Journal of Physics: Conference Series

PAPER • OPEN ACCESS

MI-RRT-Connect Algorithm for Quadruped Robotics Navigation with Efficiently Path Planning

MI-RRT-Connect Algorithm for Quadruped Robotics Navigation with Efficiently Path Planning

Yulong Zhang [1], Haoyu Jiang [1], Xungao Zhong [1,*], Xunyu Zhong [2], Jing Zhao [1]

[1]Electrical Engineering and Automation, Xiamen University of Technology, Xiamen, China

[2]School of Aerospace Engineering electrical Engineering, Xiamen University, Xiamen, China

E-mail: {zyl_5206, zhongxungao}@163.com, leaf0228@yeah.net,

zhongxunyu@xmu.edu.cn, ztulipwork@139.com

*Corresponding author: zhongxungao@163.com

Abstract- Autonomous navigation is playing an increasingly important role in quadruped robotic systems. However, providing safe and reliable path planning for robots is still an open problem. In this paper, we propose a sampling-based path planning algorithm fused with a dual-tree structure, here called Multiple Informed RRT-Connect (MI-RRT-Connect). The proposed MI-RRT-Connect can overcome the disadvantage that the Informed RRT* algorithm takes a long time in the initial search path, by using the RRT-Connect algorithm and the target deviation method, and the initial search path can be obtained quickly. Then, for the optimization of the initial search path, the multi-level parent node selection strategy and the method of calculating the path cost for the parent node are used. The simulation and quadruped robot experimental results show that our method can find an optimal path in a shorter time, and it has been well applied in a real quadruped robot.

1. INTRODUCTION

The discrete foot positions of the footed robot can be applied to different terrain transformations, so it has stronger terrain adaptability and more flexible maneuverability than crawler and wheeled robots, [1], [2], [3]. When designing autonomous navigation functions for quadruped robots in different environments, it must provide safe and reliable path planning.

The current research on the plan-planning methods of legged robots in autonomous navigation tasks mainly focuses on foothold planning and torso-based planning methods. Foothold planning considers the foot placement of each leg as the legged robot moves in complex environments [4], [5], and the Torso-based planning method will avoid complex robot dynamics modeling and separate planning and control problems. The existing Torso-based works using RRT search [6] and A* search [7] for the development of path-planning algorithms for legged robots.

Sampling-based methods are usually built on the RRT* algorithm [8] or introduce heuristic search methods [9] for global path planning of legged robots. When path planning has to go through a narrow channel to find a solution, RRT*-Connect [10] proposes a method to combine RRT-Connect with RRT* to obtain a bidirectional path close to the optimal solution. KB-RRT* [11] proposes a new pruning method, which finds the optimal parent node by expanding the state set of the parent node and maintaining the kinematic characteristics, so as to obtain the optimal solution. However, all of the above methods need to randomly search the entire environment space, which is not an effective method to reduce costs. After obtaining the initial search path, Informed RRT*[12] restricts the search area to a subset of the ellipsoid space to iteratively obtain a better global path. Li [13] uses the idea of a sliding window to cut the ellipsoid space subsets with the start point and the target point as the focus, and perform path optimization in the small space subsets after cutting. Informed RRT*-Connect [14] uses RRT*-Connect to solve the initial solution, after which the path is optimized in a subset of the ellipsoid space.

In this study, a path planning algorithm of MI-RRT-Connect (Multiple Informed RRT-Connect) is proposed to overcome some challenges brought by the Informed RRT* method and obtain the optimal collision-free path. The whole path planning process is divided into two stages, solving the initial search path and optimizing the initial search path. First, an initial path solution is performed using a bidirectional random search tree algorithm (RRT-Connect) with multiple heuristic sampling strategies. In the optimization stage, a multi-level parent node selection strategy is designed, and a cost function is set to reduce unnecessary node collision detection calculations, which can shorten the optimization time. Finally, environments and simulations to well validate the effectiveness of the proposed algorithm in the autonomous navigation of quadruped robots.

