

Article

SLAM, Path Planning Algorithm and Application Research of an Indoor Substation Wheeled Robot Navigation System

Jianxin Ren , Tao Wu *, Xiaohua Zhou, Congcong Yang, Jiahui Sun, Mingshuo Li, Huayang Jiang and Anfeng Zhang

Faculty of Mechanical and Electrical Engineering, Kunming University of Science & Technology, Kunming 650500, China; 20212203017@stu.kust.edu.cn (J.R.); 20192203090@stu.kust.edu.cn (X.Z.); 20182203001@stu.kust.edu.cn (C.Y.); 20202203082@stu.kust.edu.cn (J.S.); 20212203068@stu.kust.edu.cn (M.L.); 20212103068@stu.kust.edu.cn (H.J.); 20172203009@stu.kust.edu.cn (A.Z.)

* Correspondence: 20030140@kust.edu.cn; Tel.: +86-132-0885-7874

Abstract: Staff safety is not assured due to the indoor substation's high environmental risk factor. The Chinese State Grid Corporation has been engaged in the intelligentization of substations and the employment of robots for inspection tasks. The autonomous navigation and positioning system of the mobile chassis is the most important feature of this type of robot, as it allows the robot to perceive the surrounding environment information at the initial position using its own sensors and find a suitable path to move to the target point to complete the task. Automatic navigation is the basis for the intelligentization of indoor substation robots, which is of great significance to the efficient and safe inspection of indoor substations. Based on this, this paper formulates a new navigation system, and builds a chassis simulation environment in the Robot Operating System (ROS). To begin with, we develop a novel hardware and sensor-based chassis navigation system experimental platform. Secondly, to conduct the fusion of the odometer and inertial navigation data, the Extended Kalman Filter (EKF) is used. The map's creation approach determines how the environmental map is created. The global path is scheduled with the A* algorithm, whereas the local path is scheduled with the Dynamic Window Method (DWA). Finally, the created robot navigation system is applied to an auxiliary operation robot chassis suited for power distribution cabinet switch and the navigation system's experimental analysis is conducted using this platform, demonstrating the system's efficacy and practicability.

Keywords: wheeled mobile robot; multi-sensor data fusion; ROS; path planning



Citation: Ren, J.; Wu, T.; Zhou, X.; Yang, C.; Sun, J.; Li, M.; Jiang, H.; Zhang, A. SLAM, Path Planning Algorithm and Application Research of an Indoor Substation Wheeled Robot Navigation System. *Electronics* **2022**, *11*, 1838. <https://doi.org/10.3390/electronics11121838>

Academic Editor: Hamid Reza Karimi

Received: 26 April 2022

Accepted: 6 June 2022

Published: 9 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The high operating voltage of the substation creates electromagnetic radiation and other unfavourable conditions, such that personnel safety is not guaranteed and accidents are not uncommon. Staff can do the work from a remote location. The robots can also eliminate many missed and erroneous inspections [1].

Substation robots belong to the class of modern intelligent mobile robots. Their main functions are to familiarize themselves with their environment and find an appropriate path to the goal point to complete the entire mission. Autonomous navigation is the basis for the realization of robot intelligence, and it is also the key to realizing the robot's autonomous walking to reach the destination and complete specific tasks. During autonomous navigation, the robot not only ensures its own safety but also does not damage the surrounding environment. A complete navigation system has three aspects of autonomous positioning, map construction, and planning control [2]; these three aspects correspond to the above three problems. When constructing an environmental map, the robot must recognize its position information, the positioning and mapping of the robot are studied and analyzed as one aspect, and the navigation path planning is analyzed as another aspect, aiming

at the safety of the robot; stopping and avoiding obstacles in the navigation process are analyzed as a third aspect. The existing navigation methods include trajectory navigation, inertial navigation, Global Position System (GPS) navigation, visual navigation and laser navigation.

Trajectory navigation lays a guideline on the ground that the robot will pass; the robot body is equipped with a guideline recognition device, and the robot can walk along the guideline without deviation through the control algorithm. The advantage is that it is stable and reliable, but its disadvantages are also obvious: the environment needs to be modified, the later maintenance cost is high, and the movement of the robot is also restricted by the guideline [3–5].

Inertial navigation is a typical relative positioning navigation method. The robot's posture information is obtained by integrating the sensor data of the Inertial Measurement Unit (IMU), including the magnetometer, accelerometer, and gyroscope. The track estimation in the mileage measurement system also belongs to inertial navigation, and the track estimation is to obtain the robot's posture information through the motor encoder and robot kinematics. Compared with the inertial measurement unit, the track calculation is more accurate. Inertial navigation can have a good effect on short-distance positioning and navigation, while longer distances will cause serious accumulated errors. Inertial navigation alone cannot achieve accurate navigation and positioning effects [6].

GPS navigation is an absolute positioning navigation method, such that there is no cumulative error. Because this kind of navigation method obtains the posture by receiving the GPS signal, the strength of the signal and the interference to the signal will have a greater impact on the method, and it is even more difficult to receive in the indoor environment, but this method is relatively stable for the outdoor environment. It has been applied to mobile phones, cars, and map navigation. It is also used as a positioning data application in some outdoor robots. It is integrated with other data to improve positioning and navigation accuracy [7].

Visual navigation [8] is the hottest research direction in recent years; the main hardware includes monocular, binocular and depth cameras. This technology is also called visual Simultaneous Localization and Mapping (SLAM) [9] technology. Due to the plentiful feature information contained in the environmental image obtained by the camera, which is similar to human eyes, advanced algorithms can construct a three-dimensional map of the environment and determine the pose of the robot's environment. Compared with LIDAR, the camera is inexpensive and reduces hardware costs. However, this method is still in the early stage of research and requires high computing capabilities; most of them are based on experiments and have not yet been truly commercialized. Vision is based on light, and all of the information about the characteristics of the night environment will be lost, such that all-weather autonomous navigation cannot be achieved by vision alone.

Laser navigation has been the most popular navigation method in recent years. Because laser navigation can use SLAM technology to build maps without modifying the environment, which greatly improves the versatility of the robot, and it can also be applied to indoor and outdoor environments. Nowadays, most robots construct navigation systems by combining multiple navigation methods. Laser navigation is usually combined with inertial navigation to realize the autonomous navigation of robots, but the price of LIDAR is higher, which greatly increases the cost of the robot [10].

In the method, predecessors did a lot of research. Guastella et al. concluded that perception plays a vital role in complex environments because it provides the necessary information to make vehicles aware of their states and surroundings, all of which have the unstructured nature of the environment under consideration. However, these methods are still in the embryonic stage; in indoor substations in this structured environment, mature use is still quite far away [11]. Seçkin et al. proposed a natural navigation method to track paths from 2D maps, which aims to utilize the memory of previous paths and then follow the same path, with the problem that mobile robots can reach a target at a lower error rate without accumulating errors, even in a multi-bend system, but a sequence of

symmetrical or repetitive objects leads to the robot detecting similar patterns and making wrong decisions [12]. Yan et al. proposed a mapless navigation method based on two-dimensional LIDAR in complex and unknown environments, which does not depend on global planning or an environmental map. However, when it comes to moving obstacles, the optimal sub-target will be changed due to the movement of obstacles, which will lead to the problem of inaccessible targets [13].

Laser SLAM is currently the most stable and mainstream positioning and navigation method. The main mapping methods for laser SLAM include Gmapping, HectorSLAM, and KartoSLAM. Gmapping is currently the most widely used 2D SLAM method; it can construct indoor maps in real time, and has good mapping effects in long corridors and low-feature scenes. The disadvantage is that it depends on odometry, it is not applicable to unmanned aerial vehicles or uneven ground areas, and there is no loop. HectorSLAM has relatively high requirements for sensors, it does not use odometry, and it can adapt to the situation of uneven ground areas. The disadvantage is that the update frequency of the LIDAR is required to be high, and the measurement noise is small. KartoSLAM is a graph-based SLAM approach that uses the mean value of the graph to represent the map; each node represents the sensor measurement dataset and a position point of the robot trajectory. In this paper [14], the algorithms were compared in the size simulation environment, the actual environment, and CPU consumption, and it was found that KartoSLAM and Gmapping have more advantages. Due to the small amount of calculation and the effective use of odometer information, this paper chooses the Gmapping mapping method.

This article focuses on the chassis of the robot, including the selection of the sports chassis model and the realization of autonomous navigation functions. Firstly, the EKF is used to realize the data fusion of odometry and inertial navigation, and the Gmapping method is used to construct the environment map. Secondly, the A* algorithm is used to plan the global path, and then the DWA is used to plan the local path to achieve dynamic obstacle avoidance. At the same time, the safety touch sensor is used to realize the robot's stop function. Finally, the researched chassis navigation system is applied to the functional indoor substation switch cabinet operation robot, and it is verified in the process of detection and evaluation.

