Path planning of the fruit tree pruning manipulator based on improved RRT-Connect algorithm

Yaya Chen[1,2,3], Yuxing Fu[3], Bin Zhang[3], Wei Fu[1*], Congju Shen[3]

(1. *Mechanical and Electrical Engineering College, Hainan University, Haikou 570228, Hainan, China*;
2. *School of Information and Communication Engineering, Hainan University, Haikou 570228, Hainan, China*;
3. *College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832003, Xinjiang, China*)

Abstract: Aiming to realize the obstacle avoidance of the fruit tree pruning manipulator in unstructured complex natural environment, an improved bidirectional fast extended random tree (RRT-Connect) algorithm was presented in this study. The manipulator and obstacles were properly simplified based on their geometrical characteristics to build collision detection models taking account of the obstacles, ground, and manipulator itself and to carry out the obstacle avoidance path planning. Goal-biased strategy and adaptive step size adjustment principle were introduced to accelerate the path search speed. Bidirectional pruning optimal strategy and cubic non-uniform B-spline interpolation method were adopted to optimize the path generated by RRT-Connect. The simulation path planning experiment was carried out in the simulation system of the fruit tree pruning manipulator and the practical obstacle avoidance path planning experiment was carried out on the real fruit tree pruning manipulator path planning experiment platform. The results showed that the path planning time and the path length of the improved RRT-Connect algorithm reduced by about 55% and 60% respectively compared with the basic RRT-Connect algorithm. The path planning success rate of the improved RRT-Connect algorithm was 100%, and the planned path was smooth, continual and executable, which could effectively guide the manipulator to avoid obstacles and lead the end effector of the manipulator to the goal point. The proposed improved algorithm not only has certain application value for obstacle avoidance of the fruit tree pruning manipulator in fruit tree pruning environment, but also has theoretical reference value for path planning of other types of robots.

Keywords: manipulator, path planning, RRT-Connect, jujube pruning, obstacle avoidance, non-uniform B-spline
DOI: 10.25165/j.ijabe.20221502.6249

Citation: Chen Y Y, Fu Y X, Zhang B, Fu W, Shen C J. Path planning of the fruit tree pruning manipulator based on improved RRT-Connect algorithm. Int J Agric & Biol Eng, 2022; 15(2): 177–188.

1 Introduction

Fruit tree pruning is a seasonal and labor-intensive work. Current fruit tree pruning methods mainly include manual single-branch pruning and mechanical whole-plant geometric pruning. Manual pruning is featured with high cost, high technical requirements and complicated working conditions. Mechanical pruning has poor robustness and the working parameters can not be adjusted continuously, which is unable to prune individual branches selectively. Therefore, intelligent pruning has become the future direction of development of fruit tree pruning[1,2]. Recently, great achievements have been made in the research of pruning robot technology with the rapid development of computer technology, artificial intelligence and other related industries. However, when pruning fruit tree in unstructured complex natural environment, the branches distributed near the branch to be pruned become obstacles to the movement of the pruning manipulator, causing big trouble for automatic pruning work by pruning robot. To solve this problem, the obstacle avoidance path planning of the pruning manipulator has become one of the key technical issues for fruit tree pruning robot[3].

Given specific obstacles and initial and target pose of the pruning manipulator, the obstacle avoidance path planning of the manipulator is to search a group of continual sequence of joint values to drive the manipulator to move from the initial pose to the target pose safely without any collision with obstacles[4]. So far, substantive researches on manipulator path planning have been conducted by domestic and foreign researchers, with propositions of some impressive solutions, such as the A*, the artificial potential field method, the ant colony optimization algorithm[5-7], etc. Van Henten et al.[8] proposed the obstacle avoidance path planning algorithm of cucumber-picking robot based on A*. This algorithm can effectively obtain the shortest path in static environment. Rostami et al.[9] proposed an improved artificial potential field method to deal with the obstacle avoidance path planning problem of pruning robot. This method has conquered the problem of unreachable target and minimum point of local potential field. Ajeil et al.[10] proposed an improved ant colony optimization algorithm to solve the optimal path planning of mobile robot. This algorithm can work out a shortest collision-free path in both static and dynamic scenarios. These path planning algorithms can effectively solve the path planning of low-freedom (2-3 degrees of freedom (DOF)) manipulators. But

Received date: 2020-10-26 **Accepted date:** 2021-07-14

Biographies: Yaya Chen, MS candidate, research interests: agriculture robot, Email: 2778366825@qq.com; **Yuxing Fu,** MS candidate, research interests: machine vision, Email: 493095247@qq.com; **Bin Zhang,** MS candidate, research interests: agriculture robot, Email: 1085810207@qq.com; **Congju Shen,** Associate Researcher, PhD candidate, research interests: agricultural machinery design, Email: shencongju@163.com.

***Corresponding author: Wei Fu,** PhD, Professor, research interests: intelligent pruning of fruit trees and agricultural aviation plant protection. Mechanical and Electrical Engineering College, Hainan University, Haikou 570228, Hainan, China. Tel: +86-13070091786, Email: fuwei001@126.com.

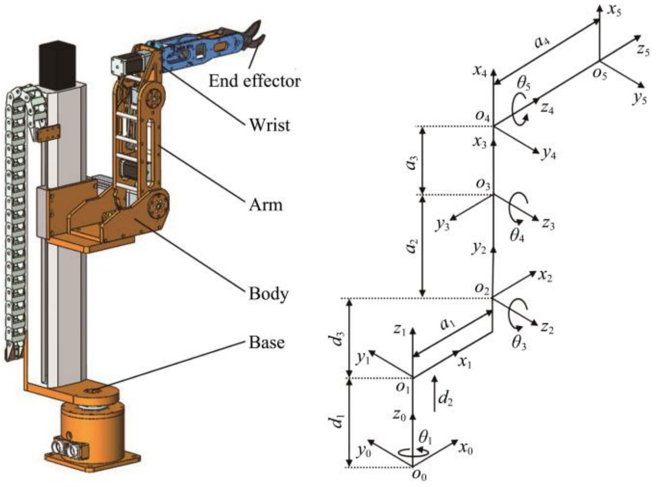
these algorithms require an accurate description of the obstacle model in the configuration space of the manipulator. The computational complexity increases exponentially with the increase of the DOF of the manipulator. These algorithms are not suitable for solving the path planning of multi-degree-of-freedom manipulator[11,12]. Therefore, algorithms based on random sampling such as the probabilistic roadmap method (PRM), the rapidly exploring random tree (RRT) algorithm and the bidirectional fast extended random tree (RRT-Connect) algorithm are proposed[13-15]. These algorithms do not need to accurately describe the obstacle model in the configuration space, but only need to obtain the sampling point through random sampling in the configuration space and carry out collision detection on the point to find the collision free path, which have small amount of calculation and high efficiency of path planning. Among these algorithms, RRT-Connect performs better on multi-degree-of-freedom manipulator, and compared with PRM and RRT, it has less path planning time and higher path planning success rate, which has been widely used to solve the path planning of multi-degree-of-freedom manipulator[16,17]. However, due to the randomness of sampling and restriction of obstacles, the RRT-Connect algorithm has slow and time-consuming convergence rate. The planned path is rough and not continual with many unnecessary points, which is not conducive to the execution of the manipulator[18,19]. So the path quality of the RRT-Connect is not satisfactory. In dealing with the shortcomings of the RRT-Connect algorithm, researchers proposed some improved algorithms. Botterill et al.[3] proposed an improved RRT-Connect algorithm to solve the path planning of the 6 DOF vine pruning manipulator in the vine pruning environment. This algorithm introduces the local optimization program into the RRT-Connect to increase the convergence rate of path planning, but the quality of the planned path is poor. Wang et al.[20] proposed an improved RRT-Connect algorithm to accelerate the convergence of path planning, which takes the center point of the connection between the starting point and the target point as the third point to realize the simultaneous expansion of four trees. Although this algorithm improves the efficiency of path planning and reduces the time of path planning, the quality of the planned path is poor. Zhao et al.[21] proposed a path planning method for the manipulator based on RRT-Connect and Bezier curve. This method uses the dichotomy merge to remove redundant path points in the collision free path searched by RRT-Connect, and uses Bezier curve to smooth transition points in the path. Although this method improves the quality of the path, it does not consider the efficiency of path planning. Zhang et al.[22] proposed an improved algorithm combining artificial potential field method and RRT-Connect algorithm to solve the path planning of UAV in complex static environment. The path planned by the improved algorithm is closer to the optimal path, but the planned path is not smooth. Liu et al.[23] proposed a goal-biased bidirectional fast extended random tree algorithm based on Bezier curve. Although this algorithm improves the efficiency of path planning and plans a smooth path, there are many unnecessary points in the path. In summary, it can be seen that only a few studies have been reported for the adoption of the RRT-Connect for path planning in an unstructured environment such as agriculture, and almost no study has been reported for the adoption of the RRT-Connect for path planning in fruit tree pruning environment. In addition, in view of the shortcomings of the RRT-Connect algorithm, although some improved algorithms have been proposed, the improved algorithm considering not only the efficiency of path planning but also the quality of path has not been proposed.

Therefore, this study proposes an improved RRT-Connect algorithm to solve the path planning problem of the fruit tree pruning manipulator in fruit tree pruning environment (unstructured and complex natural environment). The 5 DOF manipulator independently developed by the research group is used as the fruit tree pruning manipulator and the jujube tree is used as the pruning object. To obtain a collision-free path, the manipulator and obstacles are reasonably simplified. Then, the RRT-Connect algorithm is adopted to plan the pruning path of the manipulator, and the goal-biased strategy and adaptive step size adjustment principle are added to the RRT-Connect algorithm to accelerate the path searching speed. The bidirectional pruning optimal strategy and cubic non-uniform B-spline interpolation method are used to optimize the path generated by the RRT-Connect algorithm, so that a path with the best or nearly the best quality can be obtained. Finally, the feasibility, validity and superiority of the proposed algorithm are verified through the path planning simulation experiment in the fruit tree pruning manipulator simulation system and the practical obstacle avoidance path planning experiment on the real fruit tree pruning manipulator path planning experiment platform.