2. PROBLEM DESCRIPTION

In a two-dimensional space, to design a global path for the autonomous navigation of a quadruped robot, the roll, and pitch of the robot can be ignored, and only forward-backward, lateral, and yaw motions are considered. And it is assumed that the quadruped robot knows the map environment information. $X \subset \mathbb{R}^n$ represents the global environment space, $X_{obs} \subset X$ represents the space where the robot collides with the obstacle, $X_{free} \subset X$ represents the space in which the robot moves freely, $x_{start} \subset X_{free}$ represents the start position of the robot, $x_{goal} \subset X_{free}$ and represents the target position of the robot. $\sigma:(0,1) \rightarrow X_{free}$ is represented as a collision-free path $C(x)$ is represented as the cost function of the collision-free path. When $C(x)$ taking the minimum value, the optimal path is Eq. (1).

$$\sigma = \arg\left\{C(\sigma) \mid \sigma(0) = x_{start}, \sigma(1) = x_{goal}, \forall s \in [0,1], \sigma(s) \in X_{free}\right\} \tag{1}$$

A random tree is represented as $T = (V, E)$. The nodes in the random tree are represented as $V \in X_{free}$. The edges $E \in X_{free}$ are represented as connections between nodes in a random tree.

3. INFORMED-RRT* AND RRT-CONNECT ALGORITHMS

3.1. Pseudocode of the Informed-RRT* Algorithms

Algorithm 1 shows the pseudocode for Informed RRT*. The algorithm proposes parent node reselection, random tree rerouting, and sampling area restriction strategies.

Algorithm 1 Informed RRT*
Input: x_{start} , x_{gool}
Output: *path*
1: $T = \{V, E\}; V = \{x_{start}\}; E = \varnothing; X_{soln} = \varnothing$;
2: **while** $i < N$ **do**
3: $c_{best} \leftarrow \min_{x_{soln}} \in X_{soln}\{c(x_{soln})\}$;
4: $x_{rand} \leftarrow InformedSample(x_{start}, x_{goal}, c_{best})$;
5: $x_{new} \leftarrow Steer(x_{near}, x_{rand}, T)$;
6: **if** $ObstacleFree(x_{nearest}, x_{new},)$ **then**

7: $U_{near} \leftarrow Near(x_{new}, T, T_{near})$;
8: $c_{\min} \leftarrow c(x_{nearest}) + Distance(x_{nearest}, x_{new})$;
9: **for** $\forall x_{near} \in U_{near}$ **do**
10: $c_{new} \leftarrow c(x_{near}) + Distance(x_{near}, x_{new})$;
11: $x_{new} \leftarrow ChooseBestParent(c_{\min}, c_{new}, x_{near}, x_{new})$;
12: **end for**
13: $E \leftarrow E \cup \{(x_{\min}, x_{new})\}$
14: **for** $\forall x_{near} \in U_{near}$ **do**
15: $c_{near} \leftarrow c(x_{near})$
16: $c_{new} \leftarrow c(x_{near}) + Distance(x_{near}, x_{new})$;
17: $T \leftarrow RewireEdge(x_{near}, x_{new}, T)$;
18: **end for**
19: $X_{soln} \leftarrow InGoolRegion(x_{new}, X_{soln})$;
20: **end if**
21: **end while**
22: **return** T ;

Parent node re-selection strategy (Alg. 1, Line 7 and 11). Constructs a circular area with a new node x_{new} as the center and r as the radius. Create a new set $U_{x_{parent}}$ of candidate parent nodes x_{new} in the circular area. Consider the establishment of a new feasible edge between $U_{x_{parent}}$ and x_{new}, and certain conditions must be met to insert the new edge into the edge set E. In particular, create an edge from the nodes in $U_{x_{parent}}$ that can be connected to x_{new} along a path with minimum cost.

Rerouting strategy (Alg. 1, Lines 7 and 17), by traversing the feasible edges with x_{new} in $U_{x_{parent}}$. There are also other conditions that need to be met before the new edge can be inserted into the edge set E. If the path through x_{new} costs less than the path through the current parent, then the edge that connects the new node to the current parent is removed, and add a new edge to the random tree.

