ROBOTIC ARMS PASSING THROUGH OBSTACLE NODES: USING AN IMPROVED INTELLIGENT ALGORITHM

Lina Guo
Shandong Institute of Commerce and Technology
Jinan, Shandong 250103, China

Abstract - In the working process, how efficiently the robotic arm passes the obstacle nodes has a great impact on its working efficiency. This paper briefly introduced the rapidly exploring random tree (RRT) algorithm and optimized it with the gravitational potential function based on its shortcomings. The B-spline curve was used to smooth the path. Experimental analyses were carried out in two-dimensional and three-dimensional environments. It was found that the improved RRT algorithm had an average running time of 0.39 s and an average path length of 93.96 mm in the two-dimensional environment; taking environment 1 as an example, the average running time and path length of the improved RRT algorithm was 1.03 s and 147.63 mm, respectively, in the three-dimensional environment, after passing through seven path nodes, which were significantly better than the standard RRT algorithm. The results prove the effectiveness of the improved intelligent algorithm for robotic arm path optimization selection. The improved RRT algorithm can be further promoted and applied in practice.

Keywords: Intelligent algorithm, Robotic arm, Path optimization, Obstacle node.

1. Introduction

As technical level develops, the application of robots in life and industry has become more and more extensive [1]. In industrial manufacturing, the robot has become an important role in promoting the development of industrial automation. At present, in the manufacturing industry, the six-degree-of-freedom robotic arm, which is flexible and extensively applicable, can accomplish many complex tasks. Due to the complexity of the working environment, how to achieve the path optimization of robotic arms to avoid obstacles efficiently [2] has become a key content in current research [3]. Sadhu et al. [4] improved the firefly algorithm (FA) and proposed an FA combined with Q-learning. They verified the superiority of the method in aspects of solution quality and running time through simulations and real experiments. Taking a two-degree-of-freedom robot as an example, Sadiq et al. [5] combined an improved ant colony optimization algorithm (ACO) with the D* algorithm to generate optimal paths and verified the accuracy and efficiency of the method through experiments. Li et al. [6] designed an improved A-star algorithm and a 3D collision detection algorithm to plan an optimal path for the blood taking needle fixed at the end of the robotic arm and found that the method was feasible through simulations and experiments. Gai et al. [7] designed a path planning algorithm based on the artificial potential field method and proved its effectiveness through experiments on the MA1440 industrial robot. The algorithm can realize the collision-free path planning of robotic arms. This paper focused on the rapidly exploring random tree (RRT) algorithm. This paper mainly studied the RRT algorithm, designed an improved intelligent algorithm, and compared it with the standard RRT algorithm to highlight its reliability in path optimization. This work provides a theoretical basis for improving the efficiency of robotic arms passing through obstacle nodes. Moreover, this paper also proved the effectiveness of improving the RRT algorithm with the gravitational potential function, suggesting that adjusting the step length could effectively improving the search ability of the RRT algorithm. It provides a new direction for the improvement of the RRT algorithm. In future research, more improvements can be considered for the fixed step length in the RRT algorithm.

2. Path Optimization Selection Method based on an Improved Intelligent Algorithm

2.1 RRT Algorithm

The RRT algorithm is an intelligent algorithm based on random sampling [8], which has the advantages of fast search and easy implementation. It has excellent performance in path problems in complex environments [9] and has also been well used in robotics [10]. The principle of the RRT algorithm is to obtain random sampling points after giving a starting point to make the tree grow toward the location of the sampling points until the target point is reached, and the tree nodes cannot collide with the danger area, as shown in Figure 1.

