Hindawi
Journal of Robotics
Volume 2022, Article ID 3477265, 9 pages
https://doi.org/10.1155/2022/3477265

Research Article

Autonomous Navigation of Robots Based on the Improved Informed-RRT* Algorithm and DWA

Jun Dai, Dongfang Li, Junwei Zhao, and Yanqin Li

Department of Mechanical and Power Engineering, Henan Polytechnic University, Century Avenue, Shanyang District, Jiaozuo, China

Correspondence should be addressed to Junwei Zhao; zjwmail@hpu.edu.cn

Received 23 December 2021; Accepted 12 January 2022; Published 8 February 2022

Academic Editor: L. Fortuna

An improved method is proposed in this investigation to solve the problems of poor path quality and low navigation efficiency of the Informed-RRT* algorithm in robot autonomous navigation. First, the greedy algorithm is introduced in the path planning procedure. When a new node is obtained, it will be judged whether it can directly reach the target point. Second, the search scope of the potential optimal parent node becomes the constructed path, instead of the node tree, which reduces the number of nodes to be searched and improves the navigation efficiency. Combined with the dynamic window approach (DWA), the improved algorithm is utilized to simulate the autonomous navigation process of the robot based on the Robot Operating System (ROS) platform. The simulation results show that compared with the original algorithm, the length of the global path is reduced by 5.15%, and the time of planning path and autonomous navigation is shortened by 78.34% and 21.67%, respectively.

1. Introduction

Significant achievements in related research on mobile robot technology have been made in recent years, and it is now widely used in a wide range of real-world applications such as firefighting and rescue, medical services, catering services, environmental disinfection, and microfluidics devices [1]. The need for mobile robot autonomous navigation is gradually growing in the market. Path planning has been a hot topic among researchers as one of the fundamental technologies in robot autonomous navigation[2, 3]. However, the current state of autonomous robot navigation has poor path quality and navigation efficiency[4, 5].

There are mainly two types of path planning algorithms in autonomous navigation: global and local path planning algorithms. Global path planning is responsible for designing a path from the starting point to the finishing point, and local path planning is responsible for dynamic obstacle avoidance as the robot advances. Traditional global path planning algorithms include the ant colony algorithm [6–8], Dijkstra algorithm [9, 10], A* algorithm [11, 12], and rapidly exploring random tree (RRT) algorithm[13]. However, the abovementioned algorithm has different drawbacks. For example, the ant colony algorithm has the disadvantage of high computational complexity; the path planned by the Dijkstra algorithm has the problem of local optimization; the A* algorithm has the problem of low path planning efficiency in a large environment; and due to an enormous number of nodes, the RRT algorithm gets a nonoptimal path. To solve the path nonoptimal problem of the RRT algorithm, Karaman et al. [14] proposed the RRT*algorithm, which performs partial pruning and rerouting of new nodes during the planning process to make the path approach produce an the optimal solution. However, there is still a problem of low path planning efficiency. Gammell et al. [15] proposed the Informed-RRT* algorithm, which improves the path planning efficiency by optimizing the sampling space [16, 17]. The local path planning algorithm is adopted forthe artificial potential field [18], the path planned by this algorithm is relatively smooth, but it still has a problem of local optimization; the dynamic window approach (DWA)[19, 20] based on robot dynamics relies on the speed that the robot can reach in a period to form a window-like local path planning. As a nonlinear

system, path planning solves optimization problems within the constraints provided by uncertain variables. External irregular control signals can be added to stimulate the hidden efficiency of path planning [21].

In the global path planning algorithm, the Informed-RRT* algorithm does not need constructing an explicit sampling space and vastly reduces the search time. However, this algorithm still has the disadvantages of poor path and low path optimization efficiency. Therefore, an improved Informed-RRT* path planning algorithm is proposed in this research. First, the greedy algorithm is introduced in the path planning process. When a new sampling point that can be added to the node tree is found, it is judged whether the node can be directly connected to the target point. Planning the path is stopped and optimization of the path is begun once the condition is established; otherwise, sampling continues. Second, the search object for the potential optimal parent node is replaced by the constructed node tree with the path to reduce the consumed time when searching for the possible optimal parent node in the path optimization process. After simulation on the ROS platform, it is found that the improved algorithm can improve navigation efficiency.

2. Informed-RRT* Algorithm

The Informed-RRT* algorithm is a path planning algorithm based on random sampling. Based on the RRT algorithm, the algorithm optimizes parent node reselection and prunes and reroutes planes in the process of generating the first path. The sampling space is then limited to an elliptical area, and as the length of the path decreases, the elliptical area is gradually reduced.