As the number of iterations increases, the ellipse space continues to shrink (Alg. 1, Lines 3-4), and the path length also decreases, and finally, the optimal path is obtained. The definition of an ellipse sampling subset is shown in Fig. 1.

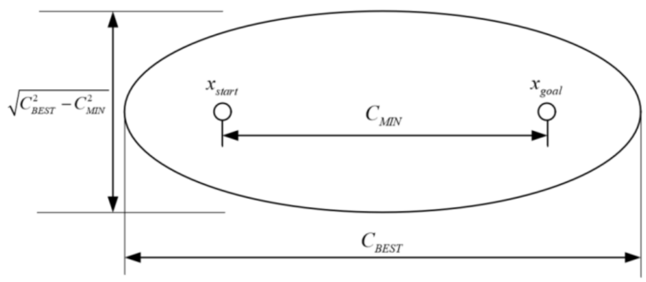


Figure 1. Ellipse sampling subset. The start position and the target position as the focus of the ellipse. The C_{BEST} represents the long axis and its value is the search path length. The C_{MIN} represents the focal length.

3.2. Pseudocode of the RRT-Connect Algorithms

Algorithm 2 shows the pseudocode of the RRT-Connect algorithm. The algorithm proposes to build two random trees T_1 and T_2. In the process of each iteration, T_1 is preferentially expanded, and the new node x_{new} added in T_1 is targeted to expand T_2. Connect function (Alg. 2, Line 4) to link T_2 and x_{new} of T_1. If there are no obstacles blocking the connection, it means that the two trees have been connected, the search process is completed, and a global path is obtained.

Algorithm 2 RRT - Connect

Input: x_{start} , x_{goal}

Output: $path$

1: $T_1 = \{V_a, E_a\}; T_2\{V_b, E_b\}; V_a = \{x_{start}\}; V_b = \{x_{goal}\}; E_a = \varnothing; E_b = \varnothing;$
2: **while** $i < N$ **do**
3: $x_{rand} \leftarrow Sample()$;
4: **if** $Extend(T_1, x_{rand}) \neq ObsCollison$; **then**
5: **if** $Connect(T_2, x_{new}) = \mathrm{Re}ached$; **then**
6: **return** $x_{rand} \leftarrow Sample()$;
7: **end if**
8: $Swap(T_1, T_2)$;
9: **end if**
10: **end while**
11: **return** $path = \varnothing$

4. ALGORITHM IMPROVEMENT STRATEGIES

4.1. Multiple Heuristic Sampling

The Informed RRT* algorithm adopts random uniform sampling in solving the initial search path, and the search direction is completely random, which affects the search efficiency. In response to this problem, the MI-RRT-Connect algorithm introduces the RRT-Connect algorithm and uses multiple heuristic sampling strategies. This method can balance the randomness and goal orientation of sampling, and improve the initial path solution speed. The algorithm pseudocode is as follows Algorithm 3.

- Set the probability P_c to select the center of the circle with the midpoint of the line connecting the start position and the target position, and the circular area formed by the radius r_{center} as the sampling area.
- Set the probability P_a to select the circle with the target position as the center and the circle formed by the radius r_{center} as the sampling area. And set the probability P_g to select the target position as the sampling point.
- Set the probability P_b to randomly and uniformly sample the environment space.

Algorithm 3 MultipleSample

Input: $Rate$, P_g

Output: x_{rand}

1: $prandom \leftarrow rand(0,100)$
2: **if** $prandom < Rate$ **then**
3: $x_{rand} \leftarrow \left[rand\left(map_{\min}, map_{\max}\right), rand\left(map_{\min}, map_{\max}\right)\right]$
4: **else if** $\{Rate < prandom < \alpha Rate\}$ **then**
5: **if** $prandom \leftarrow rand(0,100) < p_g$ **then**
6: $x_{rand} \leftarrow x_{goal}$
7: **else**
8: $x_{rand} \leftarrow x_{goal} + r_{center}\left(\cos\left(2\pi N\right), \sin\left(2\pi N\right)\right)$
9: **else**
10: $x_{rand} \leftarrow x_{center} + r_{center}\left(\cos(2\pi N), \sin\left(2\pi N\right)\right)$
11: **return** x_{rand}

The sum of the above probabilities is equal to 1, where $P_a + P_b + P_c = 1$, where α is a constant, where N is selected from 0 to 1. $map_{\min}$ and $map_{\max}$ are the minimum and maximum values of the map edges, respectively.