2 Establishment of kinematic model of manipulator and obstacle model

2.1 Kinematic model of fruit tree pruning manipulator

The three-dimensional model of the fruit tree pruning manipulator is shown in Figure 1a. This is a 5 DOF tandem manipulator consisted of base, body, wrist, arm and end effector, which can realize the rotational motion of the base, the lifting motion of the body, two types of rotational motion of the arm and the rotational motion of the wrist. The end effector is fixed to the manipulator link 5 and is used to cut the branch to be pruned. The first four joints of the manipulator control the position of the end effector and the last joint controls the attitude of the end effector. The linkage model of the manipulator is established using standard D-H parameter method, as shown in Figure 1b. The specific linkage parameters are listed as Table 1.



a. Three-dimensional model b. Simplified D-H linkage model

Note: o_0 is the base coordinate system, o_i (i=0, 1, 2, 3, 4) are the linkage coordinate systems, o_5 is the end effector coordinate system.

Figure 1 Fruit tree pruning manipulator

Table 1 D-H parameters of the manipulator

Linkages i	Length of linkages a_i /mm	Twist angles α_i /(°)	Offset of linkages d_i /mm	Joint angle θ_i /(°)	Joint variables	Note/mm
1	0	0	d_1	θ_1	–150 ° to 150 °	d_1=250
2	a_1	90	d_2+d_3	0	0-500 mm	a_1=150 d_3=76
3	a_2	0	0	θ_3	–30 ° to 180 °	a_2=260
4	a_3	90	0	θ_4	–90 ° to 120 °	a_3=80
5	0	0	a_4	θ_5	–160 ° to 160 °	a_4=390

According to the simplified D-H linkage model and D-H linkage parameters of the manipulator, the overall pose transformation matrix of the end effector in relative to the base coordinate system can be obtained through Equation (1):

$$ {}^5_0T = {}^1_0T\,{}^2_1T\,{}^3_2T\,{}^4_3T\,{}^5_4T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where, ${}^{i\text{-}1}_{i}T$ is the transformation matrix of the linkage i in relative to the linkage i–1, as shown in Equation (2):

$$ {}^{i\text{-}1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Skip intermediate steps, the spatial pose of the end effector of the fruit tree pruning manipulator can be obtained, as shown in Equation (3):

$$\begin{cases} n_x = s_1s_5 + c_1c_5c_{34} \\ n_y = -c_1s_5 + s_1c_5c_{34} \\ n_z = c_5s_{34} \\ o_x = s_1c_5 - c_1s_5c_{34} \\ o_y = -c_1c_5 - s_1s_5c_{34} \\ o_z = -s_5s_{34} \\ a_x = c_1s_{34} \\ a_y = s_1s_{34} \\ a_z = -c_{34} \\ p_x = c_1(a_1 + a_4s_{34} + a_2c_3 + a_3c_{34}) \\ p_y = s_1(a_1 + a_4s_{34} + a_2c_3 + a_3c_{34}) \\ p_z = d_1 + d_2 + d_3 + a_2s_3 + a_3s_{34} - a_4c_{34} \end{cases} \tag{3}$$

where, $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{34} = \cos(\theta_3 + \theta_4)$, $s_{34} = \sin(\theta_3 + \theta_4)$.

When known the three-dimensional coordinates of the jujube pruning point (target point), the position of the end effector can be obtained through the coordinates of this point, and the attitude of the end effector can be acquired according to the growth posture of the branch to be pruned. In this way, both the position and attitude of the end effector can be obtained, i.e. the (n, o, a, p) is known. When solving inverse kinematics, the parameterized d_2 can be assumed as a known value, hence the problem of the inverse kinematics for the manipulator is to calculate the joint variables θ_i[24]. The joint variables were obtained by using inverse transformation method as shown in Equation (4):

$$\begin{cases} \theta_1 = \arctan\dfrac{p_y}{p_x} \\ \theta_3 = \arctan\dfrac{t_2 + a_4c_{34} - a_3s_{34}}{t_1 - a_4s_{34} - a_3c_{34}} \\ \theta_4 = \arctan\dfrac{a_3(c_3t_2 - s_3t_1) + a_4(s_3t_2 + c_3t_1 - a_2)}{a_3(s_3t_2 + c_3t_1 - a_2) - a_4(c_3t_2 - s_3t_1)} \\ \theta_5 = \arctan\dfrac{s_1n_x - c_1n_y}{s_1o_x - c_1o_y} \end{cases} \tag{4}$$

where, $t_1 = c_1p_x + s_1p_y - a_1$, $t_2 = p_z - d_1 - d_2 - d_3$, $s_{34} = c_1a_x + s_1a_y$, $c_{34} = -a_z$.

2.2 Obstacle model

A jujube tree consists of the main trunk, main branch, side branch, and bearing branch[25]. These branches distributed near the branch to be pruned (target branch) would become obstacles to the movement of the pruning manipulator which have to steer away from the obstacles to prune the target branch. The bearing branch of jujube is abscission fruit branch, which has no influence on the movement of the manipulator. So the obstacles during jujube pruning are the main trunk, main branch, side branch distributed near the target branch. The branches may be bending or forked, which can hardly be expressed by a general representation method. This study applies cylindrical envelope method to approximately describe unstructured obstacles. The branches are described as several sections of cylinders which are connected to form the branches. To visualize the cylinder, the two-points-radius method was proposed to describe a cylinder with only two endpoints and one radius. The obstacle model was established with the bending point, forking point and tip point of the branches as the endpoint of the cylinder and the largest radius of each section of branches as the radius of the cylinder, as shown in Figure 2. o_i (i = 1, 2, 3, 4, 5) are the endpoints. o_2 is the forking point. o_3 is the bending point. o_4 is the tip point. r_i (i = 1, 2, 3, 4, 5) are the radii. l_1 is the length of the cylinder. Although this modeling method has enlarged the obstacle area, it simplified the collision detection calculation between the manipulator and obstacles, which can effectively improve the path planning efficiency and further ensure the security between the manipulator and obstacles.

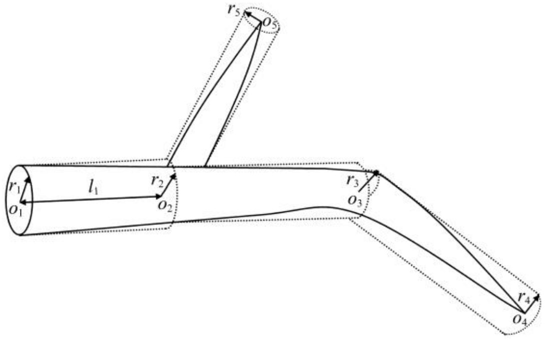
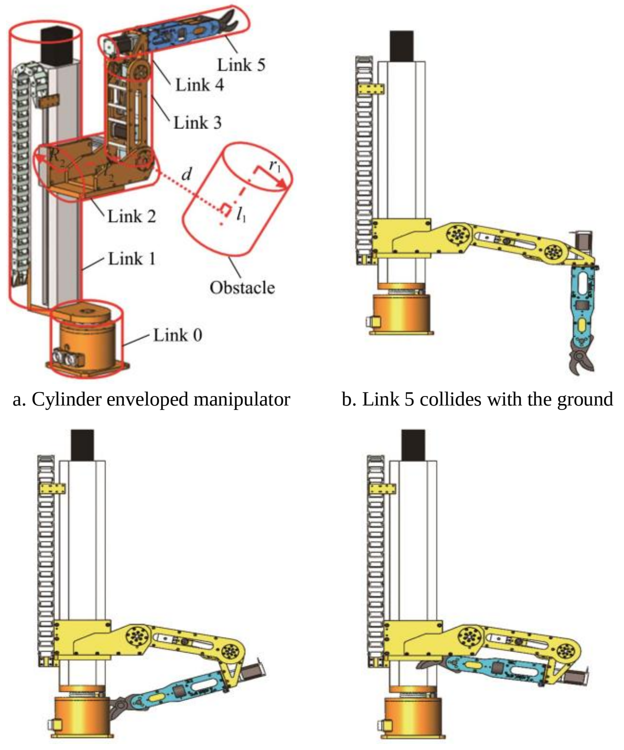


Figure 2 Schematic diagram of branch modeling

3 Obstacle avoidance path planning of fruit tree pruning manipulator

3.1 Collision detection

Collision detection is a very crucial part of the RRT-Connect algorithm, which is used to determine whether the manipulator is in a collision free state when the manipulator is at a certain point. As mentioned above, the obstacles are simplified as cylinders. As shown in Figure 3a, the linkages of the manipulator can be also simplified to cylinders according to the structural characteristics of the manipulator. As a result, the collision detection between the manipulator and obstacles is transformed into the collision detection between the cylinder and the cylinder, and then into the judgment of the size relationship between the shortest distance d between the central axes l_1 and l_2 of the two cylinders and the sum R of the radiuses of the two cylinders. The manipulator link 0 is a fixed link, which does not participate in the collision detection between the manipulator and obstacles. So the collision detection between the manipulator and each obstacle is performed five times. Only when the manipulator's five links do not collide with each obstacle, it is considered that the collision between the manipulator and obstacles does not occur.



a. Cylinder enveloped manipulator

b. Link 5 collides with the ground

c. Link 5 collides with link 0

d. Link 5 collides with link 2

Figure 3 Cylindrical envelope and collision of the fruit tree pruning manipulator

