$

In this case, μ is the expected value of the normal distribution; σ is the standard deviation of the normal distribution.

III. E. 3) Reference to the SHO algorithm

In the Informed-RRT* algorithm, the performance of path planning is further optimized by incorporating the principles of the SHO algorithm. The traditional Informed-RRT* algorithm requires the continuous generation and connection of nodes, but in high-dimensional spaces or complex environments, this leads to a rapid increase in computational complexity and may result in paths with significant randomness. To overcome these challenges, the predatory behavior of the SHO algorithm is introduced, making path planning more efficient and reliable.

By incorporating predator nodes from the SHO algorithm into the Informed-RRT* algorithm, these predator nodes act as the most fit individuals in the population, naturally guiding other predator nodes toward the target point. Thus, the introduction of predator nodes enhances the directionality of path exploration, making paths more likely to reach the target point, thereby reducing the search space and improving path planning efficiency.

Second, predator nodes can respond more flexibly when encountering obstacles, promptly adjusting paths to proactively avoid collisions, thereby reducing potential risks. This adaptability and flexibility enable the algorithm to better adapt to complex environments, including crowded environments and sudden events, without relying on predefined paths.

In the improved algorithm proposed in this paper, predator nodes are continuously updated, similar to the proliferation behavior in the SHO algorithm. This update mechanism enables the algorithm to continuously optimize paths, especially when encountering insurmountable obstacles upon reaching the target point. Predator nodes can act as global search agents to help find alternative paths. Therefore, the integration of the SHO algorithm with the Informed-RRT* algorithm not only improves path planning efficiency and reliability but also enhances the algorithm's ability to handle complex situations, enabling it to better address various challenges.

IV. Analysis of simulation results

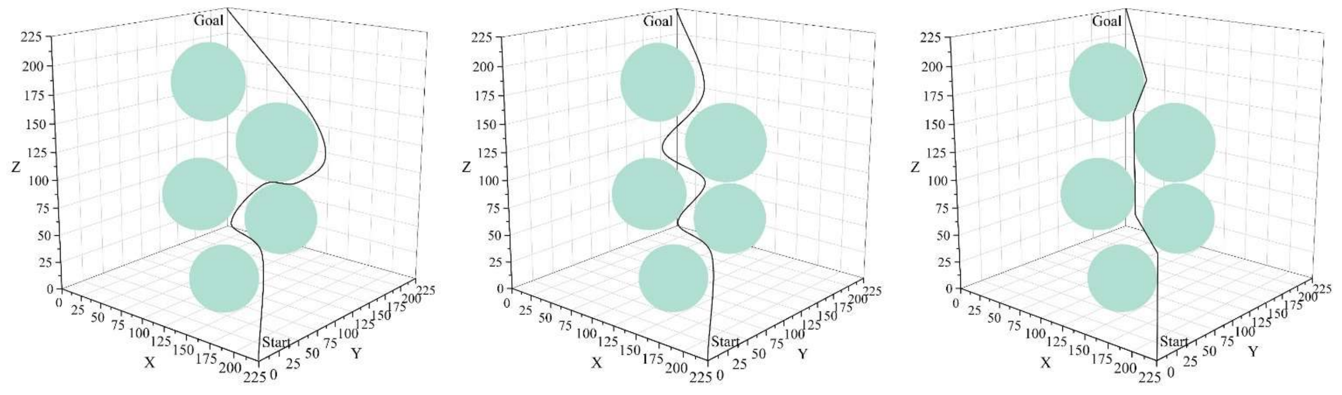
IV. A. Path simulation experiment

To validate the effectiveness and feasibility of the improved Informed-RRT* algorithm, simulation experiments were conducted on the Matlab platform for both the improved and original Informed-RRT* algorithms, as well as the related RRT algorithm. The simulation experiment platform was a DELL Inspiron 5577, with Matlab version R2018b.

