Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 14, No. 2, 2024

Multi-point Navigation Method for Intelligent Inspection Robots

Zhicheng Wang [1, 2, 3], Qingzhang Wang [1, 2,3], Lingling Shen [1, 2, 3, *], Xiaojun Qian [1,2,3], Jianhua Cui [1,2,3], Liwei Zeng [1,2,3], Zewei Ye [1,2,3], Md Hasanul Kabir [1,2,3]

[1] School of Computer and Electronic Information Nanjing Normal University, Nanjing, China
[2] School of Artificial Intelligence, Nanjing Normal University, Nanjing, China
[3] Artificial Intelligence Research Institute, Nanjing Normal University, Nanjing, China
[*] **Corresponding author**: Lingling Shen (Email: llshen509@163.com)

Abstract: A multi-point navigation method is introduced for intelligent inspection robots, aimed at enhancing efficiency and safety across various industries. The newly proposed WW algorithm improves the robots' fault tolerance and positioning accuracy. The study utilized tools such as the ROS operating system, Rviz, and Gazebo, as well as the Agilex Bunker MINI intelligent trolley and SCOUT MINI sensor platform. It discussed SLAM map construction with Cartographer and Gmapping algorithms and the AMCL positioning system. Additionally, the DWA algorithm for dynamic obstacle avoidance was introduced. The system design includes a Qt interface and Rviz interface for multi-point navigation and obstacle avoidance. The effectiveness of the WW algorithm was verified through simulations and experiments, which enhances navigation stability by setting a maximum standby time and alternative point strategy. The experimental results show that the WW algorithm can prevent the robot from entering a false dead state. The paper concludes with suggestions for further optimization of the algorithm and the integration of more complex intelligent functions using deep learning.

Keywords: Inspection robot; ROS; Multi-point navigation; WW algorithm.

1. Introduction

Artificial intelligence, as the current research hotspot, empowers the development of intelligent robots which are widely used in the fields of equipment inspection. On October 29, 2022, a fire broke out at Jinsheng Store in Nanjing, China. Such fires could have been greatly reduced if equipment inspections had been adopted early inside such building facilities. Inspection Robots are poised to replace traditional, inefficient manual inspections. Among them, fixed-point inspection robots generally use multi-point navigation algorithms. Traditional fixed-point navigation algorithms have certain limitations, and when a fixed obstacle exists at a target point that makes the target point unreachable, the inspection task at that point will be skipped. In order to improve the completion of inspection tasks, this paper proposes an optimized multi-point navigation method. The fault tolerance and stability of the robots have been effectively enhanced, as demonstrated through both simulation and real-world experimental results.

2. Multi-point Navigation Interface System

In this paper multi-point navigation and dynamic obstacle avoidance were implemented through ROS [1] extended functionality package. However, it was not possible to troubleshoot complex environments during use. Robot navigation [2] is directly interrupted and the robot needs to be recommissioned manually. And this project is based on this optimization to solve the above problems and develop a software to optimize the user experience.

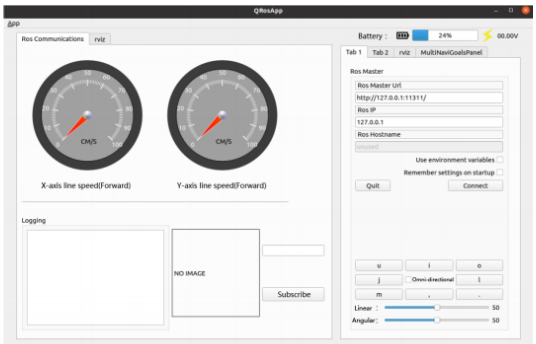

Figure 1. Robot Speed Dashboard

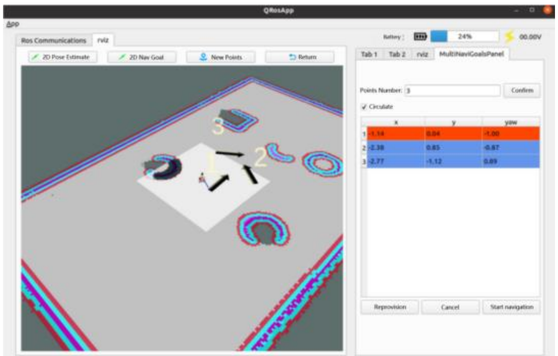

Figure 2. Multi-point Navigation Interface

The interface showed in Figure 1 includes a robot speed dashboard to operate the inspection robot via a manual keypad. Figure 2 shows the map information in the Rviz interface. The '2D Pose Estimate' is used to correct the vehicle position on the map. The '2D New Goal' is created in the 2D map and the position information of the target point is set. Finally, 'Start' is clicked to make the vehicle perform a multi-point cruise task.