Besides, according to the D-H parameters of the manipulator, the manipulator link 5 may collide with the ground, the link 0 and the link 2, as shown in Figures 3b-3d. Therefore, collision detection includes not only the collision detection between the manipulator and obstacles, but also the collision detection between the manipulator and the ground and the self-collision detection of the manipulator. The specific collision detection process is shown in Figure 4. When the manipulator is at a certain point, the first step is to obtain the radius R_5 of the manipulator link 5 and the end point coordinates (X_5, Y_5, Z_5) of the link 5 by the matrix transformation of the linkage coordinate system. The second step is to judge the size relationship between Z_5 and R_5. If $Z_5 > R_5$, there is no collision happens between the manipulator and the ground and execute the next step. Otherwise, there is collision between the manipulator and the ground, i.e. the manipulator collides when the manipulator is at this point and collision detection ends. The third step is to obtain the radius R_0 of the link 0, obtain the Z value Z_2 of the end point coordinates of the link 2 by the matrix transformation of the linkage coordinate system and calculate the sum of squares S_5 of X_5 and Y_5 and the square S_0 of R_0. The fourth step is to judge the size relationship between Z_2 and Z_5 and the size relationship between S_5 and S_0. If $Z_5 \leq Z_2$ and $S_5 \leq S_0$, the manipulator happens self-collision, i.e. the manipulator collides when the manipulator is at this point and collision detection ends. Otherwise, there is no self-collision of the manipulator and execute the next step. The fifth step is to set obstacles O_i (i=1, 2, …, n) and links L_j (j = 1, 2, …, 5) to be detected. i is the order of obstacles and j is the order of links. Let i=j=1. The sixth step is to calculate the shortest distance d between the central axes of O_i and L_j and the sum R of the radiuses of O_i and L_j and judge the size relationship between d and R. If $d \leq R$, the collision occurs between the manipulator and obstacles, i.e. the manipulator collides when the manipulator is in this point and collision detection ends. If $d > R$, let $j = j+1$ and continue to perform the sixth step until j is not less than 6. In the seventh step, judge whether i is less than n. If so, let i=i+1, j=1 and perform the sixth step. If not, there is no collision between the manipulator and obstacles, i.e. the manipulator is in a collision free state when the manipulator is at this point, and the collision detection is completed.

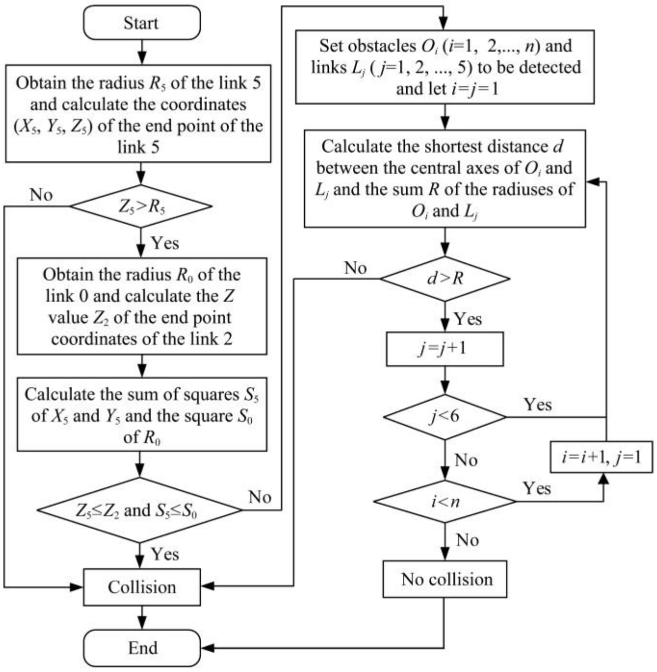

Start
Obtain the radius R_5 of the link 5 and calculate the coordinates (X_5, Y_5, Z_5) of the end point of the link 5
No
Z_5>R_5
Yes
Obtain the radius R_0 of the link 0 and calculate the Z value Z_2 of the end point coordinates of the link 2
Calculate the sum of squares S_5 of X_5 and Y_5 and the square S_0 of R_0
Z_5≤Z_2 and S_5≤S_0
No
Yes
Collision
End
Set obstacles O_i (i=1, 2,..., n) and links L_j (j=1, 2, ..., 5) to be detected and let i=j=1
Calculate the shortest distance d between the central axes of O_i and L_j and the sum R of the radiuses of O_i and L_j
No
d>R
Yes
j=j+1
j<6
Yes
No
i=i+1, j=1
i<n
Yes
No
No collision

Figure 4 Flow chart of collision detection

3.2 Basic RRT-Connect

The basic RRT-Connect algorithm constructs two parallel fast extended random trees at the start point and goal point respectively to search the configuration space. In each iteration, one tree is extended first, and then the other tree is tried to extend to the new point of the current tree expansion. The two trees T_1 and T_2 extend alternately in each other's direction until they meet[15]. As shown in Figure 5, in each iteration of the RRT-connect algorithm. First, obtain a sampling point q_{rand} by random sampling and search the nearest point $q_{nearest}$ to the sampling point q_{rand} in the random tree T_1. Next, select a new point q_{new} on the line segment connected by q_{rand} and $q_{nearest}$ to make the distance between $q_{nearest}$ and q_{new} equal to the search step size ε. Then, determine whether the manipulator is in a collision free state when the manipulator is in q_{new}. If not, discard q_{new} and enter the next iteration process. If so, determine whether the manipulator is in a collision free state when the manipulator is in m equal-part points of the line segment connected by q_{new} and $q_{nearest}$. If not, discard q_{new} and enter the next iteration process. If so, keep q_{new} and add this point to T_1. After that, T_2 starts to expand in the direction of q_{new}. During the extension of T_2, if the manipulator does not collide and the current extended point $q_{new'}$ does not meet q_{new}, T_2 calls the Connect function to quickly expand to q_{new} until the manipulator collides or

$q_{new'}$ meets q_{new}. When the manipulator collides, T_2 stops the extension. Call the Swap function to exchange two trees and enter the next iteration process. When $q_{new'}$ meets q_{new}, two trees meet and connect and obtain a collision free path.

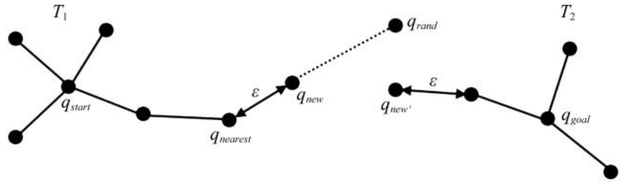


Figure 5 Extended schematic diagram of RRT-Connect algorithm

The specific steps of the algorithm are shown in Algorithm 1, which consists of three parts: Build RRT-Connect (q_{start}, q_{goal}), Connect (T, q), and Extend (T, q). In each iteration of the algorithm. First, obtain a sampling point q_{rand} by random sampling. Then, call the Extend function to extend in T_1 to generate a new point q_{new}. If the return value of the Extend function is Trapped, discard q_{new} and enter the next iteration process. If the return value of the Extend function is not Trapped, keep q_{new} and add q_{new} to T_1. T_2 starts to extend in the direction of q_{new}. In the expansion process of T_2, if the return value of the Extend function is Advanced, T_2 calls the Connect function to quickly expand to q_{new} until the return value of the Extend function is Trapped or Reached. When the return value of the Extend function is Trapped, T_2 stops the extension. Call the Swap function to exchange two trees and enter the next iteration process. When the return value of the Extend function is Reached, it means that T_1 and T_2 meet and connect, the construction of the random tree is completed, and the collision-free path is obtained.

Algorithm 1 Basic RRT-Connect

```
Build RRT-Connect ($q_{start}$, $q_{goal}$)
$T_1$.init ($q_{start}$); $T_2$.init ($q_{goal}$);
While $i < N$ do
    $q_{rand} \leftarrow$ Sample ();
    if Extend ($T_1$, $q_{rand}$) $\neq$ Trapped then
        if Connect ($T_2$, $q_{new}$) = Reached then
            Return Path ($T_1$, $T_2$);
    Swap ($T_1$, $T_2$);
Return Failed;

Connect ($T$, $q$)
repeat $g \leftarrow$ Extend ($T$, $q$);
until $g \neq$ Advanced;
Return $g$;

Extend ($T$, $q$)
$q_{nearest} \leftarrow$ Nearest ($T$, $q$);
$q_{new} \leftarrow$ Steer ($q_{nearest}$, $q$, $\varepsilon$);
if Obstacle Free ($q_{new}$) then
    if Obstacle Free ($q_{nearest}$, $q_{new}$) then
        if $q_{new} = q$ then
            $T \leftarrow q$; Return Reached;
        else
            $T \leftarrow q_{new}$; Return Advanced;
else
    Return Trapped;
```

Notes: T_1 and T_2 represent the random trees generated with the start point q_{start} as the initial point and the goal point q_{goal} as the initial point respectively.

Sample () is a sampling function, which obtains a sampling point by random sampling in the configuration space.

Swap (T_1, T_2) is the swap function, which swaps the trees T_1 and T_2 and makes the two trees expand alternately.

Nearest (T, q) is the nearest point function, which searches the nearest point $q_{nearest}$ to the sampling point in the random tree.

Steer ($q_{nearest}$, q, ε) is a function to generate a new point. It starts from the nearest point and toward in the direction of the sampling point moves forward one step size ε to obtain a new point q_{new}.

Obstacle Free (q_{new}) is a point collision detection function. When the manipulator is in q_{new}, the manipulator is in a collision free state. The function return value is 0. Otherwise, the function return value is 1.

Obstacle Free ($q_{nearest}$, q_{new}) is a line segment collision detection function. When the manipulator is in m equal-part points of the line segment connected by q_{new} and $q_{nearest}$, the manipulator is in a collision free state. The function return value is 0. Otherwise, the function return value is 1. Here, the value of m is 10.

