International Core Journal of Engineering Volume 12 Issue 6, 2026
ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

Research on Manipulator Path Planning based on Improved RRT*

Jiaqi Zhu

School of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, Shanxi, China

Abstract

Aiming at the drawbacks of traditional RRT* algorithm in 3D path planning for manipulators, such as blind random sampling, low search efficiency and redundant initial paths, an improved RRT* path planning method integrating goal-biased sampling, adaptive step-size expansion, path pruning and B-spline smoothing is proposed. Firstly, goal-biased sampling is adopted to drive the search tree to expand toward the target area and avoid random exploration in irrelevant regions. Secondly, an adaptive step size is introduced on the basis of target guidance to further reduce redundant search. Finally, path pruning and B-spline smoothing are applied to optimize path compactness and trajectory continuity. A 3D static obstacle environment is established on MATLAB, and 100 repeated simulation experiments are conducted on RRT, RRT* and the improved RRT* algorithm. The experimental results show that the success rate of the improved algorithm reaches 100% in all three scenarios. In complex environments, its average planning time is 0.039727 s, which is lower than 0.122270 s of RRT and 1.201800 s of RRT*. Meanwhile, the average number of iterations and tree nodes are also significantly reduced. Ablation experiments verify that goal-biased sampling is the main factor for the improvement of search efficiency; the adaptive step size further enhances the expansion efficiency based on target guidance, and path pruning and smoothing effectively improve the quality of the final path.

Keywords

3D Path Planning; Improved RRT*; Goal-Oriented Sampling; Manipulator; Path Optimization.

1. Introduction

The primary task of manipulator path planning is to generate a collision-free motion trajectory for the end-effector from the start point to the target point within the workspace. On the premise of satisfying obstacle avoidance constraints, it is essential to comprehensively optimize planning efficiency, path length and trajectory smoothness. With the increasing complexity of intelligent manufacturing, automated assembly, welding, handling, sorting and other industrial scenarios, manipulators are required to not only complete obstacle avoidance search rapidly in three-dimensional space, but also guarantee favorable continuity and executability of trajectories. Accordingly, the real-time performance, stability and path quality of path planning algorithms have become key research focuses.

Among sampling-based path planning algorithms, the Rapidly-exploring Random Tree (RRT) has emerged as a classic method for manipulator obstacle avoidance. It features probabilistic completeness, dispenses with the explicit construction of a complete search map, and performs well in high-dimensional continuous spaces. LaValle [1] first proposed the RRT algorithm, which combines random sampling with tree expansion to address path planning problems in high-dimensional spaces. On this basis, Kuffner and LaValle [2] developed RRT-Connect, and adopted a bidirectional fast tree connection mechanism to improve the efficiency of single-query path planning.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

Although the conventional RRT can quickly obtain a feasible path, its strong randomness easily leads to redundant search, resulting in overly long paths and unsatisfactory trajectory quality. To solve this problem, Karaman and Frazzoli [3] put forward the RRT* algorithm. By introducing parent node re-selection and path rewiring mechanisms, RRT* achieves asymptotic optimality, laying a solid foundation for subsequent research on algorithm improvement.

Nevertheless, RRT* still suffers from slow convergence, blind sampling and heavy reliance on post-processing for path optimization. A great number of studies have been conducted to improve its performance from the perspectives of sampling strategy, search mechanism and path optimization. Nasir et al. [4] proposed RRT*-Smart, which accelerates convergence via intelligent sampling and path optimization. Gammell et al. [5] presented Informed RRT*, which conducts heuristic sampling within potential optimization regions after acquiring an initial solution to boost search efficiency. Jeong et al. [6] designed RRT*-Quick to enhance convergence by expanding the candidate set of parent nodes and adopting branch pruning. Janson et al. [7] proposed FMT* to improve the search efficiency of optimal paths based on implicit random geometric graphs. Gammell, Barfoot and Srinivasa [8] further systematically analyzed informed sampling for asymptotically optimal path planning. Otte and Frazzoli [9] developed RRTX, which extends sampling-based planning to scenarios requiring rapid replanning. In a comprehensive review, Gammell and Strub [10] pointed out that sampling constraint, heuristic guidance and path quality improvement remain the core challenges for asymptotically optimal sampling-based planning methods. Overall, existing studies have made remarkable progress in goal-oriented sampling, convergence acceleration and path smoothing. However, when applied to complex three-dimensional environments with obstacles, traditional RRT* still has obvious limitations: random sampling remains highly blind; the fixed step size fails to balance global efficiency and local accuracy; the generated raw path contains numerous redundant nodes and sharp corners, resulting in poor smoothness. These drawbacks make it difficult to meet the practical requirements of manipulators for high precision and stable operation.

Aiming at the above deficiencies, this paper proposes an improved RRT* path planning method. In the search phase, goal-biased sampling and adaptive step-size expansion are adopted to reduce the blindness of random sampling and redundant expansion. After the initial path is generated, path pruning is used to remove redundant nodes, and B-spline smoothing is implemented to enhance trajectory continuity. A series of comparative simulations and ablation experiments are carried out on RRT, original RRT* and the proposed algorithm under multiple scenarios. The results fully verify the superior performance of the improved algorithm and clarify the contribution of each optimization strategy.

2. Problem Description and Modeling of Path Planning

2.1 Modeling of Workspace and Obstacles

This paper focuses on geometric path planning for the end-effector of a manipulator in a three-dimensional workspace. To highlight the performance of the algorithm, the motion of the end-effector is simplified into a point trajectory searching problem in 3D space. The boundaries of the workspace are defined as follows:

$$\Omega = (x, y, z) \mid 0 \le x \le 100, 0 \le y \le 100, 0 \le z \le 100 \tag{1}$$

The starting point is set as:

$$q_{\text{start}} = (5,5,5) \tag{2}$$

The target point is set as:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

$$q_{\text{goal}} = (95,95,90) \tag{3}$$

Obstacles are approximated using axis-aligned cuboids. The i-th obstacle is defined as:

$$O_i = [x_{\min}, y_{\min}, z_{\min}, d_x, d_y, d_z] \tag{4}$$

Then its occupied region satisfies:

$$x_{\min} \le x \le x_{\min} + d_x, y_{\min} \le y \le y_{\min} + d_y, z_{\min} \le z \le z_{\min} + d_z \tag{5}$$

This modeling method is easy to implement and facilitates collision detection and 3D visualization, which can meet the research requirements of geometric path planning in static obstacle environments in this paper.

