


RESEARCH ARTICLE

Path planning for robots in preform weaving based on learning from demonstration

Zhuo Meng, Shuo Li , Yujing Zhang and Yize Sun

College of Mechanical Engineering, Donghua University, Songjiang, Shanghai, China

Corresponding author: Zhuo Meng; Email: mz@dhu.edu.cn

Received: 9 November 2023; **Revised:** 22 December 2023; **Accepted:** 8 January 2024;

First published online: 1 February 2024

Keywords: weave; industrial robots; path planning; extreme learning machine; rapidly expanding random trees

Abstract

A collision-free path planning method is proposed based on learning from demonstration (LfD) to address the challenges of cumbersome manual teaching operations caused by complex action of yarn storage, variable mechanism positions, and limited workspace in preform weaving. First, by utilizing extreme learning machines (ELM) to autonomously learn the teaching data of yarn storage, the mapping relationship between the starting and ending points and the teaching path points is constructed to obtain the imitation path with similar storage actions under the starting and ending points of the new task. Second, an improved rapidly expanding random trees (IRRT) method with adaptive direction and step size is proposed to expand path points with high quality. Finally, taking the spatial guidance point of imitation path as the target direction of IRRT, the expansion direction is biased toward the imitation path to obtain a collision-free path that meets the action yarn storage. The results of different yarn storage examples show that the ELM-IRRT method can plan the yarn storage path within 2s–5s when the position of the mechanism changes in narrow spaces, avoiding tedious manual operations that program the robot movements, which is feasible and effective.

1. Introduction

The growing demand for high-performance fiber reinforcements in various industries has increased the need for automated weaving equipment, such as in aerospace, shipbuilding, and automotive [1, 2]. As a result, robots are gradually being integrated into preform weaving, replacing manual operations of yarn storage. This shift toward automation not only significantly improves efficiency but also reduces labor costs and ensures product consistency [3]. However, path planning is crucial to fully utilize robot technology. Selecting an appropriate path planning method based on the object being processed and the application scenarios is key to enhancing production quality and efficiency.

The robot must move from its starting point to the ending point without encountering any obstacles to ensure safe path planning. To achieve this, researchers have developed various algorithms suited for different environments, which can be categorized into four types based on their implementation principles. The first type includes graph search methods like the Dijkstra algorithm [4, 5], A* algorithm [6], and D* algorithm [7]. The second type is local planning methods like the artificial potential field method [8, 9] and elastic bands [10]. The fourth type refers to intelligent optimization methods, which include particle swarm optimization algorithm [11, 12], ant colony algorithm [13], genetic algorithm [14], and reinforcement learning [15]. Finally, the fourth type consists of random sampling methods, such as the probabilistic roadmaps (PRM) algorithm [16, 17] and the rapidly expanding random trees (RRT) algorithm [18–20]. The first three types of methods can be widely applied to mobile robots in two-dimensional planes, but they are difficult to be applied to path planning in the joint space for robots with high degrees of freedom. The basic RRT algorithm and its variants, such as RRT* [21] and Bi-RRT [22], are tree structure algorithms based on random sampling, which are widely used in motion planning

for multidegree-of-freedom robots. However, the RRT algorithm has drawbacks such as strong randomness, long planning time, and poor path smoothness. Researchers have made improvements to the RRT algorithm to enhance its rapid expansion capability. One of these enhancements involved modifying the extension direction by adding the vector sum of random gravity and target gravity [23]. This alteration increased the probability of reaching the target point, leading to faster convergence and reduced computational time. Nevertheless, using a fixed gravity coefficient would lead to a higher planning failure rate when dealing with complex obstacles. Some works focused on improving the step size by introducing an adaptive step size RRT algorithm [24], which calculated the maximum allowable step size for each iteration in the joint space under the constraint that the maximum distance of each link's movement is always less than the minimum obstacle width, thereby improving the accuracy of obstacle avoidance. Based on this, this paper introduces an Improved rapidly expanding random trees (IRRT) approach with adaptive direction and step size to enhance the planning performance of robots in challenging weaving environments with complex obstacles.

With the rise of complex application environments for robots, tasks like polishing [25], welding [26], and playing table tennis [27] have become more difficult, requiring improved path planning. To address this, researchers have explored learning from demonstration (LfD) methods that can learn task actions and skill features from manual demonstrations. One such method is the dynamic motion primitives (DMP) based on nonlinear dynamic systems [28, 29]. However, DMP requires precise models and struggles to account for uncertainty in teaching results. To account for uncertainties in teaching results, researchers have put forth the Gaussian mixture model (GMM) method [30, 31]. However, this approach requires extensive teaching data. As a result, researchers have developed and enhanced these methods. In literature [32], a motion model was created using DMP, and GMM was utilized to estimate the nonlinear function of the motion model, allowing for the extraction of more skill features in robot motion. In addition, the neural network is a common method for LfD, which includes the multilayer neural network method [33], radial basis function neural network [34], and extreme learning machines (ELM) [35]. Among them, the ELM, a single hidden layer neural network algorithm [36], is particularly advantageous as it can learn faster while maintaining accuracy with limited teaching data. Therefore, ELM is utilized in this paper to learn the action of yarn storage.

Paths containing action features can be obtained using LfD; however, changes in the application environment may lead to robots colliding with surrounding objects. To address this issue, literature [37] employed a manual teaching approach that involved wearing a magnetic tracker to enable humanoid robots to learn action features. The RRT* planning algorithm was then utilized to bias tree growth toward the teaching path. Unfortunately, the method was not suitable for industrial robots. The case discussed in this paper pertains to large-scale weaving equipment with varying mechanism positions and numerous robots. These robots operate with flexible yarn, necessitating high levels of motion. Given the complexity of the manual teaching process, the ELM-IRRT path planning method is proposed to meet the requirements of yarn storage motion and prevent collisions. By providing the corresponding starting and ending points based on the position of mechanism, a path that satisfies the yarn storage motion requirements can be rapidly planned, which significantly reduces the need for laborious manual teaching and provides a good solution to path planning for robots in preform weaving.

2. The robots for preform weaving

As shown in Fig. 1, the weaving system consists of a circumferential array of 16 basic weaving units. One end of the yarn is fixed to the basic weaving unit, and the other end is fixed to the weaving point of the mandrel.

The basic weaving unit is shown in Fig. 2, and the end of the robot is equipped with electric grippers for gripping yarn. As a result, there will be 16 robots working closely together, and multirobot path planning has been proposed in our previous work [3]. The main task of the robot is to store the sorted yarn in the yarn storage mechanism 2 and the yarn winding mechanism 5 to form a yarn opening. The robot model is ABB-IRB2600ID, and its D-H parameters are shown in Table I.

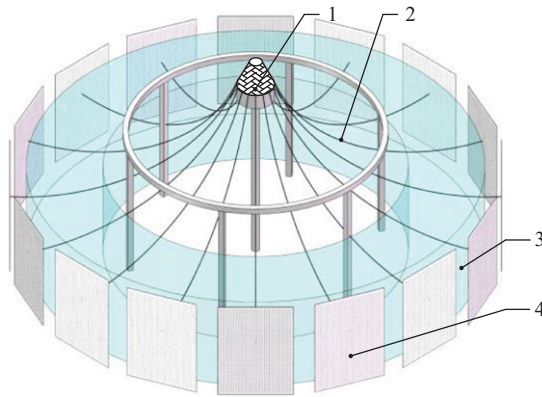


Figure 1. The model of the weaving system (1-Mandrel; 2-Yarns; 3-The workplace of robots; 4-Basic weaving unit).

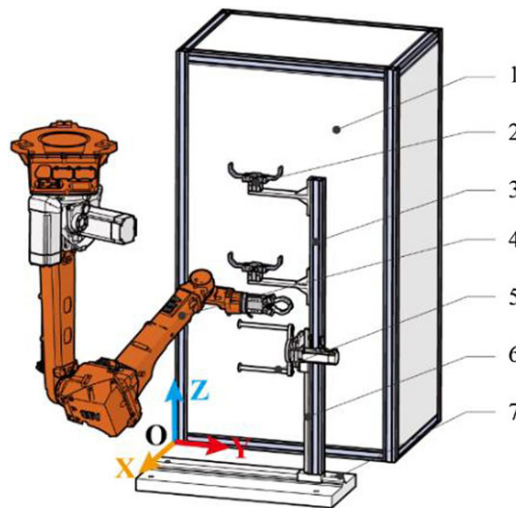


Figure 2. The simplified basic unit of the weaving system (1- yarn fixing unit; 2- yarn storage mechanism; 3- support rod; 4- robot; 5- yarn winding mechanism; 6- vertical sliding guide; 7- horizontal sliding guide; 8- obstacle mechanism).

Facing the challenge of meeting requirements using manual teaching, it is crucial to research a path planning method that works well for weaving robots. Considering the characteristics of the weaving system, the objectives and challenges of path planning are as follows:

1. The planned path must meet the requirements of yarn storage movements. The action of storing yarn by robots is complex, and each mechanism has its own unique storage path. Fig. 3 displays how the robot places yarn into the yarn storage mechanism 2 and the yarn winding mechanism 5, respectively. The curve shows the path of the robot storing yarn, and the arrow shows the direction of movement. Successful yarn storage can only be achieved by employing actions like lifting, avoiding obstacles, storing, and stretching.
2. The planned path should adapt to changes in the position of the mechanism. With the continuous weaving of the preform, the yarn storage mechanism 2 and the yarn winding mechanism 5 move along the guide rail according to a certain law. The winding mechanism 5 will move along

Table I. D-H parameter of robotic arm.