2. Construction of the Robot Chassis

2.1. Overall Design of the Robot Chassis

A robot application scenario was developed in an indoor substation; the environment has flat ground, the corridors are narrow, and the operating robot needs a pan-tilt operating module and mobile module, which are heavy and have high centres of gravity. The robot has high flexibility requirements such that it can adjust the posture of its body in a small range in-real time. Based on the above conditions, an omnidirectional mecanum wheel was chosen, as it meets the load requirements, has high versatility, and can flexibly change its posture.

Figure 1 shows the navigation structure of the robot chassis. The hardware system mainly includes an industrial controlling computer, motor drive, encoder, IMU, LIDAR sensor, safety edge sensor, battery power supply, transformer, and Controller Area Network (CAN) analyzer. The industrial control computer is the robot's brain, where all data processing is realized. The IMU [15] sensor performs data fusion with the position and posture for more accurate positioning. The LIDAR sensor is mainly used to sense and extract environmental information in an unknown environment to achieve autonomous navigation. The safety edge sensor can realize automatic obstacle avoidance and stop.

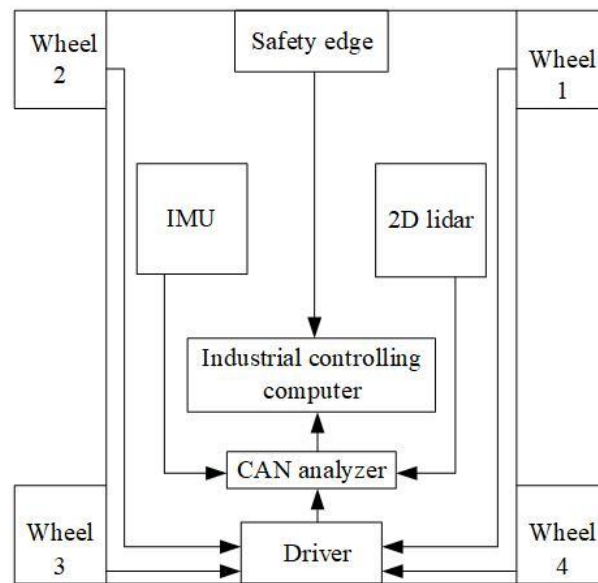


Figure 1. Navigation system framework of the robot chassis.

Positioning and navigation are accomplished using 2D LIDAR, which does not need to change the environment when acquiring environmental information. The fusion of LIDAR and other sensor information enables the construction of a more accurate environment map following a certain proportion. The robot's posture is determined when the laser beam coincides with the map. The robot body can be protected by the safety edge sensor when the laser scan is not available for irregularly shaped obstacles. Functions such as autonomous navigation can be achieved in the Gazebo simulation environment, and the code is migrated to the experimental platform to verify and improve the navigation system and to test its stability.

2.2. Mecanum Wheel Chassis Kinematics and Odometer Model

Stable control is very important for mobile robots in subsequent autonomous navigation. In order to control the motion of a mobile robot, we must first understand its kinematics. For a single Mecanum wheel, its motion analysis is shown in Figure 2, and the oblique line in the figure is the roller with which the Mecanum wheel contacts the ground.

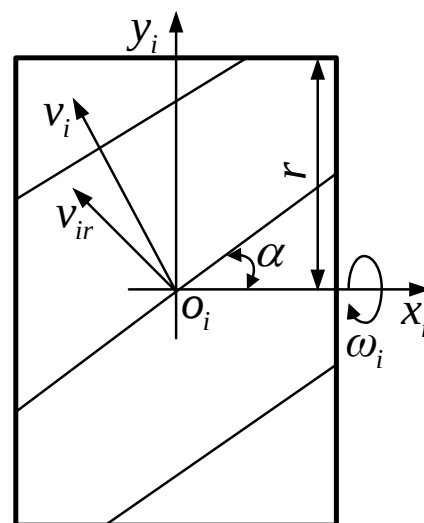


Figure 2. Motion analysis diagram of a single Mecanum wheel.

Where r is the radius of the wheat wheel, α is the offset angle of the roller, ω_i is the rotation speed of the hub ($i = 1, 2, 3, 4$), v_{ir} is the rotation speed of the roller, and v_i is the speed of the center position of the wheat wheel:

$$v_i = \begin{bmatrix} 0 & \sin \alpha \\ r & \cos \alpha \end{bmatrix} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} \quad (1)$$

The motion analysis of the entire chassis structure of the mobile robot is shown in Figure 3. Where l_y is half the distance between the two Mecanum wheels in the longitudinal direction, l_x is half the distance between the two Mecanum wheels in the transverse direction, v_x is the speed of the chassis on the axle, v_y is the speed of the chassis on the axis, and ω is the angular velocity of the chassis, then the speed of each wheel is

$$v_1 = \begin{bmatrix} 1 & 0 & -l_y \\ 0 & 1 & l_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}, v_2 = \begin{bmatrix} 1 & 0 & -l_y \\ 0 & 1 & -l_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}, v_3 = \begin{bmatrix} 1 & 0 & -l_y \\ 0 & 1 & -l_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (2)$$

$$v_4 = \begin{bmatrix} 1 & 0 & l_y \\ 0 & 1 & l_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

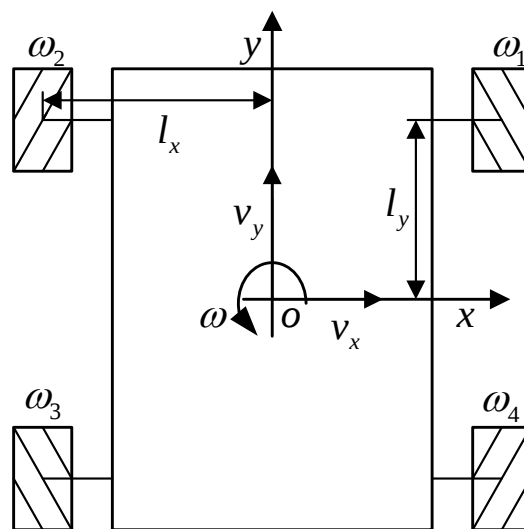


Figure 3. Mecanum wheel chassis motion analysis diagram.

Take wheel 1 as an example: combine Equation (1) with Equation (2) (usually the most-used angle is 45°), then

$$\omega_1 = \frac{1}{r} \begin{bmatrix} -1 & 1 & l_x + l_y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (3)$$

The speed of the other three Mecanum wheels can be obtained in the same way, and the inverse kinematics of the chassis as shown in Equation (4):

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -1 & 1 & l_x + l_y \\ 1 & 1 & -l_x - l_y \\ -1 & 1 & -l_x - l_y \\ 1 & 1 & l_x + l_y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (4)$$

After transformation from Equation (5), the linear velocity and angular velocity of chassis motion can be obtained:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{r}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (5)$$

During the movement of the mobile robot, the motor encoder can record the moving distance and rotation angle of the mobile robot. The higher the accuracy of the motor encoder, the more accurate the odometer of the mobile robot. Odometers provide position and attitude information for mobile robots. The method and principle of the encoder providing the odometer is shown in Figure 4:

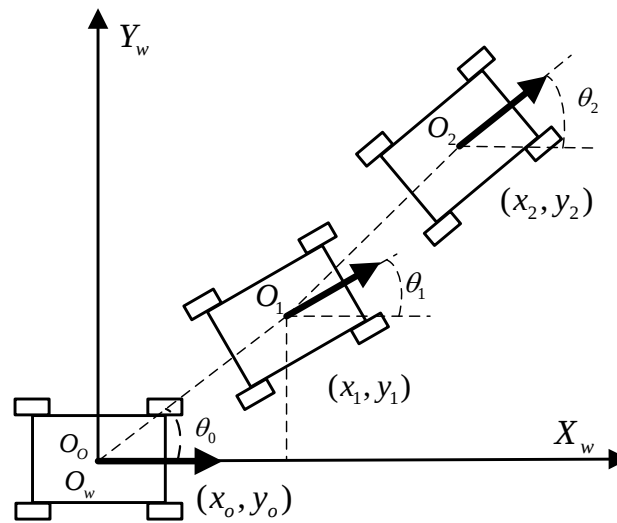


Figure 4. Odometer principle.