2.2 Collision Detection Method

For sampling-based path planning algorithms, it is necessary to judge not only whether discrete nodes lie inside obstacles, but also whether the line segments connecting nodes intersect obstacles. Therefore, this paper adopts the segment discrete sampling method for collision detection. Let the two endpoints of the detected line segment be $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$, then any point on the segment can be expressed as:

$$p(t) = p_1 + t(p_2 - p_1), t \in [0,1] \tag{6}$$

Uniform sampling is performed on the line segment at equal intervals. If any sampling point exceeds the workspace boundary or falls inside an obstacle, the line segment is determined to be in collision.

This study focuses on geometric path planning for the manipulator end-effector within the workspace. Accordingly, collision detection is carried out for the reference point of the end-effector and the connecting line segments. This simplification highlights the comparative analysis of algorithm search efficiency and path quality. Nevertheless, the actual size of the end-effector, the occupied volume of manipulator links and dynamic constraints are not fully taken into account. These issues will be further investigated in future research combined with the physical model of the manipulator.

2.3 Performance Metrics

To comprehensively evaluate the performance of different path planning algorithms, the following metrics are adopted in this paper: success rate, average planning time, average number of iterations, average total number of tree nodes, average number of nodes on the original path, average length of the original path, average number of nodes on the pruned path, average length of the pruned path, and average length of the smoothed path. Among these indicators, the success rate, planning time, number of iterations and total tree nodes mainly reflect the performance in the search phase, while the lengths of the original path, pruned path and smoothed path characterize the performance of the path optimization phase. Let the total number of experiments be $\boldsymbol{N}$ and the number of successful planning runs be $\boldsymbol{N_s}$. The success rate is defined as:

$$R_s = \frac{N_s}{N} \tag{7}$$

Let the path node sequence be $\boldsymbol{P} = \{\boldsymbol{p_0}, \boldsymbol{p_1}, \ldots, \boldsymbol{p_n}\}$. The total path length is defined as:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

$$L(P) = \sum_{i=1}^{n} \|p_i - p_{i-1}\| \tag{8}$$

The above metrics can effectively evaluate algorithm performance from multiple dimensions including search efficiency, planning stability and path quality. Meanwhile, the standard deviations of planning time, iteration count, total tree nodes and original path length are calculated to quantify the performance fluctuation across repeated experiments.

3. Improved RRT* Algorithm

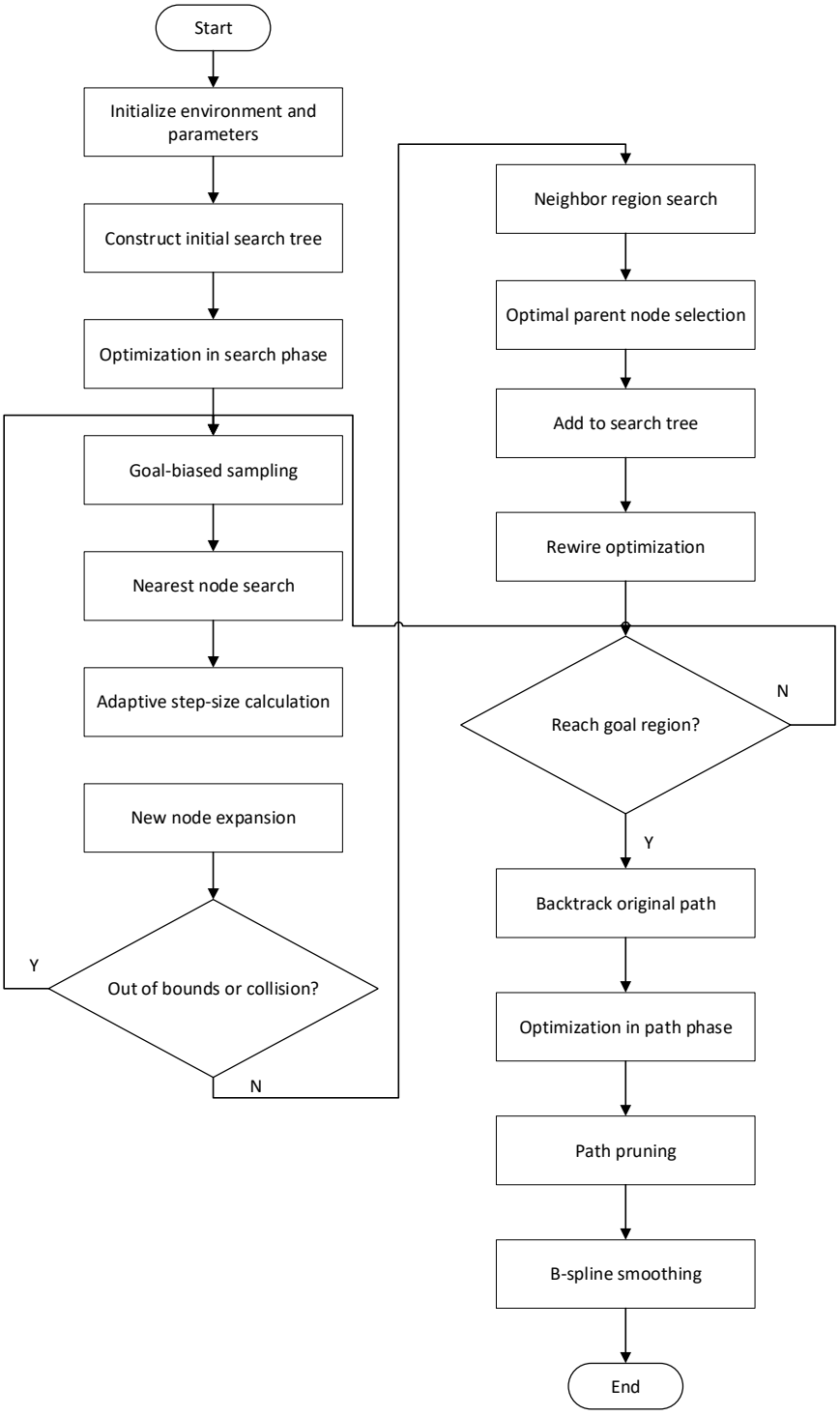

Figure 1. Overall flow chart of improved RRT* algorithm

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

Traditional RRT* improves path quality progressively via parent node reselection and rewiring, yet it suffers from blind exploration, low expansion efficiency and heavy post-processing overhead in complex three-dimensional environments. To tackle these drawbacks, the improvements proposed in this work are divided into two parts: optimization at the searching stage and optimization at the path refinement stage.