2.1. RRT Algorithm. As shown in Figure 1(a), the RRT algorithm randomly samples to obtain the Xrand node in the space and finds the Xnearest node closest to the Xrand node on the path tree and can be connected to it without obstacles. As shown in Figure 1(b), the RRT algorithm takes the Xnearest node as the parent node of the Xrand node, connecting the Xnearest node and the Xrand node, and adds the node Xrand to the node tree (it becomes the node Xnew). As shown in Figures1(c) and 1(d), until the endpoint Xgoal appears in the unit circle area centered on the Xnew node and connects the Xnew node and the Xgoal node, the plane can be obtained inversely according to the relationship between the child node and the parent node.

2.2. Reselecting Parent Node and Pruning and Rewiring. As shown in Figure 2(a), the new node Xnew is obtained according to the sampling point Xrand and the nearest node Xnearest node in the node tree by using this node as the parent node. As shown in Figure 2(b), when the Xnew node is added to the node tree, we draw a small circle with the node Xnew as the center and consider whether there is a better parent Xmin node in the circle to make the distance between the Xrand point and the Xnew node shorter. If there are more suitable parent nodes, they are connected and the original connection is removed. As shown in Figures 2(c) and 2(d), after completing the reselection of the parent node, it is judged whether there is a certain kind of node in the circle. The cost of the node passing through the Xnew node to the initial node is smaller than the cost of the original path from the node to the initial node. If this type of node exists, the Xnew node is taken as the parent node, and the node is disconnected from the original parent node.

2.3. Optimization of Sampling Space. As shown in Figure 3, the Informed-RRT* algorithm optimizes the sampling space accordingly. Xstart represents the starting point, Xgoal represents the endpoint, Cmin represents the distance between two nodes, and Cbest represents the length of the path obtained by the first iteration of the Informed-RRT* algorithm. When the first path is obtained, the length of the path Cbest is calculated. As shown in equations (1) and (2), Cbest is the distance 2*a between the two vertices of the ellipse, and Cmin is the distance 2*c between the two focal points of the ellipse. The Informed-RRT* algorithm can obtain an ellipse sampling space according to the two parameters, Cbest and Cmin. As the path of the Informed-RRT* algorithm is continuously optimized, Cbest is constantly shortened, the sampling space is continually reduced, and the convergence speed is continually accelerated.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{1}$$

$$a^2 - c^2 = b^2. \tag{2}$$

2.4. Informed-RRT Algorithm.* The pseudocode of the Informed-RRT* algorithm is shown in Figure 4. M represents the experimental map, T represents the node tree, V represents the node of the node tree, E represents the connection between the nodes in the node tree, Ls represents the list of vertices near the Xnew node, and StepSize represents the sampling step size. Sampling was performed on the map to obtain the sampling node Xrand (line 3). The Xnearest node closest to the Xrand node in the node tree (lines 4-5) is found. If the distance between the Xrand node and the Xnearest node is less than StepSize, the Xrand node becomes the Xnew node. If the distance between Xrand and Xnearest is greater than StepSize, a new Xnew node is obtained by intercepting a distance equal to StepSize along the direction of Xnear to Xrand according to the step size StepSize. A circle is drawn with Xnew as the center and StepSize as the radius, the potential optimal parent node is searched for in the node tree within the circle, and it is saved in Ls (line 6). The node Xmin is found in the Ls, and the minimum cost from Xinit to Xmin to Xnew (line 7) is guaranteed. The relationship between the Xnew node and Xmin is added to the node tree, the relationship between the Xnew node and the original parent node is disconnected, and then, rewiring is performed according to Ls (lines 9-10). This procedure first checks each node of Ls for node x; if the cost from Xinit to Xnew then to x is less than the original cost from Xinit to x

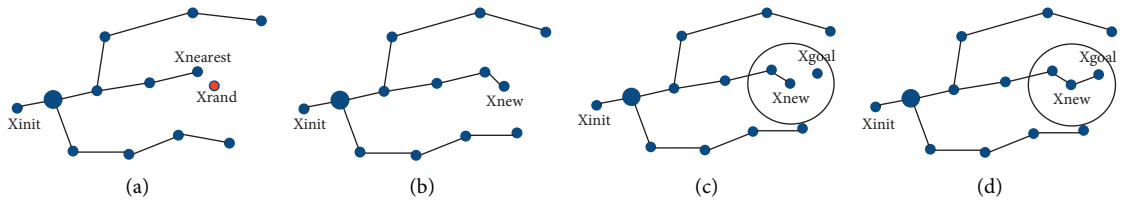

Figure 1: RRT algorithm schematic diagram.

