Contents lists available at ScienceDirect

Journal of King Saud University – Science

journal homepage: www.sciencedirect.com

Original article

Obstacle avoidance path scheme of snake robot based on bidirectional fast expanding random tree algorithm

Kenglung Hsu

Pratt School of Engineering, Duke University, Durham, NC 27708, United States

ARTICLE INFO

Article history:
Received 30 October 2021
Revised 13 January 2022
Accepted 12 March 2022
Available online 16 March 2022

Keywords:
Serpentine robot
Obstacle avoidance path planning
Gravitational component
Two-way rapid expansion of random tree

ABSTRACT

With the continuous innovation and development of related technologies in the field of robotics, various robots have emerged in large numbers, the application scenarios of robots are becoming more and more complex, and the requirements for robot technology are also getting higher and higher. As an indispensable part of robotics research, path planning technology has very important research and application value. As a new movable snake-shaped robot, the snake-shaped robot has always been a research hotspot in the field of snake-shaped robots due to its multi-gait movement ability and strong environmental adaptability. Among the various motion forms of the snake-shaped robot, the winding motion has the highest efficiency, and the snake-shaped robot in this motion form can also assist itself in its movement by contact with obstacles. However, there are few researches on the obstacle-assisted motion of snake-like robots, and the algorithm for solving collision dynamics is relatively simple. Therefore, it is necessary to conduct in-depth research on obstacle-assisted motion snake-like robots. The trajectory planning algorithm of the multi-NAO snake robot is studied. First, it analyzes the basic principles and implementation steps of the RRT algorithm and evaluates its advantages and disadvantages. On this basis, an improved fast-expanding random algorithm is proposed for the shortcomings of the RRT algorithm. Combining the advantages of the global search of the RRT algorithm, it is necessary to introduce an appropriate local search algorithm. Under the premise of ensuring that the path can be generated, a certain algorithm is added to improve the smoothness of the path, in order to reduce the turning time of the snake-like robot when it walks, and improve the search efficiency. Aiming at the randomness of RRT when generating random tree nodes, other algorithms are purposefully introduced to enable it to grow toward the target point. Path planning simulations verify that the improved two-way rapid expansion random tree algorithm has significantly improved search speed and search efficiency compared with Basic-RRT and Bi-RRT algorithms, with shorter average planning time and higher success rate, the generated path is smoother. Improved algorithm can not only avoid static global obstacles in a dynamic environment, but also efficiently avoid sudden dynamic obstacles. The real-time dynamic obstacle avoidance experiments based on the FANUC snake-shaped robot guiding the anthropomorphic obstacles to move and rotate in translation and the snake-shaped robot dynamically avoiding real human arms verify that the proposed algorithm can dynamically avoid regular and irregular moving obstacles online in time. When the number of obstacles is 3, the number of successful plans is 49, and the planning time is 55 ms; when the number of obstacles is 12, the number of successful plans is 40, and the planning time is 88.9 ms. Obviously, the increase in the number of obstacles leads to a decrease in the number of successful planning and an increase in planning time.

1. Introduction

With the increasingly obvious service trend of mobile snake-shaped robots, it has gradually penetrated into every corner of social production and life (Albini et al., 2021). Mobile snake-shaped robots can use their own sensing equipment to detect the surrounding environment and avoid obstacles in real time, in a mode similar to human intelligence. Compared with other types

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

E-mail address: kenglunghsu@163.com

https://doi.org/10.1016/j.jksus.2022.101975
1018-3647/© 2022 Published by Elsevier B.V. on behalf of King Saud University.

of snake-shaped robots, mobile snake-shaped robots have unique mobility characteristics that can bring more and more widely applicable scenarios, so they have extremely high military and civilian market value (Liu et al., 2019; Naughton et al., 2021; Jia and Ma, 2021). At the same time, the research and application of mobile snake robot technology can also reflect the national industrialization development level and science and technology development level. It is an important manifestation of national scientific and technological innovation. Therefore, most countries currently have the technical level and independent innovation ability of mobile snake robot industry. The investment of the company is also increasing, and its application prospects are attracting widespread attention from all walks of life (Amanov et al., 2021; Pavlik et al., 2021).

Due to the wide variety of application scenarios of the mobile snake-like robot, and many uncertain factors such as the diversity of its task targets and the complex and changeable operating environment, the requirements for the performance of the driving trajectory planned by the algorithm itself are becoming higher and higher (Foerster and Schmid, 2019). However, due to the shortcomings of traditional path planning algorithms, such as slow convergence speed, high computational complexity, and poor applicability, it has been difficult to meet the adaptability requirements for different environments (Tang et al., 2020). Therefore, a high efficiency and strong accuracy is sought. And the motion planning algorithm that meets the real-time requirements is very urgent and necessary. At the same time, with the continuous development and maturity of the theory of snake-like robots, path planning has become an indispensable core technology for mobile snake-like robots towards the artificial intelligence stage (Li et al., 2020). At present, mobile snake-like robots can only acquire the unknown environment by attaching sensors to their body, and use sensor scanning recognition technology to model the free state space in the unknown environment. The mobile snake-like robot needs to be in the process of motion planning. To master real-time information in an unknown environment and complete real-time obstacle avoidance, the requirements for the path planning algorithm of the mobile snake robot are also greatly improved (Véras et al., 2019). In addition, as mobile snake-like robots have been widely used in service areas such as industrial, agricultural, military and medical services, the application range of path planning algorithms has gradually increased (Tanaka et al., 2021). Therefore, it is necessary to study the path planning of mobile snake-like robots in uncertain environments.

In this paper, the trajectory planning algorithm of the snake robot is designed, and the experimental verification is carried out using the SimRobot simulation software and the real snake robot. Specifically, the novelty of this paper can be summarized as follows:.

First: An improved RRT algorithm is proposed for the shortcomings of the basic RRT algorithm.

Second: A hybrid trajectory planning algorithm is proposed based on the dynamic characteristics of the competition environment. We conduct simulations with SimRobot simulation software, and verify the accuracy and efficiency of the hybrid algorithm through experiments on a solid snake-like robot.

