ROS-based Library Book Precise Positioning Guidance Robot

Zhiteng Wu, Yuanjia Ma*
School of Electronic Information Engineering, Guangdong University of Petrochemical Technology
Maoming, Guangdong, China
**Corresponding Author*

Abstract: In order to solve the problem of book search and positioning in the library, this paper introduces a library book precise positioning guidance robot based on robot operating system (ROS). The robot has functions such as synchronous positioning and mapping, autonomous navigation and obstacle avoidance. First of all, the library book precision positioning guidance robot uses sensors such as lidar and cameras to realize the perception of the library environment, and at the same time uses image recognition technology to identify books. Secondly, the two-dimensional raster map of the library is drawn by using the SLAM algorithm, and the global and local optimal path planning and obstacle avoidance function are realized by using the Move-base function package, so as to realize the navigation and positioning tasks in the library.

Keywords: Robot Operating System; Simultaneous Positioning and Mapping; Autonomous Navigation and Obstacle Avoidance

1. Introduction

With the development of society and the advancement of science and technology, libraries play an important role as places for knowledge dissemination and learning. However, with the continuous expansion of the scale of libraries and the increase of the number of books, the traditional way of searching and locating books can no longer meet the needs of readers. Therefore, the introduction of robot technology to achieve the precise positioning and navigation of library books has become a new idea for solving problems[1].

This paper aims to design and implement a library book precision positioning and guidance robot based on Robot Operating System (ROS) to improve the service level and efficiency of libraries. The robot has functions such as synchronous positioning and mapping, autonomous navigation and obstacle avoidance, senses the library environment through sensors such as lidar and cameras, and uses image recognition technology to complete the recognition of books. The two-dimensional raster map of the library is drawn by the SLAM algorithm, and the global and local optimal path planning and obstacle avoidance function are realized through the Move_base function package, so as to realize the navigation and positioning tasks in the library [2].

Compared with the existing library positioning system, this design has the following advantages: First, the robot can accurately and quickly search and locate the books required by the reader, which greatly improves the reader's experience and satisfaction. Second, the robot can navigate autonomously in the library, avoiding the trouble of readers getting lost in an unfamiliar environment[3]. In addition, with the help of the obstacle avoidance function, the robot can actively avoid collisions with people and obstacles, ensuring safety and reliability.

This article will introduce the hardware architecture and software system design of the robot respectively, and elaborate on the positioning, navigation and obstacle avoidance algorithms of the robot. Finally, the feasibility and performance of the system are verified by experiments.

Through the introduction of the library book precise positioning guidance robot, we believe that we can improve the management level of the library, provide readers with more convenient and efficient book search and positioning services, and further promote the construction and intelligent development of library informatization.

2. Hardware System Design

2.1 Library Books are Precisely Positioned to Guide the Robot Hardware Composition

In the realm of robotics, precision holds the key, particularly when it comes to the critical task of positioning library books. A standout tool in this domain is the Robot Operating System (ROS), a widely acclaimed open-source platform. ROS offers a multitude of function packages meticulously crafted for the purpose of robot control and management. When it comes to crafting robot programs, ROS not only allows for parameter adjustments but also offers the flexibility of direct referencing, as highlighted by Martinez and Vega[4]. In the context of our library, a cutting-edge robotic system has been meticulously engineered to achieve pinpoint accuracy in book positioning and guidance. At its core lies the Raspberry Pi 4B, which assumes the role of the upper-level controller, overseeing the array of functions performed by the robot. Directing the motors and ensuring the precise movements of the book-bearing trolley is the STM32 microcontroller, which serves as the heart of this intricate system.In the development of our library's robotic assistant, we harnessed the potential of the Gmapping function package, integrating state-of-the-art Simultaneous Localization and Mapping (SLAM) algorithms. This cutting-edge technology empowered us to craft intricate 2D raster maps, laying the foundation for advanced functionalities. Building upon this foundation, we meticulously implemented both global and local optimal path planning, coupled with autonomous obstacle avoidance capabilities. Figure 1 beautifully showcases the hardware system resulting from our endeavors, a testament to our commitment to precision and guidance in the library. This intricate robotic system, featuring ROS at its core, embodies the potential for highly accurate and autonomous book positioning, setting a new standard for library automation and efficiency.