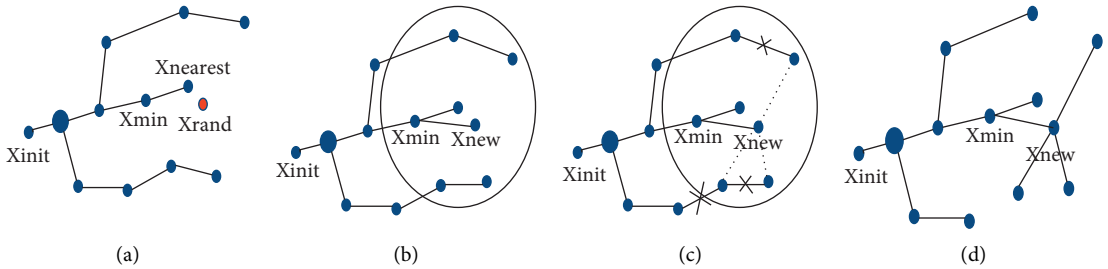

Figure 2: The informed-RRT* algorithm reselects the parent node and rewiring schematic diagram. (a) Extended random tree. (b) Reselecting parent node. (c) Replanning. (d) Replanned.

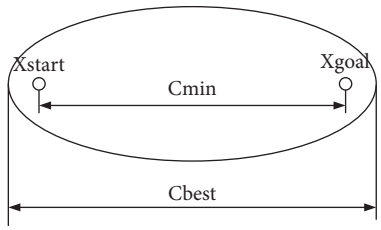

Figure 3: Schematic diagram of the Informed-RRT * sampling area.

```
1   V = {Xinit}, E = Ø,T = (V,E), StepSize = 0.05, Cmin
    = dis(Nstart,Ngoal), Cbest = 0,goal = false ;
2   for I 0 to N do
3     Xrand ← Sample() ;
4     Xnearest ← NearestVertices(Xrand,T) ;
5     Xnew ← Steer(Xnearest, Xrand,, StepSize) ;
6     Ls ← NearVertices(Xnew,T,StepSize) ;
7     Xmin ← ChooseBestParent(Xnew,Ls) ;
8     if X min = Ø then
9       T ← InserVertex(Xnew,Xmin,T) ;
10      T ← RewireVertices(Xnew,T,Ls) ;
11      goal ← Checkgoal(Xnew,Nodegoal)
12      if goal
13        Cbest ← dis(Nstart,Ngoal,T) ;
14        return Cbest ;
15    return path ← path(Nstart,Ngoal,T) ;
```

Figure 4: Informed-RRT * algorithm pseudocode.

and Xnew can be connected to x, then x is disconnected from the original parent node and Xnew is connected as the parent node of x to the tree, and a new node tree T is published. Detecting whether the target point is reached (line 11), if the target point is detected, the planning of the first path is completed, and the length of the path Cbest could be obtained (line 13–14). We return to the next round of sampling iteration, but the sampling area becomes an elliptical area adapted to the two parameters, Cbest and Cmin.

3. Optimization of Informed-RRT* Algorithm

The Informed-RRT* algorithm judges whether it should be connected to the target point by the threshold of the distance from the current node to the target point, then ends the path planning procedure, and enters the path optimization procedure. In the current environment, new sampling points can directly be connected to the target point even though the distance from the target point is greater than the threshold. In the case of this situation, the Informed-RRT* path planning algorithm will continue sampling and will not enter the path optimization procedure until the sampling node is less than the threshold. In contrast, the greedy algorithm directly connects the new sampling point with the target point, then ends the path planning, and enters the path optimization process. In addition, in the path optimization process of the Informed-RRT* algorithm, the search object when searching for the potential optimal parent node is the entire node tree, which consumes considerable time during the search process, which seriously affects the efficiency of path planning.

3.1. Introduction of the Greedy Algorithm. The greedy algorithm is a simpler and faster design technique for optimal solution problems. The algorithm is often based on the current situation to make an optimal choice according to a procedure optimization purpose, without considering all possible overall conditions, so it saves considerable time in preparing to find the optimal solution.

As shown in Figure 5 (line 12 of Figure 6), the goal node represents the target point, the New_node node represents the newly obtained sampling point, Dalte is the search radius of the node searching for obstacles, and Dis is the distance from the sampling point to the target point. As shown in Figure 5(a), once the distance Dis between the New_node and the target point is less than that of Dalte, the original algorithm is judged to find the target point. After planning the first path, the algorithm stops to enter the path optimization process. As shown in Figure 5(b), the improved Informed-RRT* algorithm introduced with the greedy algorithm determines whether the target point is found depending on whether the New_node can be directly connected to the target point.