Where $O_W X_W Y_W$ is the world coordinate system and O_i is the center of the chassis. The sampling time is very short, the motion is regarded as linear motion, the rotation speed of the wheel is considered unchanged, and the pose of the mobile robot at the moment of t is (x_t, y_t, θ_t) ; then, the odometer is

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} \cos \theta_t & -\sin \theta_t & 0 \\ \sin \theta_t & \cos \theta_t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix} \quad (6)$$

Let the sampling cycle time be Δt , the angle that the wheel turns during the sampling cycle interval is $\Delta \gamma_i$, r is the radius of the wheel, l_x is half the distance between the centers of the horizontal two wheels, and l_y is half the distance between the centers of the two vertical wheels. From Equation (5), we can obtain

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \Delta t = \frac{r}{4} \begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} \Delta t = \frac{r}{4} \begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{pmatrix} \begin{pmatrix} \Delta \gamma_1 \\ \Delta \gamma_2 \\ \Delta \gamma_3 \\ \Delta \gamma_4 \end{pmatrix} \quad (7)$$

The mobile robot position is estimated using the encoder data by Equation (7). Let Δe_i be the data change of the encoder in the sampling period, and let $\Delta \zeta$ be the number of pulses. Therefore, the pose of the robot at any time in the world coordinate system can be obtained from Equations (6) and (7):

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \frac{2\pi r}{4 \times \Delta \zeta} \begin{pmatrix} \cos \theta_t & -\sin \theta_t & 0 \\ \sin \theta_t & \cos \theta_t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} \begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{pmatrix} \begin{pmatrix} \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \\ \Delta e_4 \end{pmatrix} \end{bmatrix} \quad (8)$$

3. Robot Multi-Sensor Data Fusion Slam

The autonomous navigation system uses multi-sensor fusion SLAM technology, robot path planning technology, and robot automatic obstacle stop and obstacle avoidance technology. The chassis includes a mileage measurement unit formed by a motor encoder and an inertial navigation unit formed by the heading sensor; the data obtained by the two navigation and positioning methods [16] are used to obtain more accurate relative position coordinates. Gmapping is used to construct an environment map, in the case of a known map, according to laser point cloud data and the robot's relative positioning method; Adaptive Monte Carlo Localization (AMCL) realizes the robot's relocation, which reduces the accumulated positioning error in the long-term movement of the robot.

3.1. Data Fusion Positioning of Mileage Measurement and Inertial Navigation Based on EKF

The combination of multiple navigation methods is most useful to increase the precision of the robot's navigation [17–19]. The fusion of multiple sensor data compensates for the sensors' shortcomings to enhance the steadiness and robustness of the navigation and positioning.

Because the wheel odometer will have accumulated errors, sudden acceleration or deceleration and wheel slippage during sharp turns will also cause errors. In order to prevent this error from interfering with the mapping, an IMU (Inertial Measurement Unit) sensor is added. The sensor data of the IMU is stable, has high precision, and has a wide range of applications. In the ROS system, the extended Kalman filter can be used to fuse the two data, reduce the cumulative error of the odometer, and provide more accurate pose information for the mobile robot.

IMU and odometer data fusion can use EKF. EKF will first predict the system state and then update the observation. The system state transition and observation model can be differentiable nonlinear functions:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}) + w_k & w_k &\sim N(0, Q_k) \\ z_k &= h(x_k) + v_k & v_k &\sim N(0, R_k) \end{aligned} \quad (9)$$

where f is the nonlinear system model, h is the measurement model, w_k is system noise, v_k is measurement noise, and w_k, v_k all are zero-mean Gaussian noise [20], the covariance matrix is

$$w_k \sim N(0, Q_k) \quad v_k \sim N(0, R_k) \quad (10)$$

Taking as input the increment of the odometer from time $k-1$ to k , then

$$\begin{pmatrix} x_{k|k-1} \\ y_{k|k-1} \\ \theta_{k|k-1} \end{pmatrix} = \begin{pmatrix} x_{k-1|k-1} \\ y_{k-1|k-1} \\ \theta_{k-1|k-1} \end{pmatrix} + \begin{pmatrix} u(1) * \cos(\theta_{k-1|k-1}) \\ u(1) * \sin(\theta_{k-1|k-1}) \\ \theta_{k-1|k-1} + u(2) \end{pmatrix} \quad (11)$$

where $u(1)$ is the odometer distance increment and $u(2)$ is odometer yaw increment.

The corresponding Jacobian matrix is

$$F_k = \begin{bmatrix} 1 & 0 & -u(1) * \sin(\theta_{k-1|k-1}) \\ 0 & 1 & u(1) * \cos(\theta_{k-1|k-1}) \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

The prediction process is

$$\begin{aligned}\hat{X}_k^- &= f(\hat{X}_{k-1}^+, u_k) \\ P_k^- &= F_k P_{k-1} F_k^T + Q_k\end{aligned}\quad (13)$$

The update process is

$$\begin{aligned}K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{X}_k^+ &= \hat{X}_k^- + K_k (z_k - h(\hat{X}_k^-)) \\ P_k^+ &= P_k^- - K_k H_k P_k^-\end{aligned}\quad (14)$$

where F_k is the last state quantity, $\hat{X}_{k-1}^+$ is the state transition matrix at the corresponding moment, u_k is the control input vector, and H_k is the observation state matrix that the actual state space is mapped to the observation space. In order to build a robot model in the Gazebo simulation environment, the odometer data of the robot are set to be “odom”, which includes the current position and posture of the robot and the speed information at that time. The inertial navigation data is “imu”, and the data information after fusion is named “odom_combined”; this node is the data fusion node of the EKF algorithm, which subscribes to the information of “odom” and “imu”.

Figure 5 shows that the robot is controlled to complete a square movement with and without EKF in the Gazebo robot simulation environment. Through the path of the simulated robot, it can be found that its posture when fused with EKF has less drift, making its positioning more accurate.

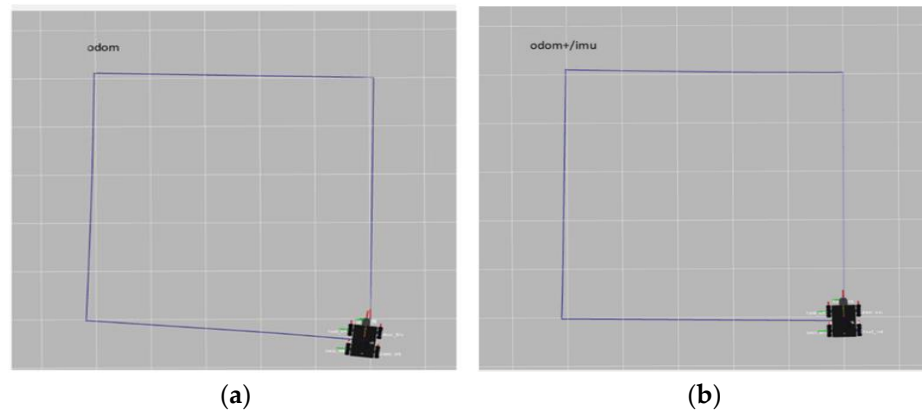


Figure 5. Comparison of multi-sensor fusion positioning effects: (a) /odom; (b) /odom + /imu.

3.2. The Method and Application of the Robot's Mainstream SLAM Mapping (Gmapping)

The environment information is acquired by 2D LIDAR, which is single-plane laser point cloud data, such that its feature information is less than that of a visual sensor, the feature map [21] representation is excluded firstly. In order to ensure stable and accurate navigation and prevent the loss of position information, the topological map [22] representation is excluded. After the above comprehensive consideration, the grid map [23] representation is used to build the robot environment map.

The SLAM is the process of sizing up the robot's location and environment information in the unknown environment from the current environment measurement, previous time position information, and control input.

The filter is indispensable to robot SLAM. The most representative is the SLAM algorithm based on a Particle Filter (PF), in which Gmapping is ameliorated based on a Rao-Blackwellized Particle Filter (RBPF) [24]. This method has two problems.

1. There is a high demand for robot posture accuracy when constructing the map. Only the posture information obtained from the motion model can construct the map, but the error will be large, which causes a mismatch between the real environment and the built environment map. The proposed distribution is too scattered, which is quite different from the real distribution; to increase the number of particles near the actual distribution requires more particles overall, which will increase the computational cost.
2. Frequent resampling leads to particle dissipation. Each particle is a posture state of the robot, which contains all of the posture and map information; frequent resampling causes the particles to gather at one point, which decreases diversity quickly, and even the correct particles tend to be discarded, causing the loss of location information.

Gmapping proposes an improved proposal distribution for the number of particles in RBPF and selective resampling for the frequent resampling of RBPF to reduce the amount of resampling [25]. Gmapping is used to build the robot simulation environment map, and to start the robot simulation environment built by ROS [26]. Figure 6 shows the map obtained.

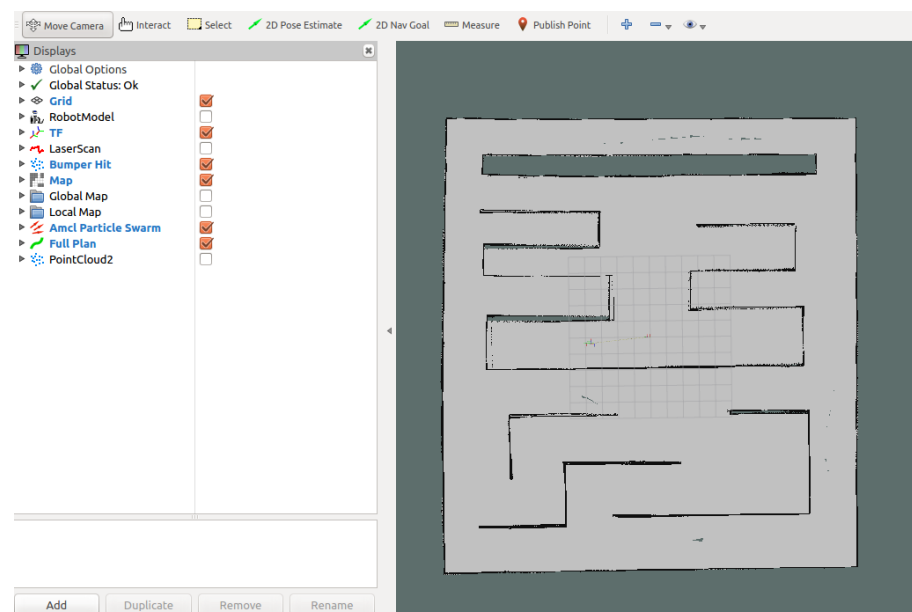


Figure 6. Map of the robot simulation environment.