Linkages i	Length of linkages a_{i-1} (mm)	Twist angles α_{i-1} ($^{\circ}$)	Offset of linkages d_i (mm)	Initial value θ_i ($^{\circ}$)	Angel range ($^{\circ}$)
1	0	0	0	0	$-180 \sim 180$
2	150	-90	0	-90	$-95 \sim 155$
3	0.9	0	0	0	$-180 \sim 75$
4	150	-90	938	0	$-175 \sim 75$
5	0	90	0	-180	$-120 \sim 120$
6	0	90	0	0	$-400 \sim 400$

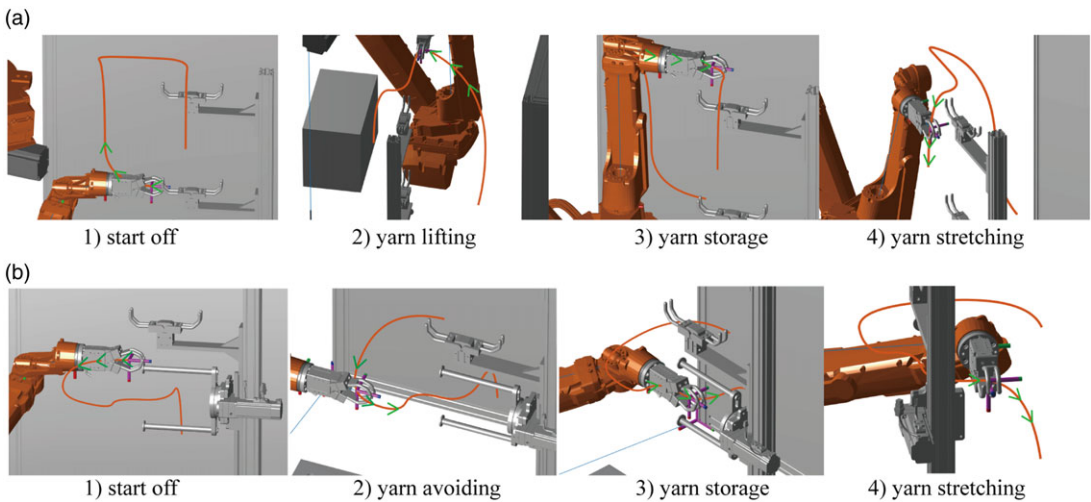


Figure 3. Yarn storage process for two mechanisms. (a) Yarn storage process for yarn storage mechanism 2, (b) yarn storage process for yarn winding mechanism 5.

the vertical sliding guide rail 6, which changes the Z value; the support rod 3 moves along the horizontal sliding guide rail 7, driving the change of Y value of the yarn storage mechanism 2 and the yarn winding mechanism 5.

3. The planned path cannot collide with surrounding institutions. It is frequent to collide with surrounding storage mechanisms during path planning due to the narrow workspace of the robot, which increases the difficulty of path planning.

3. ELM-IRRT path planning method

According to the path planning requirements of the weaving robot, the path planning method, namely ELM-IRRT, is proposed as shown in Fig. 4, which is composed of the teaching and learning phase and the expansion phase. The LfD phase employs ELM to establish a mapping relationship between the starting and ending points and path points, allowing for the learning of various paths of yarn storage. The starting and ending points of current task are input into the mapping relationship to obtain an imitation path with similar yarn storage action.

In the second expansion phase, the spatial guidance points can be extracted from imitation path, which are used to guide the expansion of the path in joint space (6-dimensional space for the weaving robot). The IRRT method with adaptive direction and step size is proposed to realize path planning in joint space. The expansion direction of the tree nodes is biased toward the spatial guide points to maintain a

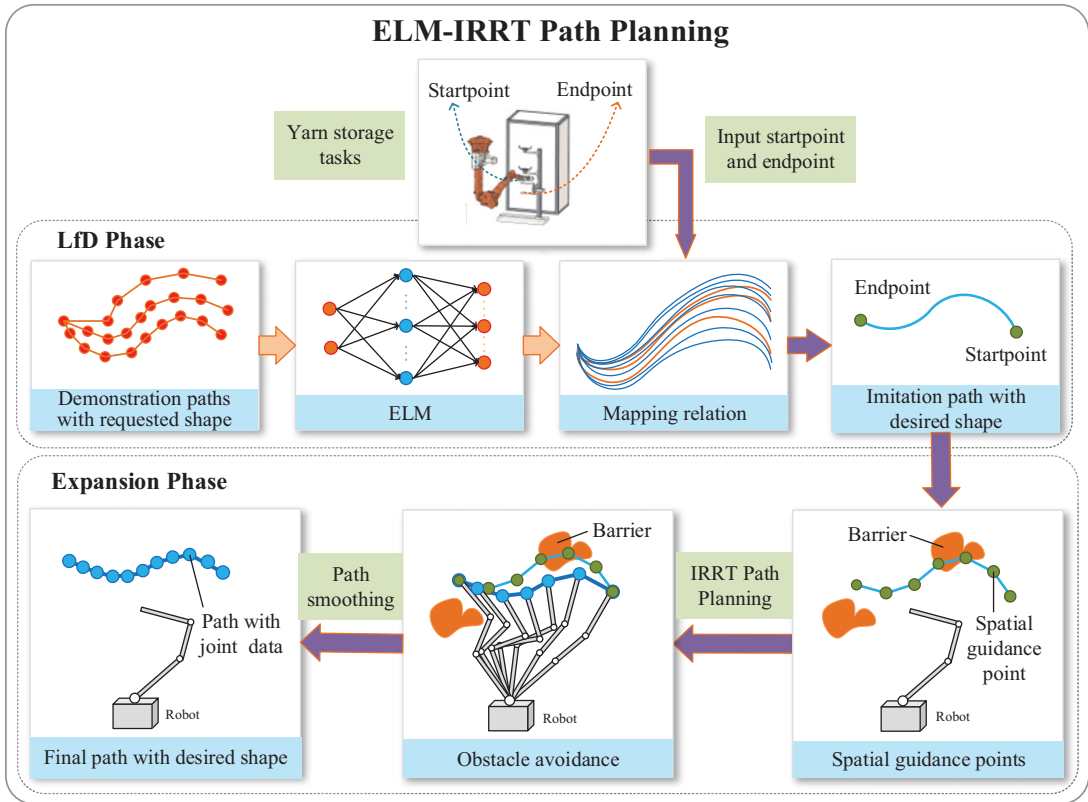


Figure 4. The principle of the ELM-IRRT path planning method.

similar path shape. At the same time, based on the obstacle avoidance ability of IRRT, each node in the three-dimensional space does not collide with obstacles, and each linkage of the robot does not collide with obstacles. Finally, the collision-free path satisfying the yarn storage action is obtained in the form of robot joint data.

3.1. The LfD phase

3.1.1. Task demonstration and data collection

In the task demonstration, the yarn storage paths of the two mechanisms at different starting points $(X_{start}, Y_{start}, Z_{start})$ and ending points $(X_{end}, Y_{end}, Z_{end})$ are taught. Each path is composed of a sequence of path points (x_i, y_i, z_i) . Collect teaching data for storing yarn process in yarn storage mechanism 2 and yarn winding mechanism 5. First, B-spline interpolation is used to convert the sparse path points into rich training data due to the sparsity of teaching path points. Second, split the coordinate values according to the coordinate direction and normalize them to obtain the coordinate sequences $t_x, t_y,$ and t_z for the directions $x, y,$ and $z,$ respectively. Finally, the training samples $D_x = \{X, t_x\}, D_y = \{Y, t_y\}$ and $D_z = \{Z, t_z\}$ for each coordinate are composed, where $X, Y,$ and Z are the sets of starting and ending points in the $x, y,$ and z directions, respectively.

3.1.2. Network model training by ELM

ELM has high learning accuracy and less network training time, and its output is that

$$t_{ij} = o = \sum_{i=1}^L \beta_i f_i(Z_j) = \sum_{i=1}^L \beta_i g(W_i^T Z_j + b_i), j = 1, \dots, N \tag{1}$$

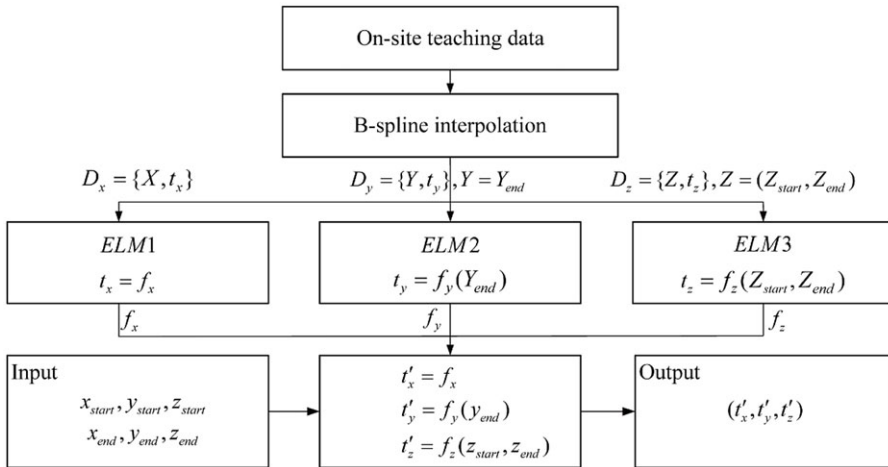


Figure 5. Training framework of ELM.

