ISCTE IUL

Instituto Universitário de Lisboa

Department of Information Science and Technology

SLAM research for port AGV based on 2D LIDAR

Zeyu Ma

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master in Telecommunications and Computer Engineering

Supervisor:

Dr. Octavian Adrian Postolache, Associate Professor with Habilitation

ISCTE-IUL

Co-supervisor:

Dr. Yongsheng Yang, Full Professor,

Shanghai Maritime University

October, 2019

Resumo

Com o aumento do comércio internacional, o transbordo de mercadorias em portos internacionais de contentores é muito movimentado. O AGV (“Automated Guided Vehicle”) foi usado como uma nova geração de equipamentos para transporte horizontal de contentores de forma automatizada. O AGV é um veículo não tripulado automatizado que pode funcionar 24 horas por dia, aumentando a produtividade e reduzindo os custos de mão-de-obra em comparação com o uso de camiões porta-contentores. A capacidade de obter informações sobre o ambiente circundante é um pré-requisito para o AGV concluir automaticamente tarefas na área portuária. Atualmente, o método de AGV baseado no posicionamento e navegação de etiquetas RFID apresenta um problema de custo excessivo. Nesta dissertação foi realizada uma pesquisa sobre a aplicação da tecnologia LIDAR de localização e mapeamento simultâneo (SLAM) num AGV. Uma plataforma de teste móvel baseada num telémetro a laser é desenvolvida para examinar o ambiente em redor em 360 graus (distância e ângulo), centrado no LIDAR, e fazer *upload* da informação para uma base de dados em tempo real para gerar um mapa do ambiente em redor. Uma estratégia de prevenção de obstáculos foi também desenvolvida com base nas informações adquiridas. A eficácia da plataforma foi verificada através da realização de testes com vários cenários e obstáculos. Por fim, com base na primeira plataforma, uma outra plataforma experimental com codificador e sensor IMU foi também desenvolvida. Nesta plataforma, a funcionalidade do SLAM é ativada pelo algoritmo *GMapping* e pela instalação do codificador e do sensor IMU. Com base no estabelecimento do ambiente circundante SLAM, foram realizadas as funções de planeamento de trajetória e prevenção de obstáculos pela plataforma.

Palavras-Chave—LIDAR, Port AGV, SLAM.

Abstract

With the increase in international trade, the transshipment of goods at international container ports is very busy. The AGV (Automated Guided Vehicle) has been used as a new generation of automated container horizontal transport equipment. The AGV is an automated unmanned vehicle that can work 24 hours a day, increasing productivity and reducing labor costs compared to using container trucks. The ability to obtain information about the surrounding environment is a prerequisite for the AGV to automatically complete tasks in the port area. At present, the method of AGV based on RFID tag positioning and navigation has a problem of excessive cost. This dissertation has carried out a research on applying light detection and ranging (LIDAR) simultaneous localization and mapping (SLAM) technology to port AGV. In this master's thesis, a mobile test platform based on a laser range finder is developed to scan 360-degree environmental information (distance and angle) centered on the LIDAR and upload the information to a real-time database to generate surrounding environmental maps, and the obstacle avoidance strategy was developed based on the acquired information. The effectiveness of the platform was verified by the experiments from multiple scenarios. Then based on the first platform, another experimental platform with encoder and IMU sensor was developed. In this platform, the functionality of SLAM is enabled by the *GMapping* algorithm and the installation of the encoder and IMU sensor. Based on the established environment SLAM map, the path planning and obstacle avoidance functions of the platform were realized.

Keywords—LIDAR, Automated Port Termina, AGV, SLAM.

Acknowledgements

Upon the completion of this thesis, I am grateful to those who have offered me encouragement and support during the course of my study. First, special acknowledgment is given to my respectable supervisors Professor Octavian Postolache and Professor Yongsheng Yang whose patient instruction and constructive suggestions are beneficial to me a lot. Second, particular thanks go to all the teachers and professors who have taught me for their instruction and generous support during these years. What is more, I would like to present my thanks to one of my special friend from whom I get tremendous love and encouragement as well as technical instruction. Finally, I would like to express the most heartfelt gratitude to my family members who have provided me great help and support.

Content

List of Figures

List of Tables

List of Acronyms

AGV	Automatic Guided Vehicle
CPU	Central processing unit
DWA	Dynamic window approach
EFK	Extended Kalman Filter
FTP	File Transfer Protocol
GPS	Global Positioning System
GPU	Graphics processing unit
HF RFID	High-Frequency Radio Frequency Identification
IMU	Inertial measurement unit
IOT	Internet of things
LIDAR	light detection and ranging
MPC	Model predictive control
ORB	Oriented FAST and Rotated BRIEF
PID	Proportional–integral–derivative
PLC	Programmable controller
PMD	Pixel-Mixed-Device Technology
PWM	Pulse Width Modulation
QC	Quayside container crane
ROS	Robotic Operating System
RPC	Remote procedure call
SD	Secure Digital
SDHC	Secure Digital High Capacity
SLAM	Simultaneous Localization and Mapping
SoC	Broadcom system on a chip
SSH	Secure shell
UAV	Unmanned aerial vehicle
USB	Universal Serial Bus
VSLAM	Visual Simultaneous Localization and Mapping
Wi-Fi	Wireless fidelity
WLAN	Wireless local area networking

WPANs	Wireless personal area networks

Chapter 1 Introduction

1.1 Motivation

In recent years, with the development of friendly relations between Portugal and China, expressed by blue partnership in 2017 to jointly build the Maritime Silk Road Economic Belt, the cooperation and development between two countries was developed. New requirements related ports have been appeared including in-port transportation capabilities and development of new automated containers terminals. In terms of improving port capacity, the following options are available: 1) Construction of a new automated container port; 2) Upgrade the traditional container terminal. At the same time, new generations of unmanned vehicles such as AGV (Automatic Guided Vehicle) have gradually replaced container truck as the main horizontal transport equipment in large international ports, greatly improving the efficiency of container transportation. However, there are some problems in the obstacle avoidance technology of AGV. In the AGV obstacle avoidance solutions, there is a scheme that is mature and widely used. That is High-Frequency Radio Frequency Identification (HF RFID) technology. This technology is mainly used to obtain position information by reading RFID tags embedded in the port operation area. However, although this technology is mature, there are some shortcomings. This technology is susceptible to electromagnetic interference. At the same time, because the tags are buried underground, they are difficult to change and cannot be quickly deployed. Moreover, in Shanghai Yangshan Port, 61,438 RFID tags are pre-buried, the construction cost is high and it is difficult to maintain. Follow the automated transformation of traditional container demand, RFID nail magnetic navigation technology is not suitable for automated transformation of traditional container terminal, with the development of IoT(Internet of Things) technology, and the coverage of wireless communication networks in the port area, a variety of sensors can be used in the port and data can be transmitted in real time via a data transfer network.

As an accurate distance measurement, the optical sensors including, light detection and ranging (LIDAR) sensor are very suitable for measuring the distance to obstacles around it. LIDAR is an optical measuring instrument with high measurement accuracy and is widely used in mobile platforms. Considering the environment of the port: large area, day and night work, electromagnetic interference, the LIDAR is an appropriate solution. Compared with ultrasonic sensors, microwave radars, cameras and other sensors, LIDAR has a long detection range, it can work at night, and it is not subject to electromagnetic interference. Additionally, LIDAR is a good solution for the environmental distance information acquisition problem in automatic port terminals with AGVs.

In the port, the working environment of the AGV is unmanned, dynamic and static. After receiving the mission command issued by the port management system, the AGV arrives at the designated location, loads or unloads the container, and arrives at another location to continue the loading and unloading work. In this process, static obstacles may appear: containers, dynamic obstacles: AGV at work. Therefore, in order to use the LIDAR to solve the obstacle avoidance problem of the port AGV, a remotely controllable platform was developed, loaded with LIDAR, to simulate the obstacle avoidance scenario encountered during the operation of the port AGV, and to test the performance of the LIDAR.

1.2 Objectives

The design and develop laser Simultaneous Localization and Mapping (SLAM) technology to the AGVs that operate in the port environment, was the main goal of the MSc paper. Thus, the development of mobile-capable laser SLAM test platform and test it to verify the effectiveness of the system was the main objective of the thesis. Additional objectives are following mentioned:

Development of a ranging system with LIDAR as the core sensor, with wireless information transmission ability and remote control. Development of scanning system based on stepper motor and mechanical platform able to achieve 360-degree scanning. The displacement of scanner is assured by DC motors controlled by Raspberry Pi computation platform.