Third: The Bi-RRT-Star dynamic obstacle avoidance path planning method based on the improved sampling function in the target direction is proposed, which is improved from three aspects: connection strategy, heuristic dense sampling and expansion of adjacent nodes.

Fourth: The effectiveness of the planning algorithm is verified through the MATLAB static simulation and ROS dynamic simulation as well as the real obstacle avoidance experiment of the snake robot in a dynamic environment.

Fifth: Based on the V-REP environment, dynamic obstacle avoidance simulation and a series of snake-like robot avoidance dynamic obstacle experiments are carried out, which verifies that the dynamic path planning method proposed in this paper can achieve online avoidance of continuous motion obstacles.

2. Related work

Relevant scholars have developed the S1 $\sim$ S7 series of snake-shaped robots (Wang et al, 2019). Among them, the S5 has a very high degree of bionics. It has a total of 8 torso units, and uses a 4-channel radio to control 64 servo motors to achieve yaw motion. It carries a battery to provide power. The S5 snake-shaped robot has many joints and a small aspect ratio, which is very similar to the aspect ratio of a real snake. The latest S7 adopts a more advanced segmented design, which can achieve lateral and longitudinal friction differences without using passive wheels. In addition, the snake-shaped robot head is equipped with a distance sensor, a bending sensor, a rotation angle sensor, a camera, etc. It can realize the functions of distance detection, motion measurement, rotation angle measurement, image acquisition and so on.

Related scholars have proposed the DRRT algorithm, namely dynamic RRT, which is suitable for dynamic environments (Nakajima et al., 2019). The core idea is to make path planning faster and more stable by retaining historical path information. The difference with ERRT is that DRRT is not a reserved path node. Instead, when the environment changes, the points and branches that become invalid due to obstacles in the current RRT are removed, and the remaining vertices and branches are still available. Search in the tree until the target point is found. However, it is possible that the tree cannot reach the target point after processing, so expand on the remaining trees until the target point is found.

Relevant scholars have proposed a comparative RRT algorithm for the dynamic environment, planning two paths per cycle: a stable path and a random path (Praserttaweelap and Kiatwanidvilai, 2020). The stable path is through pruning and re-planning the historical path to make it as close as possible to the path of the previous cycle. The random path is derived from the basic RRT method, and the path is random. By comparing the two paths, a better path is obtained. This algorithm can prevent the path from falling into a non-optimal stable state, and make it gradually approach a better path.

Randomized Path Planner (RPP) is recognized as the first randomized algorithm for motion planning (Li et al., 2021). This algorithm opens a new door for motion planning. This algorithm combines the idea of artificial potential field, by constructing an artificial potential field in the configuration space, searching the path trajectory along the negative gradient direction from the initial point to the target point. Because this type of algorithm can avoid local minima by random walk method, RPP will not fall into the local minima, search the path for the global optimum, and have a very good effect on the completeness of the probability of path planning (Hua et al., 2021).

Genetic algorithm is also one of the most used intelligent algorithms for path planning of snake-shaped robots (Liu and Guo, 2019; Agarwal and Bharti, 2018). It can achieve good planning results in the static workspace of single-snake robots and the dynamic space of multi-snake robots. Relevant scholars have proposed a knowledge-based genetic algorithm (GA), and used it in the path planning of a mobile snake-like robot (Asghar et al., 2021). They proposed a knowledge-based genetic algorithm to integrate domain knowledge into its dedicated domain. Others used genetic algorithms to complete the path planning of the snake-shaped robot in a discrete space, and the planned path achieved good results (Wang et al., 2018; Girija and Joshi, 2019). However, this method is carried out under a certain environmental model. If the obstacles in the environment change, this method is not applicable. They applied the improved genetic algorithm-genetic simulated annealing method to the path planning problem of the snake-like robot, which effectively improved the calculation speed of path planning (Malik and Ahmad, 2017).

3. Method

3.1. Serpentine robot navigation architecture

The overall framework of the snake-shaped robot for navigation is shown in Fig. 1. It can be seen from Fig. 1 that the key to snake robot navigation is path planning. The path planning of snake robot mainly includes two parts: planner and cost map. The path planning function package move_base has an interface for configuring and running the navigation of the snake-shaped robot, and can interact with it. Due to the existence of the path planning function, the snake-shaped robot only needs to obtain the target position of the navigation and some necessary sensor information. Path planning in ROS mainly relies on two planners, the global path planner and the local path planner.

3.2. Fast expanding random tree algorithm

Rapidly-Exploring Random Tree (RRT) is a data structure and a fast global search algorithm. It is proposed to solve the path planning problem of snake-shaped robots. The RRT algorithm not only has the advantages of the probabilistic graph method, but is also suitable for solving non-integrity constrained dynamics and kinematics problems, and in the case of integrity constraints, the performance is better than that of the probabilistic graph method.

Mobile snake-shaped robots often encounter such a situation: in the process of moving from one point in space to another, due to obstacles in the surrounding environment or the constraints of the snake-shaped robot's own capabilities, a reasonable method is required. The path bypasses these obstacles to reach the target point. This is the path planning problem in the continuous domain. Currently, there are many methods that can be used to solve such problems, but almost every method has limitations, such as the need for state discretization or a compromise between search efficiency and search accuracy.

The RRT algorithm uses a point in space as the root node. Usually this point is the coordinates of the snake-shaped robot's own position. The leaf nodes of the tree are continuously expanded in the surrounding space. The path composed of the nodes of the random tree from the leaf node closest to the target point to the root node is the path generated by the path planner.

In the algorithm, both the start position and the target position are mapped to nodes in the random tree, the start position is mapped to the start node, and the target position is mapped to the target node. At the beginning, the initial node is used as the root node of the random tree growth. If the Euclidean distance between the initial node and the target node is less than a set critical value ε, then there is no need to continue the random tree search and jump directly out of the search. The connection with the end point is the planned path. Otherwise, a random node is selected in the motion space, from which a new tree node is generated from the random node. If the new node generated at this time does not collide with a known obstacle, it will be inserted into the node closest to the new node in the random tree. Otherwise, you select a random node again and repeat the above process until a leaf node of the random tree is close enough to the target node, or the number of searches reaches the preset search limit. At this time, a random tree that can reach the target node has been generated. The nearest leaf node of the target node inquires its root node in turn until it reaches the root node of the random tree. These nodes are taken out to generate the planned path.