Reached means that the sampling point q is added to the random tree.

Advanced means that the new point q_{new} is added to the random tree.

Trapped means that when the manipulator is in q_{new} *or* m equal-part points of the line segment connected by q_{new} and $q_{nearest}$, the manipulator collides.

3.3 Goal-biased strategy

The sampling by the RRT-Connect algorithm is random and lacks orientation. Therefore, the goal-biased strategy was introduced into the Build-RRT-Connect (q_{start}, q_{goal}) by referring the idea of heuristic search algorithm to improve the orientation of the RRT-Connect algorithm and make the random sampling point more oriented to the goal point. Specifically, set a threshold value p_{best}. Each time the sampling point is obtained, a probability value p_g is randomly obtained first. When p_g is less than p_{best}, judge whether the point q_{goal} is in the random tree T_1. If it is, the sampling point is q_{start}. If it is not, the sampling point is q_{goal}. When p_g is not less than p_{best}, the sampling point is obtained by random sampling. Parameter p_{best} can be used to adjust the selection of sampling point. p_{best} is set as 0.5 to ensure the sampling randomness and the balance using the goal point as the sampling point.

3.4 Adaptive step size adjustment principle

When searching the barrier-free space using the RRT-Connect algorithm, the constant step size in the algorithm would restrict the extending speed of the random tree. Therefore, the adaptive step size adjustment function is introduced into the Extend (T, q) of the RRT-connect algorithm to obtain the Improve-Extend (T, q). Specifically, during the random tree's exploration of unknown space using the initial step size ε, if the return value of the current Improve-Extend function is Advanced, i.e. the extended new point by the current Improve-Extend function has no connection with another random tree and the manipulator is in a collision free state when the manipulator is in m equal-part points of the line segment connected by the new point and the nearest point, in the next extension, the random tree will expand by adding a step size ε to the initial step size. If the return value of the next Improve-Extend function is still Advanced, in the next extension, the random tree will expand by adding two steps size ε to the initial step size, and so on. If the return value of the next Improve-Extend function is Trapped, i.e. the manipulator has collision when the manipulator is in m equal-part points of the line segment connected by the extended new point by the next Improve-Extend function and the nearest point, in the next extension, the random tree will expand according to the initial step size.

The improved RRT-Connect algorithm that introduces the goal-biased strategy and the adaptive step size adjustment principle is called the BZRRT-Connect algorithm in this study. The specific steps of the BZRRT-Connect algorithm are shown in Algorithm 2. In each iteration of the BZRRT-Connect algorithm. First, obtain a sampling point q_{rand} by the goal-biased strategy. Then, call the Improve-Extend function to extend in T_1 to generate a new point q_{new}. If the return value of the Improve-Extend function is Trapped, discard q_{new} and enter the next iteration process. If the return value of the Improve-Extend function is not Trapped, keep q_{new} and add q_{new} to T_1. T_2 starts to extend in the direction of q_{new}. In the expansion process of T_2, if the return value of the Improve-Extend function is Advanced, T_2 calls the Connect function to quickly expand to q_{new} until the return value of the Improve-Extend function is Trapped or Reached. When the return value of the Improve-Extend function is Trapped, T_2 stops the extension. Call the Swap function to exchange two trees and enter the next iteration process. When the return value of the Improve-Extend function is Reached, it means that the trees T_1 and T_2 meet and connect, the construction of the random tree is completed, and the collision-free path is obtained.

Algorithm 2 BZRRT-Connect

```
Build BZRRT-Connect ($q_{start}$, $q_{goal}$)
$T_1$.init ($q_{start}$); $T_2$.init ($q_{goal}$);
While $i < N$ do
  $p_g \leftarrow$ rand (0, 1);
  if $p_g < p_{best}$ then
  if $q_{goal} \in T_1$ then
    $q_{rand} = q_{start}$;
  else $q_{rand} = q_{goal}$;
  else $q_{rand} \leftarrow$ Sample ();
  if Improve-Extend ($T_1$, $q_{rand}$) $\neq$ Trapped then
  if Connect ($T_2$, $q_{new}$) = Reached then
  Return Path ($T_1$, $T_2$);
  Swap ($T_1$, $T_2$);
  Return Failed;

Connect ($T$, $q$)
repeat $g \leftarrow$ Improve-Extend ($T$, $q$);
until $g \neq$ Advanced;
Return $g$;

Improve-Extend ($T$,$q$)
  $step \leftarrow \varepsilon$;
  $q_{nearest} \leftarrow$ Nearest ($T$, $q$);
  $q_{new} \leftarrow$ Steer ($q_{nearest}$, $q$, $step$);
  if Obstacle Free ($q_{new}$) then
  if Obstacle Free ($q_{nearest}$, $q_{new}$) then
  if $q_{new} = q$ then
  $T \leftarrow q$; Return Reached;
  else
  $step \leftarrow step + \varepsilon$; $T \leftarrow q_{new}$; Return Advanced;
  else
  $step \leftarrow \varepsilon$; Return Trapped;
```

3.5 Bidirectional pruning optimal strategy

Although the BZRRT-Connect algorithm can improve the path planning efficiency, the quality of the path planning is not satisfactory due to the production of many unnecessary points. Therefore, redundant points must be deleted. In order to effectively delete redundant points, the bidirectional pruning optimal strategy is used to optimize the path planned by the BZRRT-Connect algorithm. As shown in Figure 6, to move from point q_1 to point q_{10}, there are three obstacles to be avoided. The black line path q_1- q_2- q_3- q_4- q_5- q_6- q_7- q_8- q_9- q_{10} is the path generated by the BZRRT-Connect algorithm. In fact, on the basis of the manipulator no collision, the manipulator can move directly from point q_2 to point q_6, and the process from point q_3 to point q_6 is unnecessary. q_3, q_4 and q_5 are redundant points. The process of optimizing the black line path using the bidirectional pruning optimal strategy is as follows: firstly, on the premise of the manipulator no collision, starting from point q_1, the redundant points in the black line path are deleted to get the blue line path q_1- q_3- q_4- q_9- q_{10}. Then, on the premise of the manipulator no collision, starting from point q_{10}, the redundant points in the black line path are deleted to get the red line path q_1- q_2- q_6- q_{10}. Finally, among the blue and red paths, the red path with short path length is selected as the optimal path by the bidirectional pruning optimal strategy. It is clear that not only are the path points fewer, but the path length becomes shorter after using the bidirectional pruning optimal strategy to deal with the path planned by the BZRRT-Connect algorithm.

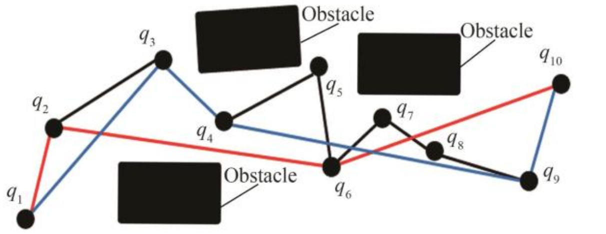


Figure 6 Schematic diagram of using the bidirectional pruning optimal strategy to optimize the path

The specific process of optimizing the path planned by the BZRRT-Connect algorithm using the bidirectional pruning optimal strategy is shown in Figure 7. The first step is to use the BZRRT-Connect algorithm to obtain the initial path N (q_0, q_1, ..., q_n), where q_i (i=0, 1, ..., n) are path points and i is the order of the path points. The second step is to set i=1, the initial point q_0 is added to the point set N_1 and q_0 is assigned to $q_{temp\text{-}1}$. The third step, start from the initial point $q_{temp\text{-}1}$ and connect the path points behind the initial point with the initial point in sequence. In the connection process, if the manipulator is in a collision free state when the manipulator is in m equal-part points of the line segment connected by the initial point and the current path point q_i, continue to connect the path points behind the current path point with the initial point in sequence until a collision occurs. In case of collision, add the previous path point $q_{i\text{-}1}$ of the current path point to the N_1 and take the previous path point of the current path point as the initial point to perform the third step again until the target point q_n is reached. In the fourth step, q_n is added to the N_1 to obtain the point set N_1 and the path length L_1 of the N_1 is calculated. The fifth step is to set i = n-1, the target point q_n is added to the point set N_2 and q_n is assigned to $q_{temp\text{-}2}$. The sixth step, start from the target point $q_{temp\text{-}2}$ and connect the path points in front of the target point with the target point in sequence. In the connection

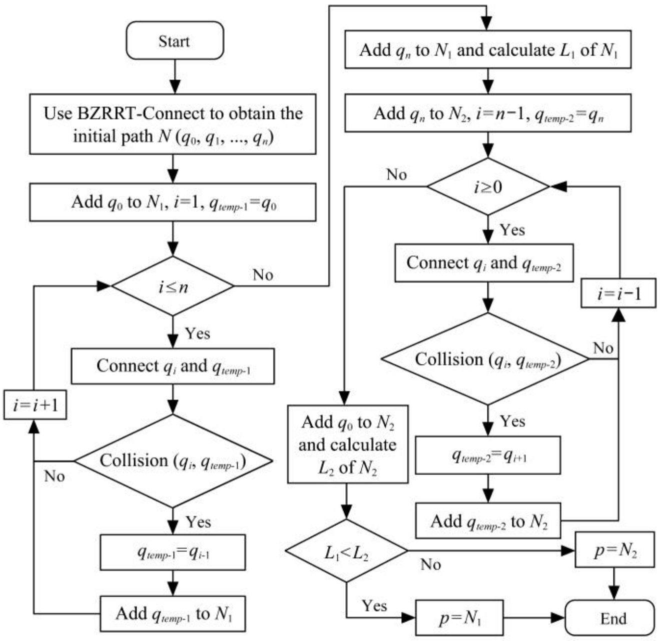

Start
Use BZRRT-Connect to obtain the initial path N (q_0, q_1, ..., q_n)
Add q_0 to N_1, i=1, $q_{temp\text{-}1}$=q_0
$i \leq n$
No
Yes
Connect q_i and $q_{temp\text{-}1}$
i=i+1
Collision (q_i, $q_{temp\text{-}1}$)
No
Yes
$q_{temp\text{-}1}$=$q_{i\text{-}1}$
Add $q_{temp\text{-}1}$ to N_1
Add q_n to N_1 and calculate L_1 of N_1
Add q_n to N_2, i=n−1, $q_{temp\text{-}2}$=q_n
No
$i \geq 0$
Yes
Connect q_i and $q_{temp\text{-}2}$
i=i−1
Collision (q_i, $q_{temp\text{-}2}$)
No
Yes
Add q_0 to N_2 and calculate L_2 of N_2
$q_{temp\text{-}2}$=q_{i+1}
Add $q_{temp\text{-}2}$ to N_2
L_1<L_2
No
p=N_2
Yes
p=N_1
End