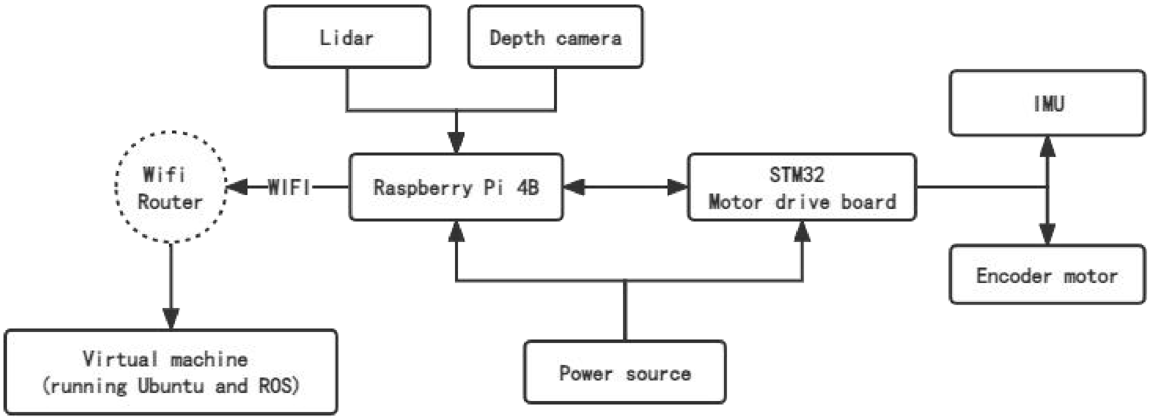

Figure 1. Library Books Are Precisely Positioned to Guide the Hardware System

The library's precision positioning and guidance robot employs a collaborative debugging methodology, seamlessly connecting the robot-side and remote workbench-side.On the robot side, a Raspberry Pi equipped with a ROS system serves as the powerhouse. It coordinates various essential components, including sensor drivers, mapping and navigation programs, and a host of other functionalities. This single-board computer is the engine that drives the robot's actions, ensuring the seamless execution of tasks with precision.In contrast, the remote workbench operates with the dexterity of a laptop running Ubuntu and ROS within a virtual machine. This virtual environment is a hub of control and monitoring, empowered by SSH remote login software and rviz graphics tools. These tools grant operators the ability to take charge of the robot's algorithm programs, facilitating real-time debugging and control. To enable remote control, a robust WiFi connection bridges the gap between the robot and the remote workbench. This connection leverages the distributed network communication capabilities of the ROS system for the smooth transmission of data. This setup creates a dynamic communication channel, where the remote workbench not only directs the robot with precise control instructions but also receives valuable feedback data in real-time. In the intricate world of library book positioning and guidance, our reliance on Lidar and depth cameras for environmental perception is paramount. Lidar is equipped with advanced Simultaneous Localization and Mapping (SLAM) capabilities, which enable real-time two-dimensional mapping of the library's surroundings. This mapping is foundational to the robot's understanding of its environment, crucial for its precise navigation. Simultaneously, the depth camera captures live video feeds from the library, a vital function that enables real-time book identification and positioning at the application layer. This real-time video stream is the robot's "eyes" within the library, allowing it to identify and locate books with remarkable accuracy. Complementing these systems, the robot's hardware is equipped with a motor drive board, a small yet pivotal component responsible for translating instructions from the Raspberry Pi into tangible movements. This board harnesses the power of the STM32 microcontroller and is well-versed in the art of PID (Proportional-Integral-Derivative) control, ensuring precise, closed-loop control over the DC coded geared motor's speed. Furthermore, data from a suite of sensors, including Inertial Measurement Units (IMUs) and motor encoders, are transmitted back to the Raspberry Pi. These sensors serve as the robot's ears and senses, providing critical information about its attitude and motion status. This intricate dance of technology and data ensures that the robot operates with utmost precision. The collaborative debugging approach between the robot-side and remote workbench-side not only guarantees precise positioning and navigation but also facilitates the fine-tuning and optimization of algorithms.