In Eq. (1), L represents the number of hidden nodes, N represents the number of training samples, and $g(x)$ represents the activation function. $W_i = [w_{i,1}, w_{i,2}]$ is the weight vector connecting the i th hidden node and the input node, $\beta_i = [\beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,m}]$ is the weight vector connecting the i th hidden node and the output node, and b_i is the deviation of hidden layer. The values of W_i and b_i are randomly given during the training process, and the minimum square method is used to minimize the error of the output result

$$\min(H\beta^T - T) \tag{2}$$

$$H(W, Z, b) = \begin{bmatrix} g(w_1^T Z_1 + b_1) & \dots & g(w_L^T Z_1 + b_L) \\ \dots & \dots & \dots \\ g(w_1^T Z_N + b_1) & \dots & g(w_L^T Z_N + b_L) \end{bmatrix} \in R^{N \times L} \tag{3}$$

In Eq. (2), H is the hidden layer output matrix, β is obtained from $\hat{\beta}^T = H^\dagger T$ where H^\dagger is the Moore–Penrose generalized inverse matrix of matrix H .

The ELM network model is trained by training samples $D_x = \{X, t_x\}$, $D_y = \{Y, t_y\}$, and $D_z = \{Z, t_z\}$ to obtain the mapping relationships of $f_x: X \rightarrow t_x$, $f_y: Y \rightarrow t_y$, and $f_z: Z \rightarrow t_z$. The training framework of ELM is shown in Fig. 5. X remains unchanged since there is no guide rail in the x direction, $f_x = t_x$; the independent variable of f_y is Y_{end} since the starting point of the system path is located at the vertical centerline of yarn fixing unit 1; and the independent variables of f_z are Z_{start} and Z_{end} .

3.1.3. The output of imitation path

According to the position of the mechanism on the vertical sliding guide rail 6 or horizontal sliding guide rail 7, input the starting point $(x_{start}, y_{start}, z_{start})$ and ending point $(x_{end}, y_{end}, z_{end})$ into the mapping relationship to obtain the imitation path (t'_x, t'_y, t'_z) that resembles the yarn storage actions in the teaching path.

3.2. Expansion phase

3.2.1. Collision detection

In order to accurately describe the robot and obstacle models, three envelope types of capsule, sphere, and cuboid are used to envelop the basic weaving unit. The algebraic method in literature [38] can be used to quickly realize the collision detection between capsule and capsule, capsule and sphere, and

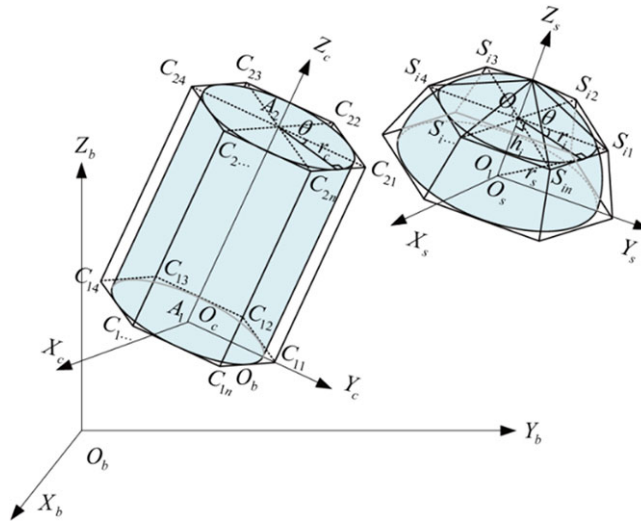


Figure 6. Convex multilateral body converted from cylinder and sphere transformation.

sphere and sphere. However, the collision detection between cuboid and capsule and between cuboid and sphere is more complicated. In such cases, the capsule and sphere need to be converted into convex multilateral bodies, and then GJK algorithm [39] can be used to detect the collision. The capsule body can be regarded as a combination of cylinder and hemisphere, as shown in Fig. 6. Similarly, the sphere is a combination of two hemispheres, and the outer tangential support point of the cylinder and hemisphere is the vertex of the convex multilateral body.

A circumscribed prism with n edges is created on a cylinder with a radius of r_c . The coordinates of the support points of the cylinder are expressed as

$${}^cC_{ij} = A_i + \frac{r_c}{\cos \theta} [\cos(2(j-1)\theta), \sin(2(j-1)\theta), 0], \quad i = 1, 2; j = 1, 2, \dots, n \quad (4)$$

In Eq. (4), $\theta = \pi/n$. For a sphere with radius r_s , the vertices of outer tangent polygon for m parallel circles are taken as the support points of the sphere, which are expressed as

$${}^sS_{ij} = O_i + \frac{r_i}{\cos \theta} [\cos(2(j-1)\theta), \sin(2(j-1)\theta), 0], \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (5)$$

In Eq. (5), $O_i = [0, 0, h_i]$, $h_i = \frac{i}{m+1}r_s$, and $r_i = \sqrt{r_s^2 - h_i^2}$. In order to detect the collision, a conversion process is carried out on the coordinates from $\{O_c\}$ to $\{O_b\}$ and $\{O_s\}$ to $\{O_b\}$, obtaining the support points ${}^bC_{ij}$ and ${}^bS_{ij}$. The GJK algorithm is then utilized for the collision detection of cuboid and sphere.

A detection queue is formed by the envelope of all obstacles. The method for collision detection is chosen based on the type of envelope to detect collision with each linkage of the robot individually. If there is any collision between the envelope of obstacle and the envelope of robot linkage, it will be flagged as a collision. Fig. 7 illustrates the basic weaving unit enveloped by capsule, sphere, and cube envelope.

3.2.2. IRRT algorithm

The RRT algorithm is a path planning method based on random sampling. The classic RRT algorithm first constructs a tree with the root node as the starting point in the joint space. Then a random point q_{rand} is generated in the joint space, and the node q_{near} closest to q_{rand} is found through traversing the tree. As the parent node, q_{near} expand a step along the q_{rand} direction to generate a new node q_{new} . Finally, the each link position of the robot in q_{new} is calculated to detect the collision. If the collision occurs, the point is discarded, otherwise, q_{new} is added to the tree. If the distance between q_{new} and q_{goal} is less than

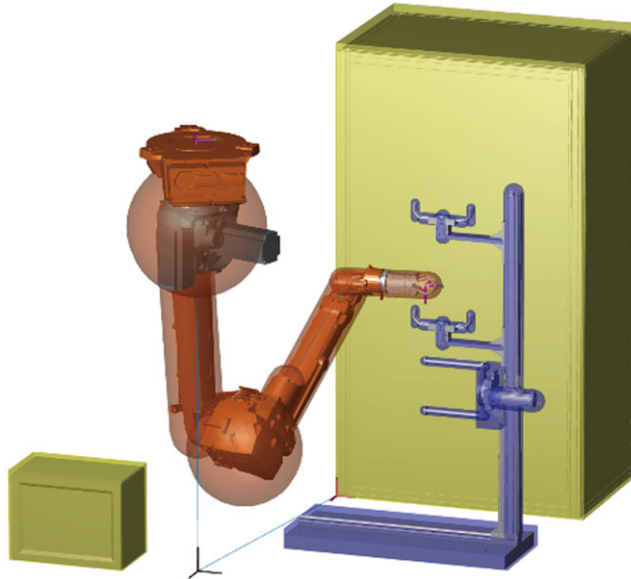


Figure 7. The basic weaving unit enveloped by capsule, sphere, and cube envelope.

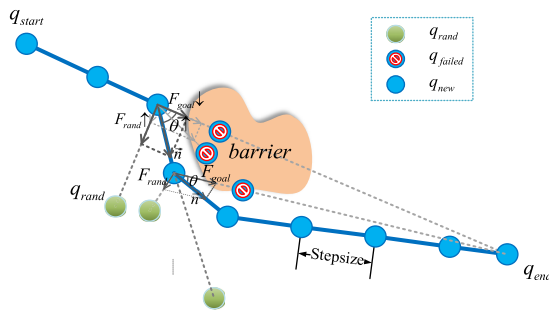


Figure 8. Schematic diagram of IRRT.

the set threshold, the path expansion is successful. The above process is repeated until the expansion is successful or the number of iterations is exceeded.

The expansion process of the basic RRT algorithm is time consuming to converge to the target point in high-dimensional space. In addition, the step length of the joint space movement requires repeated trial and error processes. For this reason, as shown in Fig. 8, an improved RRT algorithm with adaptive direction and step size, namely the IRRT algorithm, is proposed to improve path quality.