Figure 7 Flow chart of using the bidirectional pruning optimal strategy to optimize the path

process, if the manipulator is in a collision free state when the manipulator is in *m* equal-part points of the line segment connected by the target point and the current path point q_i, continue to connect the path points in front of the current path point with the target point in sequence until a collision occurs. In case of collision, add the next path point q_{i+1} of the current path point to the N_2 and take the next path point of the current path point as the target point to perform the sixth step again until the initial point q_0 is reached. In the seventh step, q_0 is added to the N_2 to obtain the point set N_2 and the path length L_2 of the N_2 is calculated. The eighth step is to compare the size relationship between the L_1 and L_2. If $L_1 < L_2$, the path *p* after the bidirectional pruning optimal strategy is applied to the path planned by the BZRRT-Connect algorithm is N_1, otherwise, *p* is N_2.

3.6 Cubic non-uniform B-spline interpolation method

Although the bidirectional pruning optimal strategy is used to optimize the path generated by the BZRRT-Connect algorithm, which makes the path points less and the path length shorter, the generated path is a spatial broken line. As the manipulator moves to the break point, the manipulator would vibrate due to the sudden change of direction, accelerating the abrasion of the motor and shortening the service life of the manipulator. The existence of obstacles will lead to produce break points, which are not conducive to the execution of the manipulator. Therefore, the path should be smoothed to produce executable path for the manipulator[18]. As an important curve in computer graphics, B-spline curve is widely used in the path planning of the manipulator owing to its continuity and locality. Due to the uneven distribution of the path points obtained by the bidirectional pruning optimal strategy, cubic non-uniform B-spline interpolation method is adopted to smooth the path and generate executable path for the manipulator[26]. As shown in Figure 8, the path after the bidirectional pruning optimal strategy is applied to the path planned by the BZRRT-Connect algorithm is the black line p_1- p_2- p_3- p_4- p_5. p_1 to p_5 are path points. In order to fit this path by the cubic non-uniform B-spline interpolation method, firstly, the non-uniform node vector is constructed according to the distribution trend of the path points. Secondly, the non-uniform B-spline basis function is constructed according to the non-uniform node vector. Then, the equations for solving the control points are established and the control points are solved. The solved control points are shown as the green points in Figure 8. Finally, according to the node vector and the control points, the path points are interpolated and fitted to obtain a smooth trajectory as shown by the red line in Figure 8. The path after the bidirectional pruning optimal strategy is applied to the path planned by the BZRRT-Connect algorithm is *p*. p_i (i = 1, 2, …, n) are the path points of *p*. The specific calculation process of using the cubic non-uniform B-spline curve to interpolate and fit the path *p* is as follows:

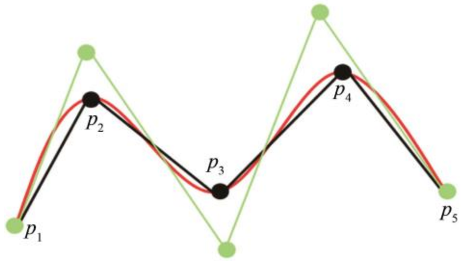


Figure 8 Schematic diagram of using the cubic non-uniform B-spline interpolation method to smooth the path

The mathematical expression of the B-spline curve is as Equation (5).

$$p(u)=\sum_{i=0}^{n} d_i N_{i,k}(u) \tag{5}$$

where, d_i (i = 1, 2, …, n) are the control points; $N_{i,k}(u)$ is the basis function of the *k* times of B-spline.

For the cubic non-uniform B-spline curve, assume the length of each side of the control polygon is $l_i=|p_i-p_{i-1}|$ (i = 1, 2, …, n), the total length is $L=\sum_{i=1}^{n} l_i$. Parameterize the node vector based on the chord length and obtain the node vector as shown in Equation (6). Then the basis function of B-spline can be calculated through the B-spline recursion Equation (7).

$$U=\left[0,0,0,0,\frac{l_1+l_2}{L},\frac{l_1+l_2+l_3}{L},L,\frac{\sum_{j=1}^{n\text{-}2} l_j}{L},1,1,1,1\right] \tag{6}$$

$$\begin{cases} N_{i,0}(u)=\begin{cases} 1, & u_i \le u \le u_{i+1} \\ 0, & others \end{cases} \\ N_{i,k}=\dfrac{u-u_i}{u_{i+k}-u_i}N_{i,k\text{-}1}(u)+\dfrac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}}N_{i+1,k\text{-}1}(u) \\ \dfrac{0}{0}=0;\ 0\times\infty=0 \end{cases} \tag{7}$$

The cubic non-uniform B-spline function of the reversely calculated control points is as Equation (8). As all the elements in the coefficient matrix are all basis function values of B-spline and are only correlated with the node vector, Equation (8) can be simplified as Equation (9), the parameters of which are shown as Equation (10).

$$\begin{bmatrix} N_{1,3}(u_3) & N_{2,3}(u_3) & & N_{0,3}(u_3) \\ N_{1,3}(u_4) & N_{2,3}(u_4) & N_{3,3}(u_4) & \\ \ddots & \ddots & \ddots & \\ & N_{n\text{-}2,3}(u_{n+2}) & N_{n\text{-}1,3}(u_{n+2}) & N_{n,3}(u_{n+2}) \\ N_{n+1,3}(u_{n+3}) & & N_{n\text{-}1,3}(u_{n+3}) & N_{n,3}(u_{n+3}) \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n\text{-}1} \\ d_n \end{bmatrix} = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n\text{-}2} \\ p_{n\text{-}1} \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n\text{-}1} & b_{n\text{-}1} & c_{n\text{-}1} \\ c_n & & & a_n & b_n \end{bmatrix} \tag{9}$$

$$\begin{cases} a_i=\dfrac{(\Delta_{i+2})^2}{\Delta_i+\Delta_{i+1}+\Delta_{i+2}} \\ b_i=\dfrac{(\Delta_i+\Delta_{i+1})\Delta_{i+2}}{\Delta_i+\Delta_{i+1}+\Delta_{i+2}}+\dfrac{\Delta_{i+1}(\Delta_{i+2}+\Delta_{i+3})}{\Delta_{i+1}+\Delta_{i+2}+\Delta_{i+3}} \\ c_i=\dfrac{(\Delta_{i+1})^2}{\Delta_{i+1}+\Delta_{i+2}+\Delta_{i+3}} \\ e_i=(\Delta_{i+1}+\Delta_{i+2})p_{i\text{-}1} \end{cases} \tag{10}$$

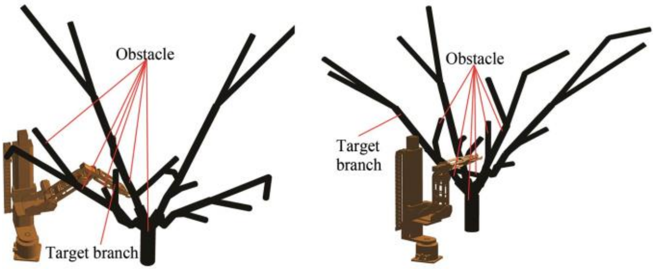
Solve Equation (8) to get all the control points, and then according to the node vector, the solved control points and Equation (5), the path points are interpolated and fitted to obtain a smooth trajectory. Although BZRRT-Connect algorithm can improve the path planning efficiency, the quality of the path needs further improvement by bidirectional pruning optimal strategy and cubic non-uniform B-spline interpolation method. Therefore, the whole improved algorithm is called BZSRRT-Connect.

4 Simulation experiment

This paper uses MATLAB 2019b to build a simulation system for the fruit tree pruning manipulator and the simulation path planning experiment of the fruit tree pruning manipulator is carried out in the built simulation system of the fruit tree pruning manipulator. The simulation experiment platform uses a computer with 3.3 GHz of main frequency and 8 GB of memory. To facilitate and visualize the research, the method of using cylindrical envelope method for the establishment of obstacle model is applied to build the jujube model in Matlab.