3.3. AMCL Relocation Based on LIDAR Data

With data fusion and the positioning of the robot's internal sensors complete, environment information is observed by LIDAR, and a more accurate map is constructed. With a known map, it is difficult to maintain positioning accuracy for a long time with only the relative positioning mode of the robot's internal sensors, such that LIDAR is used to obtain the environment information and then match with the known map; the position and posture information are obtained according to the external environment in order to correct the accumulated error of relative positioning, so as to ensure the accuracy of positioning. Only when autonomous positioning is realized can path planning and navigation be better realized. This process is called the robot's relocation process; this process also solves the problems of robot wheels slipping, the robot drifting, and being moved in a small range by a human. AMCL is used to realize this process.

The small black line in Figure 7 represents the particle swarm estimated by the robot's position, the large white arc represents the mean value of the particle swarm, the small white circle represents the robot's true position relative to the map, and the grey robot mark in the figure finally coincides with the real position through this process to realize autonomous positioning. Figure 4 shows the AMCL process.

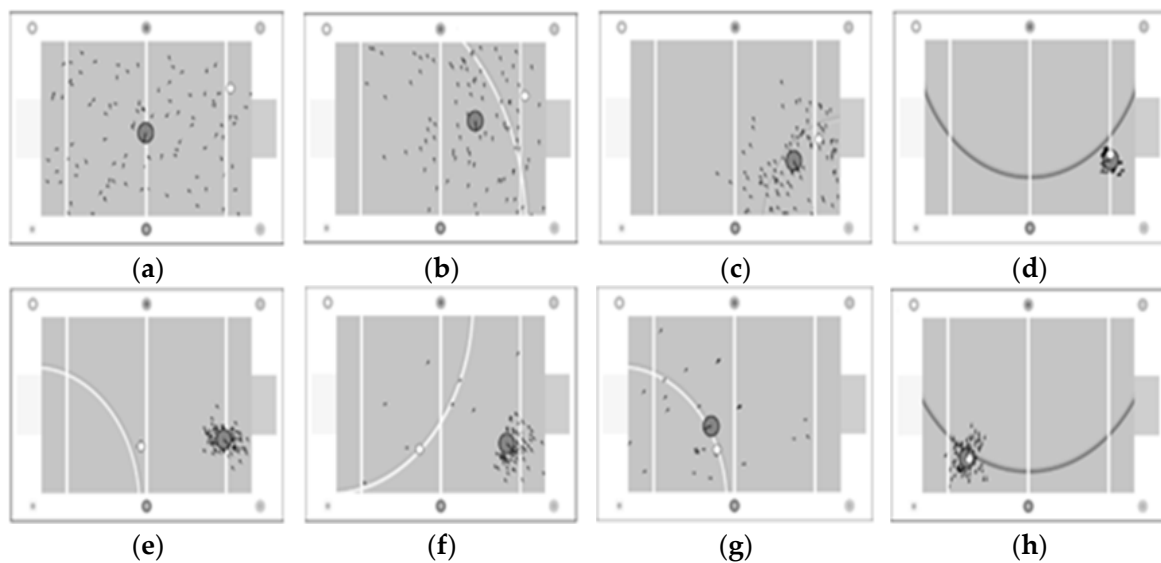


Figure 7. Adaptive Monte Carlo localization. (a–d) The global positioning of the robot; (e–h) the relocation process. Figure 4 shows the AMCL process.

Firstly, as shown in Figure 7a, each particle represents the position of the robot, and each has the same confidence. Then, bring every particle into the motion equation of the robot, make all particles move with the robot, and eliminate some particles that have moved out of the map, as shown in Figure 7b. Next, compare the remaining particles with the observed map environment information, change the confidence of each particle, improve the confidence of highly matched particles, and redistribute particles according to confidence to increase the number of accurate particles, as shown in Figure 7c. Finally, by repeatedly going through the above two steps, all of the particles are gathered around the robot, as shown in Figure 7d, thus completing the global localization process.

If a wheel drifts or the robot is moved by a human, it is separated from its real position in the environment, as shown in Figure 7e, and this is corrected by AMCL repositioning. The process is described as follows: First of all, the sudden decrease of the average particle score indicates that the robot has deviated from its intended real position; at this time, some particles are scattered around the robot map environment, as shown in Figure 7f. Afterwards, through the scattered particles, the map information is rematched for each particle, its confidence is calculated, and the number of particles with high matching degrees is increased, as shown in Figure 7g. Ultimately, through the above iterative process, the robot can obtain its position information again in order to realize the robot relocation, as shown in Figure 7h.

According to the analysis, AMCL needs the robot's motion information and LIDAR data, which also needs map information, Transform Frame (TF) conversion, and the mean and variance of the initialization particle filter for robot positioning.

The method is verified in the simulation environment and displayed by the RViz visualization tool of ROS. Figure 8a shows the initial state—the particles will gather slowly as the robot moves—and Figure 8b shows the final result. Each small arrow represents a particle, and all of the particles are updated iteratively such that they gather around the robot. By the above methods, the robot cannot obtain the real position quickly, such that an initial position of AMCL is usually given, which doesn't need to be very accurate; AMCL enables the robot to quickly complete the initialization in order to meet real-time requirements.

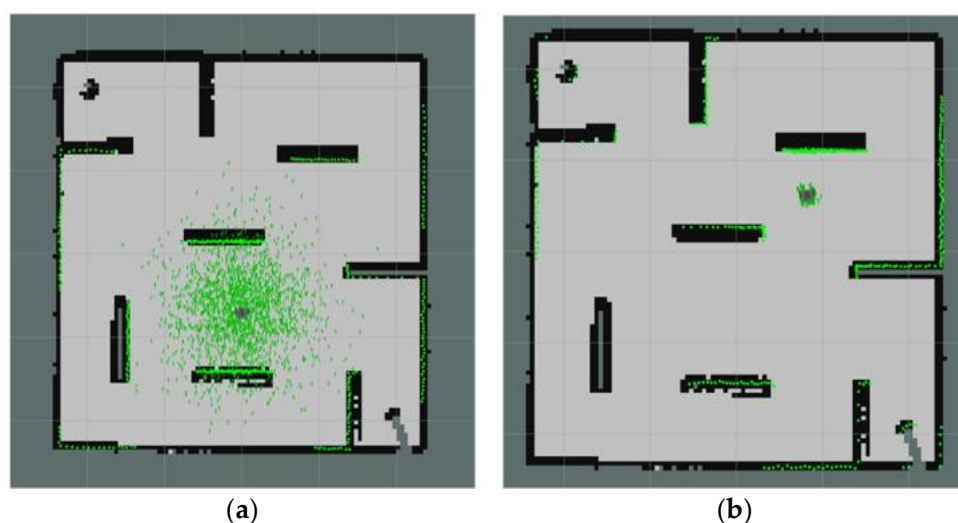


Figure 8. Effect of adaptive Monte Carlo localization. (a) AMCL positioning initial state; (b) AMCL positioning final state.

4. Research on the Path Planning of a Robot Autonomous Navigation System

Path planning analysis is performed from two aspects: global and local path planning. Common global path planning algorithms include Dijkstra, A* [27], the Probabilistic Road Map (PRM), and Rapidly exploring Random Tree (RRT). The Dijkstra algorithm is a classic breadth-first search algorithm. That is to say, the algorithm starts from the initial point and searches the entire free space layer by layer until it reaches the target point; this algorithm can obtain the shortest distance between two points, but it will greatly increase the time and data cost. The A* algorithm adds a heuristic function on the basis of Dijkstra, thereby improving the efficiency of Dijkstra's algorithm. The PRM algorithm first obtains sampling points in the environment and connects them to form a probabilistic road map. When setting the start and final points, it will find the sampling points closest to the start and endpoints. By contrast, we find it is relatively simple to search for the global path through the A* or the Dijkstra algorithm, but the route is not the shortest for the Dijkstra algorithm. Furthermore, it is also difficult to sample in narrow and long areas. The RRT algorithm takes the starting point as the root node and randomly samples the posture, and the tree is repeatedly expanded according to the reachable condition until the destination is added to the branch of the tree. It is strange that although this method directly considers the control, the route is not the shortest. In summary, A* was selected as the familiar global path planning algorithm.

The Dijkstra algorithm and the A* algorithm are provided in the navigation function package in ROS. Table 1 is a comparison of the running time of Dijkstra's algorithm and the A* algorithm.

Table 1. Dijkstra algorithm and A* algorithm traversal times.