3.2. Optimization of Search Objects. In the path optimization process of the Informed-RRT* algorithm, the search object for the potential optimal parent node is the node tree formed in the path planning process. The number of possible optimal parent nodes is only few, and the detection process requires detecting all nodes, with a problem of inefficient path planning caused by detecting redundant nodes. This paper considers replacing the search object for searching for potential optimal parent nodes from the node tree to the path.

As shown in Figure 6 (lines 19–27), we define the planned path to be optimized as a path and sample the point in the space as Xrand (line 19). If it is detected that the Xrand point is on the block, the node is discarded, and the iteration ends. If the Xrand point is not on the obstacle, we search for the Xnearest node closest to the Xrand node in the path list (line 20). After finding Xnearest, it is recorded as path(c), the distance from Xrand to the path(c) node is calculated, and it is recorded as delta. If the delta is greater than the step size, the length of SizeStep along the direction from path(c) to Xrand is cut, and one new point is selected, which is denoted as Xnew. If the delta is less than SizeStep, the Xrand node is the node Xnew to be updated (line 21). After obtaining the Xnew node, it is checked whether path(c) is at the head or tail of the path list. If it is at the head or tail of the path list, Xmin is empty; if not, Xmin is not empty (line 22). As shown in Figure 7, if Xmin is not empty and the length path(b)-Xnew-path(d) is less than the length path(b)-path(c)-path(d), the path(c) node in path is replaced with the Xnew node and rewired to obtain the new path. Then, Cbest is recalculated to reduce further the sampling space (lines 24 to 25).

4. Dynamic Window Approach (DWA)

When the robot moves along the path planned by the global path planning algorithm, the local path planning algorithm is needed to perform the dynamic obstacle avoidance function. The method used in this paper to the plane local path is the dynamic window approach (DWA). The dynamic window method is a local path planning algorithm based on robot dynamics. The algorithm relies on the speed that the robot can reach in a period to form a dynamic window for local path planning. The dynamic window method searches for the linear velocity v and angular velocity w that the robot can reach under the constraint conditions in the dynamic window at all times. Each group's velocities and angular velocities (v, w) are simulated to produce the robot's trajectory within a certain period, evaluate, and select the best trajectory.

4.1. Dynamic Window Approach Kinematics Model. First, it is assumed that the linear velocity v and angular velocity w of the robot in the dynamic window can be independently controlled. Because the dynamic window approach uses instantaneous speed, the robot's trajectory at two adjacent moments in the moving process can be regarded as a straight line. Therefore, the posture state of the robot in the working environment at adjacent moments can be expressed by the following equation:

$$\begin{cases} \Delta x = v * \Delta t * \cos(\varphi_t), \\ \Delta y = v * \Delta t * \sin(\varphi_t). \end{cases} \tag{3}$$

From equation (3), the pose change of the robot can be obtained in the time of

$$\begin{cases} x = x + v * \Delta t * \cos(\varphi_t), \\ y = y + v * \Delta t * \sin(\varphi_t), \\ \varphi_t = \varphi_t + w\Delta t. \end{cases} \tag{4}$$

4.2. Dynamic Window Speed Sampling. During the movement of the robot, the DWA calculates the speed at which the robot reaches the target point in the two-dimensional space of (v, w). However, the speed of the robot is restricted by the corresponding constraint:

(i) The restriction of the maximum and minimum speed of the mobile robot itself

(ii) The motor performance restricts the moving speed of the mobile robot

(iii) Environmental factors limit the robot's moving speed

The speed sampling space is shown in equation (5). V_s represents the speed space obtained by the mobile robot under its maximum and minimum speed constraints, V_d represents the speed space obtained by the mobile robot under the motor performance constraints, and V_a represents the speed space obtained under the constraints of factors from the environment the robot moves.