2.2 Lidar

In order to realize the function of accurately locating and guiding robots for library books, we chose lidar as the main sensor for environment perception. LiDAR uses infrared laser signals to obtain depth information of the surrounding environment, which can realize a fast and accurate mapping process, and has a high cost-effective mapping. We chose the A1 lidar as the robot's perception device. The lidar has the ability to scan 360 degrees, scanning the surrounding environment in real time and measuring distances. Its effective ranging radius reaches 12 meters, and the scanning frequency can reach more than 8000 times / second. This enables lidar to obtain rich environmental depth information and provide important data support for subsequent map construction and positioning. LiDAR collects depth information about the environment and combines SLAM (simultaneous localization and mapping) algorithms to fit and process the data to complete the map construction of the robot's area. As shown in Figure 2, the SLAM algorithm can use the environmental data obtained by lidar to accurately construct a map by establishing the relative relationship between the robot and the environment. By fusing other sensor data, such as IMUs and encoders, the accuracy and stability of maps can be further improved.

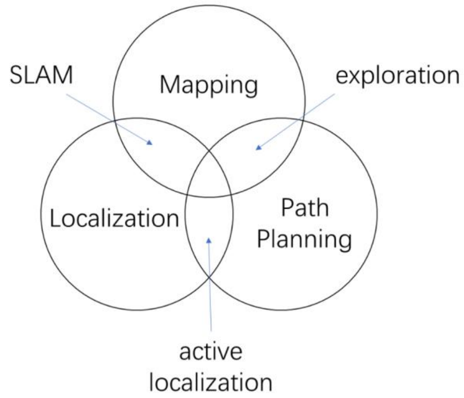

Figure 2. Conceptual Framework of the Study

2.3 Depth Camera

The depth camera is regarded as the "eye of the terminal and robot" in the library book precise positioning guidance robot, which can provide accurate depth information, help the robot perceive the library environment and realize the function of visual positioning and mapping at the same time. The depth camera can measure the distance between the points in the plane image and the camera, and combine the plane coordinates to obtain the three-dimensional coordinates of the points in the image corresponding to the actual scene.

In this system, we chose the ORBBEC Astra Pro depth camera as the primary vision sensor. The camera has a high-resolution RGB camera capable of providing clear image data. At the same time, the camera is also equipped with a depth sensor, which can obtain information about the distance between the object and the camera.

Through depth cameras, robots can capture image data from libraries in real time to identify and locate books. Using the information of depth images, we can reconstruct and map the location of books in the library in 3D. At the same time, the depth camera can provide accurate distance information, providing an important reference for the robot's navigation and obstacle avoidance.

2.4 Motor Drive Board

The motor drive board at the heart of the library's precision positioning guidance robot serves as an economical and fully integrated motor control solution. Its versatility extends to a wide range of applications, including the precise control of ROS robot differential chassis, the empowerment of intelligent remote-controlled vehicles, and the facilitation of coded motor systems. With a keen focus on compatibility, this motor drive board readily supports the GM37 coded geared motor and offers IO expansion pins to accommodate high-power coded motor requirements. Notably, it is complemented by a dedicated ROS host computer driver, granting users seamless control within the ROS environment and access to invaluable coded odometer information.Constructed around the STM32 microcontroller and harnessing the power of the TB6612 motor driver chip, this motor drive board integrates advanced features. Utilizing the PID algorithm, it attains precise closed-loop control over the speed of the DC coded geared motor. Furthermore, this versatile board has the capability to receive motion control instructions from the host computer via serial communication, providing precise motor encoding mileage feedback in real-time.The connection between the motor drive board and the host computer is facilitated by a USB cable, utilizing serial port communication to ensure swift and accurate data transfer. The extensive feature set of the motor drive board's driver includes chassis motor control, innovative odometer solutions, customizable PID parameter settings, and velocity calibration for both odometer lines and angular movements. This array of functionalities ensures consistent motor control and the acquisition of precise odometer data, thereby underpinning the core functions of the library's precision positioning guidance robot.