(1) Adaptive direction.

The adaptive direction aims to synthesize the expansion direction according to the target gravitational vector and the random gravitational vector. The directions of the target gravitational vector and the random gravitational vector are determined by the target point and the random sampling point, respectively. The gravitational force of q_{goal} is defined as F_{goal} , and the gravitational force of q_{rand} is defined as F_{rand} . The expansion direction is defined as

$$n = F_{rand} \left(\frac{q_{rand} - q_{near}}{\|q_{rand} - q_{near}\|} \right) + F_{goal} \left(\frac{q_{orient} - q_{near}}{\|q_{orient} - q_{near}\|} \right) \tag{6}$$

Taking the expansion failures number N_{failed} of its parent node as the independent variable, the new functions of F_{rand} and F_{goal} are defined as

$$F_{rand} = e^{\frac{N_{failed}}{2}} - 1, F_{goal} = e^{-\frac{N_{failed}}{2}} \tag{7}$$

The initial value of N_{failed} is zero, and N_{failed} is accumulated once when a collision is detected during expansion or the joint value exceeds the joint range. It can be seen from Eq. (7) that \mathbf{n} is toward the target point and the new point directly is expanded toward the target point when $N_{failed} = 0$. The greater the number of N_{failed} , the greater the value of F_{rand} , and the smaller the value of F_{goal} . Then the overall expansion direction tends to be more random. The new point after expansion is

$$q_{new} = q_{near} + stepsize \frac{\mathbf{n}}{\|\mathbf{n}\|} \tag{8}$$

In Eq. (8), *stepsize* represents the step size of expansion, and its value is calculated by the adaptive step size method.

(2) Adaptive step size.

According to the current expansion direction and joint values, the adaptive step size of the joint space is obtained to ensure that the maximum movement distance of each joint is less than the minimum obstacle width. The Jacobian matrix reflects the influence of small changes in joint angles on the speed of the end effector in the form of

$$v = [w_x, w_y, w_z, v_x, v_y, v_z]^T = J(q)\dot{q} \tag{9}$$

In Eq. (9), $J(q)$ represents the geometric Jacobian matrix, and v represents the spatial velocity composed of angular velocity w and linear velocity v . Set $J(q) = [J_w(q), J_v(q)]^T$, then the linear velocity of the i – th joint can be expressed as

$$\dot{p}_i = [v_{xi}, v_{yi}, v_{zi}]^T = J_{vi}(\dot{q}_i), i = 1, 2, \dots, 6 \tag{10}$$

In Eq. (10), \dot{p}_i represents the linear velocity of the i th joint, $\dot{q}_i = [\dot{\theta}_1, \dots, \dot{\theta}_i]^T \in R^{i \times 1}$, and $J_{vi}(q) \in R^{3 \times i}$ represents the linear velocity Jacobian matrix that regards i th joint as terminal joint, which is related to the rotation angle of the first i joint.

After obtaining the \mathbf{n} by Eq. (6), the maximum step length in this direction is found: set $n_j = \max(|n_1, \dots, n_i|)$ where j is the maximum subscript. Assume that each joint reaches a new point at the same time, that is, $\dot{\theta}_1 : \dot{\theta}_2 : \dots : \dot{\theta}_i = n_1 : n_2 : \dots : n_i$, then $\dot{q}_i = [\frac{n_1}{n_j}, \frac{n_2}{n_j}, \dots, \frac{n_i}{n_j}]^T \dot{\theta}_j$ is substituted into Eq. (10). Multiply both sides by Δt to get that

$$\Delta p_i = J_{vi}(q_i) \left[\frac{n_1}{n_j}, \frac{n_2}{n_j}, \dots, \frac{n_i}{n_j} \right]^T \Delta \theta_j \tag{11}$$

In order to simplify the expression, let $\Delta p_i = A_i \Delta \theta_j$. It can be seen that A_i is related to the expansion direction \mathbf{n} and the rotation angle of the first i joint. Use the compatibility of the norm $\|Ax\| \leq \|A\| \|x\|$ to get that

$$\|\Delta p_i\| \leq \|A_i\| \|\Delta \theta_j\| \tag{12}$$

Since $\|\Delta p_i\|_2 = \sqrt{\Delta p_{ix}^2 + \Delta p_{iy}^2 + \Delta p_{iz}^2}$ represents the distance that i th linkage moves in Cartesian space, the 2 norm is selected for the Eq. (12). Suppose the width of the smallest obstacle in the working space is ∇ , that is, the maximum value on the right side of Eq. (12) is ∇ . We can get that

$$\|\Delta p_{i,max}\|_2 \leq \max \|A_i\|_2 \|\Delta \theta_j\|_2 = \nabla \tag{13}$$

$$\Delta \theta_j = \|\Delta \theta_j\|_2 = \frac{\nabla}{\max \|A_i\|_2} \tag{14}$$

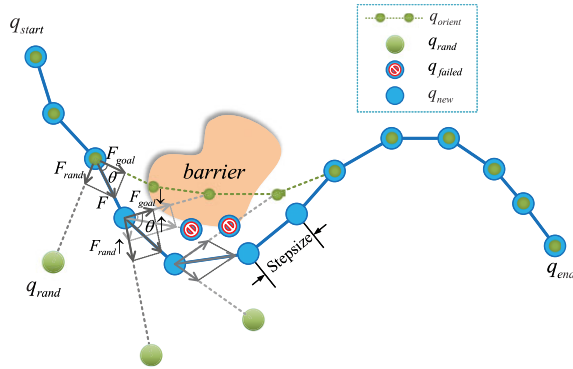


Figure 9. Schematic diagram of path point expansion.

From Eq. (14), the maximum expansion step in the n direction is calculated as

$$Stepsize = \frac{\Delta\theta_j}{n_j} \tag{15}$$

The distance of movement of the linkage of the robot is less than the width of the minimum obstacle under the step size.

3.2.3. Expansion of path points

The imitation path obtained in section 3.1 cannot guarantee that each joint of robot does not collide with the surrounding mechanism in the narrow space. Therefore, the path point expansion method is designed based on the advantage of fast expansion without collision in IRRT. Regarding the spatial guidance points of imitation path as the guidance for IRRT, the path shape of the yarn storage is obtained by expanding along the guiding path when there is no collision. The expansion angle is adjusted continuously to ensure that each linkage of the robot does not collide with surrounding mechanisms when there is collision. The principle of path points expansion is shown in Fig. 9.

(1) After inserting m path points uniformly between the imitation path points, the initial pose and end pose are interpolated and the inverse kinematics of the robot is solved. The obtained joint data $Orient\{i\} = [q_1, q_2, \dots, q_6]$ are used as the spatial guidance points for the expansion point where $i = 1, 2, \dots, m$. The spatial guidance point is the instantaneous target point of IRRT, which leads IRRT to expand along the imitation path.

(2) Random sampling is conducted within the angle range of the robot to obtain sampling point q_{rand} and a random number is generated between 0 and 1. If the random number is less than 0.9, the index $i_{closest}$ of the guidance point closest to the end of the expansion tree is found and $q_{orient} = Orient\{i_{closest} + 1\}$ is regarded as the target point q_{goal} for this expansion. The node q_{near} closest to q_{goal} is defined as the parent node; otherwise, the node q_{near} closest to q_{rand} is defined as the parent node. The index $i_{closest}$ of the guidance point closest to the q_{near} is calculated, and $q_{orient} = Orient\{i_{closest} + 1\}$ is used as the target point q_{goal} of this expansion.

(3) IRRT algorithm is used to calculate the expansion direction and step size, and the new node q_{new} is calculated.

(4) Collision detection is carried out at the angle of q_{new} . If there is a collision or joint is out of range, $N_{failed} = N_{failed} + 1$, and return step (2). Otherwise, q_{new} is expanded on the parent node q_{near} (to speed up the convergence speed, this paper defines that when the $N_{failed} > 50$ for this parent node, it will not generate a new node).

(5) The distance between q_{new} and q_{end} is calculated. If it is less than the threshold, the path expansion is successful. Otherwise, steps (2)–(5) will be repeated until the maximum number of iterations is reached.

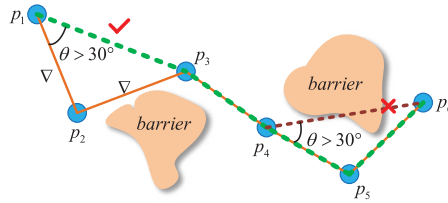


Figure 10. Schematic diagram of smooth path.

Sharp path points reduce the stability of the robot movement and a smooth path is required. The principle of smooth path is shown in Fig. 10. Suppose that p_{i+1} is a sharp point when $\angle p_{i+1}p_i p_{i+2} > 30^\circ$ and p_{i+1} is deleted if there are no obstacles between p_i and p_{i+2} . $p_1 p_2 p_3 p_4 p_5 p_6$ is an unsmoothed path. $\angle p_2 p_1 p_3 > 30^\circ$, p_2 is a sharp point. Since there is no obstacle between p_1 and p_3 , p_2 is deleted; $\angle p_5 p_4 p_6 > 30^\circ$ and p_5 is a sharp point. Since there is no obstacle between p_4 and p_6 , p_5 is reserved. The smoothed path is $p_1 p_3 p_4 p_5 p_6$. Based on this, a smooth path method is designed.