Starting Point Coordinates	End Point Coordinates	Running Time of Dijkstra's Algorithm without Obstacles	The Running Time of the A* Algorithm without Obstacles	Running Time of Dijkstra's Algorithm with Obstacles	The Running Time of the A* Algorithm with Obstacles
(1, 1)	(18, 18)	2.8 s	1.9 s	3.8 s	2.1 s

It can be seen from Table 1 that the time taken by Dijkstra's algorithm is always longer than that of the A* algorithm, whether there is no obstacle or there is an obstacle.

4.1. Global Path Planning for Robots Based on A*

Global path planning can be described as the planning of a complete path from the starting spot to a goal spot when the environment map is completely known. The A* is depicted as follows.

Assume that the starting spot of the robot is O, the goal spot is G, and the current node is N. A* uses a cost evaluation function:

$$f(n) = g(n) + h(n) \quad (15)$$

where n is the search node; $f(n)$ is the cost of the robot to move from O through n to G; $g(n)$ is the actual cost to move from O to N, which is called the dissipation function; and $h(n)$ is the estimated cost to move from N to G, which is called the heuristic function.

A* improves the Dijkstra algorithm [28] by adding a heuristic function; Figure 9 shows the influence of $h(n)$ on A*. When $h(n) = 0$, the heuristic function does not work, and A* is the Dijkstra algorithm; the larger $h(n)$, the more restricted its search space, and the harder it is to ensure an optimal search. The $h(n)$ is usually determined by the Euclidean distance and the Manhattan distance from the starting spot to the goal spot. The specific flow chart of A* is shown in Figure 10.

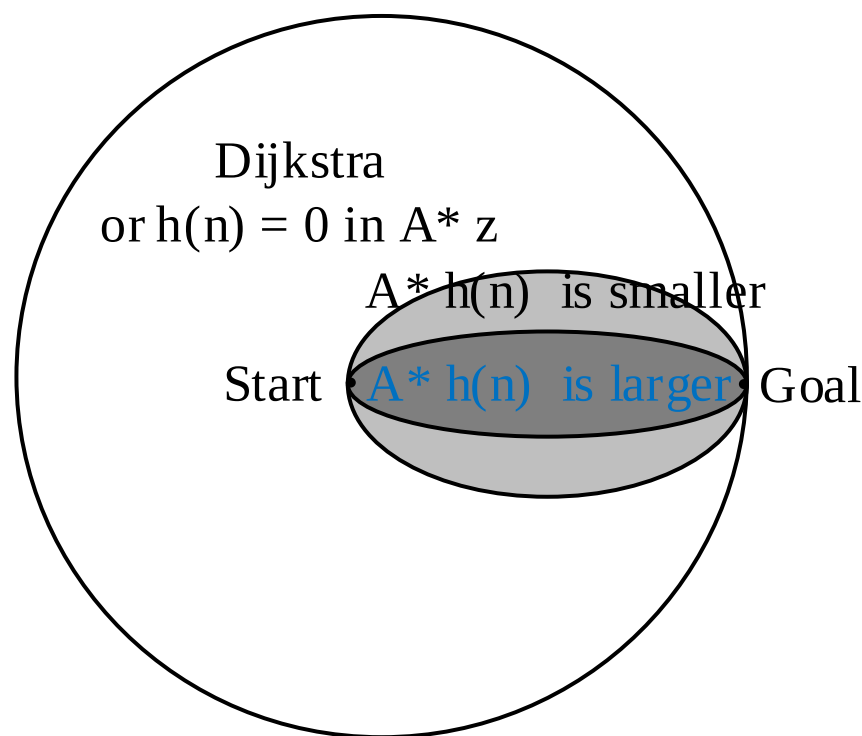


Figure 9. Impact graph of the heuristic function.

Figure 11 shows the process in the form of a graph, where the upper-left, lower-left, and lower-right corners of each square represent $f(n)$, $g(n)$ and $h(n)$, respectively. Among them, yellow represents the starting point, gray represents the ending point, red represents the optimal path, and black represents the obstacle.

Based on the above analysis, Figure 12 shows the simulation results of the A* algorithm, where the black square represents the wall, dark green is the starting spot, red is the goal spot, light blue-green represents the search space traversed during the path planning, and the yellow solid line represents the optimal path planned from the starting spot to the goal spot.

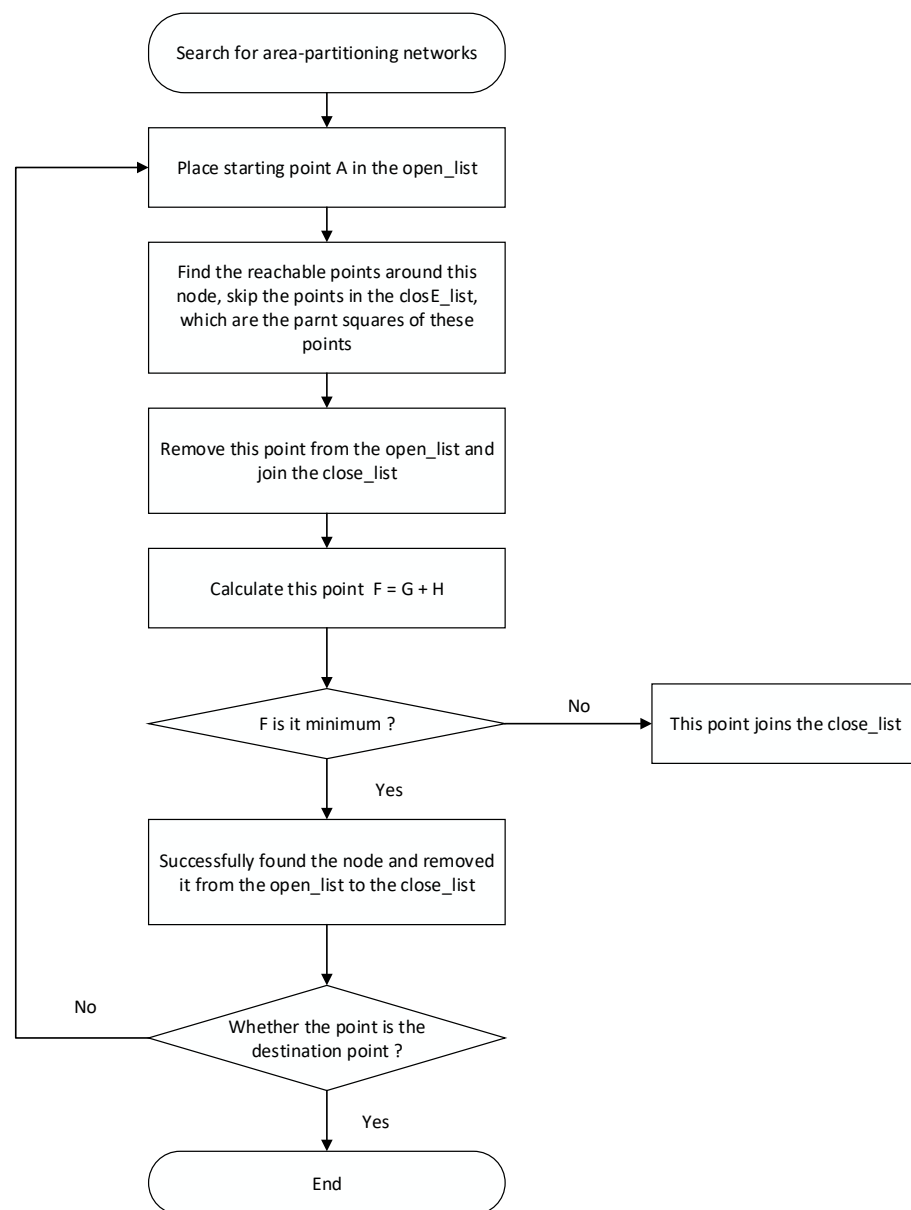


Figure 10. The A* algorithm flowchart.

108	94	80	74				
28	80	24	70	20	60	24	50
94		74		60		54	
24	70	14	60	10	50	14	40
80		60	Initial State	54			82
20	60	10	50	Initial State	14	40	72
94		74		60		54	74
24	70	14	60	10	50	14	40
108	94	80		74		74	
28	80	24	70	20	60	24	50
		108	94	88	88	88	88
		38	70	34	60	38	50

Figure 11. The A* schematic diagram of the path planning.

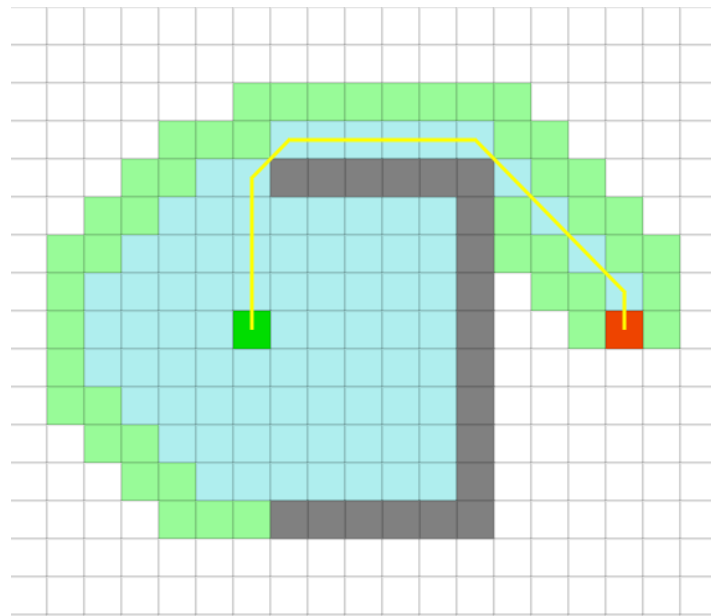


Figure 12. A* simulation result.