First, simulation experiments were conducted in a three-dimensional space, with two types of maps established. One was a simple environment, where there were no obstacles on the connection line between the starting point and the endpoint or in the space, as shown in Figure 3. The other was a complex environment, as shown in Figure 4, where (a) to (c) represent the path planning effects of the RRT* algorithm, the Informed-RRT* algorithm, and the SHO-Informed-RRT* algorithm, respectively. The improved algorithms were compared with the RRT* algorithm and the Informed-RRT* algorithm. The average path length, average search time, average number of path sampling points, and average number of path nodes for each set of experiments were compared, with the results shown in Table 1. The spatial setting is 225×225×225, with obstacle regions randomly set as spherical areas. Paths are represented by thick solid lines. The starting point coordinates are set to (225, 0, 0), and the target point coordinates are (0, 225, 225). The expansion step size is 10, the threshold is set to 10, and the maximum iteration count is 10,000. Each experiment was conducted 50 times, and the average value was calculated for comparison to reduce the impact of randomness on the experimental results.

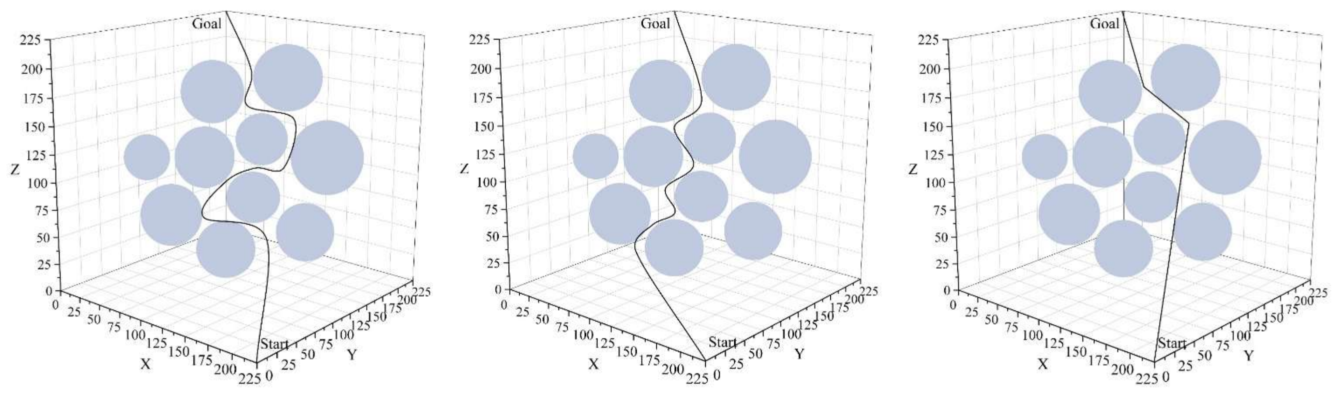
Table 1 shows that, under this map condition, the improved algorithm takes the least time, has the fastest path planning speed, and produces the shortest path length compared to other algorithms. It also has the fewest average sampling points and path nodes. In an obstacle-free environment or when the line connecting the starting point and the destination point is obstacle-free, unnecessary searches are avoided. Analyzing the results in Table 1, it can be concluded that compared to other algorithms, the improved algorithm (SHO-Informed-RRT*) spends the least amount of time on path planning, has the fastest speed, the shortest planned path length, and the fewest average sampling points and path nodes. The improved algorithm (SHO-Informed-RRT*) reduces the average path search time by 80.45% and 94.04%, the average path length by 44.64% and 16.26%, the average number of sampling points by 98.69% and 98.95%, and the average number of path points by 93.12% and 84.27%, respectively.

The results show that the improved algorithm reduces the number of path nodes while obtaining shorter paths. Compared with other algorithms, the improved algorithm has shorter search times and smoother paths, proving the superiority of the improved algorithm.