(1) The distance between point p_i and point p_{i+1} is record as $d_{i,i+1}$. The initial value of i is 1, and $d_{i,i+1}, d_{i,i+2}, d_{i+1,i+2}$ is calculated.

(2) Determine whether p_{i+1} is a sharp point. If $d_{i,i+2} + \delta < d_{i+1,i+2} + d_{i,i+1}$, p_{i+1} is a sharp point, and m points are inserted uniformly between p_i and p_{i+2} for collision detection, and then enter step (3). Otherwise, p_{i+1} is not a sharp point, and let $i = i + 1$, steps (1) and (2) are repeated. Among them, $\delta = 2(\nabla - \nabla \cos 30^\circ)$, where ∇ is the width of the minimum obstacle, and $m = \text{ceil}\left(\frac{d_{i,i+2}}{\nabla}\right)$ where ceil is a round-up function.

(3) If there is no collision, delete the point $i + 1$. Otherwise, let $i = i + 1$ and repeat the process of (1) (2) (3).

4. Experiments and result analysis

4.1. LfD phase evaluation

To verify the effectiveness of the imitation paths obtained during the LfD phase, 12 paths with different starting and ending points were obtained by teaching the yarn storage path that mechanism 2 and the yarn winding mechanism 5 at different positions. Eight paths are randomly selected as the training set, and 4 paths are selected as the testing set. The number of hidden nodes is set to 1000, and the activation function is set to *sigmoid* function to train the network models in x , y , and z directions, respectively. After that, different starting and ending points were input to the trained model. The weaving system has no guide rail in the x direction, that is, the starting and ending points are fixed in the x direction. This means that f_x has no independent variable, and the imitation path is the same as t_x of teaching path; the path always starts at the vertical center line of yarn fixing unit 1, fixing the starting point in the y direction. Therefore, Y_{end} is the independent variable for f_y ; The z direction has variable starting and ending points, with Z_{start} and Z_{end} as the independent variables for f_z . Fig 11(a) shows the imitation results of the trained network model in y direction with different inputs of Y_{end} , which is observed in the xoy plane. Fig. 11(b) shows the imitation results of the trained network model in z direction with different inputs of Z_{start} and Z_{end} , which is observed in the xoz plane. Fig. 12(a) and 12(b) show the three-dimensional imitation paths of the yarn storage process of the yarn storage mechanism 2 and the yarn winding mechanism 5, respectively. From Figs. 11 and 12, it can be seen that the shape of the imitation path is consistent with the shape of the teaching path. Even if the starting and ending points that do not appear in the input teaching data are input, similar imitation paths can be obtained, indicating the effectiveness of the imitation path.

To demonstrate the reliability of the imitation path, the same starting and ending points as the test set are input into the network model to obtain the corresponding imitation path. The similarity between the

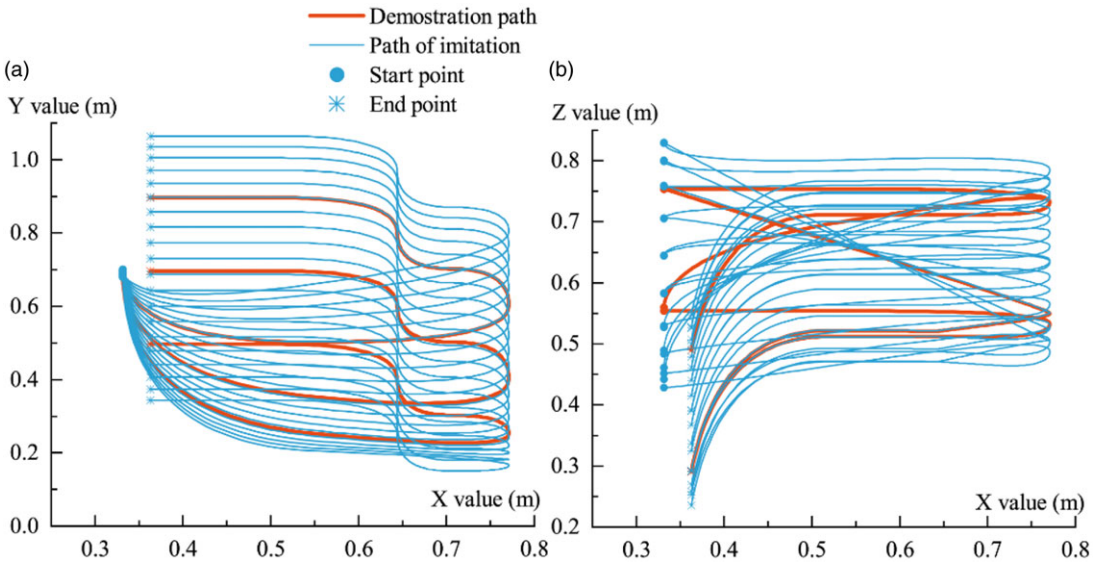


Figure 11. Result of learning from demonstration. (a) Imitated result of y-direction; (b) imitated result of z-direction.

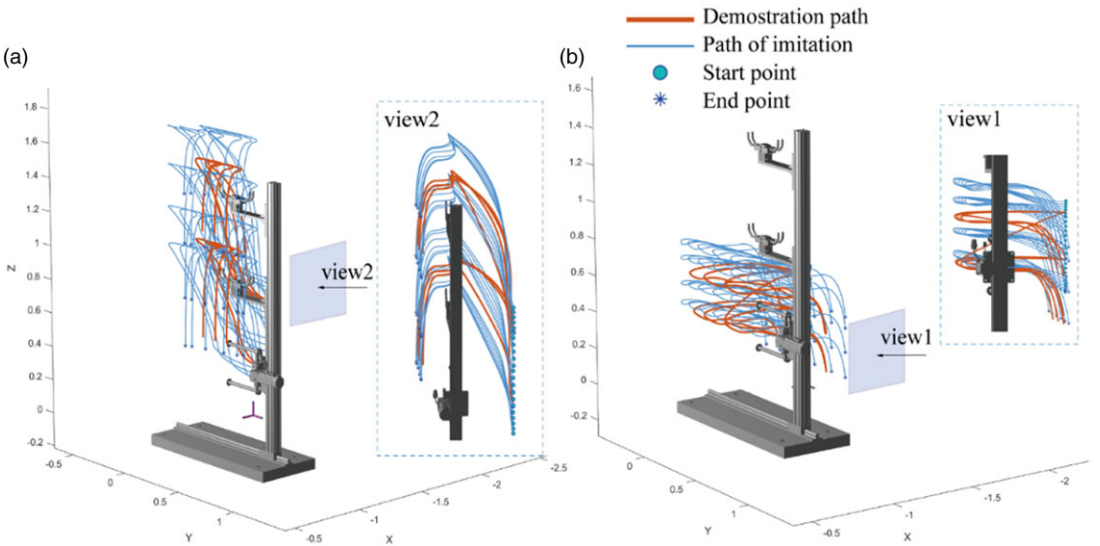


Figure 12. The path of imitation in 3D space. (a) Paths of imitation for yarn storage mechanism 2; (b) path of imitation for yarn winding mechanism 5.

imitation path and the test set path is quantitatively analyzed of using the sweep error area (SEA) defined in ref. [40], where SEA is the area enclosed by the imitation path and the test set path. The smaller the area, the closer the imitation path and the test set path are. The SEA function is expressed as

$$E = \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^{m-1} A(p(i), p(i+1), p_{demo}(i), p_{demo}(i+1)) \tag{16}$$

In Eq. (16), $A(p(i), p(i+1), p_{demo}(i), p_{demo}(i+1))$ represents the area of a quadrilateral surrounded by four points $p(i), p(i+1), p_{demo}(i)$ and $p_{demo}(i+1)$.

Table II. Performance of imitation path.

	Time (sec)	SEA (cm ²)
Average value	0.075	2.617
Range value	0.063–0.297	2.518–2.681

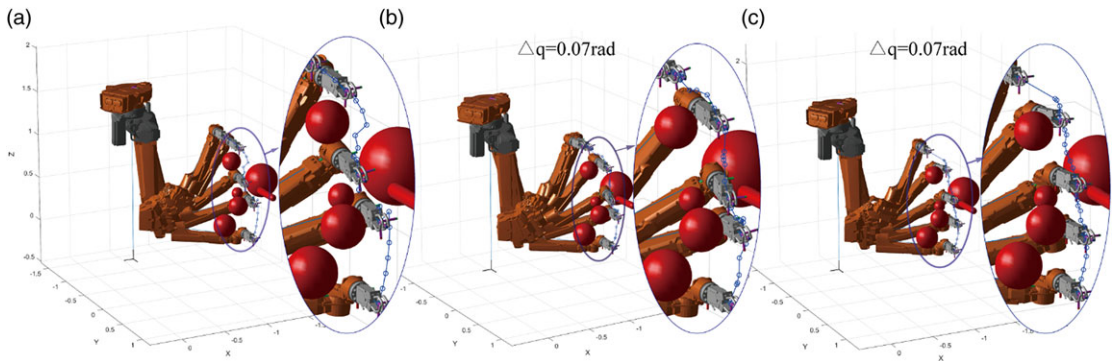


Figure 13. Comparison of planning results of three algorithms. (a) IRRT; (b) target gravity RRT; (c) GO-RRT. The blue line represents the planned path, and the dots represent the locations of the nodes of the random tree mapped in cartesian space.