During the searching stage, a goal-biased sampling strategy is introduced to guide the tree expansion toward the target, and an adaptive step-size scheme based on target distance is adopted to balance global searching efficiency and local searching accuracy. Meanwhile, the neighborhood search, optimal parent selection and rewiring mechanisms of original RRT* are retained to continuously optimize path cost. At the path refinement stage, greedy pruning is performed on the raw path to eliminate redundant intermediate nodes that can be omitted by direct connection; subsequently, B-spline curves are used to smooth key path waypoints for generating trajectories with superior continuity. The overall framework follows the common "sampling improvement plus post-processing refinement" paradigm widely adopted in existing improved RRT* literatures, while the proposed method features better simplicity and easier implementation in simulation.

As can be seen from Figure 1, the core of the improved RRT* proposed in this paper does not lie in simply stacking multiple functional modules. Instead, the algorithm first reduces invalid node expansion via goal-oriented searching, and then improves the quality of the final trajectory through post-processing optimization of the obtained path.

3.1 Goal-Biased Sampling Strategy

Traditional RRT* adopts fully random sampling. Despite its favorable exploration capability, the algorithm tends to generate massive invalid searches in regions irrelevant to the target. To mitigate such sampling blindness, this paper introduces a goal-biased strategy in the sampling phase: in each iteration, the target node is directly selected as the sampling point with a fixed probability, while random sampling within the free configuration space is performed otherwise. Let $\boldsymbol{p_g}$ denote the goal-biased probability; the generation rule for random sampling point $\boldsymbol{q_{rand}}$ is defined as:

$$q_{\text{rand}} = \begin{cases} q_{\text{goal}}, & \text{rand} < p_g \\ q_{\text{free}}, & \text{rand} \geq p_g \end{cases} \tag{9}$$

where $\boldsymbol{q}_{\text{goal}}$ denotes the target node and $\boldsymbol{q}_{\text{free}}$ represents the random sampling point in free space. In the experiments of this paper, $\boldsymbol{p_g}$ is set to 0.20. While retaining the global exploration capability of the search tree, the proposed strategy drives the tree to expand preferentially toward the target region, so as to reduce invalid iterations and shorten path planning time.

3.2 Adaptive Step-size Expansion based on Goal Distance

Fixed-step expansion features easy implementation, yet it fails to balance global search efficiency and local search accuracy at different stages of path planning. To address this issue, an adaptive step-size strategy based on goal distance is proposed in this paper. Let $\boldsymbol{q}_{\text{near}}$ be the current nearest node and $\boldsymbol{q}_{\text{goal}}$ be the target node; the Euclidean distance between them is defined as:

$$d(q_{\text{near}}, q_{\text{goal}}) = \| q_{\text{goal}} - q_{\text{near}} \| \tag{10}$$

Furthermore, the normalized distance coefficient is defined as:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

$$\lambda = \min\left(\frac{d(q_{\text{near}}, q_{\text{goal}})}{D}, 1\right) \tag{11}$$

where $\boldsymbol{D}$ denotes the reference distance. Accordingly, the current expansion step size can be formulated as:

$$\text{step} = \text{step}_{\min} + (\text{step}_{\max} - \text{step}_{\min})\lambda \tag{12}$$

where $\boldsymbol{step_{min}}$ and $\boldsymbol{step_{max}}$ denote the minimum and maximum step sizes, respectively. It can be observed that the step size approaches $\boldsymbol{step_{max}}$ when the node is far away from the target, which accelerates the searching speed; as the node gets close to the target region, the step size converges to $\boldsymbol{step_{min}}$ to improve local searching precision. In the simulations of this paper, $\boldsymbol{\text{step}_{\min} = 2.5}$, $\boldsymbol{\text{step}_{\max} = 8.0}$ and $\boldsymbol{D = 120}$ are adopted.

3.3 Path Pruning Strategy

Even after path searching with the improved RRT*, the backtracked raw path may still contain a large number of redundant nodes. To eliminate unnecessary sharp turns, a greedy path pruning strategy is employed in this work. Let the original path be:

$$P = \{p_0, p_1, p_2, \dots, p_n\} \tag{13}$$

If a collision-free straight connection exists between the current node $\boldsymbol{p_i}$ and a subsequent node $\boldsymbol{p_j}$, the intermediate nodes $\boldsymbol{p_{i+1}, \dots, p_{j-1}}$ can be removed. After iterative segment-by-segment pruning, the pruned path is obtained as:

$$P' = \{p'_0, p'_1, \dots, p'_m\}, m < n \tag{14}$$

This strategy can drastically reduce the number of path nodes without compromising path feasibility and provides a more compact set of key control points for subsequent smoothing.

3.4 B-Spline Smoothing Strategy

After pruning, redundant nodes are greatly eliminated, yet the resultant path remains piecewise linear. To enhance trajectory continuity, B-spline curves are utilized to smooth the pruned key control points. Let the sequence of control points for the pruned path be:

$$P' = \{p'_0, p'_1, \dots, p'_m\} \tag{15}$$

Then the cubic B-spline curve is expressed as:

$$C(u) = \sum_{i=0}^{m} N_{i,3}(u) p'_i \tag{16}$$

where $\boldsymbol{N_{i,3}(u)}$ denotes the cubic B-spline basis function. Uniform sampling over the parameter interval yields the set of smoothed trajectory points:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

$$S = \{s_1, s_2, \ldots, s_k\} \tag{17}$$

In this paper, $\boldsymbol{k = 100}$ is selected as the number of sampling points for the smoothed path. It should be noted that B-spline smoothing mainly aims to improve the geometric continuity of trajectories rather than minimize path length further. Accordingly, the length of the smoothed path may be slightly larger than that of the pruned path, yet the resulting reference trajectory possesses superior quality.

3.5 Algorithm Procedure

Combining all the above improved measures, the implementation procedure of the improved RRT* algorithm is described as follows. After initializing environmental parameters, sampling points are generated with a target-biased probability. The nearest node is searched from the existing tree, and the current expansion step size is calculated according to the distance between this node and the target. A new node is produced along the sampling direction followed by collision detection. If no collision occurs, neighborhood search, optimal parent selection and rewiring operations are carried out. Once the newly generated node falls within the target threshold range, it connects to the target point and the raw path is obtained via backtracking. Afterwards, greedy pruning is applied to the original path, and B-spline fitting is adopted for path smoothing before the final reference trajectory is output. This workflow simultaneously improves the searching efficiency and optimizes path quality, forming the core of the proposed algorithm.

4. Simulation Analysis

4.1 Parameter Settings

To verify the effectiveness of the proposed improved RRT* algorithm for 3D path planning of robotic manipulators, a 3D static obstacle environment is established on the MATLAB platform, and comparative experiments are conducted among the basic RRT, basic RRT*, and improved RRT* algorithms. The experimental workspace is set to $\boldsymbol{100 \times 100 \times 100}$, with the start point $\mathbf{q}_{\text{start}} = (\mathbf{5, 5, 5})$ and the goal point $\mathbf{q}_{\text{goal}} = (\mathbf{95, 95, 90})$. Obstacles are modeled as axis-aligned cuboids. To reduce randomness caused by stochastic sampling, each algorithm runs independently 100 times in each scenario, and metrics including success rate, average planning time, standard deviation of planning time, average number of iterations, average total number of tree nodes, average number of original path nodes, and average original path length are recorded.