$$V_r = V_s \cap V_d \cap V_a. \tag{5}$$

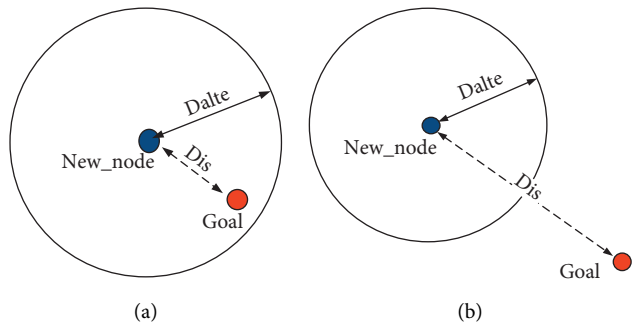

Figure 5: Schematic diagram of the different search targets. (a) The original algorithm search target; (b) the improved algorithm search target.

```
0  V = {Xinit} ,E = Ø,T = (V,E),Cmin =dis(Nstart,Ngoal),
   Cbest = 0,goal = false,StepSize=0.05 ;
1  While()
2      If ~goal
3          for I 0 to N do
4              Xrand ← Sample() ;
5              Xnearest ← NearestVertices(Xrand,T) ;
6              Xnew ← Steer(Xnearest,Xrand,StepSize) ;
7              Ls ← NearVertices(Xnew,T,StepSize) ;
8              Xmin ← ChooseBestParent(Xnew,Ls,T) ;
9              if Xmin = Ø then
10                 T ← InserVertex(Xnew,Xmin,T) ;
11                 T ← RewireVertices(Xnew,T) ;
12             goal ← CheckGotogoal(Xnew,Nodegoal)
13             if goal
14                 Cbest ← dis(Nstart,Ngoal,T) ;
15             return Cbest ;
16         return path ← path(Nstart,Ngoal,T) ;
17     If goal
18         for I 0 to N do
19             Xrand ← Sample(Cmin,Cbest) ;
20             Xnearest ← NearestVertices(Xrand,path) ;
21             Xnew ← Steer(Xnearest,Xrand,StepSize) ;
22             Xmin ← ChooseBestParent(Xnew,path) ;
23             if Xmin = Ø then
24                 path ← InserVertex(Xnew,Xmin,path) ;
25                 path ← RewireVertices(Xnew,path) ;
26             Cbest ← dis(Nstart,Ngoal,path) ;
27             return Cbest ;
28         return path ;
```

Figure 6: Improved Informed-RRT* algorithm pseudocode.

4.3. Dynamic Window Approach Trajectory Evaluation Function. By sampling the speed, the DWA algorithm can obtain several sets of speeds for obstacle avoidance.

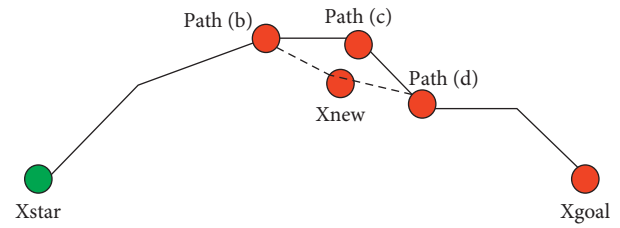

Figure 7: Schematic diagram of the optimized path by improved Informed-RRT*.

Therefore, a corresponding evaluation function is needed to evaluate the obtained path trajectory, and the optimal motion trajectory is selected to guide the robot to avoid obstacles dynamically. The trajectory evaluation function is expressed by the following equation:

$$G = \sigma(\alpha \cdot h(\,) + \beta \cdot d(\,) + \gamma \cdot v(\,))\cdot \tag{6}$$

As shown in Figure 8, the $h()$ in equation (5) is used to evaluate the angle difference between the robot and the target pose after reaching the position at the current speed. The score is judged based on the angle value of θ. A higher score is obtained when the value is larger, to ensure that the mobile robot moves in the target direction. $d()$ represents the distance from the nearest obstacle on the current trajectory of the mobile robot. If there is no obstacle on the current trajectory, $d\,()$ is set as a constant. $v()$ keeps the robot moving at a reasonable speed. α, β, and γ represent the weight coefficients. Finally, the evaluation function is obtained after the data are normalized, and the appropriate movement speed is selected to make the robot move.

5. Simulation Experiment

To verify the effectiveness of the improved algorithm, the investigation uses the ROS software platform to carry out the robot autonomous navigation simulation experiment. The experimental simulation area is a rectangular area of 17 m * 13.5 m, the starting point is set as (2 m, 2 m), and the endpoint is set as (15 m, 6 m). The boxes in the figure represent obstacles in the environment. The experimental results are compared in three aspects: the length of the planned path, the time of planning the path with the two Informed-RRT* algorithms before and after the

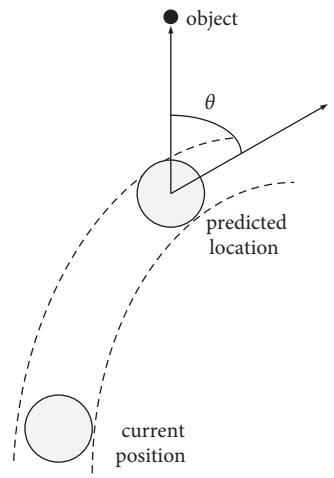

Figure 8: $h()$ evaluation.