The performance of imitation path is shown in Table II. It can be seen from Table II that the corresponding imitation path is obtained in a relatively short time, and the mean value of SEA is 2.617 cm². In other words, the area enclosed by the imitation path and the path of test set is small, indicating that the imitation path is close to the test set path so the simulated path has reliability.

4.2. IRRT algorithm evaluation

To verify the effectiveness of IRRT for obstacle avoidance, the performance of IRRT, Target-Gravity-RRT [23] and GO-RRT [41] were compared in the same starting point, endpoint, and obstacle environment. The step size of Target-Gravity-RRT and GO-RRT are set to 0.07 rad and the minimum obstacle width is equal to the diameter of the sphere 0.07m. It is defined that the number of iterations exceeds 500 or the number of expansion failures exceeds 100 as a planning failure. Fig. 13 shows the path planning results of the three algorithms. The nodes of Target-Gravity-RRT are unevenly distributed, and the distance between nodes near the endpoint of sharply increases in GO-RRT. Compared to Target-Gravity-RRT and GO-RRT, the adaptive step size feature of IRRT makes the distribution of path points more uniform.

To further demonstrate the advantages of adaptive step size, the displacement of each linkage of robot is compared with adaptive step size and fixed step size of 0.07 rad. It can be seen from Fig. 14 that the displacement of the link changes sharply when expansion with a fixed step size so that the displacement of the linkage-6 and the end effector sometimes exceeds ∇ ; the corresponding step size is solved based on the current joint value and expansion direction when expanding with adaptive step size, which makes the displacement of each link of the robot does not exceed ∇ and ensures the uniformity of path points.

The experiment is repeated 100 times for the three algorithms, and the performance of algorithms is shown in Table III. Target-Gravity-RRT and GO RRT have both 0.06 rad and 0.07 rad step size Δq . The maximum link displacement $\|\Delta p_{\max}\|_2$ exceeds the minimum obstacle width ∇ under the two step size. Although $\|\Delta p_{\max}\|_2$ of 0.06 rad step size is smaller than that of 0.07 rad step size, the number of iterations and planning time of 0.06 rad step size significantly increase. Therefore, Target-Gravity-RRT and GO-RRT need to repeatedly adjust the step size based on the obstacle width to obtain reasonable planning

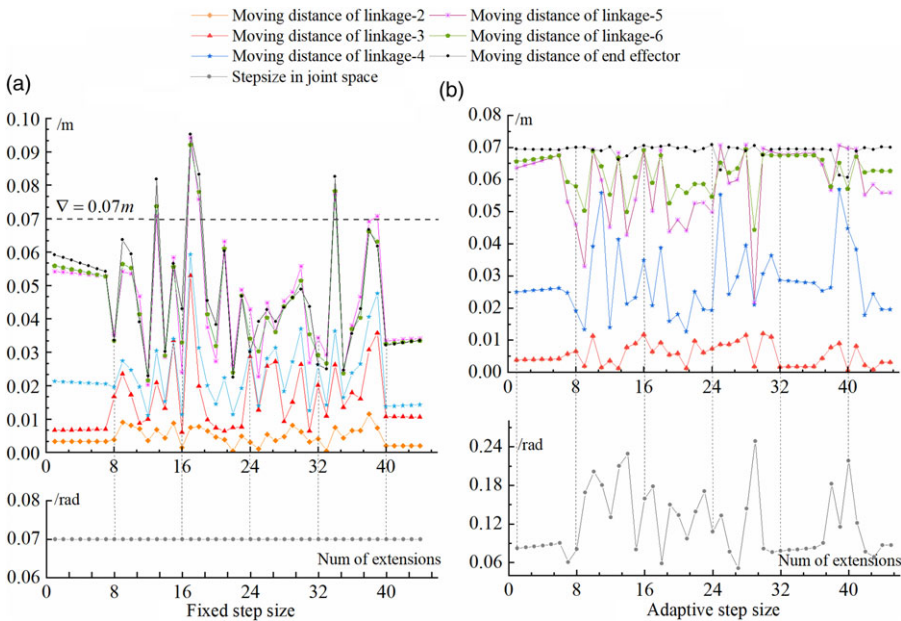


Figure 14. The comparison between fix step size and adaptive step size. (a) Fixed step size; (b) adaptive step size.

results. The adaptive step size of IRRT ensures that the maximum linkage displacement $\|\Delta p_{\max}\|_2$ does not exceed the minimum obstacle width ∇ with fewer iterations and shorter planning time. Especially in multiobstacle environments, a relationship between the expansion direction and the number of collisions is constructed in IRRT, forming an adaptive direction and improving the avoidance ability for expansion direction toward obstacles. Compared to Target-Gravity-RRT and GO-RRT algorithms, IRRT has a higher planning success rate.

4.3. Expansion phase evaluation

Guided by the spatial guide points of imitation path obtained in the LfD phase, the expansion phase of ELM-IRRT utilizes the expansion ability of IRRT to generate paths that match the path shape and are collision-free in the joint space. To verify the validity of the ELM-IRRT method in the expansion phase, the method is applied to the planar map shown in Fig. 15. The map has complex terrain containing many obstacles, large obstacles, and narrow gaps, and even some guidance points are located in obstacles, which brings challenges to path planning. Fig. 15(a) and Fig. 15(b) show the path planning results under two kind of guidance ways, respectively. The expansion direction of the tree nodes moves forward along the guide point. When large obstacles are encountered, nodes can gradually bypass the obstacles based on the expansion ability of the proposed method. At the same time, in the case that the guide point does not correctly cross the narrow gap, the proposed method can still find the correct exit through the narrow gap, avoid getting trapped in the local area, and finally successfully reach the end point. Therefore, the proposed method is validity.

To further verify the validity in high-dimensional space planning, the ELM-IRRT method is applied to a 7-DOF planar linkage robot with redundant degrees of freedom, where each link is 2.5 units long. The tree nodes expressed as the rotation angles of each joint are expanded in high-dimensional space. As shown in Fig. 16(a), six sets of joint configurations are given as spatial guidance points to guide the robot to move in an environment with two circular obstacles, and the path planning results of the proposed method are shown in Fig. 16(b). In the case of collision between the linkage position configured by

Table III. The comparison of algorithm performance.

Algorithms	Δq (rad)	∇ (m)	$\ \Delta p_{\max}\ _2$ (m)	Number of iterations (time)	Planning time (sec)	Success rate (%)
Target gravity RRT	0.060	0.070	0.083 ± 0.012	51.112 ± 9.113	2.311 ± 1.342	91.125
Target gravity RRT	0.070	0.070	0.097 ± 0.011	44.634 ± 8.675	2.101 ± 1.325	93.348
GO-RRT	0.060	0.070	0.074 ± 0.003	51.824 ± 12.836	2.151 ± 0.968	82.673
GO-RRT	0.070	0.070	0.082 ± 0.008	47.422 ± 9.469	1.892 ± 0.630	80.101
IRRT	–	0.070	0.070 ± 0.001	44.801 ± 12.220	1.967 ± 1.011	99.253

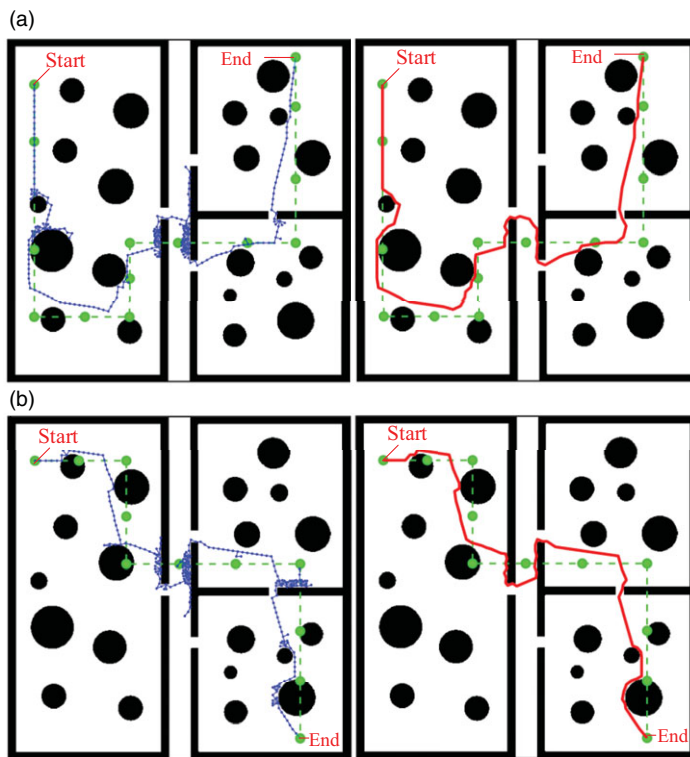


Figure 15. Path planning results in two-dimensional space under two kinds of guidance way. Green points represent guide points, green dotted lines represent the guidance way formed by guide points, blue points represent generated tree nodes, and red lines represent the final planned path.

the guidance point and the obstacle, each linkage of the robot still bypassed the large obstacle, and the moving distance of each linkage did not exceed the size of the minimum obstacle to reached the target position. Therefore, the proposed method can obtain the joint data of the robot that conforms to the desired action, which is valid in the path planning of the high degree of freedom robot.