3. Simulation experiments

3.1. Constructing map simulation experiment

ROS provides a rich simulation environment for experiments, and whether it is simulation or experiment, building the diagram is the first step.

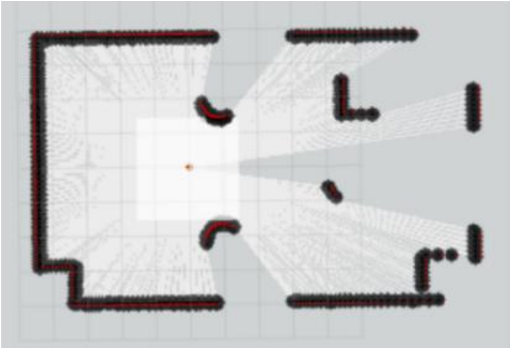

Figure 3. Map Initial Scan Display

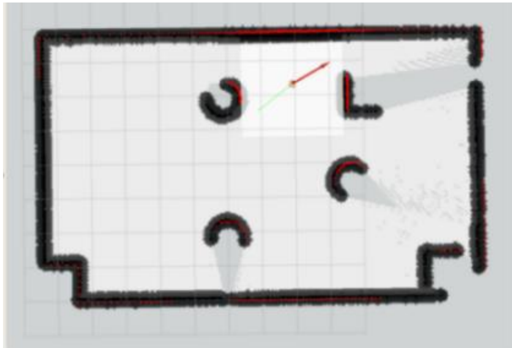

Figure 4. Map Construction Process Display

Figure 3 shows the incomplete outline of the map scanned by the initial radar for SLAM mapping, in which the vague red line at the edge of the map is the area scanned by the radar, the white area is the passable area, and the black outline is from the wall or obstacle.

In order to build a complete map, the movement of the inspection robot with the radar scanning uninterruptedly to the place where the image is not built out completely can build a complete black and white outline as shown in Figure 4.

There are two main ways to move, the first is to start the keyboard control node to control the robot's movement to build the diagram, the second is to start the move_base node, and set the target point through 2D Nav Goal in Rviz, then the robot will start to move and build the diagram autonomously.

Finally, the built map is saved by starting the map_server node, so that it can be used in further multi-point navigation experiments and there is no need to build the map again.

3.2. Navigation and dynamic obstacle avoidance experiment

The map constructed by Gmapping [3] is used in the navigation experiment. AMCL [4] node realizes real-time localization by subscribing to the information from LIDAR, move_base node carries out path planning by subscribing to the information from LIDAR and the map, and publishes the cmd_vel information to control the robot movement. Dijkstra's algorithm [5] and DWA algorithm [6] are used to achieve the robot's path planning.

The robot chassis node, navigation node and Rviz tool are started sequentially. The 2D Nav Goal tool in Rviz is used to set a goal point for the robot, then the move_base node will carry out path planning and select the optimal path for navigation. The simulated environment navigation experiment is shown in Figure 5. where the red line is the local path planning and the green line is the global path planning, the cylinders and squares on the way are the obstacles, their outlines are surrounded by the black lines and there are a few circles of coloured outlines around the black outlines, which are the expansion radius. A large number of experiments are used to detect the accuracy of the navigation in this environment, and the expansion radius is set to appropriate value. The obstacles scanned by the robot are appropriately made larger to effectively prevent the robot from hitting the obstacles.

When an obstacle suddenly appears, as shown in Figure 6 below, the robot immediately recognizes the obstacle's contour and replans the global and local paths in real time. However, since the sudden appearance of an obstacle causes the robot to fail to recognize the full contour of the obstacle immediately, the real-time map is also being reconstructed in the process of re-routing.