(a) RRT* algorithm (b) Informed-RRT* algorithm (c) SHO-Informed-RRT* algorithm

Figure 3: Path search diagram of three algorithms in map 1



(a) RRT* algorithm (b) Informed-RRT* algorithm (c) SHO-Informed-RRT* algorithm

Figure 4: Path search diagram of three algorithms in map 2

Table 1: DH parameters of the manipulator

Map	Algorithm	Search time/s	Path length	Sampling point number	Path node number
Map 1	RRT*	3.516528	593.5623	1824.53	62.31
	Informed-RRT*	10.156567	405.651	2487.24	23.16
	SHO-Informed-RRT*	0.002035	325.749	4	6
Map 2	RRT*	3.315597	598.168	1854.29	58.12
	Informed-RRT*	10.875264	395.486	2314.24	25.43
	SHO-Informed-RRT*	0.648342	331.172	24.35	4

Then, to demonstrate the effectiveness of the improved algorithm in this paper, a model of the robotic arm was created using the Matlab software platform. In three-dimensional space, obstacles were placed at the coordinates (120, 40, 60), (150, -25, 75), and (90, 150, 80), with each obstacle represented by a sphere with a radius of 40. The results show that the robotic arm can effectively avoid obstacles in the space and obtain a collision-free path under the improved algorithm, making it an effective path planning algorithm.

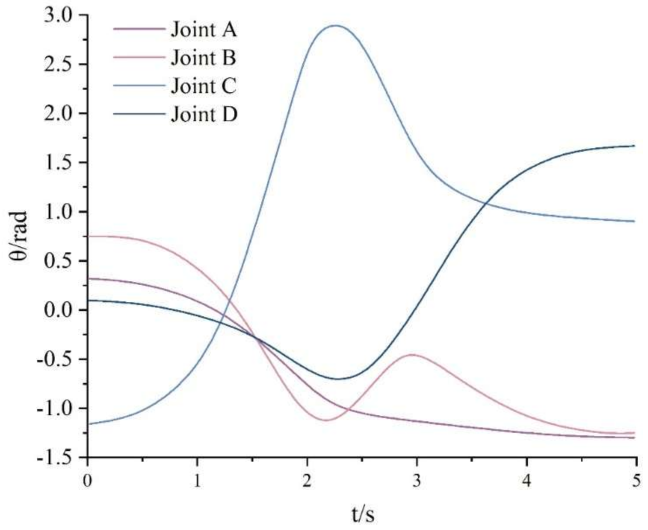
This paper addresses the issue of path planning using the RRT* algorithm by proposing an improved algorithm. It employs a direct connection method to determine whether the starting point and target point can be reached directly, thereby avoiding unnecessary searches when there are no obstacles between the starting point and target point. It then uses a target bias strategy to enhance path search speed and finally simplifies the obtained path by removing redundant points. The results show that the improved algorithm enhances the convergence speed of the algorithm, effectively eliminates redundant nodes in the path, and generates shorter and smoother paths. Therefore, the improved algorithm proposed in this paper is effective and feasible.

IV. B. Joint simulation experiment

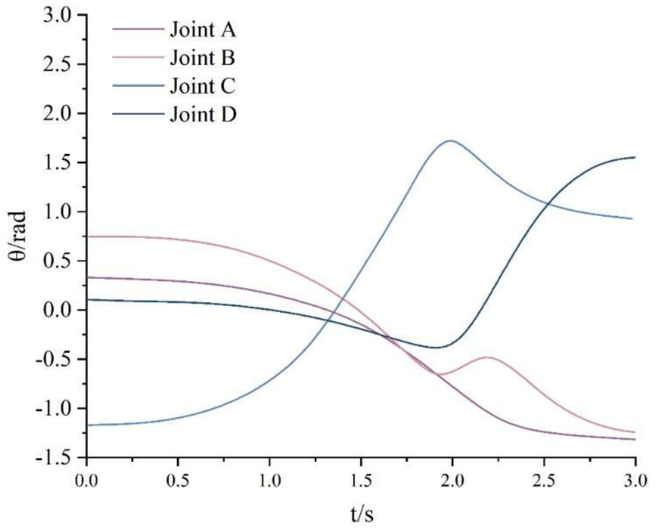
By simulating the motion time of each joint of the forestry robot before and after improving the Hippocampus Optimization Algorithm (referred to as the algorithm used), the angle change curves, angular velocity change curves, and angular acceleration change curves of each joint of the robotic arm before and after improving the algorithm used are shown in Figures 5 to 7, respectively.