Software development including:

1) Data acquiring and processing capability:

The ability to read and store data sequentially in a 0-360 degrees perspective, the ability to send data to a firebase database in real time, the ability to visualize the angle-distance data into an environmental map.

2) Platform motion control

Development of control software that provides the ability to control the motion platform based on sensor information;

Development of specific obstacle avoidance strategy based on angle-distance data.

Perform tests regarding obstacle avoidance using the platform with LIDAR that emulates the AGV behavior in port environment.

1.3 Structure of the dissertation

Chapter 2 includes the literature review about modern automated ports, also sensors for distance measuring in the port and the applicable scenarios for sensors' usage, Simultaneous positioning and map construction (SLAM) technology are also mentioned. Chapter 3 introduced the entire development process of the first experimental platform and platform characteristics. The 4th chapter describes new platform with encoder and IMU sensor, associated the Simultaneous positioning and map construction (SLAM) function and the obstacle avoidance algorithm DWA (dynamic window approach). The Chapter 5 presents the experimental results and discussion. The 6th chapter includes the conclusions and the future work.

Chapter 2 State of the art

This section describes the research area of the project. First, it introduces the development of container port and the specific operation process of automated container port. Then is introduced the description of several distance measuring sensors. The generally used communication protocol, the robot operating systems that are commonly used are also mentioned.

2.1 Automated Container Port

General speaking, the container terminal [1] is a two-way logistics node with the function of storing containers. It connects sea and land through the usage of two interfaces. These interfaces are the dock area for loading and unloading ships, and the landside for loading and unloading trucks and trains. Containers are stored in the yard to facilitate separation of the terminal and landside operations [2]. After the container ship arrived at the port, it was assigned to a berth equipped with a QC (quayside container crane) to load and unload the container. The unloaded import container is transported to the dock location near the transfer point. Containers arriving at the terminal by road or rail are handled in the truck and train operating area. They are picked up by internal devices and distributed to the assigned locations in the yard. In the port area, the horizontal handling of containers is carried out by container trucks, which are only active in the port area and are responsible for loading the imported containers from the sea side cranes, transporting them to the yard, and also responsible for transporting the containers from the yard to the sea side.(Figure 2.1)

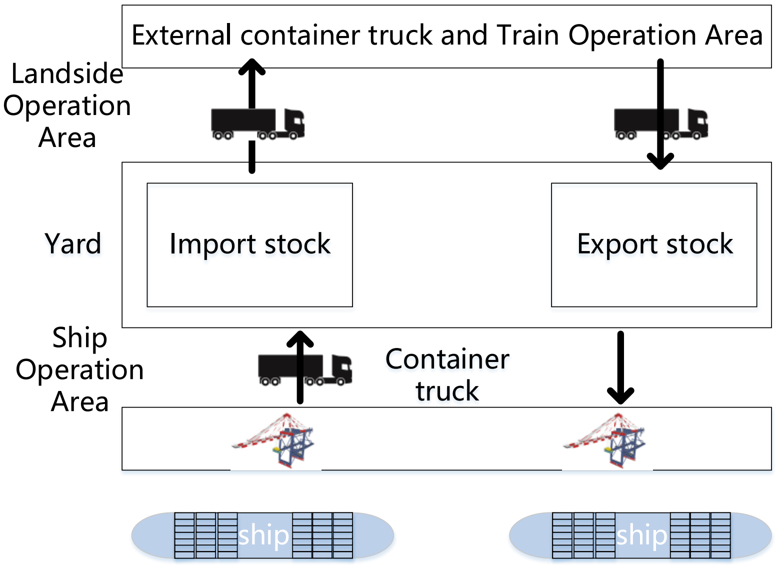


Figure 2.1 - Container terminal operation block diagram

With the rapid development of science and technology, the theoretical basis of automated science and technology is also constantly developing. New technologies formed on the basis of following technologies microelectronics technology, power electronics and energy conversion technology, computer control technology, network and radio communication technology, etc. These technologies [3] have changed the field of automation. New solutions appear in the port, such as new motor speed control equipment which is applied to the terminal loading and unloading machinery using Pulse Width Modulation (PWM) and advanced control equipment: based on programmable logic controller (PLC).

AGV is an unmanned automatic handling equipment, which has been gradually applied to container automatic handling in large international ports in recent years. AGV is a high-tech vehicle that works completely under computer control. It is fully automated equipment and provide automatic container handling functionalities. It can effectively reduce labor intensity and save manpower. In the working process of the AGV system, each step has a series of digital information communication exchanges, and the background has strong database support, eliminating human interference factors, fully ensuring the reliability of the AGV operation process, the timeliness of completing tasks, and the accuracy of information data. Moreover, AGV can not only work independently, but also has good integration and compatibility with other management systems. The AGV can work in situations where manual operations are difficult, such as the AGV can be drilled under the container for lifting operations. As shown in Figure 2.2, the crane is loading a container onto the AGV.

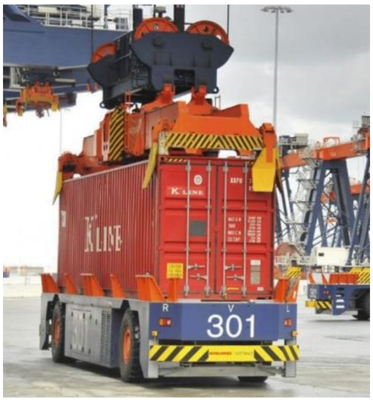


Figure 2.2 - The AGV is loading a container

2.2 Sensor Solution

Today, Internet of Things (IoT) technology is rapidly evolving, and as port automation requirements increase, various types of sensors are gradually being applied in ports. Considering the complexity of the working environment of the port, there may be scenarios where the humidity is high and the visibility is low, which is not suitable for the sensor work. AGV needs to work continuously, which requires a well-performing, stable operating obstacle avoidance solution that can be applied in a port environment. In this article [4], the authors describe the requirements of each of the world's multiple ports, as well as the corresponding sensor solutions in this work LIDAR was chosen as the experimental object, but many sensors are also used in the port. The characteristics and defects of various sensors, and why the choice of LIDAR is described in detail below:

The sensor solutions used in the field of unmanned vehicle navigation and positioning are: magnetic sensor, camera, infrared, ultrasonic, LIDAR, microwave radar. Various solutions have their own suitable occasions and advantages and disadvantages. In [5, 6] they describe the performance evaluation of obstacle detection and segmentation algorithms for automatic guided vehicle (AGV) navigation using a 3D real-time ranging camera. In [5] the author verified measurements obtained using a partially accurate 2D scanning laser range finder. In [6] they use Pixel-Mixed-Device Technology (PMD) camera to measure the obstacle, PMD offers a small, light-weight camera generating 3-D images, based on time-of-flight. The authors analyzed the sensor characteristics and application potential of mobile robots and discussed specific aspects related to parameter optimization and measurement limitations based on performance testing. In [7], a navigation and control system that uses a differentially driven magnetic point to guide the AGV is described. In addition, Hall effect sensors, encoders and counters are used for control and continuous guidance of AGV The specific implementation of the magnetic spot navigation method in Shanghai Yangshan Port is introduced in this article [8]. Combining engineering practice, the author focuses on the location method of port AGV, and proposes an analysis method of static control under wide-range conditions, using GPS static control and total station polar coordinates for positioning. The author verifies that the AGV navigation system meets the accuracy requirements of positioning through analysis and calculation methods. In this doctoral thesis [9], a prototype forklift AGV research platform capable of operating in an indoor environment was developed. The platform is equipped with a camera, laser scanner, IMU and a powerful on-board computer. Can be used for fully automated research. In [10], an extended Kalman filter (EKF) method was proposed for the problem of positioning of automated guided vehicles (AGVs) based on road sign detection, odometer and IMU measurements. This method has been tested on an AGV experimental platform equipped with various sensors. As an inexpensive and simple distance measurement sensor, the ultrasonic sensors are widely used in unmanned vehicles. In this paper [11], The authors designed an artificial neural network with supervised learning for classification and pattern recognition by using data collected by ultrasonic sensors. And apply the algorithm to agricultural machinery to improve the efficiency of precision agriculture. A method for obtaining obstacle information by filtering and clustering LIDAR point cloud data is reported in [12]. In addition, the method generates the forward angle and velocity of the robot based on the principle of least cost function. This article [13] proposes a method based on sensors to avoid local obstacles. Obstacle detection and obstacle avoidance algorithms have the advantages of simple data model and good real-time performance. In this paper [14], the authors discuss the use of 77 GHz millimeter-wave radar as a guidance sensor for autonomous land vehicle navigation. In addition to maintaining vehicle status, the Extended Kalman Filter (EKF) is also used to preserve estimates of map features. The polarization of the returned radar signal is used to identify natural features. This method does not require prior knowledge of the environment or the target infrastructure.