For the basic RRT algorithm, the fixed expansion step size is set to 5, the maximum number of iterations is set to 4000, the goal judgment threshold is set to 8, and the number of sampling points for line segment collision detection is set to 25. The basic RRT* algorithm incorporates neighborhood search, optimal parent selection, and rewiring mechanisms based on RRT, with a neighborhood search radius of $\mathbf{r = 12}$. The improved RRT* algorithm introduces goal-biased sampling and adaptive step size strategies on the basis of basic RRT*, where the goal-biased probability is $\mathbf{p_g = 0.20}$, the minimum step size is $\mathbf{step_{min} = 2.5}$, the maximum step size is $\mathbf{step_{max} = 8.0}$, the reference distance is $\mathbf{D = 120}$, and the number of sampling points for the smoothed trajectory is 100.

The experimental scenarios are divided into three categories: Simple, Medium, and Complex. The Simple scenario contains a small number of obstacles, mainly used to verify the basic planning capability of the algorithms. The Medium scenario features an increased number of obstacles and higher distribution complexity, designed to evaluate the general obstacle avoidance performance of the algorithms. The Complex scenario has denser obstacles, aiming to further compare the search efficiency, planning stability, and path quality of different algorithms in complex environments. The experimental analysis in this paper focuses on three key questions: whether the improved RRT* searches faster than RRT and RRT*, whether it maintains satisfactory path quality while accelerating the search, and which improved modules mainly contribute to its performance enhancement.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

4.2 Analysis of Search Efficiency Advantages

The primary advantage of the improved RRT* over RRT and standard RRT* lies in search efficiency. Table 1 summarizes the search performance comparison of the three algorithms across different scenarios.

Table 1. Performance Comparison of Three Algorithms under Different Scenarios

Scenario	Algorithm	Success Rate	Average Planning Time /s	Standard Deviation of Planning Time /s	Average Number of Iterations	Average Total Number of Tree Nodes
Simple	RRT	0.97	0.072941	0.041651	1355.10	1310.80
Simple	RRT*	1.00	0.770840	0.829300	1355.60	1308.10
Simple	Improved RRT*	1.00	0.015751	0.003597	95.66	88.49
Medium	RRT	1.00	0.081472	0.046743	1326.60	1228.80
Medium	RRT*	1.00	0.871430	0.633210	1395.20	1295.40
Medium	Improved RRT*	1.00	0.024811	0.015092	133.81	105.14
Complex	RRT	0.99	0.122270	0.072777	1520.20	1311.80
Complex	RRT*	0.99	1.201800	0.912650	1468.70	1265.50
Complex	Improved RRT*	1.00	0.039727	0.017358	182.66	119.90

As shown in Table 1, in terms of success rate, the improved RRT* achieves 100 successful planning runs in all three scenarios (Simple, Medium, and Complex), with a success rate of 1.00. The success rates of the basic RRT are 0.97 and 0.99 in the Simple and Complex scenarios, respectively, while the success rate of the basic RRT* is 0.99 in the Complex scenario. This indicates that all three algorithms are generally capable of searching for feasible paths in the current static obstacle environment, but the improved RRT* exhibits better planning stability.

In terms of planning time, the average planning times of the improved RRT* in the three scenarios are 0.015751 s, 0.024811 s, and 0.039727 s, respectively, which are significantly lower than those of the basic RRT (0.072941 s, 0.081472 s, and 0.122270 s) and much lower than those of the basic RRT* (0.770840 s, 0.871430 s, and 1.201800 s). Taking the Complex scenario as an example, the average planning time of the improved RRT* is reduced by approximately 67.51% compared with the basic RRT and by about 96.69% compared with the basic RRT*. This demonstrates that the improved algorithm not only outperforms RRT* in speed but also achieves significantly faster planning than the basic RRT while retaining the path optimization mechanism of RRT*.

The fundamental reason for this advantage is the significant reduction in search scale. In the Complex scenario, the average number of iterations for the basic RRT, basic RRT*, and improved RRT* is 1520.20, 1468.70, and 182.66, respectively; the average total number of tree nodes is 1311.80, 1265.50, and 119.90, respectively. It can be seen that the improved RRT* accomplishes path connection with fewer iterations and a smaller search tree scale, rather than improving the success rate by increasing the number of search nodes. Goal-biased sampling drives the search tree to expand toward the target region, reducing random explorations in irrelevant free space; the adaptive step size further eliminates redundant expansions based on target guidance, making the search process more compact.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

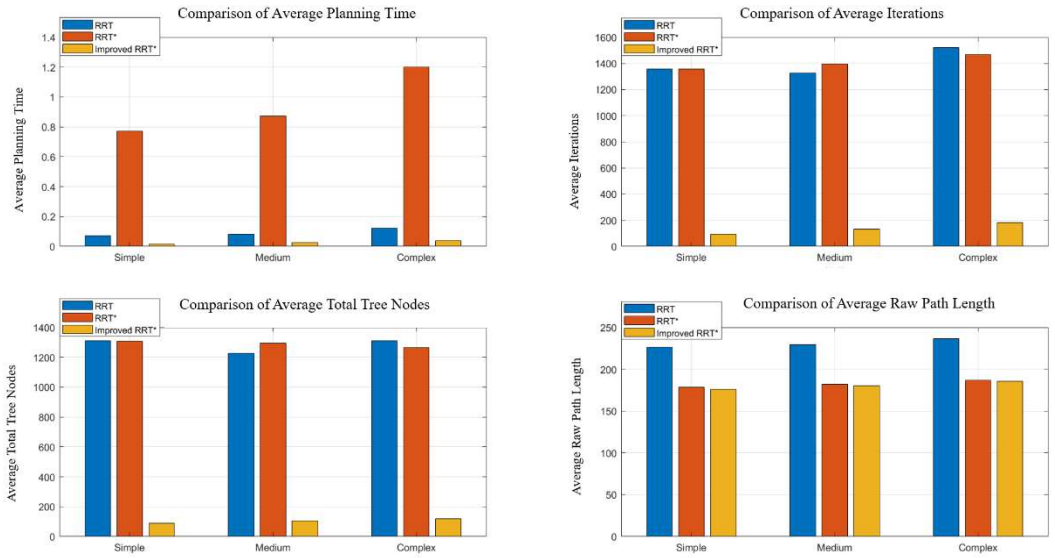

Figure 2. Performance Index Comparison of Three Algorithms in Different Scenarios

It can be further observed from Figure 2 that the improved RRT* remarkably outperforms the basic RRT and standard RRT* in terms of average planning time, average iterations and average total tree nodes. Especially in the Complex scenario, the number of tree nodes and iterations of the improved RRT* decrease drastically, which demonstrates its highly directional searching and greatly reduced blindness in random expansion. In addition, the standard deviation of planning time for the improved RRT* is 0.017358 s under the Complex scenario, lower than 0.072777 s of basic RRT and 0.912650 s of standard RRT*, implying smaller runtime fluctuation and superior planning stability in repeated experiments.