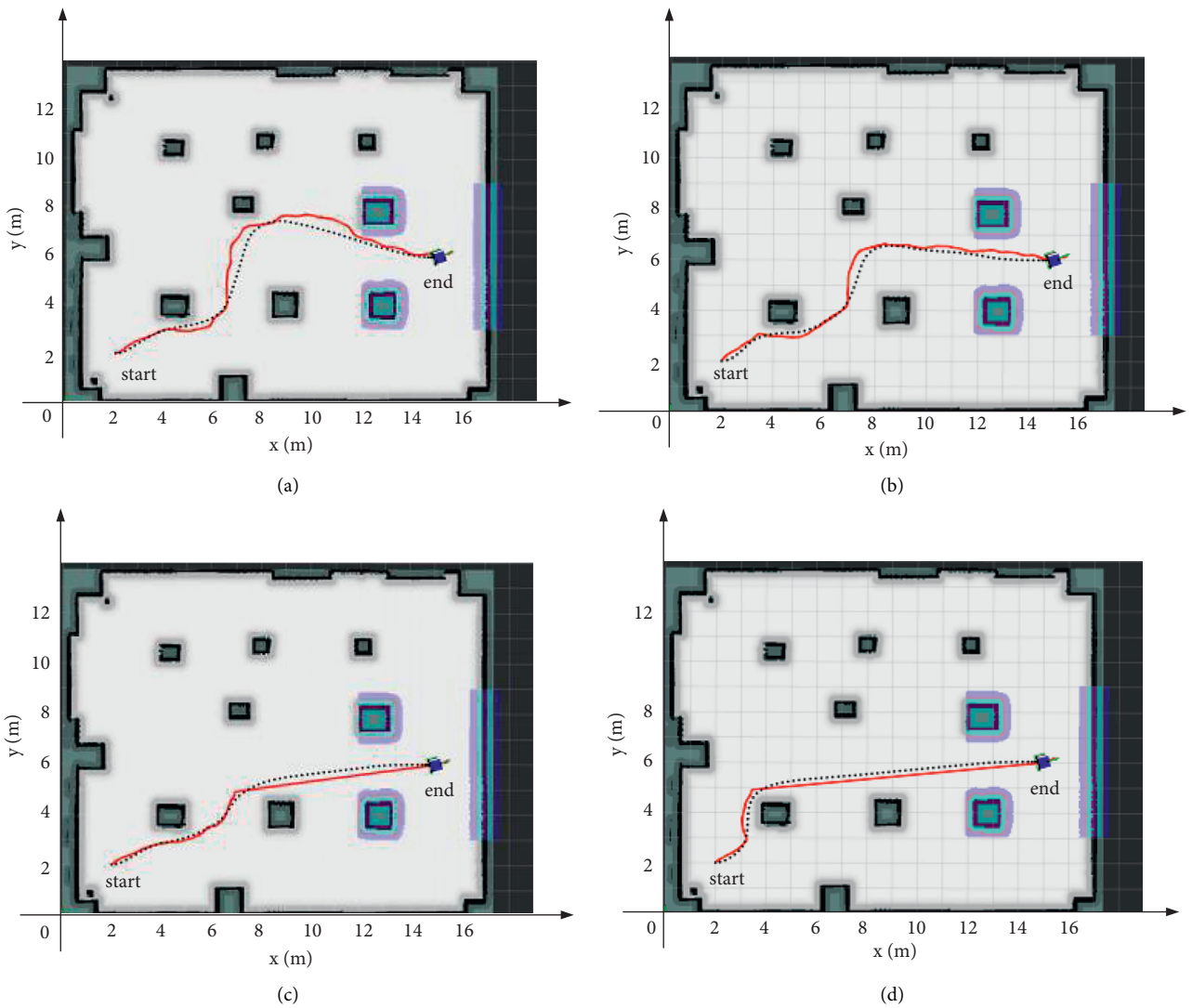

Figure 9: Autonomous navigation maps based on the two Informed-RRT* algorithms before and after the improvement.

Table 1: Experimental results of path length planning in raster maps using the improved Informed-RRT* algorithm.

	Informed-RRT* algorithm		Improved Informed-RRT* algorithm	
	Path length/m	Average length (m)	Path length/m	Average length (m)
1	15.9120		14.4086	
2	14.5151		15.0866	
3	15.1106	15.7334	15.5101	14.9216
4	15.9320		15.2948	
5	17.1973		14.3080	

Table 2: Experimental results of the time consumption of path planning in raster maps with and without the improved-RRT* algorithm.

	Informed-RRT* algorithm		Improved Informed-RRT* algorithm	
	Path planning time (s)	Average time (s)	Path planning time (s)	Average time (s)
1	11.39		2.88	
2	10.41		2.50	
3	13.44	12.19	2.91	2.64
4	10.81		2.45	
5	14.91		2.46	

Table 3: The experimental results of the autonomous navigation time of the robot in the raster map with and without the improved Informed-RRT* algorithm.