Considering the port environment, the sensors are compared and their advantages and disadvantages are listed in the Table 2.1.

Table 2.1 – Advantages and Disadvantages of sensing solutions in automated port

sensor solution	**advantage**	**disadvantage**
Magnetic sensors	High precision independent of temperature and humidity	Need to pre-lay magnetic spot, high construction and maintenance costs
camera	Efficient, precise, low cost	It is greatly affected by the light weather and does not work well at night.
infrared	High precision	Easy to be disturbed and short propagation distance
Ultrasonic	Compact, lightweight, cheap and good rangefinder	Poor range measurement at long distances
LIDAR	High measurement accuracy, free from electromagnetic waves, light interference, wide measurement range, can work at night	Affected by weather (rain, fog), huge amount of information, high cost
microwave radar	High precision, medium cost	Affected by weather (rain, fog)

2.3Wireless Communication Protocols

In this work, Wi-Fi and SSH (secure shell) are used to implement remote control of the test platform. Elements about Wi-Fi (wireless fidelity) communication protocol are following presented.

2.3.1 Wi-Fi communication protocol

Wi-Fi is a family of radio technologies that is commonly used for the wireless local area networking (WLAN) of devices which is based around the IEEE 802.11 family of standards. Wi-Fi is a trademark of the Wi-Fi Alliance, which restricts the use of the term Wi-Fi Certified to products that successfully complete interoperability certification testing. [15] Wi-Fi uses multiple parts of the IEEE 802 protocol family and is designed to seamlessly interwork with its wired sister protocol Ethernet.

Devices that can use Wi-Fi technologies include desktops and laptops, smartphones and tablets, smart TVs, printers, digital audio players, digital cameras, cars and drones. Compatible devices can connect to each other over Wi-Fi through a wireless access point as well as to connected Ethernet devices and may use it to access the Internet. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can be as small as a single room with walls that block radio waves, or as large as many square kilometers achieved by using multiple overlapping access points.

The different versions of Wi-Fi are specified by various IEEE 802.11 protocol standards, with the different radio technologies determining the ranges, radio bands, and speeds that may be achieved. Wi-Fi most commonly uses the 2.4G Hz Ultra High Frequency (UHF) and 5G Hz Super high frequency (SHF) ISM radio bands; these bands are subdivided into multiple channels. Each channel can be time-shared by multiple networks. These wavelengths work best for line-of-sight.

There are many different versions of Wi-Fi: 802.11a, 802.11b, 802.11g, 802.11n (Wi-Fi 4[16]), 802.11h, 802.11i, 802.11-2007, 802.11-2012, 802.11ac (Wi-Fi 5[16]), 802.11ad, 802.11af, 802.11-2016, 802.11ah, 802.11ai, 802.11aj, 802.11aq, 802.11ax (Wi-Fi 6[16]), 802.11ay. In this article, the Wi-Fi version used is 2.4g bg mixed version.

Table 2.2 – Versions of Wi-Fi

generation	IEEE Standard	Maximum Link-rate
Wi-Fi 6	802.11ax	600-9608 Mbit/s
Wi-Fi 5	802.11ac	433–6933 Mbit/s
Wi-Fi 4	802.11n	72–600 Mbit/s

2.3.2 IEEE 802.15.4 - ZigBee

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network [17].

The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10–100 meters line-of-sight, depending on power output and environmental characteristics. Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. Zigbee is typically used in low data rate applications that require long battery life and secure networking (Zigbee networks are secured by 128-bit symmetric encryption keys.) Zigbee has a defined rate of 250 Kbit/s, best suited for intermittent data transmissions from a sensor or input device.

Zigbee was conceived in 1998, standardized in 2003, and revised in 2006. The name refers to the waggle dance of honey bees after their return to the beehive.

2.3.3 IEEE 802.15.1 - Bluetooth

Bluetooth technology [18]is an open global specification for wireless data and voice communications based on low-cost, short-range wireless connectivity to establish a special connection between fixed and mobile device communication environments. The essence is to establish a common radio air interface (Radio Air Interface) for the communication environment between fixed devices or mobile devices, and further combine communication technology with computer technology to make various 3C devices connect without wires or cables. In this case, mutual communication or operation can be achieved in a close range. Simply put, Bluetooth technology is a technology that uses low-power radio to transfer data between various 3C devices. Bluetooth operates in the global 2.4 GHz ISM (ie, industrial, scientific, medical) band, using the IEEE 802.15.1 protocol. As an emerging short-range wireless communication technology, it is driving the development of low-rate wireless personal area networks.

2.3.4 Conclusions

The selected wireless network is based on the platform's range of motion and the stability of data transmission. The working area of the AGV is in the port is very large (the working area of Yangshan Port is 2.14 square kilometers), which requires long range and stable real-time data transmission. Taking into account these factors, Wi-Fi can achieve full coverage in the port, with a stable transmission characteristics, compared to Bluetooth, ZigBee, more suitable for a small range of indoor environment, and no low power, low energy consumption restrictions, due to the above factors , Wi-Fi 2.4g bg mix version was chosen as the platform's wireless network protocol.

2.4 Network Protocols

2.4.1 Secure Shell (SSH)

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.[19] In this paper, SSH(secure shell) is used for the communication between PC and test platform. Typical applications include remote command-line login and remote command execution, but any network service can be secured with SSH.

SSH provides a secure channel over an unsecured network in a client–server architecture, connecting an SSH client application with an SSH server. The protocol specification distinguishes between two major versions, referred to as SSH-1 and SSH-2. The standard TCP port for SSH is 22. SSH is generally used to access Unix-like operating systems, but it can also be used on Microsoft Windows. Windows 10 uses OpenSSH as its default SSH client. [20]

SSH was designed as a replacement for Telnet and for unsecured remote shell protocols such as the Berkeley rlogin, RSH, and REXEC protocols. Those protocols send information, notably passwords, in plaintext, rendering them susceptible to interception and disclosure using packet analysis. The encryption used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the Internet, although files leaked by Edward Snowden indicate that the National Security Agency can sometimes decrypt SSH, allowing them to read the contents of SSH sessions.

2.4.2 File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network. FTP is built on a client-server model architecture using separate control and data connections between the client and the server [21]. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content.

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[22,23] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as HTML editors.

2.4.3 Conclusions

The choice of network protocol is based on the platform's control method: point-to-point, in which the platform is controlled by another terminal. Real-time connection and stable information transmission are required. Considering that the ftp protocol needs to transmit information through the server, the SSH protocol is selected as the platform's network communication protocol.

2.5 Robotic Operating System (ROS)

In the experimental platform, Robotic Operating System (ROS) is adopted as the software platform, which is described as follows: The robotic operating system (ROS or ros) [24] is a robotic middleware (i.e. a collection of software frameworks for robot software development). Although ROS is not an operating system, it provides services designed for heterogeneous computer clusters, such as hardware abstraction, low-level device control, implementation of common functions, messaging between processes, and package management.

The primary design goal of ROS is to increase code reuse in the field of robotics. ROS is a distributed processing framework (Nodes). This allows executables to be designed separately and loosely coupled at runtime. These processes can be packaged into packages and stacks for easy sharing and distribution. ROS also supports a federated system of code bases. Collaboration can also be distributed. This design from the file system level to the community level makes it possible to independently decide on development and implementation. All of the above functions can be implemented by the basic tools of ROS. The data is transmitted between the node and the function package through the topic, and the ROS function package subscribes to the node information through the topic, and then implements the function. The structure of the system is as follows: the package subscribes to the topic to obtain data, and the node publishes the topic to send data. The relationship between nodes is presented in Figure 2.3.

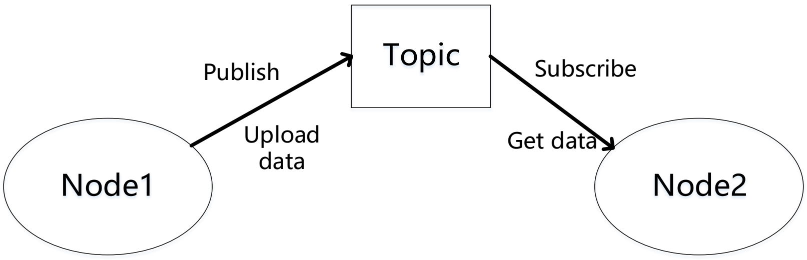


Figure 2.3 - Relationship between nodes in ROS

In order to achieve the primary goal of "sharing and collaboration," people have developed other supporting goals in the ROS architecture:

“Lightweight”: ROS is designed to be as convenient and convenient as possible. You don't have to replace the main frame with the system, because the code written by ROS can be used in other robot software frameworks. There is no doubt that ROS is easier to integrate with other robot software frameworks. In fact, ROS has completed integration with OpenRAVE, Orocos and Player.