Considering that the actual snake-like robot performs many tasks, the search process of the random tree cannot be carried out indefinitely, otherwise all other tasks will be blocked and the system will be paralyzed. When the number of searches reaches the preset upper limit, the search stops. If the path is not found

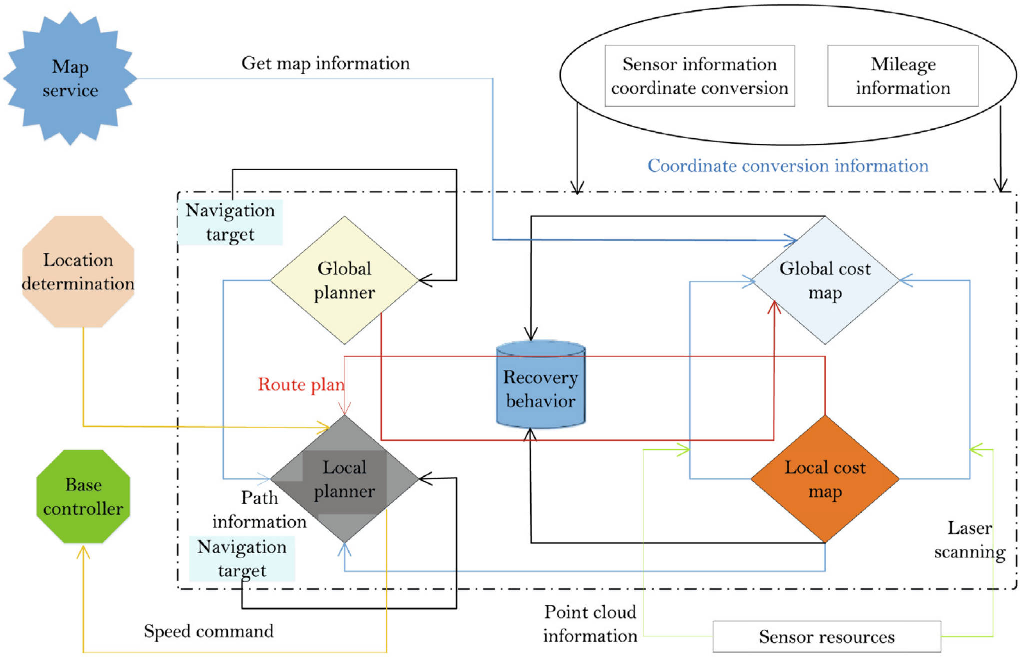

Fig. 1. Navigation frame diagram of the snake-shaped robot.

at this time, the search fails and can only be searched again the next time the planner is called.

3.3. Improved RRT algorithm

Although the basic RRT algorithm has many advantages compared with many existing path planning methods, the inherent randomness of the basic RRT algorithm also determines that this method cannot be directly applied in a static environment or in a dynamic moving obstacle environment. Path planning is in progress. Therefore, this paper uses the advantages of RRT algorithm in global search, and adds some ideas of local search methods to the basic RRT algorithm to improve the shortcomings caused by its randomness. The advantage of the RRT algorithm is that it provides many interfaces in the algorithm structure, and different implementation methods can be adopted according to the actual situation to improve the algorithm at different levels. Therefore, this article first makes different improvements in the interface implementation, and then tries to change the structure of the random tree to observe its effect. In addition, based on the realization of static environment simulation, further improvements are needed to adapt to the needs of the dynamic environment for the dynamic mobile obstacle environment.

The fast-expanding random tree algorithm tends to search the unsearched area. In this case, although the search range will become very large, this randomness will cause a lot of calculations to search for the target point or target area. In order to improve this situation, the method of selecting the target point with a certain probability is added, that is, each time a new node is selected, the end point is selected with a certain probability, so that the entire random tree will grow toward the target area.

1) Increase the gravitational component.

In order to enable the random tree to be generated toward the target point or target area, it is not necessary to perform a comprehensive search of the global environment, thereby reducing the search time and improving the real-time performance of the search. In the RRT algorithm, the idea of virtual gravitation and repulsion is introduced.

The idea of this improved algorithm is to add a target gravitational function G(n) that grows towards the target point or target area for each node n in the path. Here, node n refers to the nth expanded from the starting point xinit. xnew node. To imitate the mechanics formulas in physics, one can assume the following formulas:.

$$F(n) = R(n) + R(n-1) \cdot G(n-1) \tag{1}$$

Among them, R(n) refers to the random growth function of node n, G(n) refers to the target gravitational function, and F(n) refers to the growth guidance function from node n to the target point or target area. The gravitational potential energy function is:.

$$U_x = 0.5k_p\left(x_{near} - k_p x_{goal}\right)^2 \tag{2}$$

It can be deduced:.

$$G_x = 0.5k_p^2\left|x_{near} - x_{goal}\right| \tag{3}$$

Among them, x refers to the position vector of the snake-like robot, kp is the gravitational coefficient, xgoal refers to the target position vector of the snake-like robot, and xgoal -xnear represents the difference between the geometric distance between the target point xgoal and the node xnear. According to the above formula, the target gravity function G(n) as shown in the following formula is constructed. Among them, p is the growth step of the random tree.

$$G(n) = k_p\left(px_{near} - x_{goal}\right)/\left|x_{near} - k_p x_{goal}\right| \tag{4}$$

When adding a new leaf node to the rapidly expanding random tree, the target gravitational function calculates the force between each node and the target node or target area to guide the selection of new nodes that grow towards the target point. According to the above, the random growth function of node n in the RRT algorithm is:.

$$R(n) = k_p\| x_{near} - x_{\text{target}} \|/\left|x_{near} - px_{\text{target}}\right| \tag{5}$$

The final F(n) is:.

$$F(n) = k_p\frac{\| x_{near} - x_{\text{target}} \|}{\left|x_{near} - px_{\text{target}}\right|} + \frac{\left|x_{near} - k_p x_{goal}\right|}{\| x_{goal} - x_{near} \|} \tag{6}$$

The formula for generating new nodes after introducing the idea of gravitation is:.

$$x_{new} = x_{near} + k_p\frac{\left|x_{near} - x_{goal}\right|}{\| x_{near} - px_{goal} \|} - p\frac{\| x_{\text{target}} - x_{near} \|}{x_{\text{target}} - k_p x_{near}} \tag{7}$$

The influence of gravitational thought on the growth of random tree is mainly embodied in that the bias of each node in the random tree to the end point can be adjusted by adjusting the gravitational coefficient kp. Because the new leaf node is actually the vector sum of the vector determining the step length and the gravitational vector in the direction of the random node, that is to say, the gravitational component can completely affect the growth direction of the random tree.