4.2. Multi-level parent node reselection strategy

The Informed RRT* algorithm optimizes the path length by reconnecting the first-level parent nodes. However, when there are few obstacles in the environment space, only the first-level parent nodes are found to reconnect with the new node, which reduces the utilization of the nodes; When there are many obstacles in the environment space, expanding the search range of the parent nodes will cause the algorithm to spend a lot of computing time on node collision detection. Therefore, this paper proposes an extended multi-level parent nodes reconnection strategy, and probabilistically eliminates some selected parent nodes that do not help improve the path. The algorithm pseudocode is as follows Algorithm 4.

Algorithm 4 MultipleNear

Input: $T = \{V, E\}, x_{new}, r_{near}, P_g$
Output: T
$U_1 \leftarrow Near(x_{new}, T, r_{near})$;
$c_{\min} \leftarrow c(x_{nearest}) + Distance(x_{nearest}, x_{new})$;
$U_2 \leftarrow FindParent(U_1)$
$U_3 = \{x_{start}, x_{goal}\}$;
$U_{all} \leftarrow PrunDate(U_1, U_2, U_3)$;
$U_{useful} \leftarrow HeuristicFunction(U_{all}, x_{new}, T)$;
for $\forall x_{near} \in U_{useful}$ **do**
  $c_{new} \leftarrow c(x_{near}) + Distance(x_{near}, x_{new})$;
  $x_{new} \leftarrow ChooseBestParent(c_{\min}, c_{new}, x_{near}, x_{new})$;
end for
for $\forall x_{near} \in U_{useful}$ **do**
  $c_{near} \leftarrow c(x_{near})$;
  $c_{new} \leftarrow c(x_{near}) + Distance(x_{near}, x_{new})$;
  $T \leftarrow RewireEdge(x_{near}, x_{new}, T)$;
end for
return T;

After adding a new node x_{new} to the random tree T, traverse to find out the set of all nodes whose distance is less than r_{near}, and record it as a set U_1. Create a set U_2 to record parent nodes in U_1. Create a set U_3 with the start position x_{start}. Create a set U_{all} that contains the U_1, U_2, U_3. The final sampling set is Eq. (2).

$$\begin{aligned} U_1 &= \{x_{near}\} \\ U_2 &= \{x_{near_{parent}}\} \\ U_3 &= \{x_{start}\} \\ U_{all} &= \{U_1, U_2, U_3\} \end{aligned} \tag{2}$$

The selection formula for the radius r_{near} is defined as Eq. (3), where *d* represents the dimension of the state space, *n* represents the number of nodes in *T*, β represents a parameter determined by the environment space.

$$r_{near} = \beta \left[\frac{(\log n)}{n} \right]^{1/d} \tag{3}$$

Using the merit function calculates the probability that each node in the U_{all} will be rejected. A node whose probability value is less than the threshold value δ_t will be eliminated. At the same time, create a set U_{useful} to record the nodes that are not removed. These removed nodes are usually very close to the new node x_{new}, but there is no obvious effect on path optimization. The merit function is Eq. (4) and Eq. (5).

$$\gamma = 2\eta(1+\frac{1}{d})^{\frac{1}{d}}(\frac{\lambda(U_{all})}{\zeta_d})(\frac{\log(N)}{N})^{\frac{1}{d}} \tag{4}$$

$$P(x) = |1 - \frac{1}{e^{\tan[\frac{d}{\gamma}\cdot(\frac{\pi}{2}-|\varepsilon|)]}}|, d = \| x_{new} - x \|_2, x \in \{U_{all}\} \tag{5}$$

The parameter γ is an adjustment parameter in the model based on r-disc RGG (random geometric graph), $\eta \geq 1$ represents the adjustment variable [15], $\lambda(\cdot)$ represents the Lebesgue measure of the set, ζ_d represents the Lebesgue measure of the d-dimensional unit sphere, N represents the number of nodes in U_{all}, ε represents an arbitrary small quantity.