	Autonomous navigation based on Informed-RRT* algorithm and DWA		Autonomous navigation based on improved Informed-RRT* algorithm and DWA	
	Autonomous navigation time (s)	Average navigation time (s)	Autonomous navigation time (s)	Average navigation time (s)
1	51.04		40.53	
2	50.16		41.98	
3	53.01	52.59	42.53	41.19
4	52.11		40.29	
5	56.66		40.63	

improvement, and the time consumed by the simulation robot autonomous navigation based on the two path planning algorithms and DWA before and after the improvement. The time of planning a path represents the time from receiving a command to the start of movement of the simulated robot; the time of autonomous navigation of the robot represents the time taken by the simulated robot from receiving a command to the moment at the endpoint.

Figure 9 shows the robot's autonomous navigation results in the grid map based on the two Informed-RRT* path planning algorithms before and after the improvement. Figures 9(a) and 9(b) show the schematic diagram of the robot autonomous navigation experiment based on the Informed-RRT* algorithm and DWA, and Figures 9(c) and 9(d) show the schematic diagram of the Improved Informed-RRT* algorithm and DWA root autonomous navigation. The solid line in the figure represents the global path planned by the two Informed-RRT* algorithms before and after the improvement, and the dashed line represents the actual trajectory of the robot in the experimental environment under the action of the global path and the DWA local path planning algorithm.

Table 1 shows the length of the path planned by the two path planning algorithms in the experimental environment before and after the improvement. Experiments 1, 2, 3, 4, and 5 in the table represent autonomous navigation experiments based on the two Informed-RRT* algorithms before and after the improvement. In the five experiments, the length of the global path planned by the Informed-RRT* algorithm is 15.9120 m, 14.5151 m, 15.1106 m, 15.9320 m, and 17.1973 m, and the average length is 15.7334 m; the improved Informed-RRT* algorithm plans the length of the global path as 14.4086 m, 15.0866 m, 15.5101 m, 15.2948 m, and 14.3080 m, and the average length is 14.9216 m. Experimental results show that the length of the path planned by the improved Informed-RRT* algorithm is 5.15% shorter than that of the path planned by the original algorithm.

Table 2 shows the time consumed by the two path planning algorithms in five autonomous navigation experiments before and after the improvement. In the five experiments, the Informed-RRT* algorithm planned the global path consumption time, and the average consumption time was 12.19 s; the improved Informed-RRT* algorithm planned the overall time consumed by the path, and the average consumption time was 2.64 s. The experimental results show that the time consumed by the improved Informed-RRT* algorithm for path planning is reduced by 78.34% compared with the original algorithm.

Table 3 shows the time consumed by the simulated robot based on the two path planning algorithms and DWA before and after the improvement in five autonomous navigation experiments. In the five experiments, the simulation robots

based on the Informed-RRT* algorithm and DWA took 51.04 s, 50.16 s, 53.01 s, 52.11 s, and 56.66 s to complete the autonomous navigation, respectively, and the average consumption time was 52.59 s; based on the improved Informed-RRT* algorithm, the time consumed by the RRT*algorithm and the DWA simulation robot to complete autonomous navigation is 40.53 s, 41.98 s, 42.53 s, 40.29 s, and 40.63 s, and the average time consumed is 41.19 s. The experimental results show that the simulation robot based on the improved Informed-RRT* algorithm and the DWA algorithm consumes 21.67% less time to complete autonomous navigation than the simulation robot based on the original algorithm and DWA.

6. Conclusions

In this paper, the problems existing in the application of the Informed-RRT* path planning algorithm in the process of robot autonomous navigation are optimized from two aspects: path planning and path optimization. The simulation robot autonomous navigation experiment on the ROS platform shows that the improved Informed-RRT* algorithm is better than the original algorithm in both the length of the planned path and the time of the planned path, and the improved algorithm can effectively improve the efficiency of the robot's autonomous navigation. As a nonlinear system, the search mode with fixed parameters will inevitably fall into a local optimum and may stop and wander at nontarget points. Therefore, the path planning method has a large space for improvement.

Data Availability

Some or all data, models, or code generated or used during the study are available from the corresponding author on request.

Conflicts of Interest

The authors have no conflicts of interest.