4.1 Experiment design

In order to verify the superiority and effectiveness of the proposed algorithm, in scene A and scene B with obstacles, three algorithms of RRT-Connect, BZRRT-Connect and BZSRRT-Connect were used respectively for the simulation experiment, and each algorithm with 20 times of path planning simulation experiment in each scenario. For each algorithm in each scene, we summarized the average path planning time, the average path length and the path planning success times with 20 s of the time limit and calculate the variation coefficient of the path length to compare the stability of the path. Scene A and scene B are shown in Figure 9. In scene A and scene B, the initial pose joint value of the manipulator, the target pose joint value of the manipulator, and the initial step size of the algorithm are listed in Table 2. The target pose joint value of the manipulator is calculated according to the pruning point and the growth situation of the branch to be pruned. The obstacle parameters are listed in Table 3.



a. Scene A with obstacle b. Scene B with obstacle

Figure 9 Planning scenes

Table 2 Main parameters in different scenes

Scene	Initial pose joint value	Target pose joint value	Initial step size
A	(18°, 150 mm, 35°, 10°, 0°)	(–18.4349°, 411.7332 mm, 76.6274°, –15.3308°, -90°)	2
B	(–18°, 60 mm, 90°, 0°, –60°)	(28.9601°, 125.5624 mm, 86.6069°, 11.0724°, –121.6061°)	2

Table 3 Obstacle parameters in different scenes

Scene	Number	Starting coordinates/mm	Termination coordinates/mm	Radius/mm
A	1	(800, 0, 300)	(730, 70, 430)	35
	2	(730, 70, 430)	(700, 100, 500)	25
	3	(700, 100, 500)	(630, 170, 600)	25
	4	(630, 170, 600)	(400, 400, 1100)	25
	5	(630, 170, 600)	(550, 170, 800)	20
	6	(550, 170, 800)	(550, 120, 1000)	20
	7	(800, 0, 300)	(730, –70, 425)	35
	8	(730, –70, 425)	(694, –106, 425)	25
	9	(694, –106, 425)	(724, –176, 610)	24
	10	(724, –176, 610)	(376, –424, 1050)	25
	11	(694, –106, 425)	(630, –170, 525)	20
	12	(630, –170, 525)	(500, 50, 600)	19
	13	(800, 0, 0)	(800, 0, 300)	50
	14	(376, –424, 1050)	(100, –400, 1500)	19
B	1	(800, 0, 0)	(800, 0, 300)	50
	2	(800, 0, 300)	(730, 70, 430)	35
	3	(730, 70, 430)	(700, 100, 500)	25
	4	(700, 100, 500)	(630, 170, 600)	25
	5	(630, 170, 600)	(550, 170, 800)	20
	6	(550, 170, 800)	(550, 120, 1000)	20
	7	(800, 0, 300)	(730, –70, 425)	35
	8	(730, –70, 425)	(694, –106, 425)	25
	9	(694, –106, 425)	(724, –176, 610)	24
	10	(724, –176, 610)	(376, –424, 1050)	25
	11	(694, –106, 425)	(630, –170, 525)	20
	12	(630, –170, 525)	(500, 50, 600)	19
	13	(724, –176, 610)	(600, -200, 700)	19
	14	(600, –200, 700)	(600, –200, 1000)	19

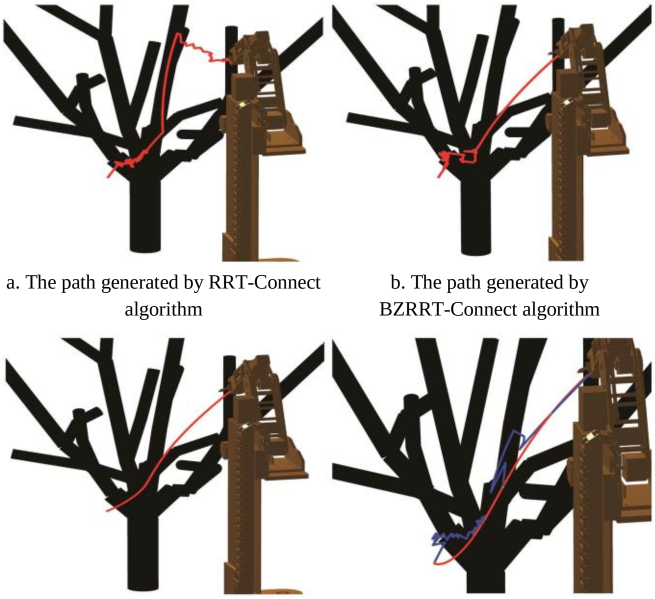
4.2 Results and analysis

Table 4 shows the statistical calculation results of using different algorithms in different scenes for the manipulator path planning simulation experiments. As shown in Table 4, the path planning time and the path length of BZRRT-Connect are reduced by 58.5% and 24.24% respectively compared with RRT-Connect in scene A. The path planning time and the path length of BZSRRT-Connect are reduced by 52.1% and 58.35% respectively compared with RRT-Connect in scene A. The path planning time and the path length of BZRRT-Connect are reduced by 56.16% and 25.18% respectively compared with RRT-Connect in scene B. The path planning time and the path length of BZSRRT-Connect are reduced by 53.16% and 68.11% respectively compared with RRT-Connect in scene B. No matter in scene A or scene B, the path planning success times of BZRRT-Connect and BZSRRT-Connect are higher than RRT-Connect. Moreover, compared with BZRRT-Connect and RRT-Connect, the variation coefficient of the path length of BZSRRT-Connect is the smallest, which are 0.06 and 0.15 respectively.

The red lines in Figure 10a–c represent the paths generated by the path planning simulation experiment of the manipulator in scene A using the RRT-Connect algorithm, the BZRRT-Connect algorithm and the BZSRRT-Connect algorithm, respectively. Figure 10d shows the path generated by the manipulator using the BZRRT-Connect algorithm in scene A and this path is optimized by using the bidirectional pruning optimal strategy and cubic non-uniform B-spline interpolation method. The blue line represents the path generated by the BZRRT-Connect algorithm, the green dots stand for the key path points obtained by the bidirectional pruning optimal strategy, and the red line refers to the path after interpolation fitting on the key path points using the cubic non-uniform B-spline interpolation method. As shown, the paths generated by the RRT-Connect algorithm and the BZRRT-Connect algorithm are spatial broken line segments, which are not smooth and contain many unnecessary points. Comparatively, the path generated by the BZSRRT-Connect algorithm is the shortest and smoothest. Figure 11 shows the changes of joint value of each joint in the process of simulation path planning experiment of the manipulator using the BZSRRT-Connect algorithm.

Table 4 Simulation results of three algorithms in different scenes

Scene	Algorithm	Average path planning time/s	Average path length/mm	Standard deviations	Variation coefficient of path length	Success times
A	RRT-Connect	19.54	1678.04	206.47	0.123	16
	BZRRT-Connect	8.11	1271.3	190.33	0.15	20
	BZSRRT-Connect	9.36	699.44	44.12	0.06	20
B	RRT-Connect	12.34	2067.47	1007.263	0.49	18
	BZRRT-Connect	5.41	1546.92	775.01	0.5	20
	BZSRRT-Connect	5.78	659.3	97.98	0.15	20


a. The path generated by RRT-Connect algorithm

b. The path generated by BZRRT-Connect algorithm

c. The path generated by BZSRRT-Connect algorithm

d. The path generated by BZRRT-Connect is optimized

Figure 10 Comparison of three algorithms

According to Table 4 and Figures 10a and 10b, the path planning time and the path length of the BZRRT-Connect algorithm are reduced by about 60% and about 25% respectively compared with the RRT-Connect algorithm. But the planned path using the BZRRT-Connect algorithm is a spatial broken line segment which is not smooth and contains many unnecessary points. The reduction of the path length is also not significant. This indicates that the BZRRT-Connect algorithm can rapidly plan a feasible path, but the quality of the path is not satisfactory.

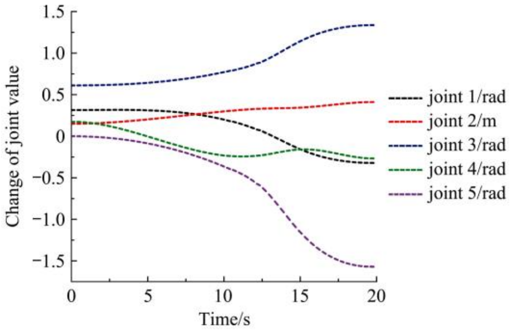


Figure 11 Changes of joint value of each joint in the process of path planning

According to Table 4 and Figure 10, the path planning time and the path length of the BZSRRT-Connect algorithm are reduced by about 55% and about 60% respectively compared with the RRT-Connect algorithm. The path planning success rate of the BZSRRT-Connect algorithm is 100%. Compared with the BZRRT-Connect algorithm and the RRT-Connect algorithm, the path planned by the BZSRRT-connect algorithm is smoother and the quality of the planned path is more stable. In addition, it can be seen from Figure 11 that in the simulation path planning experiment of the manipulator, the joint value changes smoothly and continuously without any sudden change or vibrate. Therefore, the BZSRRT-Connect algorithm can quickly plan a shorter path that is smooth, continuous and executable. The results of the simulation experiment have verified the superiority and effectiveness of the BZSRRT-Connect algorithm.

5 Practical experiment

To verify the feasibility and effectiveness of the proposed algorithm on actual fruit tree pruning manipulator, a real path planning experiment platform of the fruit tree pruning manipulator is built as Figure 12. The platform consists of the fruit tree pruning manipulator, jujube tree, jujube clamping mechanism, and 3D high-speed camera. The experiment process is as follows:

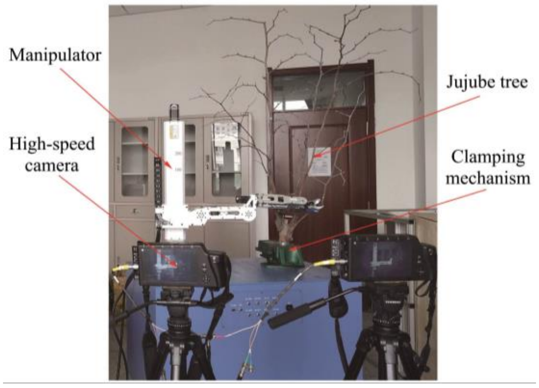


Figure 12 Experiment platform

Firstly, scan the experiment platform with LIDAR to obtain the three-dimensional coordinates of the jujube pruning point (goal point) and the obstacle parameters. Secondly, establish the base coordinate system with the point where the installation position of the manipulator is perpendicular to the working table as the origin to obtain the three-dimensional coordinates of the pruning point and the obstacle parameters under the base coordinate system. Then, make the obstacle avoidance path planning of the fruit tree pruning manipulator with the initial point, the goal point and the obstacle parameters under the base coordinate system as the input parameters of the algorithm to generate a series of path points. Finally, drive the manipulator to the goal point according to path points and use 3D high-speed camera to record the whole movement process of the manipulator.