Algorithm 4 selects the parent node process as shown in Fig. 2 (a). A-B-D-K is the original path and K is the node to be optimized. The node E is removed from the result of the merit function calculation and the parent node set $U_{useful} = \{U_1 = \{\mathrm{D,E,H,J}\}, U_2 = \{B,C,G,I\}, U_3 = \{A\}\}$ about K is obtained. Calculate the path cost connecting the nodes in U_{useful} to node K and get the new path A-F-G-K with the minimum path cost, which is shown in Fig. 2 (b).

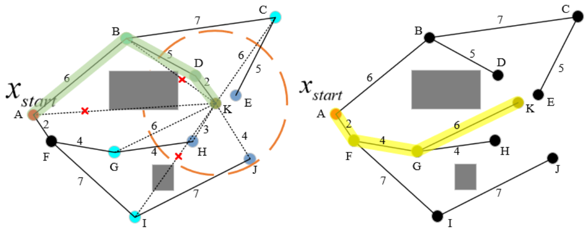


Figure 2. Reselect the parent node process. The green line is the original path, and the yellow line is the new path after modifying the parent node.

The theoretical analysis is as follows. Assuming that in the two-dimensional space, there are nodes n in the T, n has an average of $m = \{m \in U_{useful} \mid m\}$ parent nodes, and the cost of each calculation is 1. When n approaches positive infinity, we can get $\gamma \to 1$. The calculation cost of Algorithm 2 is $\Delta_{old} = n \times (m-k), k < m$, while the calculation cost of Algorithm 4 is $\Delta_{new} = n \times m \times (1-P), P \in (0,1)$. Comparing Δ_{old} and Δ_{new} approaching positive infinity, the calculation cost formula is Eq. (6). The formula shows that as the sample size increases, the computational cost of Algorithm 4 is much less than that of Algorithm 2.

$$\lim_{n \to +\infty, m \to +\infty} \frac{\Delta_{old}}{\Delta_{new}} = \infty \tag{6}$$

5. EXPERIMENT RESULTS

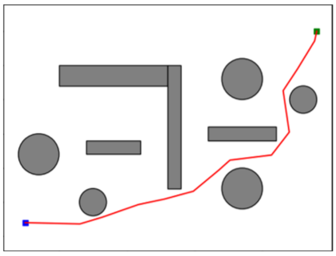
5.1. Simulation

To validate the advantages of our proposed algorithm, we design a series of simulations to test the performance of the algorithm in several different scenarios. Because the simulation in this paper is performed in a two-dimensional state space, we may ignore geometry and dynamics. In the simulation, we create various environments and establish the corresponding environment parameters as shown in Table 1.

Table 1 Simulation map parameters

Scene	Map Size	Start position	End position	Iterations
Simple	50*35	(5,2)	(48,30)	2000
Complex	50*35	(5,2)	(48,30)	2500
Maze	50*35	(5,2)	(48,30)	3000
Narrow	50*45	(5,2)	(48,30)	2000

In the first simulation, we set up a simple map to compare the efficiency of Informed-RRT*, Informed RRT-Connect, and MI-RRT-Connect in finding the initial path. Fig. 3 shows the initial path obtained by MI-RRT-Connect on a simple map. Fig. 4 shows the average of the total time and iterations obtained by running the three algorithms 50 times in a simple map. Compared with the other two algorithms, the time it takes for MI-RRT-Connect to find the initial path is 0.168 s and the number of iterations is 518, which are both lower than the other two algorithms, so MI-RRT-Connect can improve the convergence speed of the initial path.



Simple Map

Figure 3. Simple map path search performance, where the blue, green, and red points are the start position, the target position, and the initial path.