Accordingly, the primary superiority of the improved RRT* can be summarized as follows: compared with RRT, it achieves faster path searching with higher stability; compared with standard RRT*, it retains the intrinsic path optimization mechanism of RRT* while drastically cutting down invalid sampling, iteration counts and tree size, hence substantially boosting the overall search efficiency.

4.3 Analysis of Path Quality Retention Performance

Higher search efficiency does not necessarily guarantee superior algorithm performance. If an algorithm accelerates goal convergence at the cost of path quality, resulting in remarkably longer routes or severe detours, it cannot satisfy the practical path planning requirements of robotic manipulators. Consequently, following the analysis of search efficiency, it is essential to compare the raw path quality among the three algorithms. Table 2 presents the raw path quality comparison of the three algorithms under various scenarios.

As shown in Table 2, the basic RRT yields the longest average raw path length across all three scenarios, reaching 226.30, 229.43 and 236.69 for Simple, Medium and Complex respectively. Since basic RRT only focuses on rapidly searching feasible trajectories without any path cost optimization, the generated paths commonly contain obvious detours and redundant broken line segments. In contrast, standard RRT* improves path quality via parent node reselection and rewiring, reducing the average raw path length to 178.57, 182.22 and 186.78 correspondingly, which outperforms basic RRT substantially.

The average raw path lengths of the improved RRT* are 175.89, 180.34 and 185.48 respectively, which are markedly shorter than those of the basic RRT and comparable to or slightly superior to the standard RRT*. Taking the Complex scenario as an example, the average raw path length of the improved RRT* is 185.48, marginally lower than 186.78 of the standard RRT* and considerably smaller than 236.69 of the basic RRT. The above results demonstrate that the improved RRT* retains

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

nearly identical path optimization performance to standard RRT* rather than degrading path quality to the level of basic RRT while achieving a substantial.

Table 2. Raw Path Quality Comparison of Three Algorithms in Different Scenarios

Scenario	Algorithm	Average Number of Raw Path Nodes	Average Raw Path Length	Standard Deviation of Raw Path Length
Simple	RRT	46.01	226.30	14.895
Simple	RRT*	21.31	178.57	7.038
Simple	Improved RRT*	25.51	175.89	6.269
Medium	RRT	46.64	229.43	14.655
Medium	RRT*	21.61	182.22	9.533
Medium	Improved RRT*	25.91	180.34	10.148
Complex	RRT	48.03	236.69	17.710
Complex	RRT*	22.24	186.78	11.495
Complex	Improved RRT*	26.14	185.48	10.089

This phenomenon conforms to the structural design of the improved algorithm. The proposed method retains the core optimization modules of standard RRT*, including neighborhood search, optimal parent selection and rewiring operations, so it maintains the inherent capability of path cost minimization. The introduced goal-biased sampling and adaptive step size mainly serve to accelerate exploration without undermining the original optimization framework of RRT*.

Although the improved RRT* produces slightly more raw path nodes than standard RRT*, e.g., 26.14 nodes versus 22.24 nodes of standard RRT* in the Complex scenario, the number of path nodes cannot directly represent path length. The improved algorithm reserves more local turning points around obstacles for flexible obstacle avoidance, yet its overall path length is still marginally smaller than that of standard RRT*, which verifies no degradation in overall path quality.

In terms of the standard deviation of raw path length, the values reach 17.710, 11.495 and 10.089 for basic RRT, standard RRT* and improved RRT* in the Complex scenario, respectively. It can be concluded that the improved RRT* achieves not only shorter average paths but also smaller fluctuations of path quality across repeated trials.

As can be observed from Figure 3, paths generated by the basic RRT exhibit prominent overall detours and severe local fluctuations. Compared with basic RRT, standard RRT* produces shorter paths yet still contains redundant segments during obstacle avoidance. In contrast, trajectories from the improved RRT* are more compact with searching directions pointing toward the target region, and its local obstacle avoidance layout is more reasonable. Such visualized results in Figure 3 are basically consistent with the statistical path-length data listed in Table 2.

In summary, the second-layer superiority of the improved RRT* lies in the fact that it does not merely pursue rapid connection between start and target nodes. While drastically cutting planning time, iterations and tree nodes, it maintains raw path quality comparable to or even slightly better than standard RRT*. In other words, the improved RRT* integrates the fast-search merit of basic RRT and the path optimization capability of standard RRT*.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

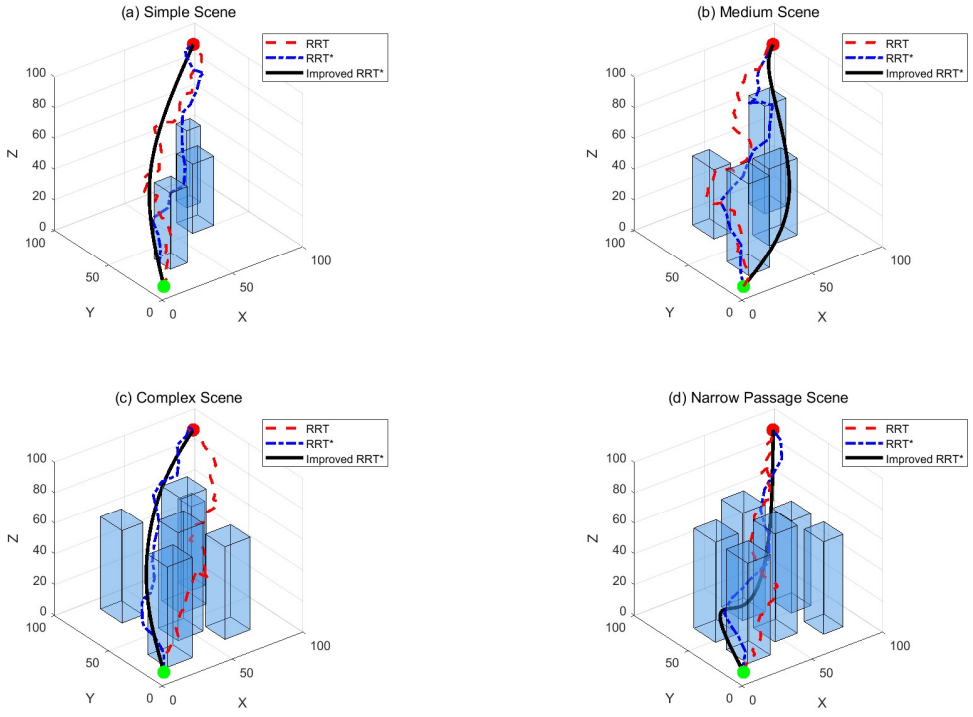

Figure 3. Comparison of End-effector Trajectory Planning Results for Manipulator with Different Algorithms under Various Scenarios