3. Software System Design

3.1 Software System Composition

The software system of this system is divided into three layers: application control layer, navigation control layer and perception execution layer. The overall software system framework is shown in Figure 3, where ROS Master acts as the ROS scheduling hub and plays the role of server, while other nodes can be regarded as publishers or subscribers, communicating through topics or services[5].

3.1.1 Application control layer

This layer is located on the Ubuntu operating system host computer applications, which are connected to the library robot through SSH remote login. In the visual application interface, users can obtain relevant information about the robot in real time and control the robot to carry out functions such as mapping and navigation.

3.1.2 Navigation control layer

This layer realizes communication and control between upper and lower layers. It is responsible for receiving instructions from the application control layer and passing the instructions to the appropriate nodes in the aware execution layer. At the same time, the navigation control layer also receives the environmental information and data transmitted by the perception execution layer and sends it to the application control layer for display or further processing.

3.1.3 Perception execution layer

This layer includes motion control nodes, environment awareness nodes, video image transmission nodes, and lidar. The motion control node is responsible for controlling the motor drive and realizing the movement of the robot. The context-aware node is used to obtain information and data about the surrounding environment to support the robot's autonomous navigation and obstacle avoidance functions. The video image transmission node can collect image information of the robot's environment for remote monitoring or other applications. Lidar, on the other hand, builds a map of the area where the robot is located in real time to provide data support for the robot's navigation and obstacle avoidance.

Through the above level division, the system design can realize the precise positioning and navigation function of the robot. The application control layer provides a user-friendly interface that allows the operator to obtain information about the robot in real time and control the behavior of the robot. The existence of a navigation control layer enables efficient communication and coordination between nodes. The perception execution layer realizes key tasks such as motion control, environment perception, video image transmission and lidar map construction of the robot through the functions of each node.