ROS-agnostic library: The proposed development model is to write the ROS-agnostic library using the clear function interface.

Language independence: The ROS framework is easy to implement in any programming language. We've been able to run smoothly in Python and C++, with Lisp, Octave, and Java libraries added.

Simple test: ROS has a built-in unit/combination set test framework called "rostest". This makes integration debugging and decomposition debugging easy.

Scalability: ROS is suitable for large real-time systems and large-scale system development projects.

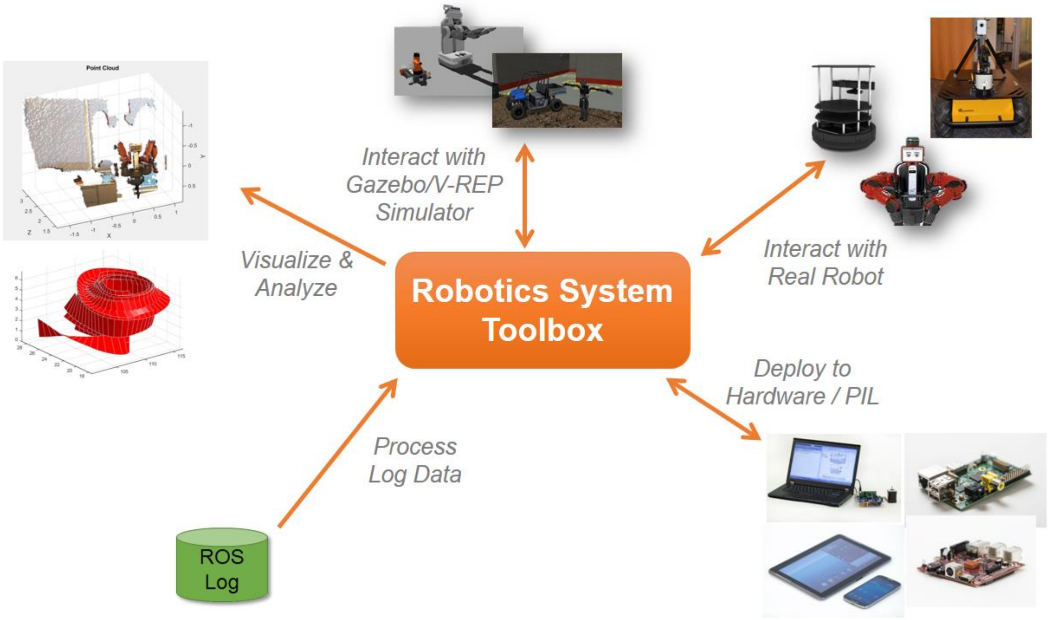


Figure 2.4 - The ROS applications

The ROS system can also be combined with MATLAB. With off-the-shelf algorithms and hardware connections that can be used to develop autonomous mobile robot applications, the Robotics Toolbox provides complete integration between interfaces and MATLAB and Simulink and Robotic Operating Systems (ROS). Now, robotics researchers and engineers can work in a single integrated design environment to design, test, and deploy robotic algorithms on ROS-enabled robots and robot simulators such as gazebos and V-REP. By reducing the manual conversion from MATLAB to ROS, this frees up more time to focus on design exploration and iteration.

2.6 Simultaneous positioning and mapping (SLAM)

Simultaneous localization and mapping (SLAM) [25] are a concept: the robot is expected to start from an unknown location in the unknown environment, and locate its position and posture by repeatedly observing the map features (such as corners, columns, etc.) during the movement. And then build the map according to its own position increment, as to achieve the purpose of simultaneous positioning and map construction.

SLAM technology is the key technology to realize the mobility of automatic intelligent devices such as robots or AGVs. It is the basis for realizing its automatic control. SLAM technology is widely used in service robots, drones, AR / VR and other fields. The current common SLAM system has two forms: LIDAR based SLAM (laser SLAM) and vision-based SLAM (Visual SLAM or VSLAM).

In this article [26], the authors propose a monocular SLAM system for Robot City Search and Rescue (USAR), using a variable to simulate a simulated disaster in environments. Combined the monocular SLAM with a 2D LIDAR SLAM to realize a 2D mapping and 6D localization SLAM system. This article presents a new approach to improving the positioning of particle filters. This paper [27] presents a new method to improve the positioning of particle filters. This method uses the CGR algorithm to extend and extend the traditional RBPF SLAM, which can be applied to various robot sensors. Combining the grid graph structure and LIDAR, a general SLAM algorithm is proposed. In this paper [28], aiming at the shortcomings of 3dslam requiring a lot of calculations, a SLAM system based on 2.5D direct height map is proposed. The SLAM method was tested on the KITTI dataset and showed excellent performance compared to the existing LiDAR SLAM method. In this article [29], a software model based on the graph-based SLAM method is developed using LIDAR technology, which allows real-time direct and automatic control of mobile robots. This paper [30] introduces various ROS-based SLAM methods and analyzes their feasibility in mobile robot applications in a unified indoor environment. During the experiment of UGV prototype motion, the trajectories obtained by processing different sensor data (conventional camera, LIDAR, ZED stereo camera and Kinect depth sensor) were compared. In this article [31], the authors design a mobile robotic system that uses LIDAR to obtain observations. Based on the robotic operating system (ROS), the LIDAR is used to obtain the matching data of the 2D laser scanning, and the open source *GMapping* software package is used for SLAM, and the RVIZ (ROS 3D visualization tool) is used to realize the mapping in the unknown environment. In this paper [32], a LIDAR scan matching real-time positioning experiment based on reference beacons is designed and implemented. The trajectory and positioning results of the robot platform prove that the method can provide accurate and reliable navigation and positioning.

2.6.1 Laser SLAM

Laser SLAM [33] was originally based on distance-based positioning methods (such as ultrasound and infrared single-point ranging). The emergence and popularity of Light Detection and Ranging (LIDAR) is related to the fact that LIDAR makes measurement faster and more accurate. The information collected by the LIDAR presents a series of scattered points with accurate angle and distance information, which is called a point cloud. Generally, the laser SLAM system calculates the distance and attitude of the relative motion of the LIDAR by matching and comparing two-point clouds at different times, thus completing the positioning of the test platform itself. LIDAR distance measurement is relatively accurate, stable in environments other than direct sunlight, and point cloud processing technology is mature. At the same time, the point cloud information itself contains direct geometric relationships, making the path planning and navigation of the platform intuitive. The laser SLAM theoretical algorithm research is very mature, and the applied products are also very rich.

There are many SLAM algorithms, such as *HectorSLAM*, *Gmapping*, *KartoSLAM*, *CoreSLAM* and *LagoSLAM*. In [34], the author analyzes in detail the characteristics of various SLAM algorithms. Regarding the *HectorSLAM* algorithm, it is not a suitable choice because it cannot use mileage information and cannot be modeled in conjunction with IMU data. The *CoreSLAM* and *LagoSLAM* take up high CPU resources but can't get accurate results. There are two classic 2D SLAM algorithm: *gmapping* [35] and *hector* [36], the advantages and disadvantages of the two algorithms through table3.1.

Table 2.3 - Comparison of SLAM algorithms in ROS

Algorithms Name	*gmapping*	*hector*
Advantages	construction indoor maps in real-time, no loopback detection, small calculation amount and high precision. Make full use of odometer information, low LIDAR frame rate require and high stability.	No need for a odometer, can estimate 6-degree-of-freedom pose, can be adapted to air or ground unevenness.
Disadvantages	Extremely dependent on odometer information, not suitable for building large scene maps, no loop detection, so the map may be misplaced when the loop is closed.	The choice of the initial value has a great influence on the result, high requirement of LIDAR frame rate.

2.6.2Visual SLAM

The SLAM that uses the camera as an external sensing sensor is called a visual SLAM (vSLAM). The camera has the advantages of rich visual information and low hardware cost. The classic vSLAM system generally includes four main parts: front-end visual odometer, back-end optimization, closed-loop detection and composition.