4.4 Analysis of Post-processing Effect for Planned Paths

Previous experimental results demonstrate that the improved RRT* can rapidly generate high-quality raw paths. Nevertheless, the end-effector trajectory of a manipulator needs not only collision-free property but also compactness, continuity and smoothness. Accordingly, path pruning and B-spline smoothing are adopted as post-processing operations after path searching, and the post-processing performance of the improved RRT* is summarized in Table 3.

Table 3. Comparison of Post-processing Effects for Paths Generated by Improved RRT*

Scenario	Average Number of Raw Path Nodes	Average Number of Pruned Path Nodes	Average Raw Path Length	Average Pruned Path Length	Average Smoothed Path Length
Simple	25.51	3.02	175.89	157.35	158.18
Medium	25.91	3.44	180.34	159.72	161.00
Complex	26.14	4.47	185.48	166.01	168.11

As shown in Table 3, path pruning significantly reduces the number of path nodes and path length. For the Simple, Medium and Complex scenarios, the average path node count of the improved RRT* declines from 25.51, 25.91 and 26.14 to 3.02, 3.44 and 4.47, representing a reduction of approximately 88.16%, 86.72% and 82.90%, respectively. Meanwhile, the average path length drops from 175.89, 180.34 and 185.48 to 157.35, 159.72 and 166.01, with a length reduction of around 10.54%, 11.43% and 10.50% correspondingly.

This reveals that the raw path still contains numerous intermediate nodes that can be removed via collision-free straight-line connection. The greedy pruning strategy effectively reserves only critical waypoints and streamlines the overall path profile.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

After B-spline smoothing on pruned paths, the resulting trajectory length rises slightly compared with the pruned version yet remains considerably shorter than the original raw path. Taking the Complex scenario as an example, the pruned path length is 166.01 and the smoothed counterpart reaches 168.11. Despite a mild length increment against the pruned path, it is approximately 9.36% shorter than the original raw path of 185.48. It proves that the core function of B-spline smoothing lies in optimizing geometric continuity and smoothing polyline trajectories rather than further minimizing path length, which renders the trajectory applicable for subsequent manipulator trajectory planning.

To prevent local obstacle intrusion during smoothing, high-density discrete secondary collision detection is implemented on smoothed trajectories. Experimental verification confirms that all smoothed trajectories stay collision-free under the given static obstacle environment. Hence, the smoothing operation improves trajectory continuity without compromising the geometric feasibility of the planned path.

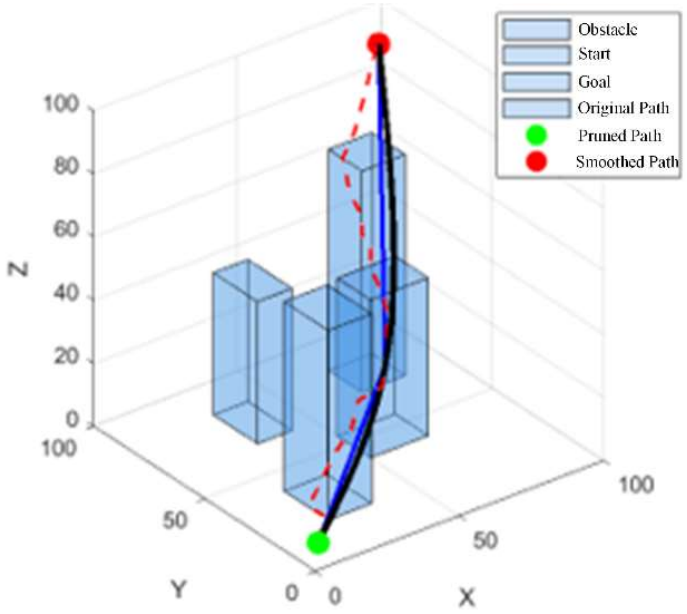

Figure 4. Secondary Collision Verification Results of Smoothed Trajectories

Therefore, path pruning and B-spline smoothing are not merely cosmetic treatments for trajectory curves; instead, they constitute critical post-processing procedures that convert raw feasible paths from path searching into compact, smooth and executable reference trajectories. These post-processing operations differ fundamentally from the optimization strategies adopted in the sampling phase. Specifically, goal-biased sampling and adaptive step size focus on accelerating feasible path searching, whereas pruning and smoothing aim to refine path morphology to satisfy the practical demands of subsequent manipulator trajectory generation.

4.5 Ablation Experiment Analysis

Previous comparative experiments have verified that the improved RRT* outperforms basic RRT and standard RRT* in overall search efficiency and path quality. Nevertheless, it remains necessary to quantify which improved modules contribute to such performance gains. To this end, ablation experiments are carried out under the Complex scenario to separately investigate the respective effects of goal-biased sampling, adaptive step expansion, as well as path pruning and smoothing. Different algorithm configurations are listed in Table 4.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

Table 4. Algorithm Configuration Settings for Ablation Experiments

Algorithm Version	Goal-biased Sampling	Adaptive Step	Pruning & Smoothing	Verification Purpose
RRT*	×	×	×	Baseline algorithm
RRT* + Goal-biased Sampling	√	×	×	Verify the effectiveness of goal-oriented sampling
RRT* + Adaptive Step	×	√	×	Verify the effectiveness of adaptive step expansion
Improved RRT*	√	√	√	Verify the comprehensive performance of the complete proposed method

The ablation experimental results of each algorithm variant under the complex environment are presented in Table 5.