As shown in Figure 5, the differences in the angular changes of the robot arm's joints before and after the improvement of the algorithm are relatively small. As shown in Figure 6, the differences in the angular velocity changes of the robot arm's joints before and after the improvement of the algorithm are significant, particularly for joint D, which exhibits a notable increase in maximum angular velocity after the improvement. After the algorithm was improved, the maximum values of the angular velocity of each joint of the robotic arm are all within the first interpolation curve, but all are less than Vmax. As shown in Figure 7, after the algorithm was improved, the angular acceleration curves of each joint of the robotic arm are more compact than before the improvement, and the increase in the amplitude of angular acceleration change has significantly improved time efficiency.

Analysis shows that after adopting the SHO-Informed-RRT* algorithm, the angle change curves, angular velocity change curves, and angular acceleration change curves of each joint of the robotic arm are relatively smooth, and the starting and ending points of the angular velocity change curves and angular acceleration change curves are both zero, meeting the actual motion requirements and set conditions of the robotic arm. During motion, the angular changes of each joint of the robotic arm are small, while the changes in angular velocity and angular acceleration are significant, achieving the optimal design goal for robotic arm motion time set in this paper.

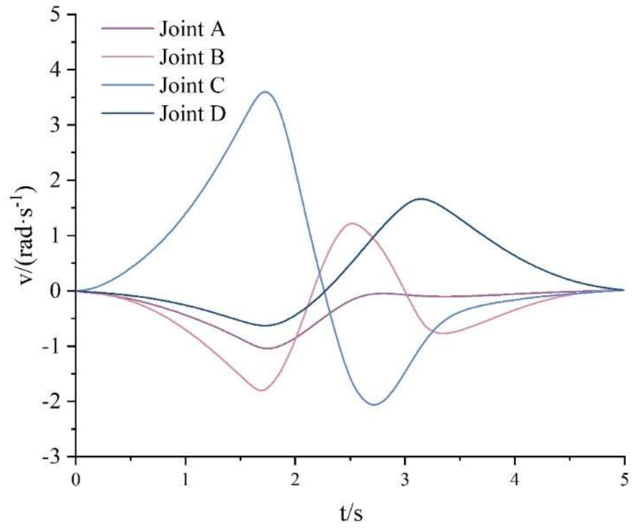


(a) Angle curve before improvement

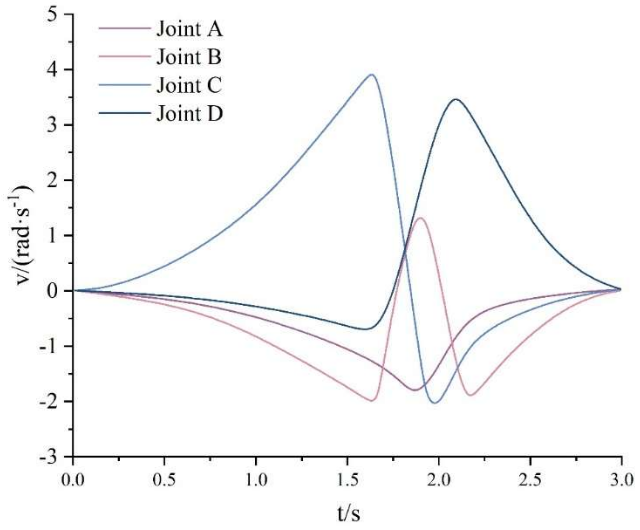


(b) Angle curve after improvement

Figure 5: Angle curve before and after algorithm improvement

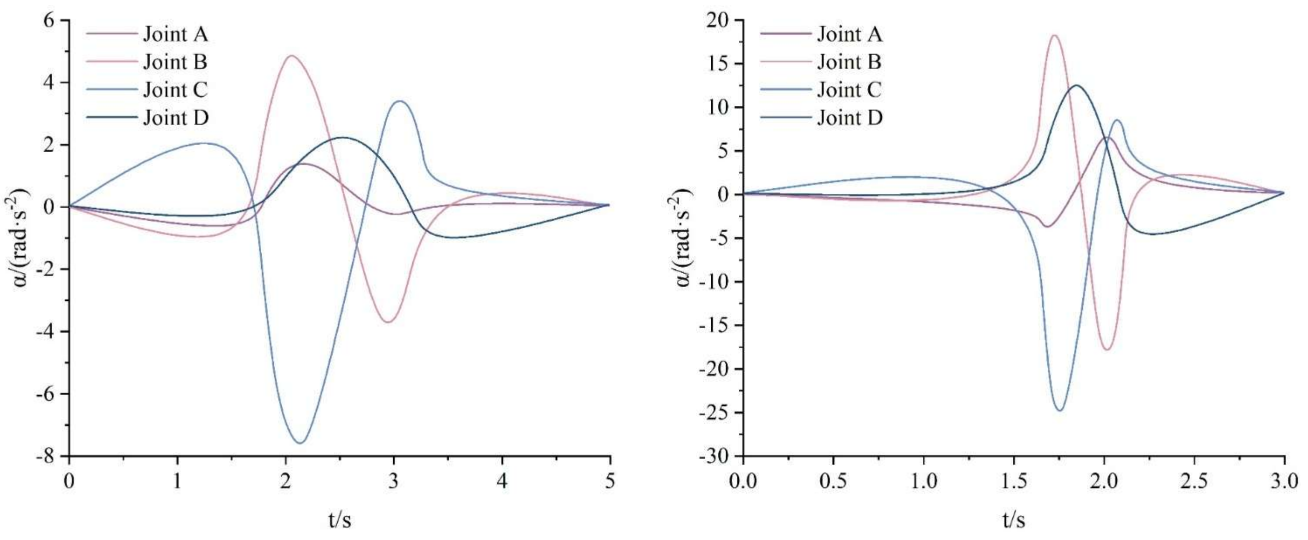


(a) Motion angular velocity curve before improvement



(b) Motion angular velocity curve after improvement

Figure 6: Motion angular velocity curve before and after algorithm improvement



(a) Motion angle acceleration curve before improvement (b) Motion angle acceleration curve after improvement