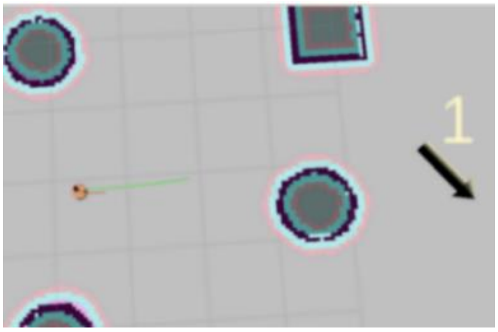

Figure 5. Accessible Path Planning Display

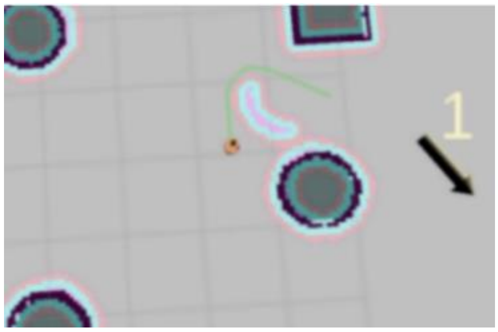

Figure 6. Re-Routing When Sudden Obstacles

4. Optimization methods

4.1. Problems with the original algorithm

On one hand, the expansion coefficient of the obstacle will be set larger for safety reasons to ensure that the intelligent inspection robot will not hit the obstacle. If the expansion coefficient is set relatively large, it will be judged as an impassable area in some narrow places, resulting in the possibility of not being able to pass some actually passable paths or not being able to reach some actually reachable points, which will lead to the failure of path planning at that point. In fact, the inspection task of a point is not only completed at a fixed point. The original algorithm only sets a fixed point, and when the situation described above occurs, the point will be directly skipped. But in reality, there are some other points around that point that can complete the inspection task. So, the inspection success rate of the original algorithm is very much affected by environmental factors.

On the other hand, when the temperature of the hardware processor is too high, its processing performance will be reduced, and the data transfer from the sensors will be slowed down. At this time, the TF coordinate transformation will be biased, and the longer the bias will be bigger, and then the sensor calibration point will be confused, which will directly lead to the failure of the path planning, and report the error of "unable to reach the point", and then get the next target point. The cycle repeats, and the intelligent inspection robot cannot work at last.

4.2. WW algorithm

The optimization algorithm shown in Figure 7 proposes the idea of setting the maximum standby time length and the strategy of alternative points. By setting the maximum standby time, the intelligent inspection robot is allowed to carry out navigation planning many times in the specified time, when there is a fixed obstacle or other unexpected situation in the location of the target point, and when the specified standby time is reached, there is no successful path planning to the target point, then it is considered that the target point is unreachable, and it will turn to get the alternative point of the target point, i.e. The WW algorithm was proposed with the maximum standby time length and the strategy of alternative points.

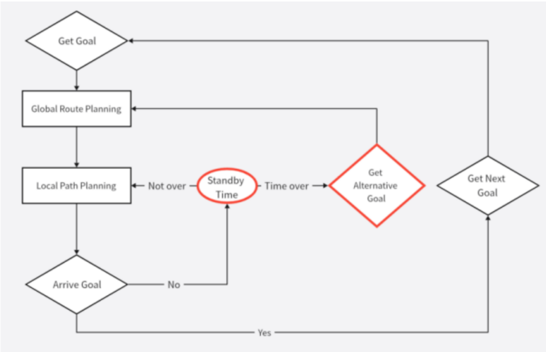

Figure 7. Core Flowchart Of The Algorithm

By setting the maximum standby time, the intelligent inspection robot is allowed to carry out navigation planning many times in the specified time. When there is a fixed obstacle or other unexpected situation in the location of the target point, and when the specified standby time is reached, there is no successful path planning to the target point, then it is considered that the target point is unreachable, and it will turn to get the alternative point of the target point, i.e. The inspection robot can automatically move backward or sideways by a certain distance according to the spatial location of the point to find an alternative point that is reachable and can complete the inspection task of the point, and then navigates.

Compared with the previous algorithm, the advantage of this optimization algorithm is that it allows the incoming parameters of the sensors to have a certain degree of error, allows a certain time difference in the conversion of the coordinate system.

This greatly improves the fault tolerance of the intelligent inspection robot's navigation and greatly reduces the cases of entering a false-dead state.