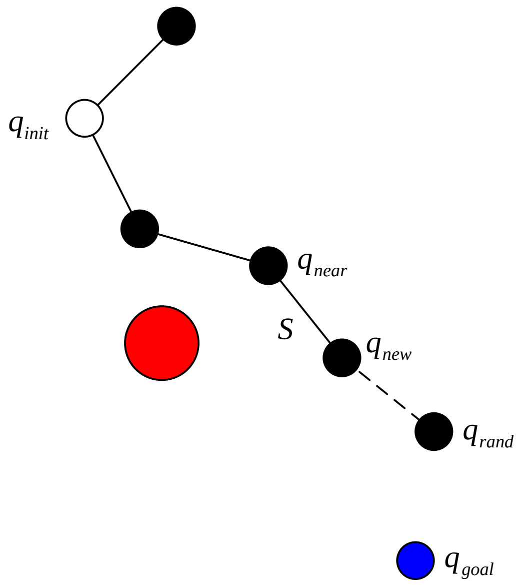

Figure 1: Schematic diagram of the principle of the RRT algorithm

In Figure 1, the red area is the obstacle node, initial node q_{init} is the starting point, and q_{goal} is the target point. The sample function is used for random sampling to obtain random point q_{rand}. Then, the nearnode function is used to find point q_{near} that is nearest to q_{rand}. The direction from q_{near} to q_{rand} is extended according to step length S to obtain a new node called q_{new}. The collision function is used to detect whether the path collides with the obstacle point. If it does, the sampling is repeated again; if not, q_{new} is added with a random number. After continuous growth, whether it has grown to the site near q_{goal} is determined; if it has, it indicates that the path has been successfully found.

The functions involved in the RRT algorithm are as follows.

(1) The sample function: new sampling point q_{rand} is obtained through random sampling.

(2) The near node function: all nodes are traversed to calculate the distance between q_i and q_{rand} and find the minimum value, i.e., point q_{near} that is nearest to q_{rand}. The distance formula is:

$$Distance = \sqrt{\left(\left(q_i(x) - q_{rand}(x)\right)^2 + \left(q_i(y) - q_{rand}(y)\right)^2 + \left(q_i(z) - q_{rand}(z)\right)^2\right)} \quad (1)$$

(3) The collision function: point q_{near} is connected to q_{new}; the set of all the points on the direction from q_{near} to q_{new} is written as: $\left\{q_j : q_{new} \in q_j, q_{new} = q_{new} + \frac{\lambda(q_{new} - q_{near})}{\|q_{new} - q_{near}\|}, \lambda \in [0,1]\right\}$; the coordinate of the obstacle point is $q_0(x_0, y_0, z_0)$, and the coordinate of q_j is $q_j(x_j, y_j, z_j)$; then, the collision-free function is:

$$q_{new} \leftarrow collisionfree(q_j, q_0, r)\{d: d_i \in d, d_i = |\overrightarrow{q_j - q_0}|\};$$

if $r \leq d$, there is no obstacle on the path; otherwise, there is an obstacle.

2.2 Improved RRT Algorithm

The standard RRT algorithm uses a fixed step length and random node expansion, which will lead to a slow convergence [11], especially in the case of many obstacle nodes [12]. In order to further improve the effectiveness of the intelligent algorithm for path optimization selection, this paper improves the RRT algorithm.

First, in order to reduce the random tree from producing too many useless nodes and to guide the tree to grow in the direction of the target point, it is improved by combining the gravitational potential function [13]. In the standard RRT algorithm, the growth direction of the new node is: $d = \frac{q_{rand} - q_{near}}{\|q_{rand} - q_{near}\|}$. Based on the gravitational potential function, the gravitational potential energy of q_{goal} at q_{near} is written as: $U = \frac{k}{2}\|q_{goal} - q_{near}\|^2$, where k is the gravitational force coefficient. The gravitational force of q_{goal} to q_{near} is: $U = k\|q_{goal} - q_{near}\|$. Then, the gravitational potential function of the target point to any point q is:

$$F(q) = \frac{k(q_{goal} - q)}{\|q_{goal} - q\|}. \quad (2)$$

Through the gravitational potential function, let q_{near} incline to q_{goal}; then, the growth direction of the new node after the improvement is:

$$d' = \frac{q_{rand} - q_{near}}{\|q_{rand} - q_{near}\|} + \frac{k(q_{goal} - q_{near})}{\|q_{goal} - q_{near}\|} \quad (3)$$

Then, k is adjusted according to the value of $\|q_{goal} - q_{near}\|$ to enable the dynamic adjustment of the step length.

The path finally obtained by the RRT algorithm is a polyline. The abrupt direction change of the robotic arm at the inflection point is not conducive to the long-term use of the arm. Therefore, this paper uses a quadratic B-spline curve [14] to smooth the path obtained by the RRT algorithm, as shown in Figure 2.