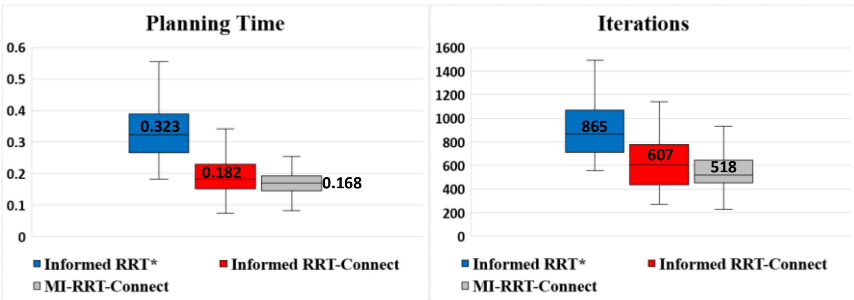


Figure 4. Reselect the parent node process. The green line is the original path, and the yellow line is the new path after modifying the parent node.

In the second simulation, the optimal paths generated by the three algorithms across different maps are compared, as shown in Fig. 5. For the given same map, compare the average values obtained after each of the three algorithms runs 50 times, including the length of the optimal path and the total planning time, and the results are shown in Fig. 6.

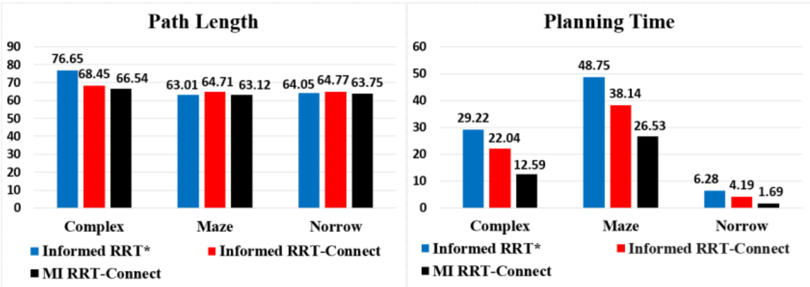


Figure 5. Results of the time and path lengths taken by the three algorithms to generate the optimal path.

Focusing on the blue cross point in Fig. 6 (a), (b), and (c), it shows that the algorithm generates the optimal path consisting of these nodes. Compared with the elliptical area in the same map, the MI-RRT-Connect algorithm uses fewer nodes to optimize the path, and it can be observed that the path in this area is simpler and smoother, which is attributed to the multi-level parent node reselection strategy and the node cost function that eliminates the nodes with poor optimization effect on the overall path.

Furthermore, Fig. 6 shows that the MI-RRT-Connect algorithm takes 12.59 s, 26.53 s, and 1.69 s to generate the optimal path on the three maps, respectively. Meanwhile, the resulting path lengths are 66.54 m, 63.12 m, and 63.75 m. From these results, it can be seen that the MI-RRT-Connect algorithm shows better performance compared to the other two algorithms.