Figure 7: Motion angle acceleration curve before and after algorithm improvement

To further demonstrate the superiority of the proposed SHO-Informed-RRT* algorithm, it is compared with the traditional RRT* algorithm and the Informed-RRT* algorithm. The algorithm parameters are as follows: maximum iteration count of 100, maximum mutation probability of 0.1, minimum mutation probability of 0.05, maximum crossover probability of 0.8, minimum crossover probability of 0.4, and allowed error rate of 0.05. Four comparative experiments (Experiments 1–4) were conducted to observe and statistically analyze the total motion time of the robotic arm when trajectory planning was performed using the traditional RRT* algorithm, the Informed-RRT* algorithm, and the SHO-Informed-RRT* algorithm. The results of the four comparative experiments are shown in Table 2.

As shown in Table 2, the results of the four comparative experiments are consistent, with no significant differences. In terms of the average overall motion time of the robotic arm, the SHO-Informed-RRT* algorithm reduces the time by 2.02089 seconds compared to the Informed-RRT* algorithm and by 3.07432 seconds compared to the traditional RRT* algorithm.

The comparison experiment results indicate that the proposed SHO-Informed-RRT* algorithm outperforms other algorithms in terms of convergence speed and the overall motion time of the robot arm along the planned trajectory, effectively achieving optimization of robot arm trajectory planning.

Table 2: Experiment results

Algorithm	The overall movement time of the mechanical arm				
	1	2	3	4	Mean
RRT*	6.18526	6.20312	5.81425	6.22748	6.10753
Informed-RRT*	4.52683	4.92458	5.21623	5.54876	5.05410
SHO-Informed-RRT*	3.04626	3.06748	3.01896	3.00013	3.03321

V. Conclusion

The paper completes image segmentation of trees based on visual imaging and image processing technologies, designs and solves the kinematic model of forestry robots, and optimizes the Informed-RRT* algorithm using the Hippocampus Optimization Algorithm to complete path planning for the mechanical operations of forestry robots.