2) Path smoothing.

The basic idea of the RRT algorithm to construct a path is to construct a random tree in free space. When a leaf node in the random tree is close to the end point, you find a branch composed of a sequence of nodes in the random tree as the final output.

There may be redundant nodes that can be deleted near the start node of the path. Similarly, there may be redundant nodes that can be deleted near the end node. Therefore, the path smoothing process does two things: one is to scan from the start node to the end node. Any node in the path is to detect whether the node will collide; the second is to detect the collision information from the termination node to each node in the path. After the two processes are detected, all the redundant nodes that can be deleted in the path can be determined. At this time, the redundant nodes in the path are deleted to obtain the planned path after the path smoothing process. Path smoothing can further optimize the planned path, shorten the distance of the path, reduce the number of times the snake-shaped robot turns, and shorten the time from the starting point to the end point.

3) Two-way rapid expansion of random trees.

The RRT algorithm itself is a method to solve the problem of trajectory planning. It can be improved to become another planner. For example, it can grow in both directions during the growth of a random tree, that is, let one tree start to grow at the starting point, and the root tree grows from the end point.

The two-way rapid expansion random tree mentioned here is a change to the algorithm structure, which is mainly reflected in: adding a tree, and defining the random tree searched from the starting point as random tree one. It is the second random tree, the two trees start searching paths from their respective starting points at the same time, and the way of generating nodes is the same as that of the single-root random tree.

When the random tree is first searched in the free space and the random tree is established, random tree two is also established.

Random tree one and random tree two alternately grow leaf nodes. It is close enough to another tree, that is, it detects whether the Euclidean distance between two nodes is less than a preset value. When the Euclidean distance between the two nodes is less than the preset value, connect the two nodes, that is, connect the random tree 1 and the random tree 2, then a path from the start point to the end point can be generated.

After each leaf node is generated, it is tested whether the current leaf node can be connected to another tree. If the condition is met, the two trees are connected through the leaf node, and the growth of the two random trees is terminated at the same time. When connecting two trees, you need to know which leaf node on the tree at this time detects that it can connect the random tree where it is with another random tree. This information can be obtained by the variable useFirstTree. When useFirstTree is true, it means the leaf node is on random tree 1. When useFirstTree is false, it means that the leaf node is on random tree 2.

In order to make path planning more efficient, the method of path caching is introduced. That is, a cache area is established to store the points on the path during the previous planning. When the random tree search path is re-established, the operation of selecting the target point will select the point in the path buffer with a certain probability.

Now define a probability probWayPoint to choose a point in the path cache area. probGoal makes the growth of the random tree bias towards the target point, and probWayPoint makes the growth of the random tree bias towards a planned path. If the sum of probGoal and probWayPoing is less than 1, the random tree will be random with the probability of 1-probGoal-probWayPoint Grow. The way of taking points represented by these three probabilities is shown in Fig. 2.

The discrete points in Fig. 2 represent the path points in the path cache point, the random tree connected by the solid line is the reconstructed random tree, and the node connected by the dashed line represents the process of selecting a new target node.

This operation first selects two random trees, where p is a random decimal number ranging from 0 to 1, and i is a random integer ranging from 0 to NumWayPoint −1, where NumWayPoint is the number of nodes in the path buffer quantity. Probabilistic selection of points in the buffer area is achieved by adjusting the value of the random number p. If the value of p is between probGoal and probGoal + probWayPoint, the function returns a point in the path buffer. The function returns the node indexed by i in the buffer area. When the selected random decimal p is between probGoal + probWayPoint and 1, the function returns a random point in free space. This operation is similar to selecting a node with a certain probability bias toward the end point.

4. Results and discussion

4.1. Static obstacle avoidance path planning experiment

The path planning of the snake robot is a motion planning in a high-dimensional manifold space to visually verify the feasibility, superiority and effectiveness of the proposed bidirectional fast expanding random tree algorithm. Firstly, the 3D plane path planning simulation in the static obstacle scene is carried out in the MATLAB environment. In the static obstacle avoidance simulation environment, the two-way fast expanding random tree algorithm proposed in this paper is compared with the traditional Basic-RRT and two-way Bi-RRT algorithms to verify the correctness and superiority of the proposed algorithm. This subsection assumes that the snake robot is a point-like snake robot without direction, that is, similar to a mobile snake robot platform. Then the pose of the snake robot in its configuration space is simplified as the two-dimensional plane position and orientation angle in the scene.

In the MATLAB environment, the size of the entire state space is 700×500, and the obstacle area is randomly set. Arbitrarily we specify the initial position and target position, and then plan the plane path. First, the pruning function is used to obtain a series of path points without collision, and then by inserting some necessary intermediate nodes, the sharp included angles in the path are smoothly transitioned. Fig. 3 shows the curvature change of the bidirectional fast-expanding random tree algorithm during the entire path planning process. It can be seen from the figure that the curvature of the planned path changes continuously.

In addition, for objective evaluation and performance, the proposed algorithm is better than the other two algorithms in terms of performance. At the same time, considering the unique random sampling of the RRT algorithm, the three algorithms are planned 100 times for the same simulation scenario. Finally, statistically we record their respective average search time, average number of sampling nodes, and successful planning times during the entire planning process, as shown in Fig. 4.

From the relevant data of the above simulation experiment, it can be seen that the search speed and search efficiency of the two-way rapid expansion random tree algorithm are significantly improved compared with the other two algorithms Basic-RRT and Bi-RRT. It is smoother, and the curvature of the path is continuously changing, which can better meet the requirements of the snake-shaped robot to move smoothly and without impact during the static obstacle avoidance process.