4.2. Robot's Local Path Planning Based on DWA

In local path planning, the robot plans the best local available path and issues speed commands to control its movement according to the real dynamic environment information from LIDAR. Universal local path planning algorithms include Artificial Potential Field (APF) [29], Vector Field Histogram (VFH) [30], and DWA. According to the robot's indoor application environment and the environmental representation in the grid map, DWA was chosen; according to the characteristics of the dynamic environment, DWA can realize the function of real-time obstacle avoidance.

DWA forecasts the trajectory based on the multi-group velocity sampling of the robot motion state according to the model of the robot chassis motion; it chooses the best-predicted trajectory so as to avoid the robot crashing, and then the speed command of the robot is issued to control it along the best local path.

5. Research on Automatic Stopping and Avoiding Obstacles

During the development of the robot navigation system, it is equally important to ensure the safety of the robot body and the surrounding environment. The local path planning based on the dynamic window method (DWA) is carried out under the dynamic environment map. The real-time information obtained by the laser can detect whether there are obstacles in front of the robot and the distance information of the obstacles [31], and then realize the autonomous navigation and obstacle avoidance function of the robot. However, because the data acquired by the 2D LIDAR comes from a single plane, it is impossible to test shorter roadblocks. Therefore, a safety edge sensor is used to detect shorter roadblocks and stop the robot's movement based on real-time feedback.

The safety edge, also known as a bumper strip, is a pressure-sensitive switch that appears as a soft, bendable ribbon. It can be fixed around the moving vehicle body to prevent collision. Generally, it is normally open logic. When triggered, these safety edges are compressed and deformed due to their softness. Then, a closed IO signal is generated through sensing and transmitted to the core controller, which feed back the information that the moving car body has collided with the object. The significance of installing the safety edge is that because the mobile robot usually uses a two-dimensional laser to perceive the scene, it can only identify and detect objects in the laser plane, but there may be three-dimensional obstacles on the user's site. It is impossible to perceive the protruding part at the bottom of the object. At this time, the protruding lower part will touch the anti-collision bar, and the robot will stop immediately after sensing the triggering of the anti-collision

bar. Therefore, the anti-collision strip is generally installed on the plane that cannot be scanned by the laser; it is generally installed relatively low, and the anti-collision strip will be installed before and after the robot.

For the robot itself, the control of the safety edge sensor is set as the highest priority, followed by the keyboard control, and finally the robot autonomous navigation process control. The robot's safety has a preference based on this priority, and when the safety edge sensor receives a collision pressure signal, the robot will stop immediately.

6. Practical Application Platform of the Autonomous Navigation System

By analyzing the operating mechanism of the robot ROS system [32], a complete navigation software system for the robot chassis was built, and it was all concentrated in a ROS workspace; multiple processes took place simultaneously to achieve the autonomous navigation function of the robot chassis and completed the chassis software system design. The researched robot navigation system was applied to actual robots, the new chassis physical parameters were recalibrated, and we analyzed and verified the system through practical application.

Figure 13 shows the robot body; the main function is to assist substation workers in the detection of a distribution switch cabinet. The robot has three main parts, including a Mecanum wheel chassis, a three-axis robot arm platform movement module, and a spindle operation pan-tilt module.

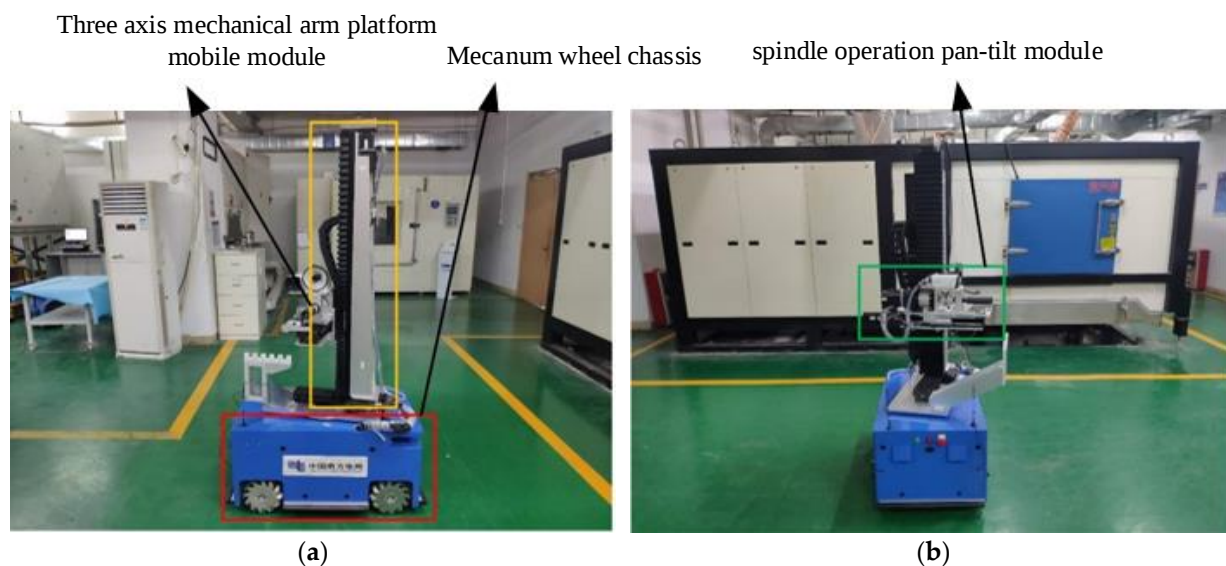


Figure 13. Application platform of the autonomous navigation system. (a) Main view; (b) left view.

We used the power distribution switchgear robot to achieve positioning and mapping; we also used the Gmapping method to build a map, and used the map_server of ROS to save the map file. The saved map file provides a reference for the later navigation process. In the same indoor environment, the map created by the mobile robot using the Gmapping algorithm through the single odometer method is shown in Figure 14a, and the map created by integrating the odometer and IMU data is shown in Figure 14b. All of the maps have a resolution of 5 cm. Through the comparison of the two maps, it can be seen that in the initial period of time, the maps created by the two are basically not much different. With the accumulation of time and the increase of the moving distance of the mobile robot, the contour of the single-wheel odometer method gradually appears tilted and deformed when the map is constructed, and the constructed map has a certain deviation compared with the actual indoor environment, as shown in the red box in Figure 14a. The map is tilted by 2° . However, the map constructed by combining the odometer and IMU data has a clear outline and can accurately estimate the pose information of the mobile robot in the indoor environment, and the constructed map can accurately and truly express the indoor

environment. Through comparative analysis, it can be seen that the map results created by the latter are more accurate.

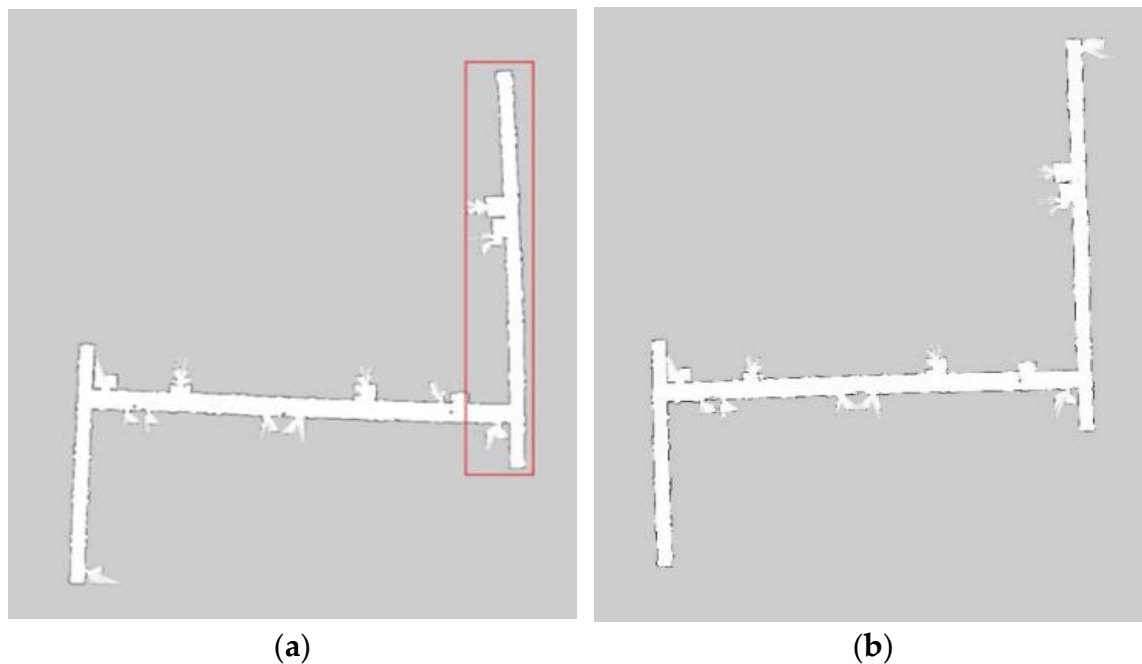


Figure 14. Multi-sensor laser SLAM and single-sensor laser SLAM. (a) Raster map created by a single-wheel odometer; (b) raster map created after fusing IMU.