(1) Compared with other algorithms, the path planning time of the SHO-Informed-RRT* algorithm in this paper is the shortest, the planning speed is the fastest, the planning length is the shortest, and the average number of sampling points and path nodes is the smallest. The SHO-Informed-RRT* algorithm reduces the average path search time, average path length, average number of sampling points, and average number of path nodes on complex maps by 80.45%, 44.64%, 98.69%, and 93.12%, respectively, compared to the RRT* algorithm, and by 94.04%, 16.26%, 98.95%, and 84.27%, respectively, compared to the Informed-RRT* algorithm.

(2) After adopting the SHO-Informed-RRT* algorithm, the angular changes of the joints of the forestry robot's manipulator were smaller, while the changes in angular velocity and angular acceleration were larger, achieving the optimal manipulator motion time target set in this paper. The average manipulator motion time was reduced by 3.07432s and 2.02089s compared

to the RRT* algorithm and Informed-RRT* algorithm, respectively.

Funding

This work was supported by the Linxia Power Supply Company, State Grid Corporation of China (B727142400XL).

References

[1] Brown, M., Ghaffariyan, M. R., Berry, M., Acuna, M., Strandgard, M., & Mitchell, R. (2020). The progression of forest operations technology and innovation. Australian Forestry, 83(1), 1-3.

[2] La Hera, P., Mendoza-Trejo, O., Lindroos, O., Lideskog, H., Lindbäck, T., Latif, S., ... & Karlberg, M. (2024). Exploring the feasibility of autonomous forestry operations: Results from the first experimental unmanned machine. Journal of Field Robotics, 41(4), 942-965.

[3] Nie, J., Wang, Y., Li, Y., & Chao, X. (2022). Artificial intelligence and digital twins in sustainable agriculture and forestry: a survey. Turkish Journal of Agriculture and Forestry, 46(5), 642-661.

[4] Talbot, B., Pierzchała, M., & Astrup, R. (2017). Applications of remote and proximal sensing for improved precision in forest operations. Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering, 38(2), 327-336.

[5] Cavalli, R., & Amishev, D. (2019). Steep terrain forest operations–challenges, technology development, current implementation, and future opportunities. International Journal of Forest Engineering, 30(3), 175-181.

[6] Tan, Z., Liu, J., Sun, B., Qin, H., & Ma, Y. (2023). Study of a chassis path planning algorithm for a forest harvester. International Journal of Forest Engineering, 34(3), 439-451.

[7] Visser, R., & Obi, O. F. (2021). Automation and robotics in forest harvesting operations: Identifying near-term opportunities. Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering, 42(1), 13-24.

[8] Jin, T., & Han, X. (2024). Robotic arms in precision agriculture: A comprehensive review of the technologies, applications, challenges, and future prospects. Computers and Electronics in Agriculture, 221, 108938.

[9] Ferreira, J. F., Portugal, D., Andrada, M. E., Machado, P., Rocha, R. P., & Peixoto, P. (2023). Sensing and artificial perception for robots in precision forestry: a survey. Robotics, 12(5), 139.

[10] Ayoub, E., Levesque, P., & Sharf, I. (2023, May). Grasp planning with cnn for log-loading forestry machine. In 2023 IEEE international conference on robotics and automation (ICRA) (pp. 11802-11808). IEEE.

[11] Xiao, X., Wang, Y., & Jiang, Y. (2024). Review of research advances in fruit and vegetable harvesting robots. Journal of Electrical Engineering & Technology, 19(1), 773-789.

[12] He, Z., Ma, L., Wang, Y., Wei, Y., Ding, X., Li, K., & Cui, Y. (2022). Double-arm cooperation and implementing for harvesting kiwifruit. Agriculture, 12(11), 1763.

[13] Zeng, R., Wen, Y., Zhao, W., & Liu, Y. J. (2020). View planning in robot active vision: A survey of systems, algorithms, and applications. Computational Visual Media, 6(3), 225-245.

[14] Wang, H., Lin, Y., Xu, X., Chen, Z., Wu, Z., & Tang, Y. (2022). A study on long-close distance coordination control strategy for litchi picking. Agronomy, 12(7), 1520.