Suppose the platform carries the sensor (camera) to move in an unknown environment. For convenience, the motion of a continuous time becomes a discrete time t=1,...k, and at these moments, the position of the platform is represented by x, then each moment The position is recorded as x1, x2...xk, which constitutes the trajectory of the platform. In terms of maps, the map is assumed to consist of a number of landmarks, and at each moment, the sensor measures a portion of the landmarks to obtain their observations. There are N landmark points, which are represented by y1, y2...yn. The positioning problem (estimation x) and the mapping problem (estimation y) are solved by the motion measurement u and the sensor reading z.

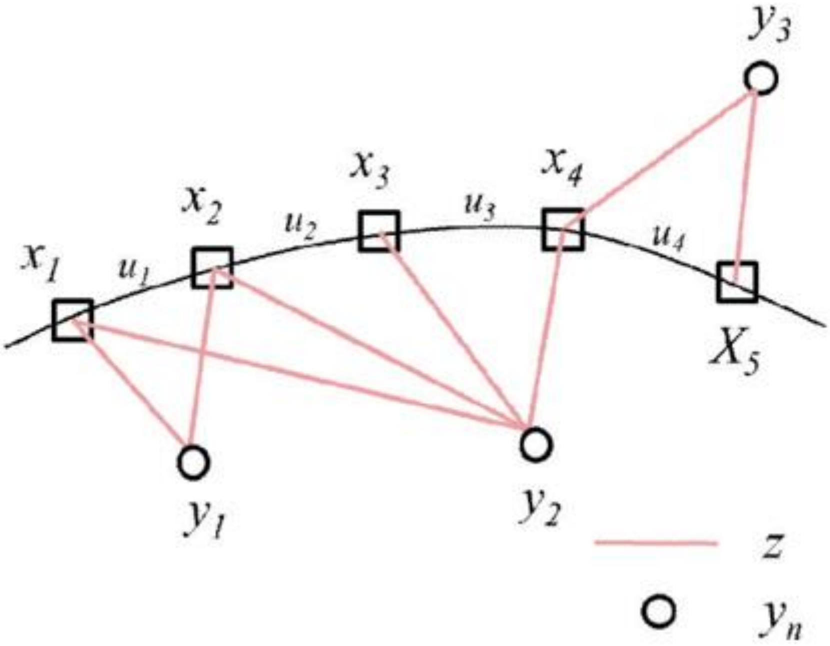


Figure 2.5 – The VSLAM system diagram

MonoSLAM is the first real-time monocular vision SLAM system [37]. *MonoSLAM* uses EKF (Extended Kalman Filter) as the back end to track the sparse feature points of the front end, and updates its mean and covariance with the camera's current state and all landmark points as state quantities. In EKF, the position of each feature point is Gaussian, and an ellipsoid can be used to represent its mean and uncertainty. The longer they are in a certain direction, the more unstable it is in that direction. The disadvantages of this method are: narrow scene, limited number of road signs, and easy to lose sparse feature points.

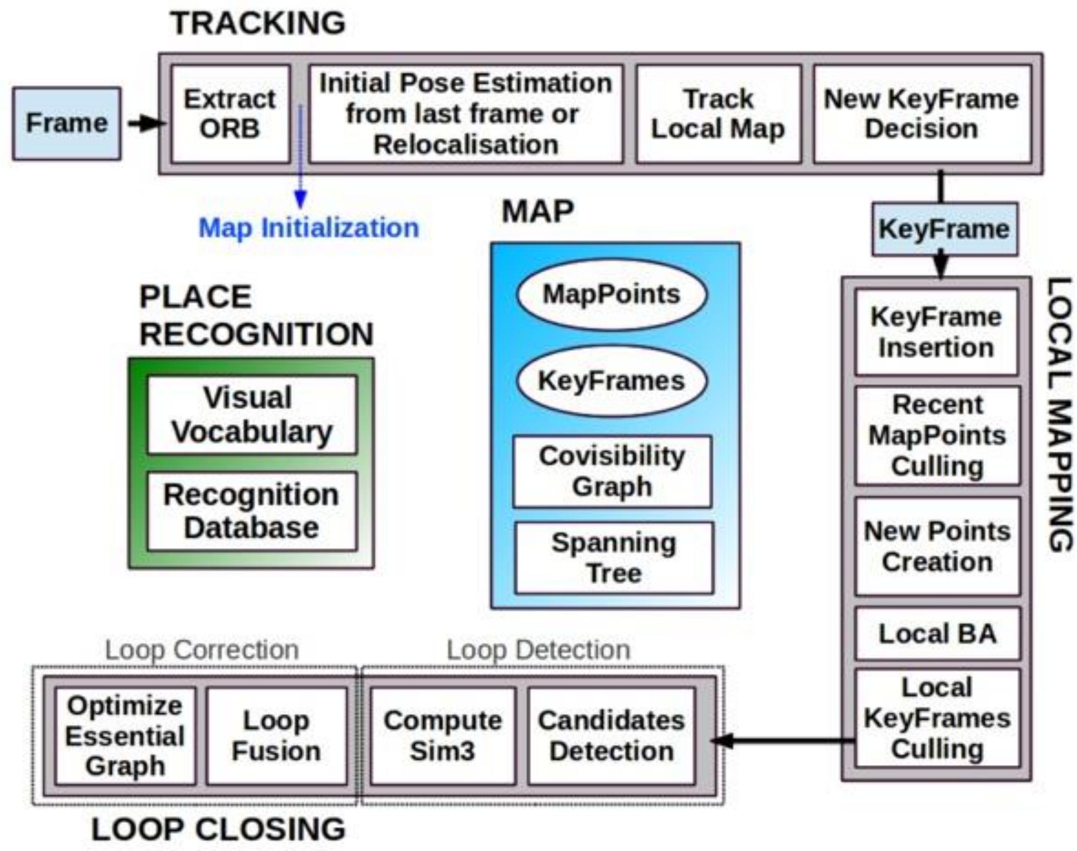


Figure 2.6 - The ORB-SLAM three-thread flow chart

The ORB-SLAM [38] is calculated around ORB features, including the ORB dictionary for visual odometer and loopback detection. Oriented FAST and rotated BRIEF (ORB) is a fast-robust local feature detector, first presented by Ethan Rublee et al. in 2011, [39] that can be used in computer vision tasks like object recognition or 3D reconstruction. It is based on the FAST key-point detector and the visual descriptor BRIEF (Binary Robust Independent Elementary Features). Its aim is to provide a fast and efficient alternative to SIFT. The ORB feature is computationally efficient compared to SIFT or SURF and has good rotation and scaling invariance. As is shown in Figure 2.6, ORB-SLAM innovatively uses three threads to complete SLAM. The three threads are: Tracking thread for tracking feature points in real time, optimized thread for local Bundle Adjustment, and loop detection and optimization thread for global Pose Graph. The disadvantage of this method is that it is very time consuming to calculate the ORB feature for each image, and the three-threaded structure imposes a heavy burden on the CPU. Sparse feature point maps can only meet the positioning requirements, and cannot provide navigation, obstacle avoidance and other functions.

2.7 Test Platform Hardware Components

The hardware of two platforms is described in this chapter. First, the Raspberry Pi is the core of the platform, responsible for controlling the motor output; data processing and transmission of the LIDAR; data collection of the encoder and Inertial measurement unit (IMU); establishing a data connection with the personal computer, in short, the Raspberry Pi is the brain of the entire platform. LIDAR, encoder, Inertial measurement unit (IMU) and other sensors are used as tools to obtain external information, all these things together form the hardware system of the platform.

2.7.1 LIDAR

LIDAR (Light Detection and Ranging) is an abbreviation for laser detection and ranging system. It is an optical remote sensing technique that measures a target's distance by illuminating a beam of light, usually a beam of laser light. LIDAR has applications in surveying, archaeology, geography, geomorphology, earthquake, forestry, remote sensing, and atmospheric physics [40]. In addition, this technology is also used in specific applications such as airborne laser map mapping, laser altimetry, LIDAR and other high-line drawing. Optical radars measure object distances similarly to what is commonly referred to as radars, which measure the distance of an object by measuring the time interval between the transmitted and received pulse signals.