Table 5. Ablation Experimental Results under Complex Scenario

Algorithm Version	Success Rate	Average Planning Time/s	Standard Deviation of Planning Time/s	Average Iterations	Average Total Number of Tree Nodes	Average Number of Raw Path Nodes	Average Raw Path Length	Average Pruned Path Length	Average Smoothed Path Length
RRT*	0.99	1.4310	1.3173	1469.00	1265.80	22.00	185.83	—	—
RRT* + Goal-biased Sampling	1.00	0.078577	0.041536	213.28	130.40	25.29	180.11	—	—
RRT* + Adaptive Step	0.95	1.9339	1.2541	2081.20	1746.20	21.842	186.59	—	—
Improved RRT*	1.00	0.037610	0.016016	186.58	121.19	26.83	187.64	165.33	167.25

As can be seen from Table 5, goal-biased sampling serves as the primary contributor to the improvement of search efficiency. Compared with the original standard RRT*, the configuration equipped only with goal-biased sampling raises the success rate from 0.99 to 1.00, cuts the average planning time from 1.4310 seconds down to 0.078577 seconds, reduces the average iteration count from 1469.00 to 213.28, and decreases the average total number of tree nodes from 1265.80 to 130.40. The sharp decline in planning time, iteration times and the scale of tree nodes demonstrates that goal-biased sampling effectively guides the expansion of the search tree toward the target area and eliminates redundant random exploration in irrelevant space.

It can be intuitively observed from Figure 5 that average planning time, average iterations and average total tree nodes are all remarkably reduced after introducing only goal-biased sampling. This indicates that the superior speed of the improved RRT* over conventional RRT* primarily originates from abandoning blind full-space random exploration; instead, the tree expands toward the target with a fixed probability to drastically cut down ineffective searching.

Meanwhile, Table 5 reveals that the adaptive step strategy cannot work well independently. When only adaptive step is adopted without goal-biased sampling, the success rate drops to 0.95, the average planning time rises to 1.9339 s, the average iteration count increases to 2081.20, and the average total

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

tree nodes grow to 1746.20, resulting in inferior overall performance compared with baseline RRT*. The underlying reason is that variable step size fails to eliminate redundant expansions when sampling directions remain scattered and may even generate more invalid extensions along wrong orientations. Only when goal-biased sampling provides reliable search guidance can adaptive step bring extra performance gains.

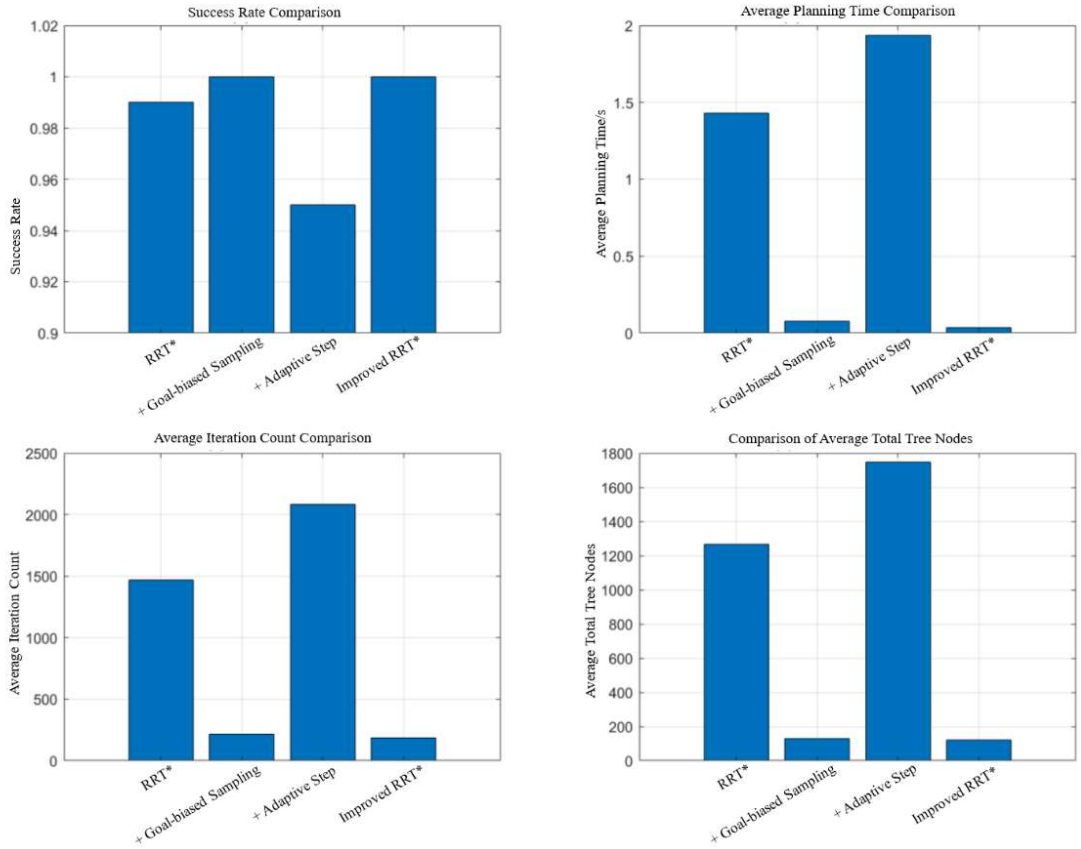

Figure 5. Comparison of Search Performance via Ablation Experiments under Complex Scenario

Equipped with both goal-biased sampling and adaptive step, the fully improved RRT* further reduces average planning time to 0.037610 s, lower than 0.078577 s of the RRT* with sole goal bias. Its average iterations fall to 186.58 and average total tree nodes decrease to 121.19, outperforming the RRT* plus goal-biased sampling variant. Such results verify their complementary effects: goal-biased sampling enhances directional searching, while adaptive step minimizes redundant expansions on the basis of definite expanding directions.

From the perspective of path quality, the average raw path length of complete improved RRT* is 187.64, slightly longer than 185.83 of original RRT* and 180.11 of RRT* with goal bias alone. It demonstrates that high searching efficiency and minimal raw path length are conflicting optimization objectives. Goal-biased sampling and adaptive step mainly focus on accelerating planning, whereas high-quality final trajectories rely on subsequent pruning and smoothing. After pruning, the average path length of the improved RRT* is shortened from 187.64 to 165.33; after B-spline smoothing, the length moderately rises to 167.25 with greatly improved trajectory continuity and remains distinctly shorter than the original raw path.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