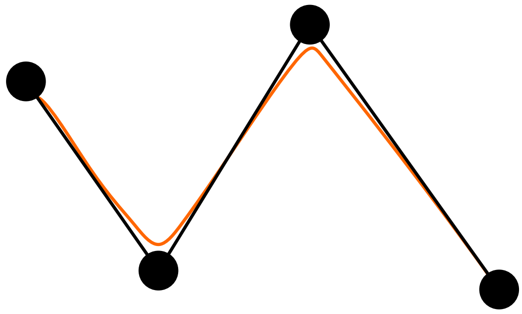

Figure 2: B-spline curve path smoothing

The formula for the k-order B-spline is written as:

$$L_{i,n}(t) = \sum_{k=0}^{n} L_{i+k} \cdot Q_{k,n}(t), \tag{4}$$

$$Q_{k,n}(t) = \frac{1}{n!}\sum_{j=0}^{n-k}(-1)^j \cdot C_{n+1}^{j} \cdot (t+n-k-j)^n, \tag{5}$$

where L_{i+k} is the vertex of the control and $Q_{k,n}(t)$ is the B-spline basis function, $0 \leq t \leq 1$, $k = 0,1,\cdots,n$. In the quadratic B-spline, the expression of the basis function is:

$$\begin{cases} Q_{0,2}(t) = \frac{1}{2}(t-1)^2 \\ Q_{1,2}(t) = \frac{1}{2}(-2t^2+2t+1). \\ Q_{2,2}(t) = \frac{1}{2}t^2 \end{cases} \tag{6}$$

Then, its segmental expression equation is written as:

$$L(t) = \sum_{k=0}^{2} Q_{k,2}(t) \cdot B_k = \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \end{bmatrix} \tag{7}$$

where B_k is the curve vertex.

3. Results and Analysis

Simulations were performed under the MATLAB environment to analyze the performance of the improved intelligent algorithm for the optimization selection of paths when passing through obstacle nodes. The analysis was first performed in a two-dimensional environment with a map size of 100 × 100 mm. In the map, the starting point was (10, 15), and the target point was (80, 80). The extended step length of random numbers was 20. The gravitational coefficient was 1.5. The red areas represented obstacles. The performance of RRT and improved RRT algorithms was compared by repeating the experiment 20 times. The path optimization results in one of the experiments are shown in Figure 3.

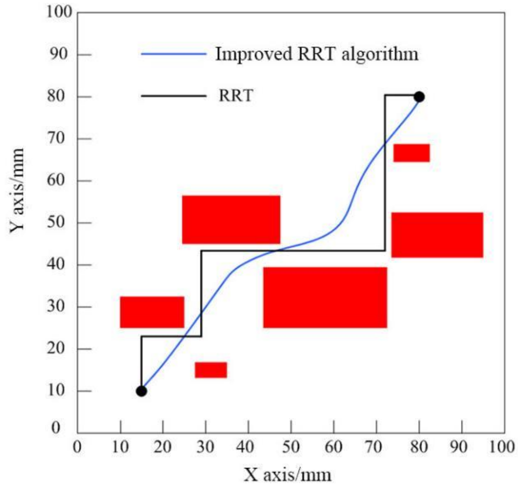

Figure 3: Results of path optimization selection for RRT and improved RRT algorithms

It was seen from Figure 3 that the path optimization selection result obtained by the standard RRT algorithm was a polyline with a total length of 133.7 mm; the path result obtained by the improved RRT algorithm after smoothing was a curve with a total length of 94 mm, which was 29.69% shorter than the standard RRT algorithm. The results of the 20 experiments were averaged to compare the running time and path length between the algorithms. The results are shown in Figures 4 and 5.

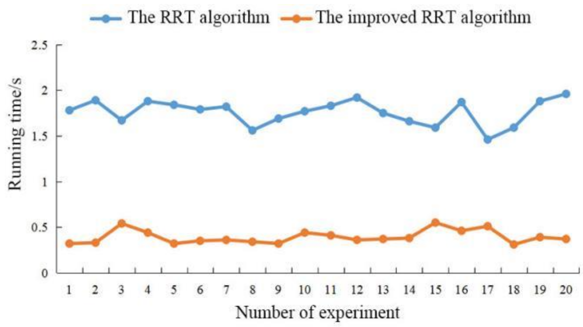

Figure 4: Comparison of the running time between different algorithms

It was observed in Figure 4 that the running time of the RRT algorithm was significantly longer than that of the improved RRT algorithm, which were all above 1.5 s, while the running time of the improved RRT algorithm was below 1 s. In comparison, the shortest running time of the RRT algorithm was 1.46 s, while the longest running time of the improved RRT algorithm was only 0.55 s. The average running time of the improved RRT algorithm was only 0.39 s after 20 times of experiment, which was 77.84% shorter than the RRT algorithm. These results suggested that the improved RRT algorithm was more efficient in optimizing the path selection for the robotic arm.