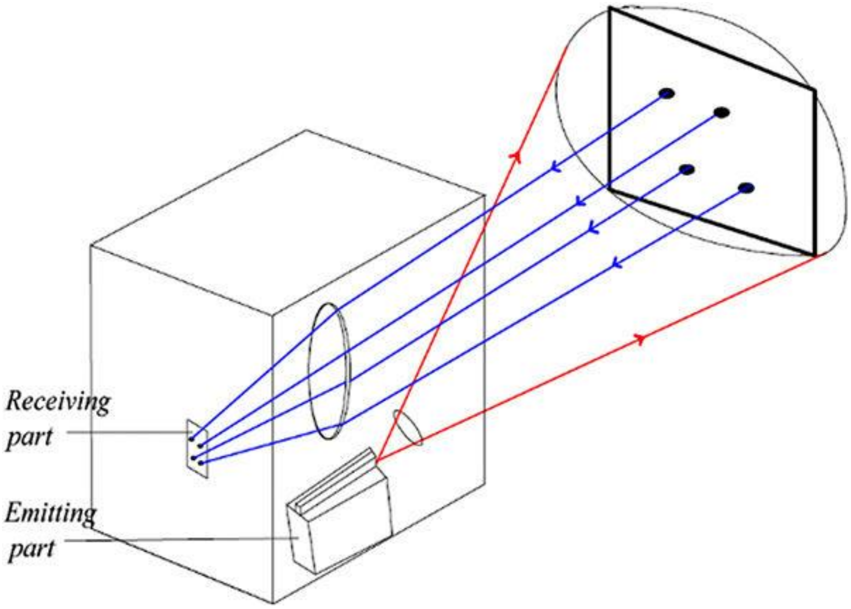


Figure 2.7 – LIDAR ranging principle diagram

At present, LIDAR can be divided into solid-state LIDAR and mechanical LIDAR. LIDAR with mechanical rotating shaft can be used as mechanical LIDAR. Commonly used LIDAR such as RPLIDAR, YDLIDAR, etc. use brushless motor to drive LIDAR rotates to scan the environment within 360. Solid-state LIDAR does not use motor to drive its rotation. The scanning area is a certain horizontal angle and height, which cannot achieve 360-degree scanning. However, due to its use of optical phased array [41] (OPA optical phased array). Phased array the transmitter consists of a plurality of transmitting and receiving units forming a rectangular array. By changing the phase difference of the light emitted by different units in the array, the purpose of adjusting the angle and direction of the outgoing wave can be achieved. Phased array technology may be adjacent to transmit an electric signal by controlling the phase difference of light in the array, the entire module can be changed to achieve lasing direction and angle, and hundreds of thousands of units a transmit array, the control unit can transmit let implement a scanning plane in 3D space, reaches a same effect as the rotating machine LIDAR.

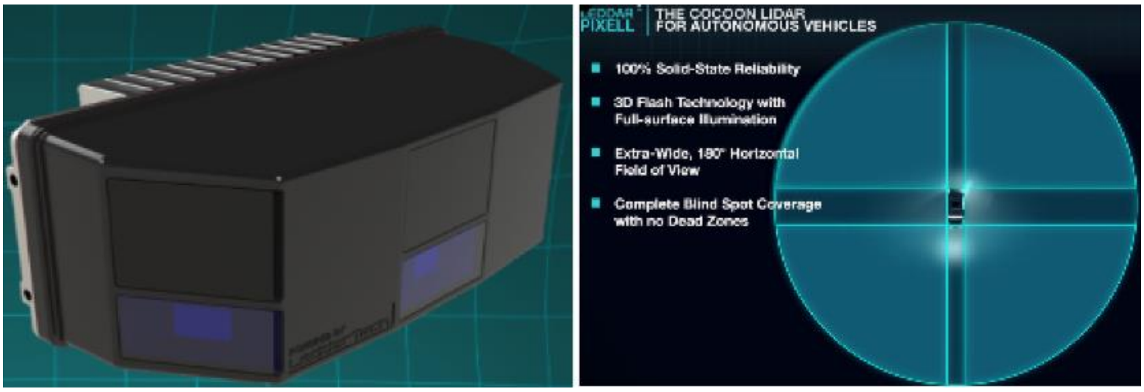

LEDDAR PIXELL
THE COCOON LIDAR FOR AUTONOMOUS VEHICLES
100% Solid-State Reliability
3D Flash Technology with Full-surface Illumination
Extra-Wide, 180° Horizontal Field of View
Complete Blind Spot Coverage with no Dead Zones

Figure 2.8 - The solid-state LIDAR: LEDDAR PIXELL

At present, the mature solid-state LIDAR products have LeddarTech's LEDDAR PIXELL [42] as shown in Figure 2.8. In the figure, four LEDDAR PIXELL solid-state LIDARs are used at the four directions to detect the space distance information in the 360-degree direction and generate 3D environment modelling.

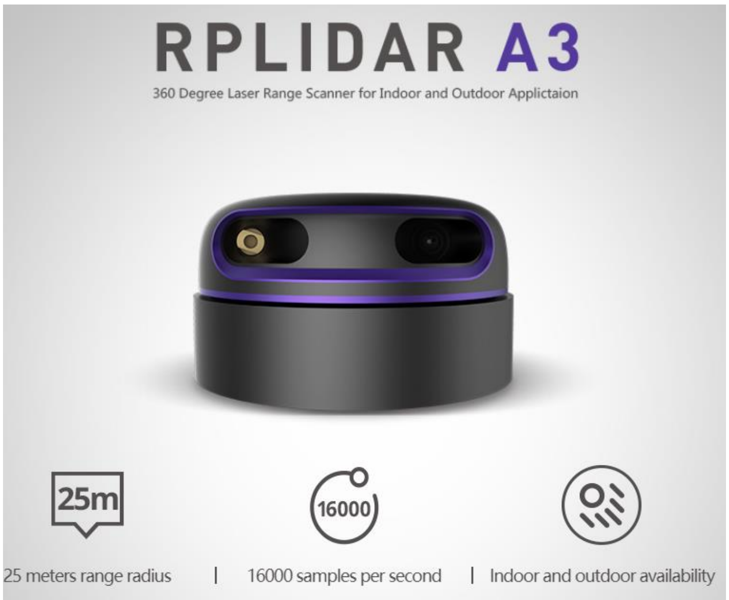

RPLIDAR A3
360 Degree Laser Range Scanner for Indoor and Outdoor Applictaion
25m
25 meters range radius
16000
16000 samples per second
Indoor and outdoor availability

Figure 2.9 - The mechanical LIDAR: RPLIDAR A3

Mechanical radar products such as mechanical LIDAR: RPLIDAR A3[43] from SLAMTEC, SLAMTEC was founded in 2013, its autonomous robot navigation and sensor core has extensive research and practical experience. This company continuously providing efficient and reliable solutions for the robot market through technology development and product iteration as shown in Figure 2.9, can detect 360-degree 2d distance information. Effective distance 25m, the highest measurement frequency can reach 16000 times per second, can be used indoors and outdoors.

Table 2.4 - Comparison between mechanical LIDAR and solid-state LIDAR

	Solid state LIDAR	Mechanical LIDAR
Advantages	No need for rotating parts, simple structure, small size, longer service life than mechanical LIDAR, Higher scanning accuracy and faster scanning speed. Multi-target monitoring.	360-degree scanning, low cost, low technical level, easy to use
Disadvantages	High technical requirements, large receiving area, more noise, and high cost	Scanning accuracy is lower than mechanical LIDAR, limited by 2d horizontal scanning

2.7.2 Raspberry Pi Platform

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. [44]

Several generations of Raspberry Pi [45] have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+; on-board memory ranges from 256 MB to 1 GB RAM. Secure Digital (SD) cards in Micro Secure Digital High Capacity (SDHC) form factor (SDHC on early models) are used to store the operating system and program memory. The boards have one to four Universal Serial Bus(USB) ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm tip-ring-sleeve jack for audio output. Lower-level output is provided by a number of General-purpose input/output (GPIO) pins, which support common protocols like I^2C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth. Prices range from US$5 to $35.

Table 2.5 – Comparison between different Raspberry Pi types

Family	Model	Form Factor	Ether-net	Wire-less	GPIO	Release data	discontinued
Raspberry pi 1	B	Standard (85.60 × 56.5mm)	Yes	No	26-pin	2012	Yes
	A		No			2013	Yes
	B+		Yes		40-pin	2014	
	A+	Compact (65×56.5 mm)	Yes			2014	
Raspberry pi 2	B	Standard	No	No		2015	
Raspberry pi Zero	Zero	Zero(65 × 30mm)	Yes	No		2015	
	W/WH		No	Yes		2017	
Raspberry pi 3	B	Standard	Yes	Yes		2016	
	A+	Compact	No			2018	
	B+	Standard	Yes			2018	
Raspberry 4	B	Standard	Yes	Yes		2019	

2.7.3 Encoder

An encoder [46] is a device that compiles or converts a geometrical quantity (angle) into a form that can be used for communication, transmission, and storage. The encoder converts an angular displacement or a linear displacement into an electrical signal. The encoder can be divided into two kinds of encoder: incremental encoder and absolute encoder.