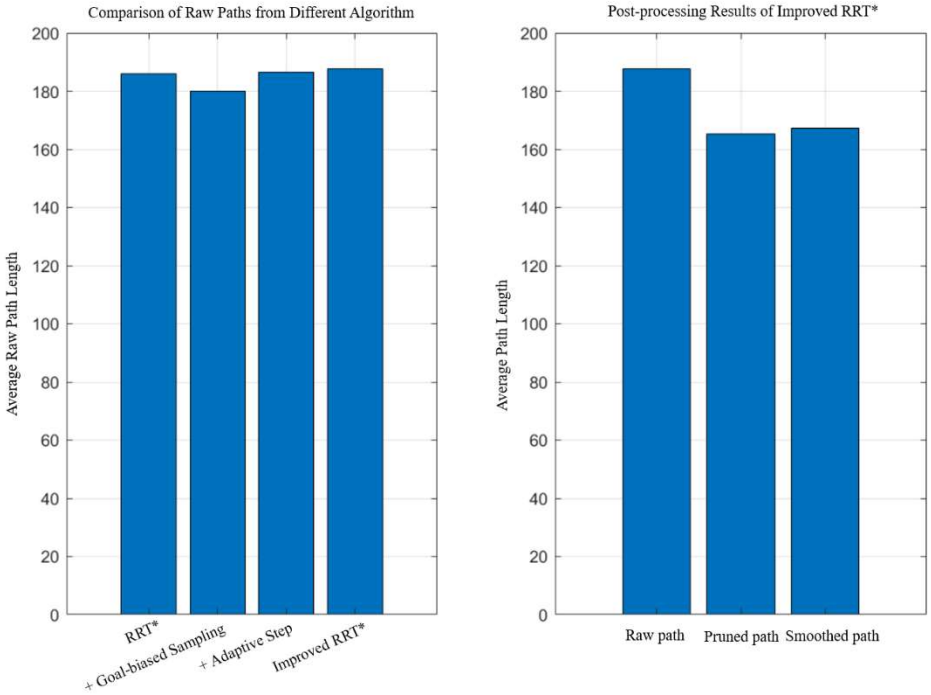

Figure 6. Comparison of Raw Path Lengths and Post-processing Results of the Improved RRT*

It can be observed from Figure 6 that the variations in the average raw path length of different algorithm versions fail to fully follow the changing trend of search efficiency, indicating that faster planning speed does not guarantee a shorter raw path. The superiority of the fully improved RRT* is reflected in two aspects: first, the combination of goal-biased sampling and adaptive step greatly accelerates the searching process; second, subsequent pruning and smoothing refine the final trajectory quality in post-processing stage.

In summary, compared with the original RRT, the improved RRT* achieves faster planning together with shorter path length and fewer redundant nodes and detours. In contrast to standard RRT*, it obtains comparable final path quality while drastically cutting planning time, iteration counts and the total number of tree nodes. Ablation results further verify that goal-biased sampling dominates efficiency improvement, adaptive step plays a synergistic optimization role under directional guidance, and pruning plus smoothing are responsible for final trajectory refinement. Therefore, the outstanding performance of the improved RRT* originates from the layered collaborative framework consisting of goal-oriented searching, adaptive expansion and path post-processing.

5. Conclusion

Aiming at the drawbacks of conventional RRT* algorithm in 3D path planning for manipulators, such as severe blindness in random sampling, low searching efficiency, excessive redundant path nodes and poor trajectory smoothness, this paper proposes an improved RRT* path planning method incorporating goal-biased sampling, adaptive step expansion, path pruning and B-spline smoothing. Three types of static obstacle environments are established on the MATLAB platform, and 100 repeated experiments are conducted on the original RRT, standard RRT* and the improved RRT*. The main conclusions are summarized as follows:

The improved RRT* achieves a 100% planning success rate in simple, medium-complex and complex environments with favorable planning stability. Compared with the original RRT and standard RRT*, the improved algorithm obtains obvious reductions in average planning time, average iterations and average total tree nodes, which verifies that the combination of goal-biased sampling and adaptive step strategy effectively eliminates invalid search and improves exploration efficiency.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202606_12(6).0003

In terms of path quality, the average raw path length of the improved RRT* is markedly shorter than that of the original RRT and comparable to or slightly superior to standard RRT*. In the complex environment, the average raw path lengths of original RRT, standard RRT* and improved RRT* are 236.69, 186.78 and 185.48 respectively. It demonstrates that the proposed algorithm maintains favorable path quality while accelerating search speed.

Path pruning dramatically removes redundant nodes and shortens path length. For three scenarios, the average path nodes of improved RRT* drop from 25.51, 25.91, 26.14 to 3.02, 3.44, 4.47 correspondingly; meanwhile, the average path lengths decrease from 175.89, 180.34, 185.48 to 157.35, 159.72, 166.01. Subsequent B-spline smoothing further optimizes the geometric continuity of trajectories, and secondary collision detection proves all smoothed paths remain collision-free.

Ablation experiments prove that goal-biased sampling dominates the improvement of search efficiency. Under the complex environment, only with goal-biased sampling enabled, the average planning time declines from 1.4310 s (standard RRT*) to 0.078577 s, and average iterations fall from 1469.00 to 213.28. Adaptive step performs poorly when used alone, yet it further cuts redundant expansions under directional guidance from goal-biased sampling, pushing the average planning time of fully improved RRT* down to 0.037610 s. Path pruning and smoothing are responsible for eliminating redundant nodes and optimizing trajectory continuity.

References

[1] LaValle, S. M. (1998). *Rapidly-exploring random trees: A new tool for path planning* (Technical Report TR 98-11). Iowa State University.

[2] Kuffner, J. J., & LaValle, S. M. (2000). RRT-Connect: An efficient approach to single-query path planning. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* (pp. 995–1001). https://doi.org/10.1109/ROBOT.2000.844730

[3] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894. https://doi.org/10.1177/0278364911406761

[4] Nasir, J., Islam, F., Malik, U., et al. (2013). RRT*-SMART: A rapid convergence implementation of RRT*. In *International Conference on Computer, Communications, and Control Technology*. InTech. https://doi.org/10.5772/56718

[5] Gammell, J. D., Srinivasa, S. S., & Barfoot, T. D. (2014). Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/IROS.2014.6942976

[6] Jeong, I. B., Lee, S. J., & Kim, J. H. (2015). RRT*-Quick: A motion planning algorithm with faster convergence rate. In *Robot Intelligence Technology and Applications 3* (pp. 79–90). Springer. https://doi.org/10.1007/978-3-319-16841-8_7

[7] Janson, L., Schmerling, E., Clark, A., et al. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International Journal of Robotics Research*, 34(7), 883–921. https://doi.org/10.1177/0278364915577958

[8] Gammell, J. D., Barfoot, T. D., & Srinivasa, S. S. (2018). Informed sampling for asymptotically optimal path planning. *IEEE Transactions on Robotics*, 34(4), 966–984. https://doi.org/10.1109/TRO.2018.2830331

[9] Otte, M., & Frazzoli, E. (2016). RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning. *The International Journal of Robotics Research*, 35(7), 797–822. https://doi.org/10.1177/0278364915594679

[10] Gammell, J. D., & Strub, M. P. (2021). Asymptotically optimal sampling-based motion planning methods. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 295–318. https://doi.org/10.1146/annurev-control-061920-093753