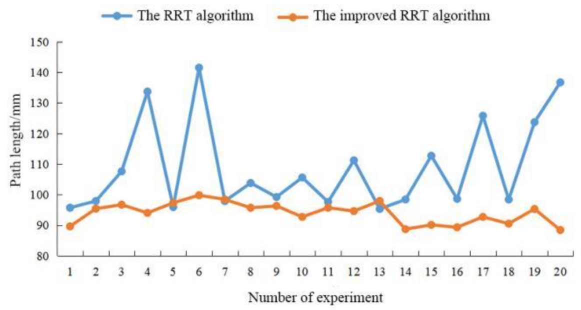

Figure 5: Comparison of the path length between different algorithms

It was seen from Figure 5 that the path results obtained by RRT had a large variation as there was a large gap between the path lengths obtained from different experiments. The longest and shortest path obtained by the RRT algorithm was 141.5 mm and 95.3 mm, respectively, while the paths obtained by the improved RRT were all below 100 mm, 99.8 mm the longest and 88.4 mm the shortest. The average path length of the improved RRT algorithm in the 20 experiments was 93.96 mm, which was 13.69% shorter than that of the RRT algorithm (108.86 mm). These results verified the effectiveness of the improved RRT algorithm in optimizing the path selection.

Then, the analysis was carried out in the three-dimensional environment. Figure 6-8 show the results of the RRT and improved RRT algorithms obtained from every time of run under three different three-dimensional environments. The map size was 90 × 90 × 50 mm. In Figure 6, the black point was the starting point, the blue point was the target point, the black line segment was the path optimization selection result obtained by the RRT algorithm, and the red line segment was the path optimization selection result obtained by the improved RRT algorithm.

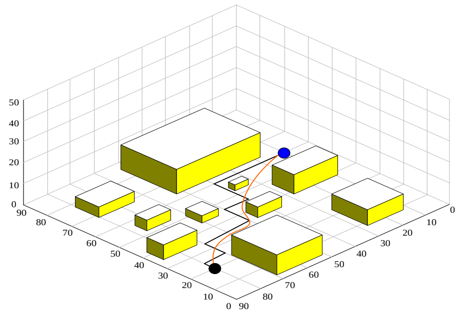

Figure 1: Three-dimensional environment 1

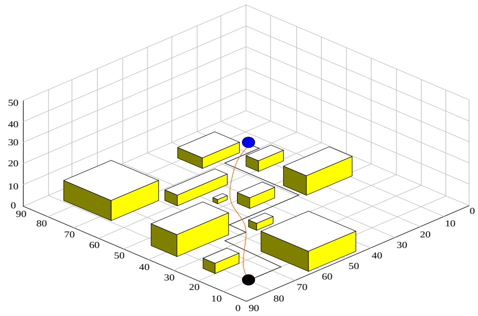

Figure 7: Three-dimensional environment 2

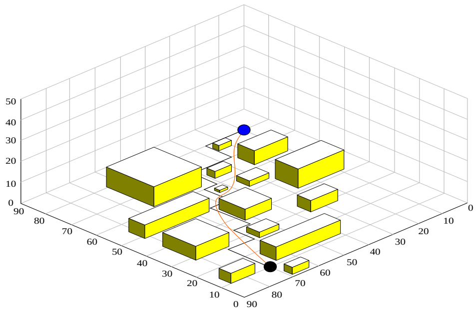

Figure 8: Three-dimensional environment 3

It was observed in Figures 6-8 that that the black line (the RRT algorithm) was more curved and the red line (the improved RRT algorithm) was smooth. In the three environments, RRT and improved RRT algorithms ran 100 times each, and the results were averaged, as shown in Table 1.

Table 1. Comparison of algorithm performance in a three-dimensional environment

	Environment 1		Environment 2		Environment 3	
	RRT algorithm	The improved RRT algorithm	RRT algorithm	The improved RRT algorithm	RRT algorithm	The improved RRT algorithm
Running time/s	6.72	1.03	6.87	1.05	7.64	1.07
Number of path nodes/n	72	7	75	8	79	8
Path length/mm	203.15	147.63	205.77	148.66	217.84	151.33

According to Table 1, the running time of the improved RRT algorithm was significantly shorter than that of the RRT algorithm in the three-dimensional environment. Taking environment 1 as an example, the running time of the improved RRT algorithm was 1.03 s, which was 84.67% less than that of the RRT algorithm. The improved RRT algorithm required less path nodes. Taking environment 1 as an example, the RRT algorithm obtained 72 path nodes, while the improved RRT algorithm only obtained seven nodes, which was about 1/10 of the RRT algorithm. As to the obtained path optimization selection results, under the three environments, the path length of the RRT algorithm was always larger than 200 mm, and the path length of the improved RRT algorithm was always larger than 150 mm. Taking environment 1 as an example, the path length of the RRT algorithm was 203.15 mm, and that of the improved RRT algorithm was 147.63 mm, which was 27.33% less than the RRT algorithm. These results verified the reliability of the improved RRT algorithm.