5.1 Experiment design

As shown in Figure 13a, the manipulator starts from the initial point has to avoid the obstacles to prune the target branch. In this scene, the three algorithms of RRT-Connect, BZRRT-Connect and BZSRRT-Connect were used respectively for the practical experiment, each algorithm with 10 times of practical path planning experiment. For each algorithm, we summarized the average path planning time and the path planning success times with 45 s of the time limit for the planning time. The joint value of the initial pose of the manipulator is (28°, 20 mm, 80°, 50°, 0°), the joint value of the goal pose of the manipulator is (–15.8529°, 43.0468 mm, 0.6895°, 85.0829°, –30.582°), and the initial step size of the algorithms is the same as simulation experiment. The obstacle parameters are shown in Table 5.

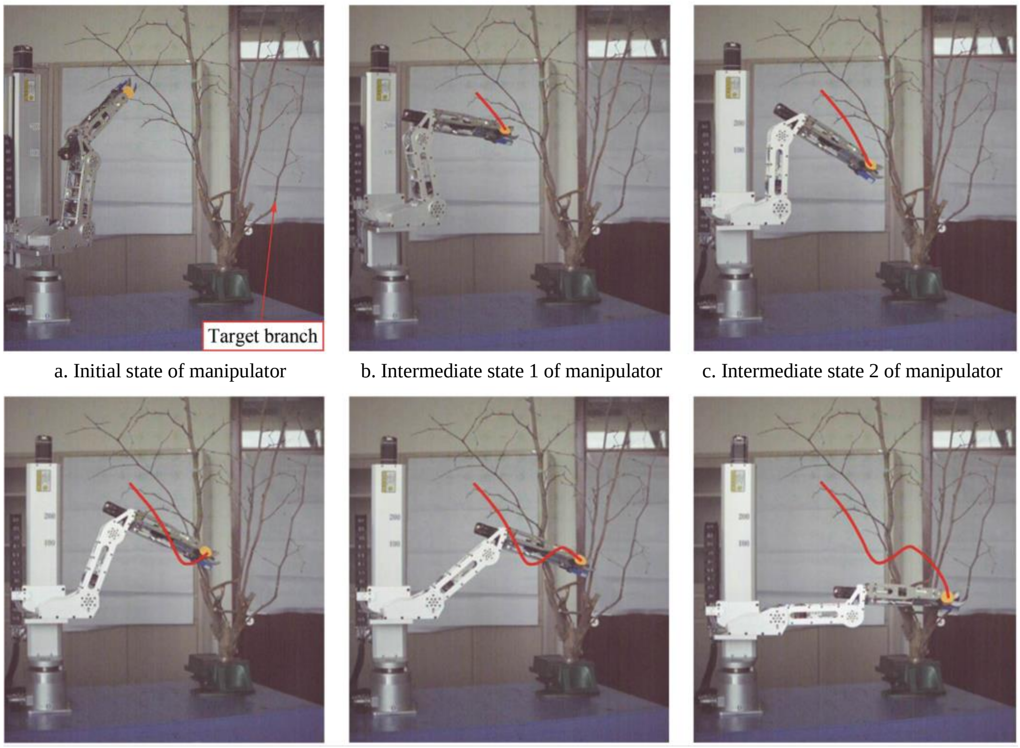
Table 5 Obstacle parameters (mm)

Number	Starting coordinates	Termination coordinates	Radius
1	(626.59, 37.60, 440)	(607, 36.4, 625)	22
2	(607, 36.4, 625)	(465, –75, 739)	22
3	(465, –75, 739)	(619.5, 80.67, 725.3)	22
4	(619.5, 80.67, 725.3)	(623, 37.2, 805.6)	22
5	(623, 37.2, 805.6)	(567, 40.18, 1019.2)	22
6	(465, –75, 739)	(455.48, –71.9, 741.1)	10
7	(455.48, –71.9, 741.1)	(329, –169, 718)	10
8	(619.5, 80.67, 725.3)	(665, 151, 809.4)	9
9	(665, 151, 809.4)	(695, 183, 894)	9
10	(623, 37.2, 805.6)	(590, –179, 864)	10
11	(590, –179, 864)	(602, –360, 725)	10
12	(748.6, –35.2, 379)	(771, –36, 502)	12.5
13	(771, –36, 502)	(928, –42, 1362.8)	12.5
14	(700, 80, 200)	(626.59, 37.6, 440)	32
15	(680, 30, 210)	(748.6, –35.2, 379)	25
16	(734, –48.49, 264.5)	(758, –166, 337)	10
17	(731, –338, 517)	(722.38, –391.7, 578)	9
18	(731, –338, 517)	(707, –312, 580)	9
19	(771, 80, 100)	(771, 80, 200)	85

5.2 Results and analysis

Figure 13 presents the movement process of the manipulator using the BZSRRT-Connect algorithm to perform obstacle avoidance path planning experiment. Figure 13a-f shows the states of the manipulator at different times in the process of moving from the initial point to the goal point while avoiding obstacles. The red lines are the movement paths of the manipulator, which are obtained by using the 3-D-Manager in ProAnalyst software to feature tracking the movement process of the manipulator recorded by 3D high-speed camera. The interface of 3-D-Manager is shown in Figure 14. The 3-D calibration panel uses a 3-D calibration board to calibrate the 3D high-speed camera and establishes a three-dimensional coordinate system with the bottom center of the 3-D calibration board as the origin. The 3-D measurement panel takes the scissors mouth center of the end effector as the feature point to automatically track the movement process of the manipulator and calculate the three-dimensional coordinates of each tracked point. The 3-D manager panel records and outputs the results of the feature tracking. Figure 15 shows the paths generated by the obstacle avoidance path planning experiments using the RRT-Connect algorithm, the BZRRT-Connect algorithm and the BZSRRT-Connect algorithm. The paths are obtained by using 3-D-Manager to feature tracking the movement process of the manipulator recorded by 3D high-speed camera, which are labeled with different colors. The red line represents the path generated by the BZSRRT-Connect algorithm, the blue line represents the path generated by the BZRRT-Connect algorithm, and the green line represents the path generated by the RRT-Connect algorithm. As shown, the paths planned by the RRT-Connect algorithm and the BZRRT-Connect algorithm are spatial broken line segments which are not smooth and contain many unnecessary points. Compared with the RRT-Connect algorithm and the BZRRT-Connect algorithm, the path generated by the BZSRRT-Connect algorithm is the shortest and smoothest.

Table 6 shows the statistical results of the practical experiments. As shown, the path planning time of BZRRT-Connect and BZSRRT-Connect decrease by 62.56% and 48.24% respectively compared with RRT-Connect, and the path planning success times of BZRRT-Connect and BZSRRT-Connect are higher than RRT-Connect. Figure 16 shows the changes of joint value of each joint in the process of practical obstacle avoidance path planning experiment of the manipulator using the different algorithms.



a. Initial state of manipulator b. Intermediate state 1 of manipulator c. Intermediate state 2 of manipulator

d. Intermediate state 3 of manipulator e. Intermediate state 4 of manipulator f. Final state of manipulator

Figure 13 Motion process of the obstacle avoidance path planning experiment of the manipulator

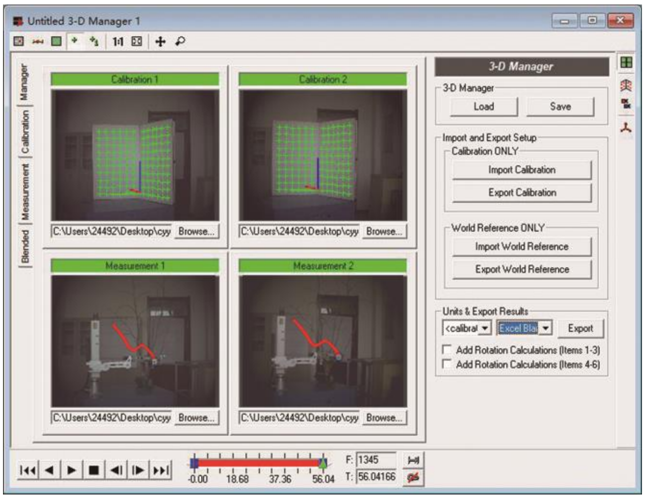


Figure 14 3-D-Manager interface

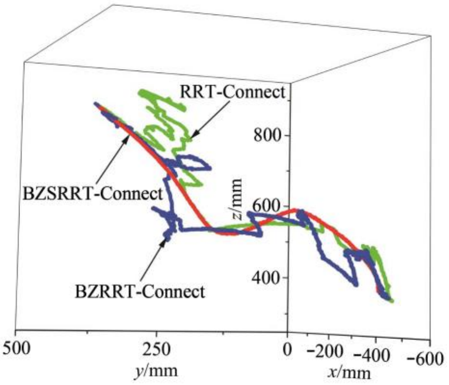


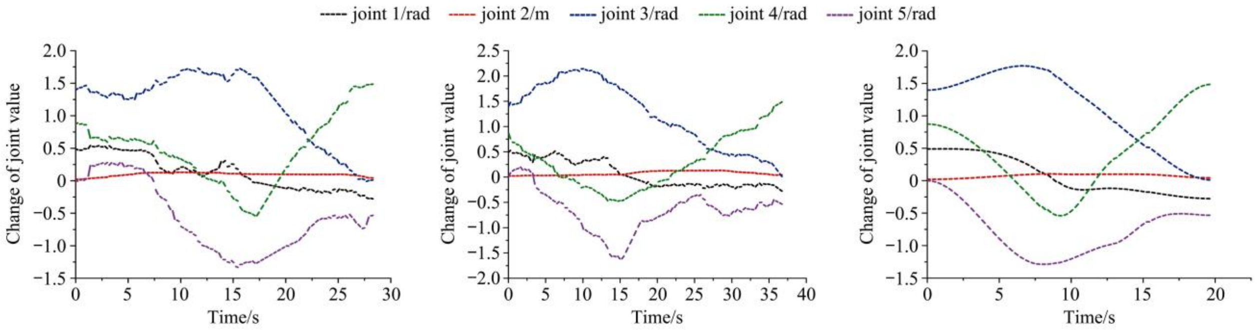
Figure 15 Results of feature tracking of the paths generated by the three algorithms