1) Incremental encoder: The incremental value refers to a change of relative position information. The calculation of the increase and decrease of the signal from point A to point B is also called “relative value”, it requires uninterrupted counting of subsequent devices, because each time the data is not independent, but depends on the previous readings, the error caused by the power outage and interference of the previous data cannot be judged, resulting in error accumulation.

2) Absolute encoder: The absolute means that after the encoder is initialized, an origin is determined, all the position information is the absolute position of the "origin". It does not require the uninterrupted counting of subsequent devices, but directly reads the current position. value. For the error that may occur due to power failure and interference, since each reading is independent of the previous influence, it will not cause error accumulation.

From the measurement principle, it can be divided into photoelectric encoder and magnetic encoder.

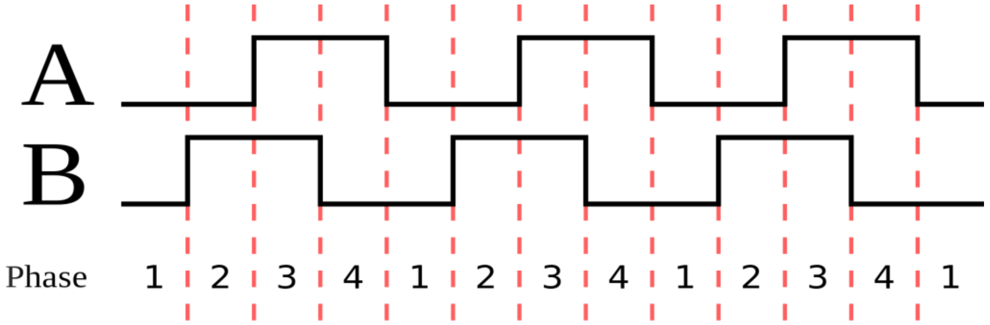


Figure 2.10 - A-B phase encoder outputs two orthogonal square waves

The magnetoelectric encoder [47] is an angle or displacement measuring device. The principle is to use a magneto resistive or component to measure the angle or displacement of a varying magnetic material. A change in the angle or displacement of the magnetic material causes a certain resistance or voltage change. After processing by the single-chip microcomputer, the pulse signal or the analog signal is output to achieve the purpose of measurement.

The main working principle of the photoelectric encoder is photoelectric conversion, which is a sensor that converts the mechanical geometric displacement of the output shaft into a pulse or digital quantity by photoelectric conversion. The photoelectric encoder is mainly composed of a grating disk and a photoelectric detecting device. In the servo motor system, the grating disk and the motor are coaxial, so that the rotation of the motor drives the rotation of the grating disk, and then the photoelectric detecting device outputs a pulse signal, according to the signal per second. The number of pulses calculates the current motor speed. The code wheel of the photoelectric encoder outputs two signals with a phase difference of 90 degrees, and the rotation direction of the motor can be determined according to the change of the state of the two-channel output optical code.

2.7.4 Inertial measurement unit (IMU)

An inertial measurement unit (IMU) [48] is an electronic device that uses a combination of accelerometers, gyroscopes, and magnetometers to measure and report specific forces, angular rates, and the orientation. IMUs are commonly used to maneuver aircraft (attitude and heading reference systems), including unmanned aerial vehicles (UAVs), as well as spacecraft, including satellites and landers. Recent developments have allowed the production of GPS devices that support IMU. The IMU allows the Global Positioning System (GPS) receiver to operate when GPS signals are not available, such as in a tunnel, inside a building, or when there is electrical interference. Usually, the IMU sensor is installed on the center of gravity of the object to be measured. The IMU sensor can measure the 6-degree-of-freedom pose information (x, z, y, roll, pitch, and yaw) of an object in three-dimensional space, as shown in Figure 2.11, using IMU sensor to measure the position of the aircraft in the air.

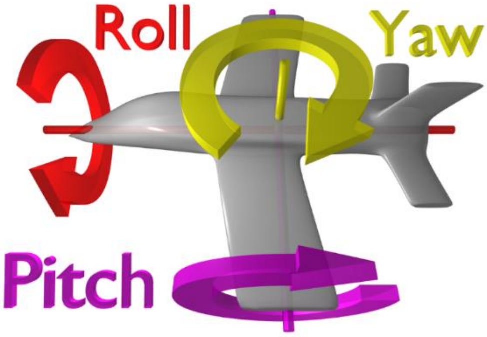


Figure 2.11 - The pose of aircraft in the air space: roll, pitch and yaw

Chapter 3 LIDAR Test platform with stepper motor

3.1 Overview

In order to study the application of LIDAR for AGV's obstacle avoidance in specific environments (ports), first of all, a LIDAR-based testplatform was developed. It relies on a rangefinder consisting of a LIDAR with a stepper motor to measure the distance information within 180 degrees in front of the platform, then the obstacle avoidance strategy based on distance information at different angles was developed, the platform will make corresponding moving actions according to the obstacle avoidance strategy. The platform system diagram can be seen in Figure 3.1.

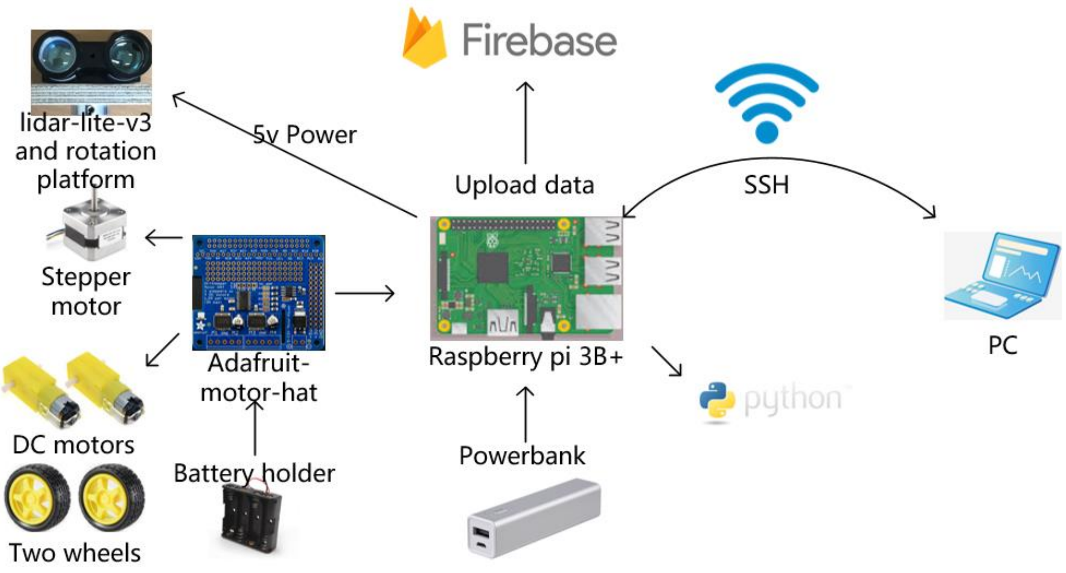


Figure 3.1 - Test system block diagram

3.2 System Architecture

The localization of obstacles is performed using LIDAR-lite v3 that is a compact optical distance measurement sensor from Garmin. This device measures the distance to object by calculating the time delay between the transmission of a Near-infrared laser signal and its reception after reflecting from the target. Effective range is 40 meters, the measurement accuracy within 5 meters is 2.5 cm. The LIDAR transmits data to the Raspberry Pi through the I2C (Inter-Integrated Circuit) communication protocol and operates at 5V DC, which means it can work well with Raspberry Pi computation platform.

A two-dimensional rotating platform for LIDAR was designed that allows LIDAR to measure obstacle position in multiple directions. The rotating platform consists of a stepper motor and a link plate. The stepper motor is controlled by the Raspberry Pi, and its rotation frequency and rotation mode can be set by code.

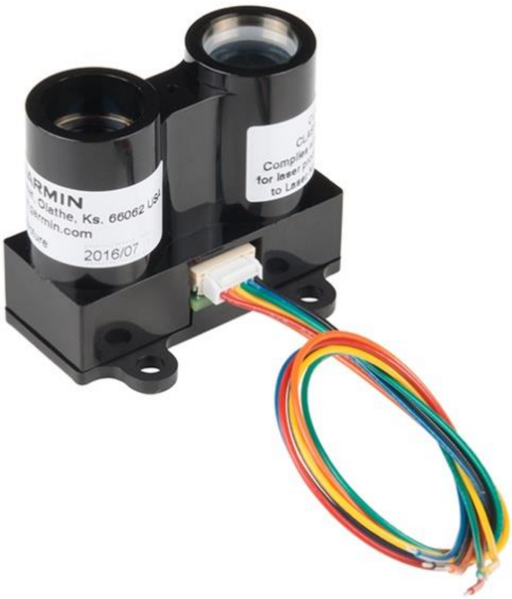


Figure 3.2 - LIDAR-lite-v3

To operate the platform, a Raspberry Pi 3B+ was selected. Because the Raspberry Pi 3b+ has a Wi-Fi connection, it is used to establish a connection with the pc, and there are matching accessories (motor drive board). The Raspberry Pi will be used to connect to the LIDAR and the motor driver board.