The widths of the two sides and three positions in the middle of the corridor were selected for measurement, and the measured values of the grid map were used for comparison. The experimental data are shown in Table 2 and Table 3. By comparing Tables 2 and 3, it can be seen that the error of the map accuracy in the early stage is relatively small. With the increase of the grid map, the error of the map created by a single odometer is larger than that of the odometer and IMU. The composition effect is better after the fusion of the odometer and IMU.

Table 2. Accuracy of a single-odometer grid map.

Location	Measurements/m	Map Values/m	Absolute Error/m	Relative Error/%
1	2.450	2.424	0.026	1.06
2	3.425	3.359	0.066	1.93
3	2.430	2.305	0.125	5.15

Table 3. Raster map accuracy of the fusion odometer and IMU.

Location	Measurements/m	Map Values/m	Absolute Error/m	Relative Error/%
1	2.450	2.428	0.022	0.89
2	3.425	3.380	0.045	1.31
3	2.430	2.366	0.064	2.63

We need to achieve autonomous navigation in the environment, without rebuilding the map. The constructed indoor substation environment map is shown in Figure 15.

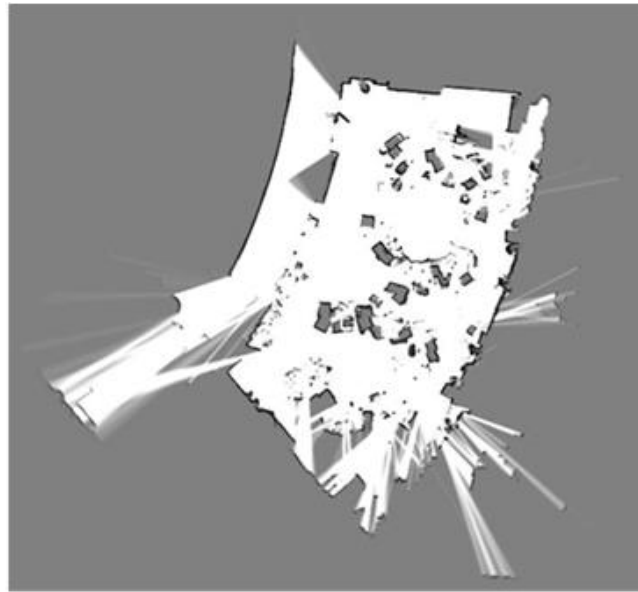


Figure 15. Indoor environment map.

For the analysis of the autonomous navigation system of the robot, we mainly analyzed the positioning accuracy of the robot reaching the target point during the cyclic navigation process. As shown in Figure 16, we set the starting point and the target point for the robot, and used tape to mark the initial pose state of the robot at the starting point and endpoint of the robot. Then, we set the robot to continuously cycle between two points, recording the position and attitude of each navigation to the target point.



Figure 16. The markers of the robot navigation point. The non-English words in the image: China Southern Power Grid.

According to the analysis of many experiments, after the mobile robot reached the target point, we used a tape measure to measure the actual position of the mobile robot, compared it with the position data in rviz, and took the maximum error, minimum error and average error in the 10 sets of data. The accuracy is shown in Table 4. We can see that the accuracy of the autonomous navigation of mobile robots meets the experimental requirements.

Table 4. Autonomous navigation accuracy.

Type	Maximum Error/cm(°)	Minimum Error/cm(°)	Average Error/cm(°)
X direction	8.5	4.0	5.5
Y direction	7.2	4.3	5.2
attitude	21.8	6.5	15.6

In order to ensure the safety of the robot, we analyzed its obstacle avoidance and stop function, as shown in Figure 17. Without affecting the LIDAR data, the robot autonomously bypasses the obstacle to avoid obstacles when the LIDAR scans an obstacle. The safety edge reads the collision signal and stops the robot from moving again when an obstacle that cannot be scanned by the radar is placed on the robot's navigation path and a collision occurs. After we manually remove the obstacle, the robot resumes navigation.

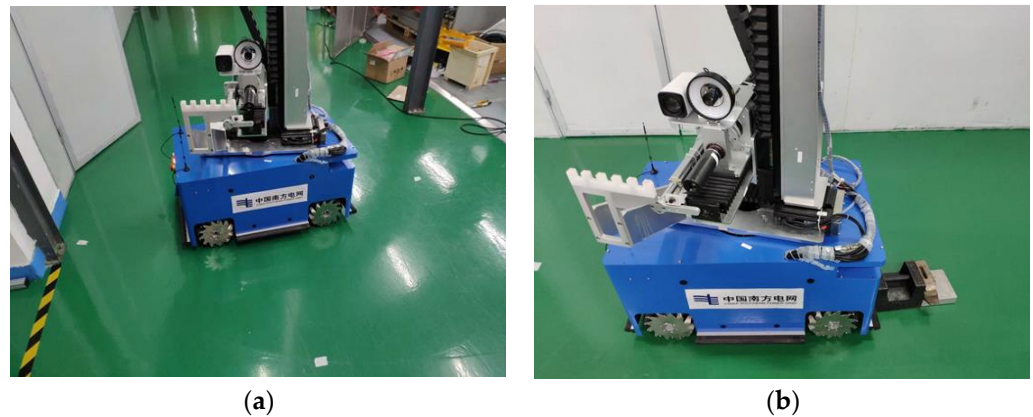


Figure 17. Robot obstacle avoidance and obstacle stopping experiment. (a) Robot obstacle avoidance; (b) robot stop barrier. The non-English words in the image: China Southern Power Grid.

Figure 18 shows the actual operation process of the robot, whereas Figure 18a shows the initial position and charging position; the initial and charging positions can be different, according to the tasks published by the server, and the robot autonomously navigates to the target point of the task assignment. Figure 18b shows the robot movement process. Figure 18c shows the robot reaching the goal spot. Through analysis, we know that the autonomous navigation system can meet the navigation and positioning requirements of the distribution switch cabinet operating robot. By using this platform, the practical value of the developed robot navigation system was verified. Finally, we tested and certified the complete functions and performance indicators of the robot through a third-party testing agency which meets the project indicators and requirements.

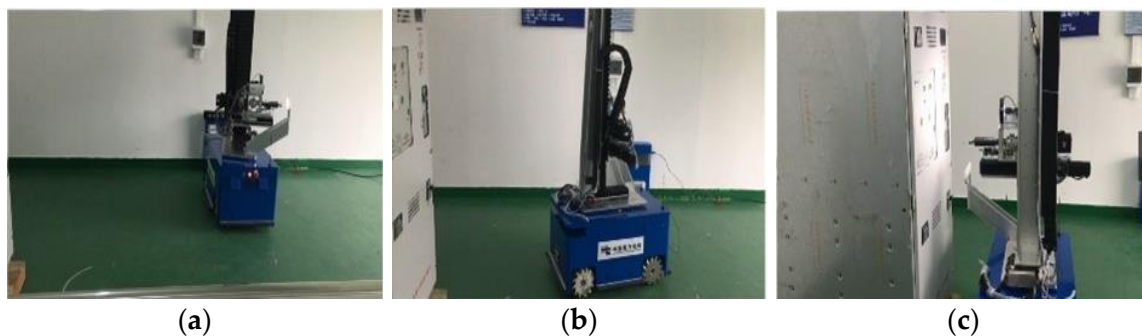


Figure 18. Automatic navigation process of the distribution switchgear robot. (a) Initial position and charging position; (b) the robot movement process; (c) the robot reaching the goal spot.

7. Conclusions

Real-time and accurate chassis navigation is a core function to enable substation robots to complete inspection tasks. The autonomous navigation system is a key technology of robot intelligence. This study focused on the research and implementation of the autonomous navigation system of an auxiliary operation robot in an indoor substation environment; our main achievements are as follows.