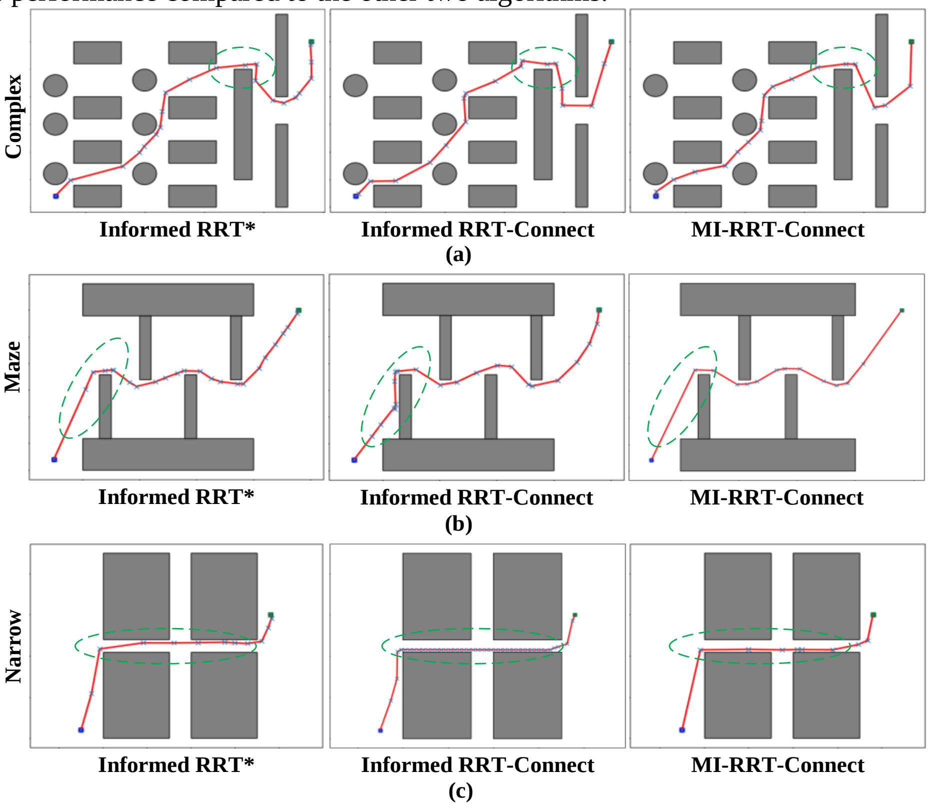


Figure 6. Results of the time and path lengths taken by the three algorithms to generate the optimal path.

5.2. Quadruped Robot Experiment

The quadruped robot platform used in the experiment is the A1 robot produced by Unitree (Fig. 7 (a)). A1 is equipped with single-line lidar and IMU for environmental perception and state estimation. All mapping and motion planning functions are built into the ROS platform. Navigation experiments are set in a structured environment with multiple obstacles. A1 robot uses a navigation system built by lidar to run the path planning algorithm, recording the total running time of the robot and the distance traveled by following the trajectory, which is used to compare the performance of MI-RRT-Connect and Informed RRT-Connect algorithms. Fig. 8 shows the start and target positions of the two groups of experiments and the state of the robot crossing obstacles.

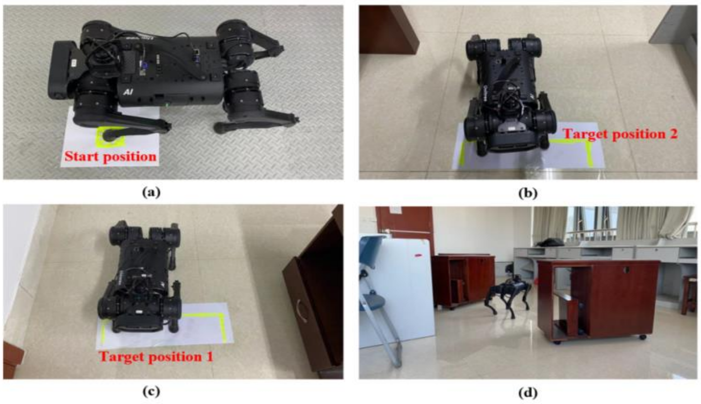


Figure 7. Set the navigation experiment position. (d) is the quadruped robot crossing the obstacle.

Table 2 Navigation Performance Indicators

	Total Running Time(s)		**Movement distance(m)**	
	MI-RRT-Connect	***Informed RRT-Connect***	***MI-RRT-Connect***	***Informed RRT-Connect***
1	56.52	130.57	8.37	9.78
2	82.39	90.74	8.53	8.81

Fig. 8 shows how the quadruped robot follows the optimal paths generated by the two path-planning algorithms. The blue line is the trajectory generated by the plan, the red line is the trajectory tracked by the quadruped robot, the green arrow is the direction of the robot's starting position, and the purple arrow is the direction of the target position. Finally, Table 2 reports the total time and distance moved for the quadruped robot to move along the trajectory. From Table 2, it can be seen that the robot takes the least distance and time to follow the trajectory generated by the MI-RRT-Connect algorithm. Therefore, it can be proved that our proposed algorithm can provide optimal path planning with the least time and the shortest distance for the quadruped robot.