4.4. An application example of path planning for a weaving robot

To test the ELM-IRRT path planning method, the weaving system is modeled in SolidWorks, teaching data is collected in Robot Studio, and the path planning strategy is programmed in MATLAB. The simulation test was run on a computer with a 2.3 GHz main frequency and 16 GB of memory.

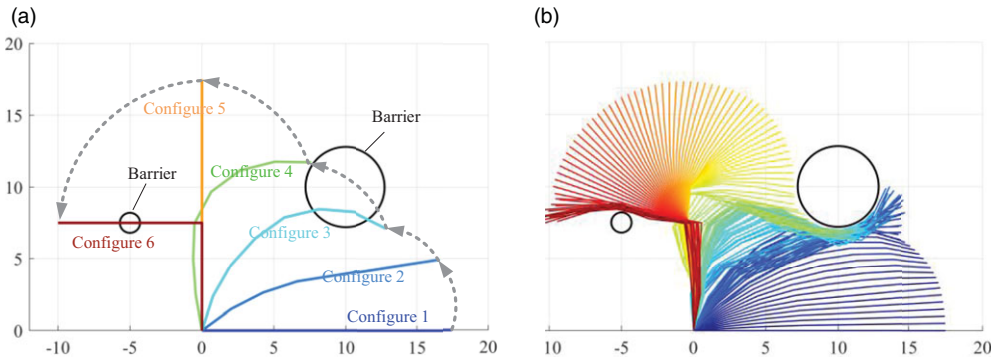


Figure 16. Path planning results of 7-DOF planar linkage robot. The black rings represent obstacles, and lines with different color represent robot with different joint configure.

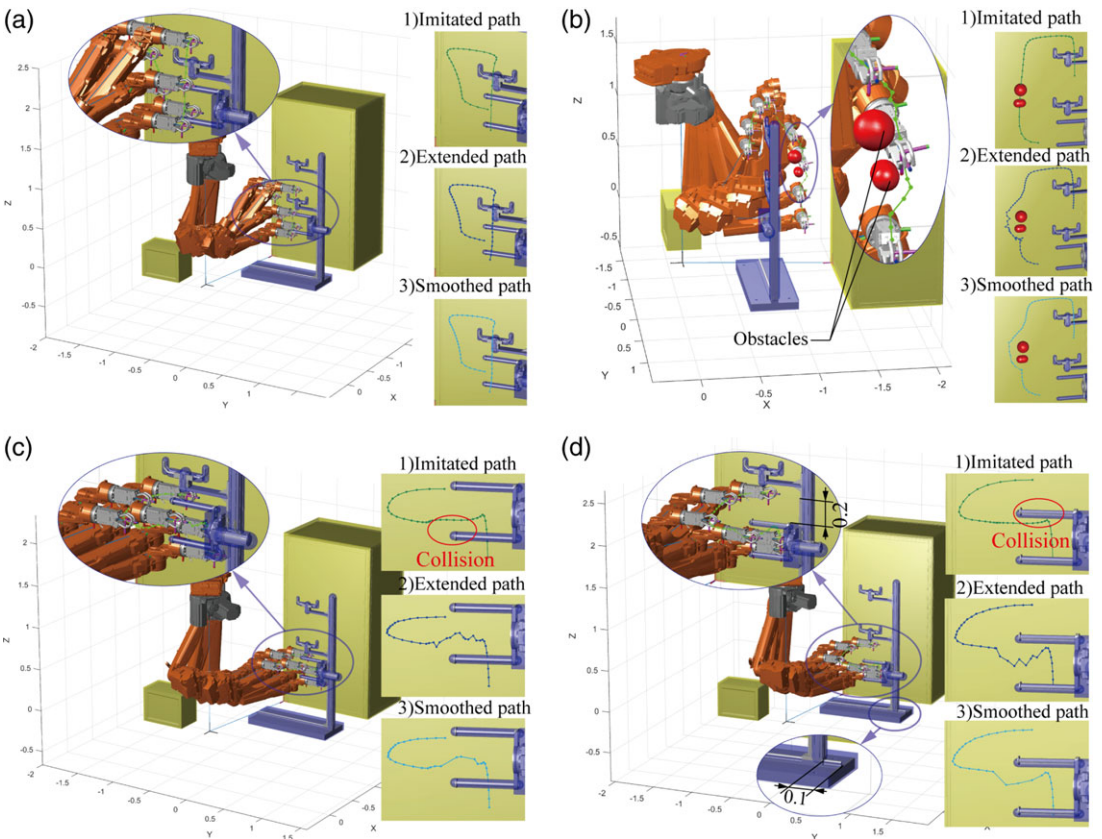


Figure 17. Path planning results under different scenes. (a) Scene A; (b) Scene B; (c) Scene C; (d) Scene D.

The ELM-IRRT method was validated for various yarn storage processes, and the experimental results are shown in Fig. 17.

Fig. 17(a) shows the scene where the robot stores the yarn in the yarn storage mechanism 2. At this time, the LfD phase obtains an imitation path that does not collide with the surrounding mechanism. Since the path point does not collide with the surrounding mechanism, the path point expands directly to the target point in the expansion stage. This results in the imitation path being almost identical to

Table IV. Efficiency of path planning in different scenes.

Scene	ELM-IRRT				Manual teaching		
	Time of imitate path (sec)	Time of expand path (sec)	Time of smooth path (sec)	Total time (sec)	Label points (min)	Test run (min)	Total time (min)
A	0.249 ± 0.012	1.216 ± 0.253	0.002 ± 0.001	1.467 ± 0.266			
B	0.250 ± 0.017	3.893 ± 0.950	0.097 ± 0.039	4.240 ± 1.006			
C	0.249 ± 0.026	1.753 ± 0.296	0.052 ± 0.026	2.054 ± 0.348	5 ± 2	2 ± 1	7 ± 3
D	0.247 ± 0.011	1.796 ± 0.341	0.062 ± 0.012	2.105 ± 0.364			

the expansion path. In order to verify the obstacle avoidance ability in a complex obstacle environment, two obstacles are artificially added to the guidance point. The path planned by ELM-IRRT is shown in Fig. 17(b). Since ELM-IRRT has the advantage of adaptive direction, the relationship between the extended direction and the number of collisions is constructed, avoiding the collision between the robot connecting rod and complex obstacles.

Fig. 17(c) shows the scene where the robot stores the yarn into the winding mechanism 5. At this time, the LfD phase obtains an imitation path of collision with the surrounding mechanism. The collision-free path obtained in the expansion stage has some sharp points, and the path that is smooth and does not collide with the surrounding mechanism is obtained. Fig. 17(d) illustrates the yarn winding mechanism 5 moving 0.2 m along the vertical guide rail 6 and 0.1 m along the horizontal guide rail 7. Despite the movement of the mechanism position, the path planned by the ELM-IRRT effectively meets the operation requirements of yarn storage while avoiding collision with the barriers.

The experiment is repeated 30 times for the above yarn storage process, and the planning efficiency is shown in Table IV. The total time of path planning using ELM-IRRT method is within 2s–5s, while the time of traditional manual teaching path needs to be about 10 min. The path planning efficiency of using ELM-IRRT method for different yarn storage processes is much higher than that of manual teaching, further indicating that ELM-IRRT method has practical advantages on path planning of large weaving systems.

5. Conclusion

Aiming at the path planning problem for robots in preform weaving, an ELM-IRRT path planning method is proposed. The main conclusions are as follows:

1. During the LfD phase of ELM-IRRT, ELM is used to autonomously learn the mapping relationship between the starting point, ending point, and path points. This process results in obtaining an imitation path with yarn storage action similar with the teaching path under different starting and ending points. The spatial guidance points of the imitation path are extracted to characterize the shape of the yarn storage process.
2. During the expansion phase of ELM-IRRT, the expansion direction of the tree node is inclined to the spatial guidance point of the imitation path, and each linkage of robot are guaranteed not to collide with obstacles in the moving process. Consequently, collision-free path with ideal shape was obtained in the joint space.
3. Based on the LfD and expansion phase, the path planned by the ELM-IRRT method not only meets the action requirements of yarn storage but also avoids the collision between the robot and the surrounding mechanisms. The proposed method can generate an ideal path within 2s–5s, avoiding the tedious manual teaching path, which is the practical and the feasible for path planning of the preform weaving robot.

Author contributions. All authors contributed to the study conception and design; Zhuo Meng conceived and designed the study; Material preparation and analysis were performed by Shuo Li; the first draft of the manuscript was written by Shuo Li and reviewed by Zhuo Meng; and the experiment data were collected by Yujing Zhang. The resources and equipment were provided by Yize Sun; all authors read and approved the final manuscript.