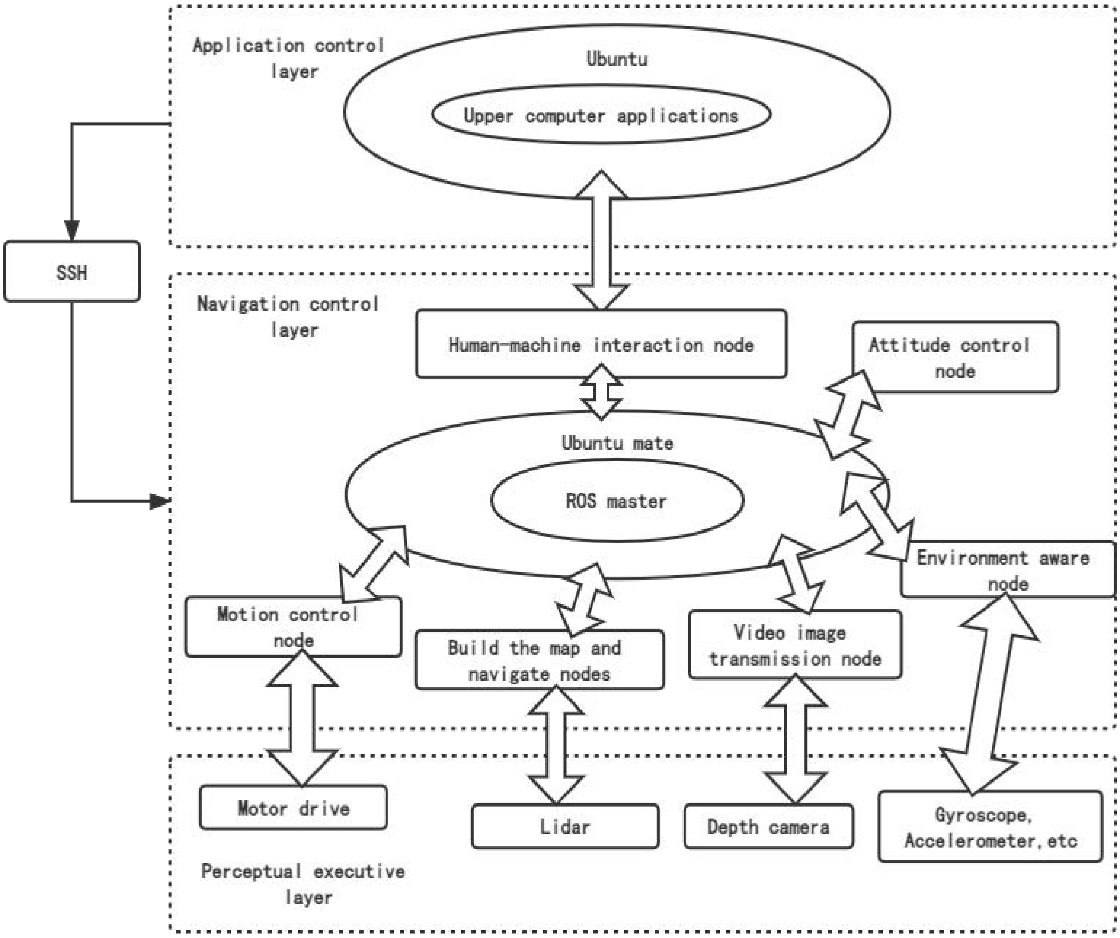

Figure 3. Wheel Layout of Typical Omni-directional Platform

In Figure 3, the robot acquires environmental data through the perception execution layer. The navigation control layer facilitates communication between the application layer and the lower layers, enabling the transmission of data and commands. Users can easily monitor the robot through the application control layer, without the need to directly interact with the lower layers.

3.2 Slam Algorithm

Simultaneous localization and mapping (SLAM) is a technology used to solve the real-time positioning, map construction, and navigation of robots in unknown environments. In order to achieve SLAM functionality, robots need to have the ability to perceive depth information about their surroundings, and lidar sensors are the key to achieving this function. At present, ROS robots have integrated a variety of mapping algorithms, such as Gmapping, Hector and Cartographer. In this project, we used lidar sensors for localization and mapping, and selected the Gmapping algorithm to build SLAM maps[6].

Gmapping is an open-source SLAM algorithm based on the Rao-Black Particle Filter (RBPF) algorithm. The algorithm separates the localization and mapping processes. First, the particle filter algorithm is used for localization, and then the laser sensor model is used to quantify and optimize the matching degree of the current particle with the generated map. Then, the error is continuously corrected by the observation information of the sensor (such as IMU, odometer, etc.), so as to achieve accurate update of the map[7]. Each particle carries a map, which increases the computational effort and memory footprint of the algorithm when building large maps. Therefore, the Gmapping algorithm shows high accuracy and good robustness in indoor small scenes and low-feature environments, while the amount of computation is small[8].

The Gmapping feature package generates 2D raster maps by subscribing to the robot's odometer, depth and IMU information and configuring the necessary parameters[9]. The following is the specific mapping process of the Gmapping feature package: "Start the radar and Gmapping node, odom information release"→ "start the Gmapping node"→ "start rivz, display pose, node, map data" → "map save", and finally draw a 2D raster map[10].

3.3 Autonomous Navigation and Obstacle Avoidance

In the SLAM algorithm, the robot can calculate its pose in space under a reference frame based on the received information about its surroundings and generate a 2D raster map. This provides a blueprint for autonomous navigation and obstacle avoidance by robots. The move_base function package is mainly used for robot path planning, which can simultaneously set the target position and direction of the robot in a certain reference frame and realize the obstacle avoidance function. Path planning for a robot is usually divided into two steps. First, the Global Planner feature is included in the move_base to calculate the optimal route from the starting point to the destination. Secondly, the Local Planner function is included in the move_base to solve the local path planning problem when the robot encounters obstacles and other situations on the actual road. In addition, the Monte Carlo Localization Method (AMCL) is used to calculate the robot's position in the map. The AMCL algorithm adopts the idea of particle filtering to estimate the probability distribution of robot pose by randomly placing a group of particles on the SLAM map and scoring these particles according to the detection information and data of the robot sensor. Through multiple iterations, the particles are clustered at an optimal estimation point, which represents the robot's position estimate. In this design, the A algorithm is used to implement global path planning, which is faster and more efficient than the earlier Dijkstra algorithm, which has been integrated in ROS with complete packaging and ease of use. The local path planning is realized using the Dynamic Window Approach (DWA), which realizes the autonomous obstacle avoidance function of the robot by evaluating multiple sets of velocity samples and selecting the route with the best evaluation result[11].