Figure 8. The path trajectories provided by the two algorithms and the real trajectories followed by the quadruped robot.

6. CONCLUSION

In this paper, we propose a novel sampling-based path planning method suitable for the autonomous navigation of quadruped robots, which is called MI-RRT-Connect. Compared with the traditional Informed RRT* algorithm, the proposed method combines the double tree structure and improves the efficiency of path optimization by evaluating the path cost of the parent node. The experimental results show that the method can generate the optimal path for robot navigation in the shortest possible time. Meanwhile, we show that the algorithm runs on a quadruped robot, demonstrating the good performance and effectiveness of the proposed algorithm.

ACKNOWLEDGEMENT

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61703356, in part by the Natural Science Foundation of Fujian Province under Grant 2018J05114 and 2020J01285, in part by the Innovation Foundation of Xiamen under Grant 3502Z2020607)

REFERENCES

[1] Ferreira J, Moreira A P, Silva M, et al. "A survey on localization, mapping, and trajectory planning for quadruped robots in vineyards"//2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). Wuhan, China. IEEE, 2022: 237-242.
[2] Wang K, Chen T, Bi J, et al. "Vision-based Terrain Perception of Quadruped Robots in Complex Environments"//2021 IEEE International Conference on Robotics and Biomimetics (ROBIO). Sanya, China. IEEE, 2021: 1729-1734.
[3] Qi S, Lin W, Hong Z, et al. "Perceptive Autonomous Stair Climbing for Quadrupedal Robots"[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic. IEEE, 2021: 2313-2320.

[4] Čížek P, Masri D, Faigl J. "Foothold placement planning with a hexapod crawling robot"//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, Canada. IEEE, 2017: 4096-4101.
[5] Ding, Liang, et al. "Fault-tolerant free gait and footstep planning for hexapod robot based on Monte-Carlo tree" *arXiv preprint arXiv:2006.07550* (2020).
[6] Pradhan S, Mandava R K, Vundavilli P R. "Development of path planning algorithm for biped robot using combined multi-point RRT and visibility graph." International Journal of Information Technology, 2021, 13(4): 1513-1519.
[7] Bellicoso C D, Bjelonic M, Wellhausen L, et al. "Advances in real-world applications for legged robots." Journal of Field Robotics, 2018, 35(8): 1311-1326.
[8] Wang H, Zheng Y F, Jun Y, et al. "DRC-hubo walking on rough terrains"//2014 IEEE international conference on technologies for practical robot applications (TePRA). Boston, USA. IEEE, 2014: 1-6.
[9] Hauser K, Bretl T, Latombe J C, et al. (2008) "Motion planning for legged robots on varied terrain" The International Journal of Robotics Research, 27(11-12): 1325-1349.
[10] Klemm S, Oberländer J, Hermann A, et al. (2015) "Rrt-connect: Faster, asymptotically optimal motion planning" In: 2015 IEEE international conference on robotics and biomimetics (ROBIO). Zhuhai, China. IEEE,2015: 1670-1677.
[11] Wang J, Li B, Meng M Q H. (2021) "Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning" Expert Systems with Applications,
[12] 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. Chicago, USA. IEEE, 2014: 2997-3004.
[13] Li C, Wang C, Wang J, et al. (2021) "Sliding-Window Informed RRT*: A Method for Speeding Up the Optimization and Path Smoothing" In: 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR). Xining, China. IEEE, 2022: 141-146.
[14] Mashayekhi R, Idris M Y I, Anisi M H, et al. (2020) "Informed RRT*-connect: An asymptotically optimal single-query path planning method" IEEE Access, 8: 19842-19852.
[15] Gammell J D, Srinivasa S S, Barfoot T D. (2015) "Batch Informed Trees (BIT): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs" In: //2015 IEEE international conference on robotics and automation (ICRA). Seattle, USA. IEEE, 2015: 3067-3074.