Financial support. This work is supported by Key R&D Program of Jiangsu Province (BE2023070).

Competing interests. The authors have no relevant financial or nonfinancial interests to disclose.

References

- [1] K. Bilisik, "Multiaxis three-dimensional weaving for composites: A review," *Text Res J* **82**(7), 725–743 (2012).
- [2] S. Li, Z. Shan, D. Du, L. Zhan, Z. Li and Y. Liu, "Automatic weaving method for three-dimensional composite preforms using vision system," *Text Res J* **91**(15-16), 1876–1888 (2021).
- [3] G. Xu, Z. Meng, S. Li and Y. Sun, "Collision-free trajectory planning for multi-robot simultaneous motion in preforms weaving," *Robotica* **40**(12), 4218–4237 (2022).
- [4] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar and A. A. Jamal, "Robotic Indoor Path Planning Using Dijkstra's Algorithm With Multi-Layer Dictionaries," *In: 2015 2nd International Conference on Information Science and Security (ICISS)*, Seoul, South Korea (IEEE, 2015) pp. 1–4.
- [5] F. Mirahadi and B. Y. McCabe, "EvacuSafe: A real-time model for building evacuation based on Dijkstra's algorithm," *J. Build. Eng.* **34**, 101687 (2021). doi: [10.1016/j.jobbe.2020.101687](https://doi.org/10.1016/j.jobbe.2020.101687).
- [6] C. W. Warren, "Fast Path Planning Using Modified A* Method," *In: [1993] Proceedings IEEE International Conference on Robotics and Automation*, Atlanta, GA, USA (IEEE, 1993) pp. 662–667.
- [7] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field D* algorithm," *J. Field Robot.* **23**(2), 79–101 (2006).
- [8] W. Chen, X. Wu and Y. Lu, "An improved path planning method based on artificial potential field for a mobile robot," *Cyber. Info. Technol.* **15**(2), 181–191 (2015).
- [9] W. Wang, M. Zhu, X. Wang, S. He, J. He and Z. Xu, "An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators," *Int. J. Adv. Robot. Syst.* **15**(5), 1729881418799562 (2018).
- [10] S. Quinlan and O. Khatib, "Elastic Bands: Connecting Path Planning and Control," *In: [1993] Proceedings IEEE International Conference on Robotics and Automation*, Atlanta, USA (IEEE, 1993) pp. 802–807.
- [11] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen and W. Deng, "Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning," *Eng. Appl. Artif. Intel.* **121**, 105942 (2023).
- [12] S. Lin, A. Liu, J. Wang and X. Kong, "An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse," *J. Comput. Sci.* **67**(7), 101938 (2023).
- [13] Q. Luo, H. Wang, Y. Zheng and J. He, "Research on path planning of mobile robot based on improved ant colony algorithm," *Neu. Comput. Appl.* **32**(6), 1555–1566 (2020).
- [14] C. Lamini, S. Benhlima and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Proc. Comput. Sci.* **127**, 180–189 (2018).
- [15] S. Chehelgami, E. Ashtari, M. A. Basiri, M. T. Masouleh and A. Kalhor, "Safe deep learning-based global path planning using a fast collision-free path generator," *Robot. Auton. Syst.* **163**, 104384 (2023).
- [16] R. Bohlin and L. E. Kavraki, "Path Planning Using Lazy PRM," *In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Francisco, CA, USA (IEEE, 2000) pp. 521–528.
- [17] G. Sánchez and J.-C. Latombe, "On delaying collision checking in PRM planning: Application to multi-robot coordination," *Int. J. Robot. Res.* **21**(1), 5–26 (2002).
- [18] R. Wang, X. Zhang, Y. Fang and B. Li, "Virtual-goal-guided RRT for visual servoing of mobile robots with FOV constraint," *IEEE Trans. Syst. Man Cybern. Syst.* **52**(4), 2073–2083 (2022).
- [19] K. Wei and B. Ren, "A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm," *Sensors* **18**(2), 571 (2018). doi: [10.3390/s18020571](https://doi.org/10.3390/s18020571).
- [20] W. Xinyu, L. Xiaojuan, G. Yong, S. Jiadong and W. Rui, "Bidirectional potential guided RRT* for motion planning," *IEEE Access* **7**, 95046–95057 (2019). doi: [10.1109/access.2019.2928846](https://doi.org/10.1109/access.2019.2928846).
- [21] I. Noreen, A. Khan and Z. Habib, "Optimal path planning using RRT* based approaches: A survey and future directions," *Int J. Adv. Comp. Sci. Appl.* **7**(11), 97–107 (2016).
- [22] G. Ma, Y. Duan, M. Li, Z. Xie and J. Zhu, "A probability smoothing Bi-RRT path planning algorithm for indoor robot," *Future Gener. Comp. Syst.* **143**, 349–360 (2023).
- [23] X. Cao, X. Zou, C. Jia, M. Chen and Z. Zeng, "RRT-based path planning for an intelligent litchi-picking manipulator," *Comput. Electr. Agric.* **156**, 105–118 (2019). doi: [10.1016/j.compag.2018.10.031](https://doi.org/10.1016/j.compag.2018.10.031).
- [24] B. An, J. Kim and F. C. Park, "An adaptive stepsize RRT planning algorithm for open-chain robots," *IEEE Robot. Auto. Lett.* **3**(1), 312–319 (2018).

- [25] D. Zhu, X. Feng, X. Xu, Z. Yang, W. Li, S. Yan and H. Dang, “Robotic grinding of complex components: A step towards efficient and intelligent machining—challenges, solutions, and applications,” *Robot. Comput. Integr. Manufactur.* **65**, 101908 (2020).
- [26] M. Shahabi, H. Ghariblu, M. Beschi and N. Pedrocchi, “Path planning methodology for multi-layer welding of intersecting pipes considering collision avoidance,” *Robotica* **39**(6), 945–958 (2021).
- [27] K. Mülling, J. Kober, O. Kroemer and J. Peters, “Learning to select and generalize striking movements in robot table tennis,” *Intl. J. Robot. Res.* **32**(3), 263–279 (2013).
- [28] A. J. Ijspeert, J. Nakanishi and S. Schaal, “Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots,” *In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH3792)*, Washington, DC, USA (IEEE, 2002) pp. 1398–1403.
- [29] T. Davchev, K. S. Luck, M. Burke, F. Meier, S. Schaal and S. Ramamoorthy, “Residual learning from demonstration: Adapting dmps for contact-rich manipulation,” *IEEE Robot. Autom. Lett.* **7**(2), 4488–4495 (2022).
- [30] S. Calinon, F. Guenter and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Trans. Syst. Man Cybern., Part B* **37**(2), 286–298 (2007).
- [31] L. Rozo, P. Jiménez and C. Torras, “A robot learning from demonstration framework to perform force-based manipulation tasks,” *Intell. Serv. Robot.* **6**(1), 33–51 (2013).
- [32] C. Yang, C. Chen, W. He, R. Cui and Z. Li, “Robot learning system based on adaptive neural control and dynamic movement primitives,” *IEEE Trans. Neural Netw. Learn. Syst.* **30**(3), 777–787 (2019). doi: [10.1109/TNNLS.2018.2852711](https://doi.org/10.1109/TNNLS.2018.2852711).
- [33] S. Liu and H. Asada, “Teaching and learning of deburring robots using neural networks,” *In: [1993] Proceedings IEEE International Conference on Robotics and Automation*, Atlanta, USA (IEEE, 1993) pp. 339–345.
- [34] M. Kaiser and R. Dillmann, “Building Elementary Robot Skills from Human Demonstration,” *In: Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, USA (IEEE, 1996) pp. 2700–2705.
- [35] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin and C. Xu, “Fast and stable learning of dynamical systems based on extreme learning machine,” *IEEE Trans. Syst. Man Cybern.: Syst* **49**(6), 1175–1185 (2017).
- [36] S. Ding, H. Zhao, Y. Zhang, X. Xu and R. Nie, “Extreme learning machine: Algorithm, theory and applications,” *Artif. Intell. Rev.* **44**(1), 103–115 (2013). doi: [10.1007/s10462-013-9405-z](https://doi.org/10.1007/s10462-013-9405-z).
- [37] N. García, J. Rosell and R. Suárez, “Motion planning by demonstration with human-likeness evaluation for dual-arm robots,” *IEEE Trans. Syst. Man Cybern.: Syst* **49**(11), 2298–2307 (2019).
- [38] M. Kong and G. Yu, “Collision Detection Algorithm for Dual-Robot System,” *In: IEEE International Conference on Mechatronics and Automation*, Tianjin, China (IEEE, 2014) pp. 2083–2088.
- [39] G. v. d. Bergen, “A fast and robust GJK implementation for collision detection of convex objects,” *J. Graph. Tool* **4**(2), 7–25 (1999).
- [40] S. M. Khansari-Zadeh and A. Billard, “Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions,” *Robot. Auton. Syst.* **62**(6), 752–765 (2014).
- [41] G. Kang, Y. B. Kim, Y. H. Lee, H. S. Oh, W. S. You and H. R. Choi, “Sampling-based motion planning of manipulator with goal-oriented sampling,” *Intell. Serv. Robot.* **12**(3), 265–273 (2019).