4. Experimental Testing

Figure 4 depicts a visual snapshot of a navigation experiment conducted within an indoor environment, offering a fascinating glimpse into robotic exploration. The image in the lower right corner captures the world from the perspective of a depth camera located at the experimental site, showcasing the technology at work. During the course of this experiment, a series of meticulously coordinated actions took place. The initial step involved establishing network communication with the robotic platform, known as the 'bot.' This connection was achieved through SSH login on a remote workbench, a practice that enables remote oversight and control. Following this, a pivotal sequence of events unfolded. All the sensor drivers residing on the robot were remotely initiated, and concurrently, the Gmapping function pack was set into motion. This function pack, a component of the ROS ecosystem, plays a critical role in creating a 2D representation of the robot's surroundings. It's the first step in the process of building an indoor map. Moving up to the application control layer, the robot was placed under the direct guidance of a remote operator. With precision and care, the operator oversaw the robot's movements, directing it through the environment to meticulously construct the indoor map. This collaborative effort, combining technology and human control, laid the foundation for subsequent stages. Upon the successful completion of the indoor map, the stage was set for the robot's autonomy. The robot's navigation system, made accessible through the user-friendly Rviz interface of the trolley control software, was meticulously configured. This interface empowers users to set navigation paths and destinations, allowing the robot to autonomously traverse the environment while deftly avoiding obstacles that may lie in its path. The fruits of this experiment revealed remarkable capabilities. The robot, guided by a combination of sensor data, remote operator input, and advanced software, was able to achieve not only synchronous positioning mapping but also seamless automatic navigation. This intricate dance of technology and human interaction showcased the potential of robotics in indoor environments, where precision and adaptability are paramount.

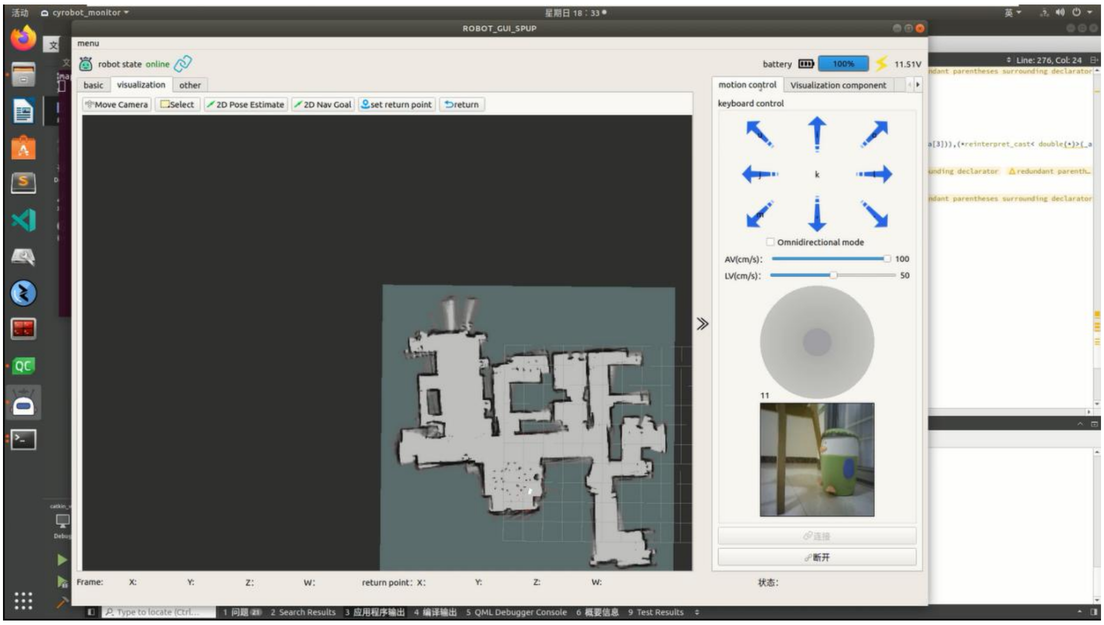

Figure 4. Comparison of Experimental Data

Through these experimental tests, the feasibility of the design and implementation of the Library Book Precise Positioning Guidance Robot is verified. The robot technology enables precise positioning of books in a library environment and guides the user to the location of the target book. The experimental results prove the effectiveness of the robot navigation and obstacle avoidance functions, and the robot can perform autonomous navigation in the indoor environment and complete the navigation task.