In order to actually verify the feasibility and effectiveness of the proposed static path planning algorithm, the ROS Moveit development environment is used to build a path planning obstacle avoidance experiment platform for the UR10 snake robot in a dynamic environment. The Kinect V2 RGB-D sensor is placed on a tripod and fixed at a fixed position in the environment to perceive static and dynamic obstacles in the global environment. The anthropomorphic obstacle is fixed at a certain position in the static environment. Kinect obtains the color map and depth map of obstacles in the environment, which are first converted into octree maps, and processed by the point cloud PCL algorithm to obtain the outline information of the obstacles. The process of scene information processing is shown in Fig. 5. Real-time collision detection is realized by calling the FCL flexible collision detection library. The angle change curve of each joint in the static obstacle avoidance process is shown in Fig. 6. It can be seen from Fig. 6 that the obstacle avoid-

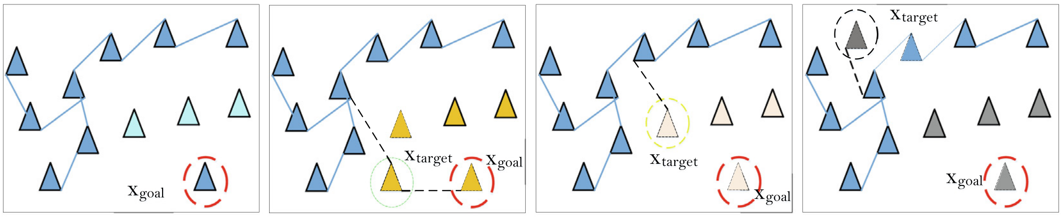

Fig. 2. Schematic diagram of growth with different probability biases.

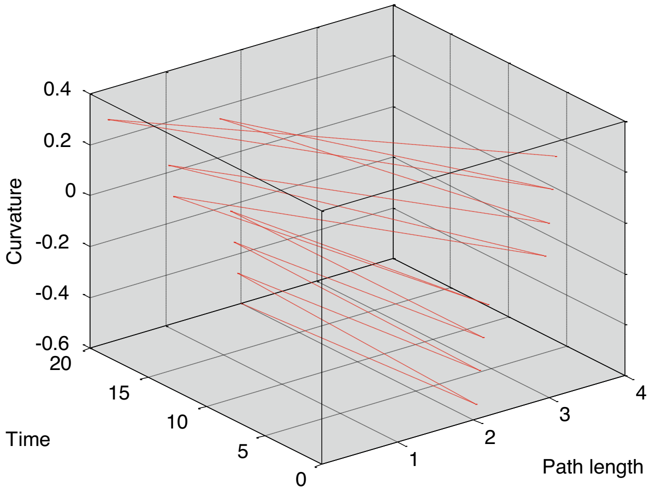

Fig. 3. Curvature change diagram of the generated path by the bidirectional fast expanding random tree algorithm.

ance time of the six joints does not exceed 8 ms, and the joint movement speed does not exceed 6 m/s, ensuring a certain stability.

4.2. Dynamic obstacle simulation based on V-REP environment

In this section, the performance indicators of the snake robot dynamic environment path planning algorithm proposed above will be tested and verified in the snake robot virtual simulation platform V-REP software. First, the geometric models of the snake robot, dynamic sphere, and human obstacles are built in V-REP. The computer for all simulation experiments is configured as CPU Intel i7 3.8 Hz octa-core, RAM 16G, GPU NVIDIA GeForce GTX 1080Ti with CUDA 10.0, all algorithm programs are written in C++ language in the form of V-REP plug-in, embedded in the virtual scene for routing planning. The Kinect V2 RGB-D sensor is used for real-time perception to obtain environmental information such as dynamic obstacles. In this simulation experiment, the performance of the dynamic path replanning algorithm Bi-RRT-Star is mainly tested in a dynamic environment with a large number of moving spherical obstacles, including the number of successful planning, planning time and trajectory cost.

1) Different moving speeds of dynamic obstacles.

There are a total of 5 spherical obstacles around the snake robot with a diameter of 0.1 m. We repeated planning experiments for 50 times, the statistical related performance indicators are shown in Table 1. From the table, it can be seen that as the speed of dynamic obstacles increases, the success rate of dynamic path planning is decreasing, and the cost of path trajectory is also following. However, the planning time remains basically unchanged. The greater the cost of the path trajectory, the less smooth the path trajectory generated by the planning.

2) Different numbers of dynamic obstacles.

There are different numbers of moving spherical obstacles around the snake robot, the diameter of which is 0.1 m, and the speed of all obstacles is set to 1 m/s. The dynamic path planning experiment of the snake-shaped robot was repeated 50 times, and the statistical related performance indicators are shown in Table 2. It can be seen from the table that with the increase in the number of surrounding dynamic obstacles, the success rate of the dynamic path planning of the snake robot is decreasing, and the planning time and the cost of the path trajectory are increasing accordingly.

4.3. Experiment of dynamic obstacle avoidance path planning based on bidirectional fast expanding random tree algorithm

In this section, we will build a real experimental platform to verify the dynamic path planning method, including three experiments based on the FANUC snake-like robot to guide the translation and rotation of anthropomorphic obstacles, and the snake-like robot to dynamically evade the human arm. In the first two experiments, the FANUC snake-shaped robot is used to guide the