Authors' Contributions

All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Dai Jun, Li Dongfang, Zhao Junwei, and Li Yanqin. The first draft of the manuscript was written by Dai Jun, and all authors commented on the previous versions of the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This work was partially supported by the Key scientific research projects of colleges and universities in Henan Province (No. 22A460020),University-Industry Collaborative Education Program (Nos. 201901202002 and 202102558013), the Henan Postdoctoral Science Foundation (No.166182), and the Startup Foundation for PhD of Henan Polytechnic University (No.660407/023).

References

[1] S. Gagliano, G. Stella, and M. Bucolo, "Real-time detection of slug velocity in microchannels," *Micromachines*, vol. 11, no. 3, p. 241, 2020.

[2] H. Fang, S. Ong, and A. Nee, "Robot path planning optimization for welding complex joints," *International Journal of Advanced Manufacturing Technology*, vol. 90, no. 9–12, pp. 3829–3839, 2017.

[3] J. Hong and K. Park, "A new mobile robot navigation using a turning point searching algorithm with the consideration of obstacle avoidance," *International Journal of Advanced Manufacturing Technology*, vol. 52, no. 5–8, pp. 763–775, 2011.

[4] Y. Y. Shi, Q. Q. Li, and S. Q. Bu, "Research on intelligent vehicle path planning based on rapidly-exploring random tree," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5910503, 14 pages, 2020.

[5] L. Z. Zhang, Z. B. Lin, J. Wang, and B. W. He, "Rapidly-exploring Random Trees multi-robot map exploration under optimization framework," *Robotics and Autonomous Systems*, vol. 131, 2020.

[6] M. Hamzheei, R. Z. Farahani, and H. Rashidi-Bajgan, "An ant colony-based algorithm for finding the shortest bidirectional path for automated guided vehicles in a block layout," *International Journal of Advanced Manufacturing Technology*, vol. 64, no. 1-4, pp. 399–409, 2013.

[7] Q. Zhou and Y. Zheng, "Long link wireless sensor routing optimization based on improved adaptive ant colony algorithm," *International Journal of Wireless Information Networks*, vol. 27, no. 2, pp. 241–252, 2020.

[8] G. N. Ambewadkar and S. P. Gajre, "Probe path planning for flatness measurement on coordinate measuring machine using ant colony optimization," *Advanced Engineering Forum*, vol. 41, pp. 86–91, 2021.

[9] M. Fromeide and A. Hansen, "Predicting motion patterns using optimal paths," *Frontiers in Physics*, vol. 9, p. 344, 2021.

[10] M. Li, F. Zhang, and J. Y. Fang, "Optimal path solution based on Dijkstra algorithm," *Frontiers in Economics and Management*, vol. 2, pp. 170–176, 2021.

[11] X. Lai, J. Li, and J. Chambers, "Enhanced center constraint weighted A* algorithm for path planning of petrochemical inspection robot," *Journal of Intelligent and Robotic Systems*, vol. 102, no. 4, p. 78, 2021.

[12] J. Santos, P. M. Rebelo, L. F. Rocha, P. Costa, and G. Veiga, "A* based routing and scheduling modules for multiple AGVs in an industrial scenario," *Robotics*, vol. 10, no. 2, p. 72, 2021.

[13] A. Leonardo, A. Víctor, and S. Jorge, "Path planning based in algorithm rapidly-exploring random tree RRT," *Advanced Science Letters*, vol. 24, no. 6, pp. 8831–8836, 2018.

[14] S. Karaman, M. R. Walter, and A. Perez, "Anytime motion planning using the RRT*," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1478–1483, Shanghai, China, May 2011.

[15] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2997–3004, Chicago, IL, USA, September 2014.

[16] H. Ryu and Y. Park, "Improved informed RRT* using grid-mapskeletonization for mobile robot path planning," *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 11, pp. 2033–2039, 2019.

[17] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmedy, and I. Ali, "Informed RRT*-Connect: an asymptotically optimal single-query path planning method," *IEEE Access*, vol. 8, pp. 19842–19852, 2020.

[18] F. Duchoň, A. Babinec, M. Kajan, and P. Beňo, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[19] L.-S. Liu, J.-F. Lin, J.-X. Yao et al., "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," *Wireless Communications and Mobile Computing*, vol. 2021, no. 4, 12 pages, Article ID 8881684, 2021.

[20] X. Zhang, J. Lai, D. Xu, H. Li, and M. Fu, "2Dlidar-based SLAM and path planning for indoor rescue using mobile robots," *Journal of Advanced Transportation*, vol. 2020, no. 3, 14 pages, Article ID 8867937, 2020.

[21] M. Bucolo, A. Buscarino, C. Famoso, L. Fortuna, and M. Frasca, "Control of imperfect dynamical systems," *Nonlinear Dynamics*, vol. 98, no. 4, pp. 2989–2999, 2019.