4. Conclusions

This paper studied the path optimization selection problem when the robotic arm passed the obstacle nodes and designed an improved RRT algorithm. The comparison between RRT and improved RRT algorithms demonstrated that the improved RRT algorithm had a shorter running time and a shorter average path length in the two-dimensional environment; in the three-dimensional environment, taking environment 1 as an example, the improved RRT algorithm had a running time of 1.03 s and a path length of 147.63 mm and obtained seven path nodes, which was significantly better than the RRT algorithm. This paper verified the effectiveness of the improved RRT algorithm in solving the path optimization selection problem. The proposed algorithm can be further applied in actual robotic arms.

References

[1] Zaidi L, Sahnoun M, Bettayeb B. “Task allocation based on shared resource constraint for multi-robot systems in manufacturing industry - ScienceDirect,” IFAC-PapersOnLine, 2019, 52(13):2020-2025.

[2] Raheem F A, Sadiq A T, Abbas N. “Robot Arm Free Cartesian Space Analysis for Heuristic Path Planning Enhancement,” International Journal of Mechanical & Mechatronics Engineering, 2019, 19(1):29-42.

[3] Xu T, Zhou H, Tan S, Li Z, Ju X, Peng Y. “Mechanical arm obstacle avoidance path planning based on improved artificial potential field method,” Industrial Robot, 2022, 49(2):271-279.

[4] Sadhu AK, Konar A, Bhattacharjee T, Das S. “Synergism of Firefly Algorithm and Q-Learning for Robot Arm Path Planning,” Swarm and Evolutionary Computation, 2018, 43:50-68.

[5] Sadiq A T, Raheem F A, Abbas N. “Ant Colony Algorithm Improvement for Robot Arm Path Planning Optimization Based on D* Strategy,” International Journal of Mechanical & Mechatronics Engineering, 2021, 21(No. 1):96-111.

[6] Li F, Huang Z, Xu L. “Path Planning of 6-DOF Venipuncture Robot Arm Based on Improved A-star and Collision Detection Algorithms,” 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019: 2971-2976.

[7] Gai SN, Sun R, Chen SJ, Ji S. “6-DOF Robotic Obstacle Avoidance Path Planning Based on Artificial Potential Field Method,” 2019 16th International Conference on Ubiquitous Robots (UR), 2019: 165-168.

[8] Wang J, Hirota K, Wu X, Dai Y, Jia Z. “Hybrid Bidirectional Rapidly Exploring Random Tree Path Planning Algorithm with Reinforcement Learning,” Journal of Advanced Computational Intelligence and Intelligent Informatics, 2021, 25(1 TN.148):121-129.

[9] Xiong J, Duan X. “Path planning for UAV based on improved dynamic step RRT algorithm,” Journal of Physics Conference Series, 2021, 1983(1):012034.

[10] Pérez-Hurtado I, Martínez-Del-Amor M, Zhang G, Neri F, Perez-Jimenez MJ. “A membrane parallel rapidly-exploring random tree algorithm for robotic motion planning,” Integrated Computer Aided Engineering, 2020, 27(2):121-138.

[11] Shi Y, Li Q, Bu S, Yang J, Zhu L. “Research on Intelligent Vehicle Path Planning Based on Rapidly-Exploring Random Tree,” Mathematical Problems in Engineering, 2020, 2020(Pt.8):5910503.1-5910503.14.

[12] Li B, Chen B. “An Adaptive Rapidly-Exploring Random Tree,” IEEE/CAA Journal of Automatica Sinica, 2022, 9(2): 283-294.

[13] Qureshi A H, Ayaz Y. “Potential functions based sampling heuristic for optimal path planning,” Autonomous Robots, 2016, 40(6):1079-1093.

[14] Kong X, Chen J. “Two Extensions of the Quadratic Nonuniform B-Spline Curve with Local Shape Parameter Series,” Mathematical Problems in Engineering: Theory, Methods and Applications, 2021, 2021(Pt.38):9980320.1-9980320.12.