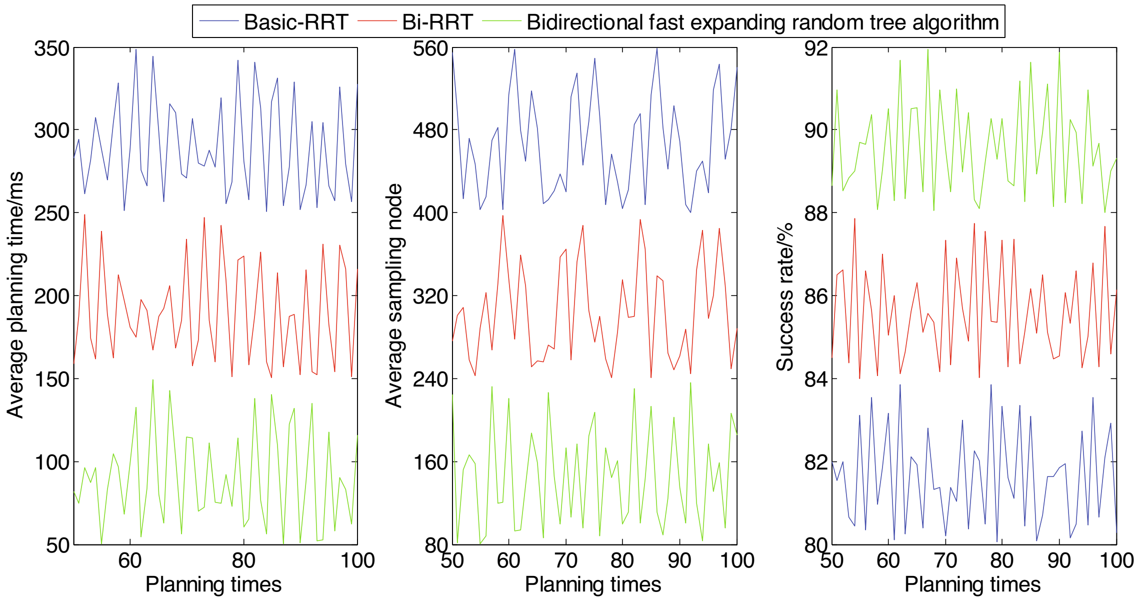

Fig. 4. Comparison of simulation results of different algorithms.

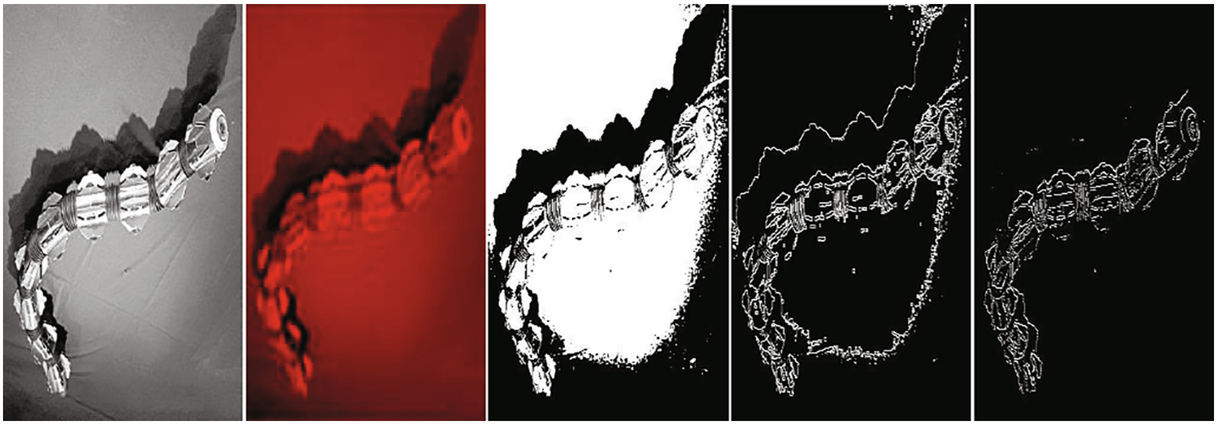

Fig. 5. Processing of scene information.

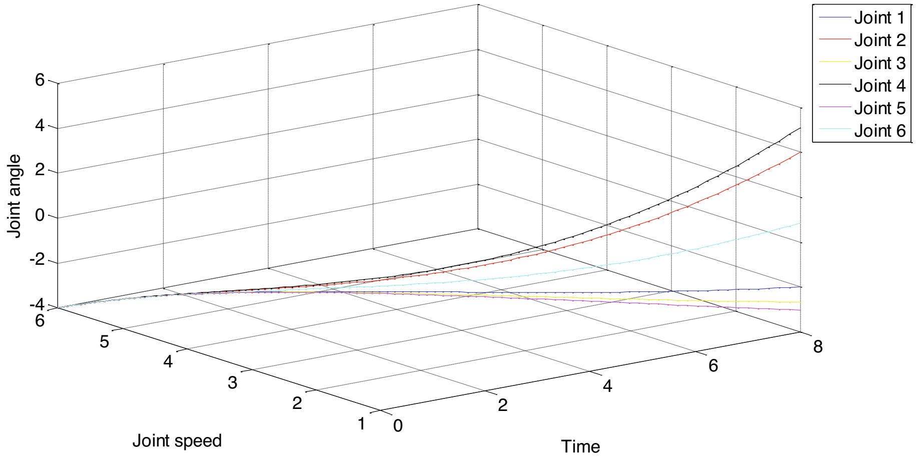

Fig. 6. The angle change curve of each joint during static obstacle avoidance.

Table 1
Results of dynamic programming under different obstacle speeds.

Obstacle speed (m/s)	Number of successful plans	Planning time (ms)	Trajectory cost
1	48	62	0.51
2	45	63	0.55
3	37	63.5	0.59
4	31	68.9	0.63

Table 2
Results of dynamic programming under different numbers of moving obstacles.

Number of obstacles	Number of successful plans	Planning time (ms)	Trajectory cost
3	49	55	0.41
6	47	68	0.53
9	44	73.5	0.57
12	40	88.9	0.64

anthropomorphic obstacles to achieve uniform translation and constant-axis rotation, while the latter snake-shaped robot evades the human arm by random motion.

1) Based on FANUC translational guidance, snake robot avoiding obstacles.

This section mainly tests the performance of the snake-shaped robot in avoiding regular translational obstacles under the guidance of the Bi-RRT-Star dynamic path planning method. Manually we teach and program the FANUC snake robot in advance, so as to realize that the anthropomorphic obstacle fixed by the end effector moves at a translation speed of 600 mm/min along the negative direction of the Y axis at a uniform speed, and records the initial pose and target of the snake robot at this time.