1. The three-dimensional model of the robot experimental chassis was developed and designed, and the robot simulation environment was built in ROS; the real navigation system experimental platform was built to prepare for navigation system research.
2. The autonomous navigation system has three parts: multi-sensor data fusion positioning and mapping, robot navigation path planning, and automatic stopping and obstacle avoidance. For multi-sensor data fusion positioning and mapping, the EKF algorithm was used for data fusion, and the Gmapping method was used to build the environmental maps. For the robot navigation path planning, the A* was used for global path planning, and the DWA algorithm was used for local path planning; the local path also realized the dynamic obstacle avoidance effect. For automatic stopping and obstacle avoidance, the safety edge sensor was used to guarantee the security of the robot as well as the surroundings, and to realize the stopping and obstacle function. According to the above research and analysis, the software development and design of the navigation system were completed with the help of ROS.
3. The designed navigation system was applied to a switchgear auxiliary operating robot, the physical structure parameters of the chassis were calibrated for the new navigation system, and experimental analysis was conducted on the complete autonomous navigation system. The results indicate that the developed system contains the indicators and demands of the distribution switchgear auxiliary operation robot project, and the practical value of the system was verified.
4. There are still deficiencies in the design of our current navigation system. When there is a fixed obstacle below the scanning range of the LIDAR, the mobile robot will stop at the obstacle, resulting in the inability to continue to run to the target point, which greatly affects the follow-up distribution box inspection work. In addition, the navigation system we designed is only suitable for structured and flat environments such as indoor substations, and it is difficult to apply in complex environments. Therefore, future research can solve the navigation disruption problem by adding a multi-line radar or a depth camera, and by adding a perception algorithm based on deep learning. Then, the wide applicability of the model can be improved by modifying the chassis structure to be more suitable for complex environments.
5. Our ultimate goal is to detect the target of the distribution box. The design of the indoor substation wheeled robot navigation system in this paper provides the basis for the subsequent target detection. In object detection algorithm [33] provides an effective and efficient framework to detect different growth stages under a complex orchard scenario which can be extended to the detection of different fruits and crops, disease detection, and different automated agricultural applications. They proposed a real-time object detection framework, Dense-YOLOv4, based on an improved version of the YOLOv4 algorithm including DenseNet in the backbone to optimize feature transfer and reuse. Furthermore, a modified path aggregation network (PANet) was implemented to preserve fine-grain localized information. Liu et al. [34] provided a comprehensive review of recently developed deep learning methods for small object detection. Firstly, they summarized challenges and solutions of small object detection, and presented major deep learning techniques. Then they discussed related techniques developed in four research areas, including generic object detection, face detection, object detection in aerial imagery, and segmentation. In addition, they also compared the performances of several leading deep learning methods for small object detection—including YOLOv3, Faster R-CNN, and SSD—based on three large benchmark datasets of small objects. The robot platform we designed can carry out the target detection

algorithm mentioned above. Therefore, in the future, the mentioned target detection algorithm can be burned into the robot platform we designed for testing, and can further complete the follow-up detection of apparent defects in distribution boxes.

Author Contributions: Conceptualization, J.R.; methodology, J.R., T.W. and X.Z.; software, A.Z., C.Y. and X.Z.; validation, J.R. and X.Z.; formal analysis, J.R.; investigation, J.R., C.Y., J.S. and H.J.; resources, H.J., M.L. and J.S.; data curation, J.R. and C.Y.; writing—original draft preparation, J.R.; writing—review and editing, T.W.; visualization, J.R. and X.Z.; supervision, T.W.; project administration, T.W.; funding acquisition, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sawada, J.; Kusumoto, K.; Maikawa, Y.; Munakata, T.; Ishikawa, Y. A mobile robot for inspection of power transmission lines. *IEEE Trans. Power Deliv.* **1991**, *6*, 309–315. [\[CrossRef\]](#)
2. Thrun, S. Probabilistic robotics. *Commun. ACM* **2002**, *45*, 52–57. [\[CrossRef\]](#)
3. Maxwell, W.L.; Muckstadt, J.A. Design of Automatic Guided Vehicle Systems. *AIIE Trans.* **2007**, *14*, 114–124. [\[CrossRef\]](#)
4. Arkin, R.C.; Murphy, R.R. Autonomous navigation in a manufacturing environment. *IEEE Trans. Robot. Autom.* **1990**, *6*, 445–454. [\[CrossRef\]](#)
5. Katz, Z.; Bright, G. A guidance technique for an automated guided vehicle. *Int. J. Adv. Manuf. Technol.* **1992**, *7*, 198–202. [\[CrossRef\]](#)
6. Fu, X.M. Research on Multi-Sensor Fusion Indoor Robot Inertial Navigation and Positioning. Master's Thesis, University of Chinese Academy of Sciences, Shenzhen, China, 2019. (In Chinese)
7. Xing, Y.Y. Research on GPS-based robot navigation technology. *Agric. Sci. Technol.* **2011**, 79–81,84. (In Chinese) [\[CrossRef\]](#)
8. Biswas, J.; Veloso, M. Depth camera based indoor mobile robot localization and navigation. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St Paul, MN, USA, 14–18 May 2012; pp. 1697–1702.
9. Gao, X. *Fourteen Lectures on Visual SLAM: From Theory to Practice*; Publishing House of Electronics Industry: Beijing, China, 2017. (In Chinese)
10. Xi, W. Development of Electric Power Inspection Robot Navigation System Based On Laser Scanning. Master's Thesis, Zhejiang University, Hangzhou, China, 2017. (In Chinese)
11. Guastella, D.C.; Muscato, G. Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review. *Sensors* **2020**, *21*, 73. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Seçkin, A.Ç. A Natural Navigation Method for Following Path Memories from 2D Maps. *Arab. J. Sci. Eng.* **2020**, *45*, 10417–10432. [\[CrossRef\]](#)
13. Yan, K.; Ma, B. Mapless Navigation Based on 2D LIDAR in Complex Unknown Environments. *Sensors* **2020**, *20*, 5802. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Santos, J.M.; Portugal, D.; Rocha, R.P. An evaluation of 2D SLAM techniques available in robot operating system. In Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linköping, Sweden, 21–26 October 2013; pp. 1–6.
15. Cho, B.S.; Moon, W.S.; Seo, W.J.; Baek, K.R. A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. *J. Mech. Sci. Technol.* **2011**, *25*, 2907–2917. [\[CrossRef\]](#)
16. Guo, Y.J. Theoretical Analysis and Experimental Research on Positioning Error Based on Mobile Robot GPS/INS Integrated Navigation Mode. Master's Thesis, Anhui University of Engineering, Wuhu, China, 2019. (In Chinese)
17. Moore, T.; Stouch, D. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems 13*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 335–348.
18. Zhou, W.; Hou, J.; Liu, L.; Sun, T.; Liu, J. Design and simulation of the integrated navigation system based on extended Kalman filter. *Open Phys.* **2017**, *15*, 182–187. [\[CrossRef\]](#)
19. Royer, E.; Lhuillier, M.; Dhome, M.; Lavest, J.-M. Monocular vision for mobile robot localization and autonomous navigation. *Int. J. Comput. Vis.* **2007**, *74*, 237–260. [\[CrossRef\]](#)
20. Zhao, H. Research on Path Planning Algorithm Based on Robot Vision Image. Master's Thesis, Hefei University of Technology, Hefei, China, 2010. (In Chinese)
21. Lee, S.J.; Cho, D.W.; Chung, W.K.; Lim, J.H.; Kang, C.U. Feature based map building using sparse sonar data. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 1648–1652.
22. Simhon, S.; Dudek, G. A global topological map formed by local metric maps. In Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190), Victoria, BC, Canada, 17–18 October 1998; pp. 1708–1714.

23. Henderson, T.C.; Linton, T. Raster map image analysis. In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009; pp. 376–380.
24. Grisetti, G.; Stachniss, C.; Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [[CrossRef](#)]
25. Yang, B.B. Research on Observation Structure and Inertial Navigation Method of Underwater Robot Hangzhou. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2008. (In Chinese)
26. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; p. 5.
27. Li, J.; Sun, X.X. Research on UAV path planning algorithm based on improved A-Star algorithm. *J. Ordnance Eng.* **2008**, *29*, 788–792. (In Chinese)
28. Seder, M.; Petrovic, I. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 1986–1991.
29. Fang, P.P.; Yang, J.F.; Shi, Y.Y.; Yu, L.Y. Unmanned vehicle obstacle avoidance method based on gradient descent method and improved artificial potential field method. *Manuf. Autom.* **2018**, *40*, 81–84. (In Chinese)
30. Zhang, F.; Zhou, S.; Zhang, J.; Gao, A.T. Robot obstacle avoidance method based on optimized vector field histogram method. *J. Sichuan Mil. Eng.* **2013**, *34*, 102–104. (In Chinese)
31. Yang, Q.; Qu, D.; Xu, F.; Zou, F.; He, G.; Sun, M. Mobile robot motion control and autonomous navigation in GPS-denied outdoor environments using 3D laser scanning. *Assem. Autom.* **2018**, *39*, 469–478. [[CrossRef](#)]
32. Zhu, D.S. Design and Implementation of ROS Indoor Service Robot Control System. Master’s Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China, 2019. (In Chinese)
33. Roy, A.M.; Bhaduri, J. Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4. *Comput. Electron. Agric.* **2022**, *193*, 106694. [[CrossRef](#)]
34. Liu, Y.; Sun, P.; Wergeles, N.; Shang, Y. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* **2021**, *172*, 114602. [[CrossRef](#)]