5. Conclusion

In the simulation environment, ROS provides multi-point navigation and dynamic obstacle avoidance functions will be fine. But in the real environment, due to the limited performance of the 2D radar and the processor, the real-time state information cannot be timely and accurately shown, and this will lead to the failure of the path planning. Current autonomous multi-point navigation robots will abandon the point and go to acquire the second target point if they are unable to carry out path planning after acquiring the target point or if they are unable to reach the target point due to the presence of fixed obstacles at the target point location.

With the original algorithm we conducted a large number of experiments, in which a situation as shown in Figure 8 occurs, in which the color radar scanning boundary is clearly misaligned with the map obstacle boundary, and the robot enters a false-dead state. This is due to the TF coordinate system conversion is not timely, the parameters have not been able to pass in correctly, but the path planning has already begun, the direct error into the next point, and at this time will be the same cannot path planning, so the robot is trapped in a cycle of false death state.

When we use the WW optimization algorithm, i.e., setting the maximum standby time and alternative points, we re-conducted a large number of experiments in the field of the arithmetic center, as shown in Figure 9 below, in which the solid points are the initial target points and the dotted points are the feasible alternative points around the target points. The experimental results show that the WW optimization algorithm can be implemented correctly, and the intelligent inspection robot will not fall into a false-dead state.

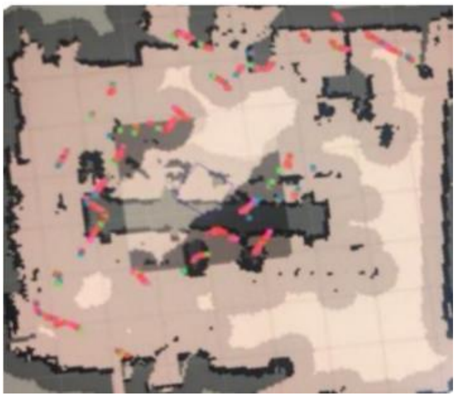

Figure 8. Radar Scanning Misalignment

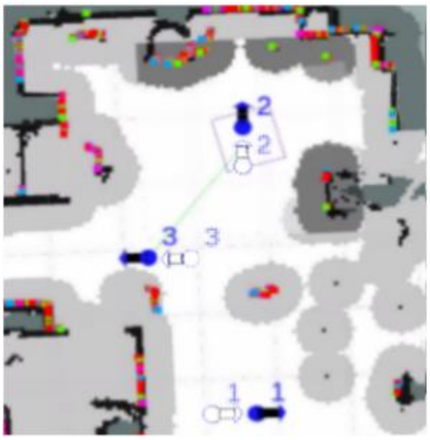

Figure 9. Radar Scanning Normally

Acknowledgment

This work was financially supported by Innovation Training Fund of Artificial Intelligence Research Institute, Nanjing Normal University.

References

[1] https://www.ros.org/

[2] Felix Nobis, Johannes Betz, Leonhard Hermansdorfer, Markus Lienkamp. Autonomous Racing: A Comparison of SLAM Algorithms for Large Scale Outdoor Environments[C].The 3rd International Conference on Virtual and Augmented Reality Simulations (ICVARS'19), Association for Computing Machinery, New York,2019:82-89.

[3] Jiaxin Wu, Xize Jin, Kai Sun, Jieyang Zhou, and Jinlin Zhang. 2024. Design of Autonomous Navigation Robot Based on ROS System. In Proceedings of the 3rd International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA '23). Association for Computing Machinery, New York, NY, USA, 39–44.

[4] Jingchao Mao, Lifeng Zhu, Huiran Hu, Linhu Wei, and Aiguo Song. 2024. Design Of Intelligent Mobile Robot System Based On Remote Operation. In Proceedings of the 2024 4th International Conference on Robotics and Control Engineering (RobCE '24). Association for Computing Machinery, New York, NY, USA, 96–101.

[5] Zhao, X.; Wang, Z.; Huang, C.K.; Zhao, Y.W. Mobility based on improved A* algorithm. Robot path planning. Robot 2018(40):903-910. (In Chinese)

[6] WenHao Zhai. 2024. Real-time Autonomous Vehicle Maneuver Generation Algorithm Based on Hierarchical Planning. In Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (DEAI '24). Association for Computing Machinery, New York, NY, USA, 968–972.