When the Kinect V2 RGB-D vision sensor senses the dynamic and continuous translation of the terminal anthropomorphic obstacle, the Bi-RRT-Star algorithm dynamically re-plans a smooth path for the snake-shaped robot to run safely and without collision and to avoid moving anthropomorphic obstacles in time. The state of the serpentine robot pose configuration sequence represents the relative position of the snake robot and the anthropomorphic obstacle at different moments. During the entire path planning process, the snake-shaped robot can always reach the target position from the initial pose safely and without collision. The angle change curve of each joint during the obstacle avoidance process is shown in Fig. 7. From the figure, the movement of the snake-shaped robot can be seen. The angle of each joint in the dynamic obstacle avoidance process based on FANUC translational guidance varies from −6rad/s to 9 rad/s, and this range of change has a certain robustness.

2) Based on FANUC rotation guidance snake robot avoiding obstacles.

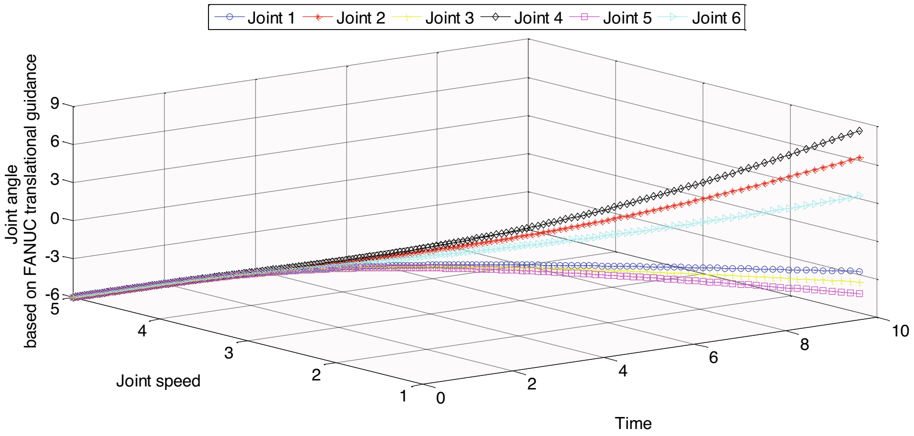

Fig. 7. The angle change curve of each joint during the dynamic obstacle avoidance process based on FANUC translational guidance.

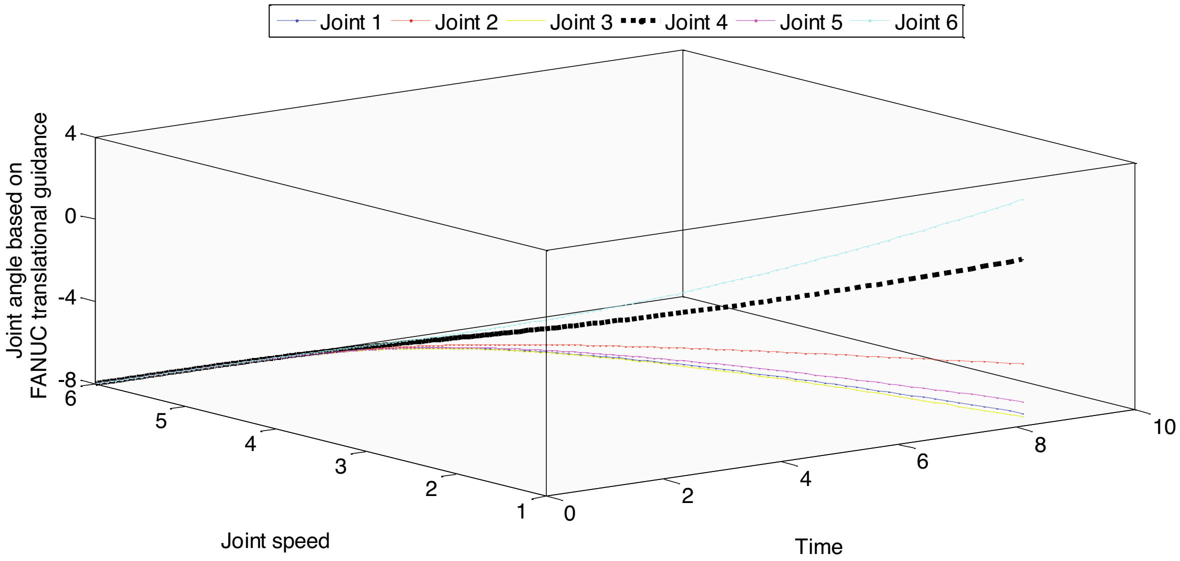

Fig. 8. The angle change curve of each joint in the dynamic obstacle avoidance process based on FANUC rotation guidance.

This section mainly tests the performance of the snake-shaped robot to avoid regular obstacles under the guidance of the Bi-RRT-Star dynamic path planning method. Similar to the regular translation, the FANUC snake-shaped robot is still manually programmed in advance to achieve the fixed anthropomorphic obstacle of the end effector at 60. The angular velocity of/min rotates at a constant speed around the positive direction of the Y axis, and records the initial pose and target pose of the snake-shaped robot at this time.

When the Kinect V2 RGB-D vision sensor senses the dynamic and continuous rotation of the end anthropomorphic obstacle, the Bi-RRT-Star algorithm dynamically re-plans a smooth path for the snake-shaped robot to run safely and without collision and the anthropomorphic obstacle that avoids movement in time. The state of the serpentine robot pose configuration sequence represents the relative position of the snake robot and the anthropomorphic obstacle at different moments. During the entire path planning process, the snake-shaped robot can always reach the target position from the initial pose safely and without collision. The angle change curve of each joint during the obstacle avoidance process is shown in Fig. 8. From the figure, you can see the motion of the snake-shaped robot.

5. Conclusion

This paper analyzes the basic ideas and implementation steps of the basic RRT algorithm, and evaluates its advantages and disadvantages. Then, in view of the large amount of calculation when constructing the path of this algorithm, the real-time performance is poor, and the path is composed of random nodes, which results in the path is not smooth and the path is not the shortest. An improved fast-expanding random algorithm is proposed. Finally, the SimRobot simulation software was used to simulate the planned path in a static environment, and the physical NAO snake-shaped robot was used to plan the trajectory in the actual dynamic process. When a moving obstacle moves to the planned path, directly discarding the previously constructed random tree itself is a waste of computing resources. Because searching and building a random tree is a very computer resource consuming thing, and the moving obstacle may only conflict with a branch in the random tree. There is no need to make such a big sacrifice. Therefore, the method is to detect whether the obstacle is in conflict with the branches of the random tree at this time, and then delete the branches. On the basis of the original random tree, dynamically we build and delete the branches of the random tree.