[15] Hua, Y., Zhang, N., Yuan, X., Quan, L., Yang, J., Nagasaka, K., & Zhou, X. G. (2019). Recent advances in intelligent automated fruit harvesting robots. The Open Agriculture Journal, 13(1).

[16] Cano, J., Yang, Y., Bodin, B., Nagarajan, V., & O'Boyle, M. (2018, October). Automatic parameter tuning of motion planning algorithms. In 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 8103-8109). IEEE.

[17] Liu, J., & Liu, Z. (2024). The vision-based target recognition, localization, and control for harvesting robots: a review. International Journal of Precision Engineering and Manufacturing, 25(2), 409-428.

[18] Singh, A., Kalaichelvi, V., & Karthikeyan, R. (2022). A survey on vision guided robotic systems with intelligent control strategies for autonomous tasks. Cogent Engineering, 9(1), 2050020.

[19] Li, T., Xie, F., Zhao, Z., Zhao, H., Guo, X., & Feng, Q. (2023). A multi-arm robot system for efficient apple harvesting: Perception, task plan and control. Computers and electronics in agriculture, 211, 107979.

[20] Rodić, A., Ilić, U., & Stevanović, I. (2024, May). AI-Based Multi-criteria Path Planning of Cartesian Robot with Telescopic Arm for Tree Fruit Picking. In International Conference on Robotics in Alpe-Adria Danube Region (pp. 451-461). Cham: Springer Nature Switzerland.

[21] Guo, Z., Fu, H., Wu, J., Han, W., Huang, W., Zheng, W., & Li, T. (2025). Dynamic Task Planning for Multi-Arm Apple-Harvesting Robots Using LSTM-PPO Reinforcement Learning Algorithm. Agriculture, 15(6), 588.

[22] Jelavic, E., Jud, D., Egli, P., & Hutter, M. (2022). Robotic precision harvesting: mapping, localization, planning and control for a legged tree harvester. Field Robotics, 2, 1386-1431.

[23] Xie, F., Guo, Z., Li, T., Feng, Q., & Zhao, C. (2025). Dynamic Task Planning for Multi-Arm Harvesting Robots Under Multiple Constraints Using Deep Reinforcement Learning. Horticulturae, 11(1).

[24] Zhang, C., Yong, L., Chen, Y., Zhang, S., Ge, L., Wang, S., & Li, W. (2019). A rubber-tapping robot forest navigation and information collection system based on 2D LiDAR and a gyroscope. Sensors, 19(9), 2136.

[25] Oliveira, L. F., Moreira, A. P., & Silva, M. F. (2021). Advances in forest robotics: A state-of-the-art survey. Robotics, 10(2), 53.

[26] Hanrui Wang,Ruoxi Sun,Cunjian Chen,Minhui Xue,Lay-Ki Soon,Shuo Wang & Zhe Jin. (2025). Iterative Window Mean Filter: Thwarting Diffusion-Based Adversarial Purification. IEEE Transactions on Dependable and Secure Computing,22(2),1827-1844.

[27] Lintao Duan,Weitao Du,Liming Wang,Wennian Yu,Zaigang Chen,Fengshou Gu & Andrew Ball. (2025). Time-varying transient EHL behaviors of spur gear pairs considering tooth waviness extracted by gaussian filter. Tribology International,210,110758-110758.

[28] Yanqin Li & Dehai Zhang. (2025). Toward Efficient Edge Detection: A Novel Optimization Method Based on Integral Image Technology and Canny Edge Detection. Processes,13(2),293-293.

[29] Fan Zhang,Feng Zhang,Hongbo Zou,Hengrui Ma & Hongxia Wang. (2024). Offshore Wind Power Foundation Corrosion Rate Prediction Model Based on Improved SHO Algorithm. Processes,12(6),1215-1215.

[30] Afroze Rahman,Anindita Kundu & Sumanta Banerjee. (2025). IQ-RRT*: a path planning algorithm based on informed-RRT* and quick-RRT*. International Journal of Computational Science and Engineering,28(3),303-313.