5. Conclusion

This paper presents a cost-effective solution for the precise positioning and guidance of library book robots. The solution utilizes Raspberry Pi 4B as the main controller and STM32 as the motion controller. Environmental data is collected using lidar and depth cameras, and the Gmapping mapping algorithm is employed for map construction. Additionally, the Move_base function package is utilized to plan paths on the 2D grid map, enabling automatic navigation.

With its focus on low cost, short development cycles, and strong scalability, this solution provides valuable insights and reference implementations for supermarkets, hospitals, banks, and other institutions seeking to deploy service robots.

The robot uses the Raspberry Pi 4B as the core controller, providing the robot with powerful computing power and flexible scalability. As a motion controller, STM32 controls the motor drive of the robot to realize motion control and navigation functions. Lidar and depth cameras are used to collect information about the surrounding environment, and 2D raster maps are constructed in real time through Gmapping mapping algorithms. Move_base function package is responsible for path planning, finding the optimal path on the map through optimization algorithms and passing them to the motion controller to realize automatic navigation and patrol functions.

The design is characterized by low cost, short cycle time and high scalability. Common hardware platforms and open source software reduce costs and enable rapid development and iteration. At the same time, the scheme can be expanded and customized according to actual needs to meet the needs of service robots in different fields such as supermarkets, hospitals, and banks.

This design scheme provides a reference case for service robots in related fields, and its low cost, short cycle time and strong scalability make it have a wide range of practical application prospects.

Acknowledgment

This article was supported by the Innovation and Entrepreneurship Project of Guangdong University of Petrochemical Technology.

References

[1] Chen MengYuan.(2018).The SLAM Algorithm for Multiple Robots based on Parameter Estimation. Intelligent Automation and Soft Computing(3).

[2] Sun Jian,Zhao Jie,Hu Xiaoyang,Gao Hongwei & Yu Jiahui.(2023).Autonomous Navigation System of Indoor Mobile Robots Using 2D Lidar. Mathematics(6).

[3] Wang Jun & Takahashi Yasutake.(2019).Particle Smoother-Based Landmark Mapping for the SLAM Method of an Indoor Mobile Robot with a Non-Gaussian Detection Model. Journal of Sensors.

[4] Martinez Javier & Vega Julio.(2022).ROS System Facial Emotion Detection Using Machine Learning for a Low-Cost Robot Based on Raspberry Pi. Electronics(1).

[5] Shih Jhih Syuan & Lin Ta Te.(2013).Reconstruction of 3D Natural Field Scenes Based on SLAM Robot. IFAC Proceedings Volumes(4).

[6] Tang Liang,Jiang Yunliang & Lou Jungang.(2016).Reliability Architecture for Collaborative Robot Control Systems in Complex Environments. International Journal of Advanced Robotic Systems(1).

[7] Hu Tianjiang,Zhao Boxin,Tang Dengqing... & Shen Lincheng.(2016).ROS-based ground stereo vision detection: implementation and experiments. Robotics and biomimetics(1).

[8] Ramil Safin, Roman Lavrenov,Edgar A. Martínez-García & Evgeni Magid.(2018).ROS-based Multiple Cameras Video Streaming for a Teleoperation Interface of a Crawler Robot. Journal of Robotics, Networking and Artificial Life (JRNAL)(3).

[9] Luo Jingwen & Qin Shiyin.(2023).An interval SLAM algorithm for leg ‑ arm mobile robots based on a CBMeMBer filter with Gaussian indicator box-particles. Measurement Science and Technology(9).

[10] Liu Weilian.(2021).Slam algorithm for multi-robot communication in unknown environment based on particle filter. Journal of Ambient Intelligence and Humanized Computing(prepublish).

[11] Li Jinyu, Yang Bangbang, Chen Danpeng, & Bao Hujun.(2019).Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality.Virtual Reality & Intelligent Hardware(4).