Simulation and experimental results show that the search speed and search efficiency are significantly improved compared with the Basic-RRT and Bi-RRT algorithms, the average planning time is shorter, the planning success rate is higher, and the generated path is smoother, and it can dynamically avoid the global environment. In terms of the multi-tree-based artificially guided RRT algorithm, the planner generates a fast-expanding random tree at the starting point, the end point, and the artificially guided point to form a multi-tree. When the obstacle speed is 4 m/s, the number of successful planning is 31, the planning time is 68.9 ms, and the trajectory cost is 0.63, which is in line with practical application requirements. In order to merge multiple trees to return a continuous solution path, the first problem is the connection of the trees. This paper uses a simple spline curve to perform fitting smoothing and obstacle avoidance processing at the connection, but occasionally sharp corners will appear. The robot will be strenuous to track because the dynamic constraints cannot be taken into account at the connection. Finding a more efficient connection fitting method to make the multi-tree connection as far as possible to meet the dynamic constraints is the next step in the improvement of the algorithm. In the future, we should explore the local feature construction method of 3D point cloud data in mixed scenes, and realize the establishment and extraction of feature descriptors with Euclidean transformation and symmetric transformation invariance, so as to obtain a general technology suitable for simultaneous recognition of all sizes and multi-targets in mixed scenes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarwal, D., Bharti, P.S., 2018. A review on comparative analysis of path planning and collision avoidance algorithms[J]. International Journal of Mechanical and Mechatronics Engineering 12 (6), 608–624.

Albini, A., Grella, F., Maiolino, P., Cannata, G., 2021. Exploiting Distributed Tactile Sensors to Drive a Robot Arm Through Obstacles[J]. IEEE Robotics and Automation Letters 6 (3), 4361–4368.

Amanov, E., Nguyen, T.D., Burgner-Kahrs, J., 2021. Tendon-driven continuum robots with extensible sections—A model-based evaluation of path-following motions [J]. The International Journal of Robotics Research 40 (1), 7–23.

Asghar, M.Z., Habib, A., Habib, A., Khan, A., Ali, R., Khattak, A., 2021. Exploring deep neural networks for rumor detection[J]. Journal of Ambient Intelligence and Humanized Computing 12 (4), 4315–4333.

Foerster, K.T., Schmid, S., 2019. Survey of reconfigurable data center networks: Enablers, algorithms, complexity[J]. ACM SIGACT News 50 (2), 62–79.

Girija, S., Joshi, A., 2019. Fast hybrid PSO-APF algorithm for path planning in obstacle rich environment[J]. IFAC-PapersOnLine 52 (29), 25–30.

Hua, X., Wang, G., Xu, J., Chen, K., 2021. Reinforcement learning-based collision-free path planner for redundant robot in narrow duct[J]. Journal of Intelligent Manufacturing 32 (2), 471–482.

Jia, Y., Ma, S., 2021. A Coach-Based Bayesian Reinforcement Learning Method for Snake Robot Control[J]. IEEE Robotics and Automation Letters 6 (2), 2319–2326.

Li, D., Pan, Z., Deng, H., Peng, T., 2020. 2D Underwater Obstacle Avoidance Control Algorithm Based on IB-LBM and APF Method for a Multi-Joint Snake-Like Robot [J]. Journal of Intelligent & Robotic Systems 98 (3-4), 771–790.

Li, L., Miao, Y., Qureshi, A.H., Yip, M.C., 2021. MPC-MPNet: Model-Predictive Motion Planning Networks for Fast, Near-Optimal Planning under Kinodynamic Constraints[J]. IEEE Robotics and Automation Letters 6 (3), 4496–4503.

Liu, G., Guo, J., 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification[J]. Neurocomputing 337, 325–338.

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

Malik, K.R., Ahmad, T., 2017. E-assessment data compatibility resolution methodology with bidirectional data transformation[J]. Eurasia Journal of Mathematics, Science and Technology Education 13 (7), 3969–3991.

Nakajima, M., Tanaka, M., Tanaka, K., 2019. Simultaneous control of two points for snake robot and its application to transportation[J]. IEEE Robotics and Automation Letters 5 (1), 111–118.

Naughton, N., Sun, J., Tekinalp, A., Parthasarathy, T., Chowdhary, G., Gazzola, M., 2021. Elastica: A compliant mechanics environment for soft robotic control[J]. IEEE Robotics and Automation Letters 6 (2), 3389–3396.

Pavlik, J.A., Sewell, E.C., Jacobson, S.H., 2021. Two new bidirectional search algorithms[J]. Computational Optimization and Applications 80 (2), 377–409.

Praserttaweelap, R., Kiatwanidvilai, S., 2020. Model of Obstacle Avoidance on Hard Disk Drive Manufacturing with Distance Constraint[J]. International Journal of Computer and Information Engineering 14 (11), 392–395.

Tanaka, M., Sawabe, H., Nakajima, M., Ariizumi, R., 2021. Redundant Control of a Planar Snake Robot with Prismatic Joints[J]. International Journal of Control, Automation and Systems 19 (10), 3475–3486.

Tang, L., Zhu, LiMin, Zhu, XiangYang, Gu, GuoYing, 2020. Confined spaces path following for cable-driven snake robots with prediction lookup and interpolation algorithms[J]. Science China Technological sciences 63 (2), 255–264.

Véras, L.G.D.O., Medeiros, F.L.L., Guimaráes, L.N.F., 2019. Systematic literature review of sampling process in rapidly-exploring random trees[J]. IEEE Access 7, 50933–50953.

Wang, H.-H., Yu, L., Tian, S.-W., Peng, Y.-F., Pei, X.-J., 2019. Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network[J]. Applied Intelligence 49 (8), 3016–3026.

Wang, W., Zuo, L., Xu, X., 2018. A learning-based multi-RRT approach for robot path planning in narrow passages[J]. Journal of Intelligent & Robotic Systems 90 (1), 81–100.