Table 6 Statistical results of the practical experiments

Algorithm	Average path planning time/s	Success times
RRT-Connect	39.8	8
BZRRT-Connect	14.9	10
BZSRRT-Connect	20.6	10

As can be seen from Table 6 and Figure 15, although the path planning time of the BZRRT-connect algorithm is reduced by 62.56% compared with the RRT-connect algorithm, the planned path using the BZRRT-Connect algorithm is a spatial broken line segment which is not smooth and contains many unnecessary points. In addition, it can be seen from Figure 16b that in the process of the path planning using the BZRRT-Connect algorithm, the joint value change of each joint of the manipulator is not smooth, discontinuity, and there are sudden change and vibrate. This shows that the BZRRT-Connect algorithm can quickly plan a feasible path, but the path quality is not satisfactory.

According to Table 6, Figure 13 and Figure 15, compared with the RRT-Connect algorithm, the BZSRRT-Connect algorithm can realize a shorter path planning time, shorter path length and smoother path, and the planned path can effectively guide the manipulator to avoid obstacles and lead the end effector of the manipulator to the goal point. It can be seen from Figure 16 that in the process of the path planning using the BZSRRT-Connect algorithm, the joint value changes smoothly and continuously without any sudden change or vibrate. Therefore, the BZSRRT-Connect algorithm can rapidly plan a smooth, continual and executable path and the path can effectively guide the manipulator to avoid obstacles and make the end effector of the manipulator reach the goal point, thus verifying the feasibility and effectiveness of the proposed algorithm for the actual fruit tree pruning manipulator.



a. The variations of each joint value generated by RRT-Connect algorithm

b. The variations of each joint value generated by BZRRT-Connect algorithm

c. The variations of each joint value generated by BZSRRT-Connect algorithm

Figure 16 Changes of joint value of each joint in the practical obstacle avoidance path planning experiment

6 Conclusions and future work

An improved RRT-Connect algorithm was proposed to solve the obstacle avoidance path planning of the fruit tree pruning manipulator in fruit tree pruning environment. Based on the basic RRT-Connect algorithm, the improved RRT-Connect algorithm introduced the goal-biased strategy and the adaptive step size adjustment principle to accelerate the path search speed and reduce the path planning time. The improved RRT-Connect algorithm used the bidirectional pruning optimal strategy and the cubic non-uniform B-spline interpolation method to optimize the path generated by the RRT-connect algorithm to improve the path quality. The path planning simulation experiment of the manipulator was carried out in the fruit tree pruning manipulator simulation system. The results showed that, compared with the basic RRT-connect algorithm, the path planning time of the improved RRT-connect algorithm was reduced by about 55%, the path length was reduced by about 60%, the planned path was smoother, and the quality of the planned path was more stable. The path planning success rate of the improved RRT-connect algorithm was 100% and the planned path was smooth, continual and executable. It was verified that the improved RRT-connect algorithm not only had shorter path planning time, but also had higher path quality. The practical obstacle avoidance path planning experiment for the fruit tree pruning manipulator was carried out on the real path planning experiment platform of the fruit tree pruning manipulator. The results showed that the improved RRT-Connect algorithm could rapidly plan a smooth, continual and executable path and the path could effectively guide the manipulator to avoid obstacles and make the end effector of the manipulator reach the goal point. The results also verified the feasibility and effectiveness of the improved RRT-Connect algorithm.

There are also limitations in the proposed algorithm. The collision detection algorithm used in this study is based on the premise that the obstacle location is fixed. When in the field environment, the location of the obstacle may change due to the wind disturbance. At this time, the proposed algorithm can’t complete real-time obstacle avoidance, which will cause some restrictions on the path planning of the manipulator. Therefore, the next step is to study the real-time obstacle avoidance path planning of the manipulator.

Acknowledgements

This study was supported by the National Natural Science Foundation of China (No. 51765058) and the Corps Regional Innovation Guidance Plan (No. 2021BB020).

[References]

[1] Fu W, Liu Y D, Kan Z, Pan J B, Cui J. Development status and trend of orchard pruning machine. Journal of Agricultural Mechanization Research, 2017; 39(10): 7–11. (in Chinese)

[2] Peng F, Zheng H N, Lu S H, Shi Z T, Liu X X, Li L. Growth model and visualization of a virtual jujube tree. Computers and Electronics in Agriculture, 2019; 157: 146–153.

[3] Botterill T, Paulin S, Green R, Williams S, Lin J, Saxton V, et al. A robot system for pruning grape vines. Journal of Field Robotics, 2016; 34(6): 1100–1122.

[4] Feng L, Jia J H. Improved algorithm of RRT path planning based on comparison optimization. Computer Engineering & Applications, 2011; 47: 210–213.

[5] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research, 1986; 5(1): 90–98.

[6] Van Henten E J, Hemming J, Van Tuijl B, Kornet J G, Bontsema J. Collision free motion planning for a cucumber picking robot. Biosystems Engineering, 2003; 86(2): 135–144.

[7] Lazarowska A. Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation. Journal of Navigation, 2015; 68(2): 291–307.

[8] Van Henten E J, Hemming J, Van Tuijl B, Kornet J G, Meuleman J, Bontsema J, et al. An autonomous robot for harvesting cucumbers in greenhouses. Autonomous Robots, 2002; 13(3): 241–258.

[9] Rostami S M H, Sangaiah A K, Wang J, Liu X Z. Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP Journal on Wireless Communications and Networking, 2019; 2019(1): 1–19.

[10] Ajeil F H, Ibraheem I K, Azar A T, Humaidi A J. Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. Sensors, 2020; 20(7): 1880. doi: 10.3390/s20071880.

[11] Cao X M, Zou X J, Jia C Y, Chen M Y, Zeng Z Q. RRT-based path planning for an intelligent litchi-picking manipulator. Computers and Electronics in Agriculture, 2019; 156: 105–118.

[12] Luo L F, Wen H J, Lu Q H, Huang H J, Chen W L, Zou X J, et al. Collision-free path-planning for six-DOF serial harvesting robot based on energy optimal and artificial potential field. Complexity, 2018; 2018: 1–12.

[13] Lavalle S M. Rapidly-exploring random trees: a new tool path planning. Iowa City: Computer Science Department of Iowa State University, 1998; 98p.

[14] Kulkarni P, Goswami D, Guha P, Dutta A. Path planning for a statically stable biped robot using PRM and reinforcement learning. Journal of Intelligent & Robotic Systems, 2006; 47(3): 197–214.

[15] Kuffner J J, Lavalle S M. RRT-connect: an efficient approach to single-query path planning. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, California: IEEE, 2000; pp. 995–1001.

[16] Nguyen T T, Kayacan E, Baedemaeker J D, Saeys W. Task and motion planning for apple harvesting robot. IFAC Proceedings Volumes, 2013; 46(18): 247–252.

[17] Ma J T. Research on obstacle avoidance motion planning of citrus harvesting robot in unstructured environment. MS dissertation, Chongqing: Chongqing University of Technology, 2019; 96p. (in Chinese)

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

[19] Zhang H J, Wang Y K, Zheng J, Yu J Z. Path planning of industrial robot based on improved RRT algorithm in complex environments. IEEE Access, 2018; 6: 53296–53306.

[20] Wang K, Huang B, Zeng G H, Li X B. Fast path planning algorithm based on improved RRT-Connect. Journal of Wuhan University, 2019; 65(3): 283–289. (in Chinese)

[21] Zhao X L, Cao Z Q, Geng W J, Yu Y Y, Tan M, Chen X C. Path planning of manipulator based on RRT-Connect and Bezier curve. Proceedings of 9th IEEE International Conference on CYBER Technology in Automation, Suzhou: IEEE, 2019; pp. 649–653.

[22] Zhang D G, Xu Y, Yao X T. An improved path planning algorithm for unmanned aerial vehicle based on RRT-Connect. 2018 37th Chinese Control Conference, Wuhan: CCC, 2018; pp. 4854–4858.

[23] Liu H Y, Zhang X B, Wen J, Wang R H, Chen X. Goal-biased bidirectional RRT based on curve-smoothing. IFAC-PapersOnLine, 2019; 52(24): 255–260.

[24] Li Z Y, Zhao D J, Zhao J S. Structure synthesis and workspace analysis of a telescopic spraying robot. Mechanism and Machine Theory, 2019; 133: 295–310.

[25] Liu Y D. Design and research of profiling and pruning device for jujube tree. MS dissertation. Shihezi: Shihezi University, 2018. (in Chinese)

[26] Yuan C Y, Zhang W Q, Liu G F, Pan X L, Liu X H. A heuristic rapidly-exploring random trees method for manipulator motion planning. IEEE Access, 2020; 8: 900–910.