Figure 3.3 - Raspberry Pi 3B+

In this platform, a set of two DC motors and an Adafruit-motor-hat as a controller were used. This Adafruit-motor-hat can control two DC motors and one stepper motor simultaneously. The power supply is based on 6V batteries and a power bank for the Raspberry Pi computation platform.

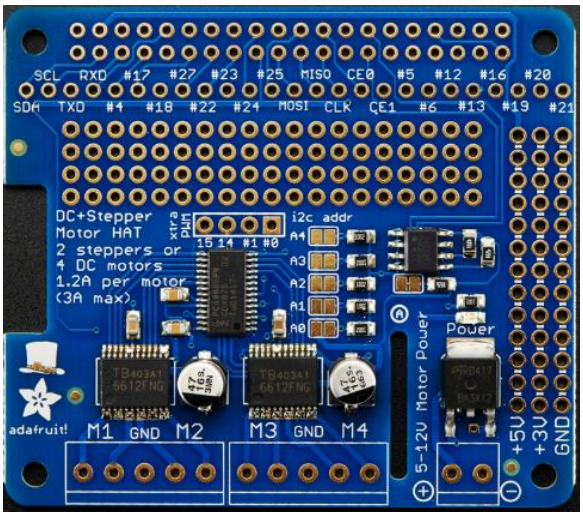


Figure 3.4- Adafruit-motor-hat – motor driver

Using the mentioned hardware components, the Robot platform was implemented (please see Figure 3.5).

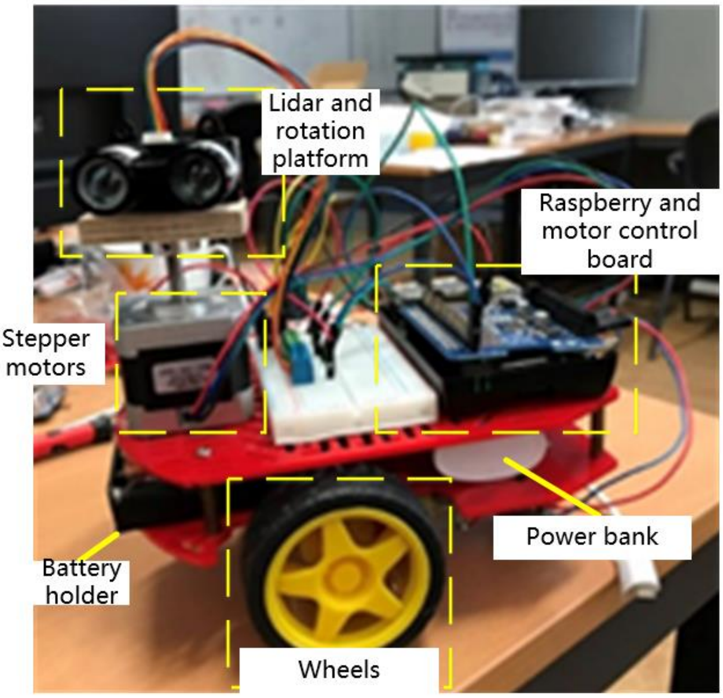

Lidar and rotation platform
Raspberry and motor control board
Stepper motors
Power bank
Battery holder
Wheels

Figure 3.5 - The fully assembled Robot platform

Regarding the implemented Robot platform can be mentioned that it is a two-wheel differential structure platform with two DC motor driven wheels and a universal wheel. The structure of the platform is divided into two layers. The lower layer is equipped with two DC motors, battery modules and a power bank. The upper layer is equipped with LIDAR, breadboard, raspberry Pi 3b+ and motor drive board.

3.3 System software

3.3.1 Firebase real-time database

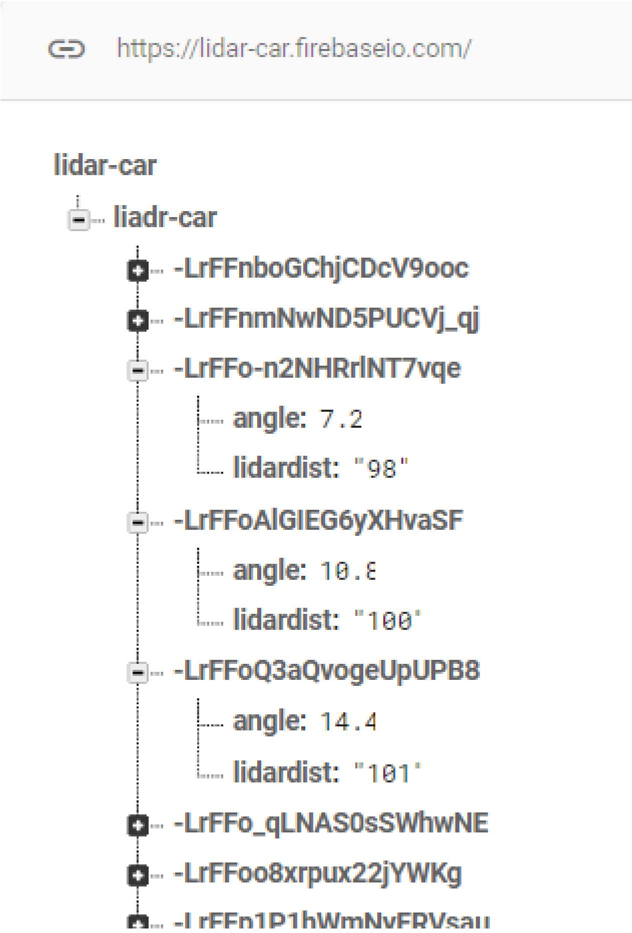
Firebase is a mobile and web application development platform developed by Firebase, Inc, acquired by Google at 2014 [49]. The firebase database was used to store the distance to obstacle data. Connect the Raspberry Pi on the platform to the Internet via WI-FI. Two properties for each data: the measurement angle and the distance value. When a measurement cycle ends, the distance values for all angles are uploaded. The specific Python code it is following presented:

Python code:

firebase=firebase.FirebaseApplication('https://lidar-car.firebaseio.com/')

result = firebase.post('lidar-car', {'angle':60,'lidist':str(dist[0])})

This code is sent to the real-time database using the firebase.post method when the LIDAR is measuring a distance of 60 degrees



```
https://lidar-car.firebaseio.com/

lidar-car
  liadr-car
    -LrFFnboGChjCDcV9ooc
    -LrFFnmNwND5PUCVj_qj
    -LrFFo-n2NHRrlNT7vqe
      angle: 7.2
      lidardist: "98"
    -LrFFoAlGIEG6yXHvaSF
      angle: 10.8
      lidardist: "100"
    -LrFFoQ3aQvogeUpUPB8
      angle: 14.4
      lidardist: "101"
    -LrFFo_qLNAS0sSWhwNE
    -LrFFoo8xrpux22jYWKg
    -LrFFp1P1hWmNvFRVsau
```

Figure 3.6– Real-time database developed firebase

3.3.2 Motion control of the platform

The motion platform will make corresponding actions according to the obstacle avoidance strategy. Through the kinematics model of the platform to explain how the platform achieves the above actions, the structure of the first platform and the second platform are the same, both of which are two-wheel differential structure, so the kinematic model to be introduced below is applicable to two platforms.

The introduction of the platform kinematics model is to clarify the relationship between the angular velocity of the left and right wheels and the overall linear velocity and angular velocity of the platform, and also provide theoretical support for the platform's motion control.

The platform used in the study is a two-wheeled platform with two drive wheels and a universal wheel. The following figure shows the pose of the mobile platform at two adjacent moments, θ_1 is the angle at which the mobile platform moves around the arc at two adjacent moments, θ_2 quals to θ_1 and θ_3 is the amount of change in the heading angle (toward the head head) of the moving machine at two adjacent moments. L is the distance between the left and right wheels, which is the distance between the right wheel and the left wheel. d is the distance that the right wheel travels more than the left wheel. R is the radius of the circular motion of the platform.

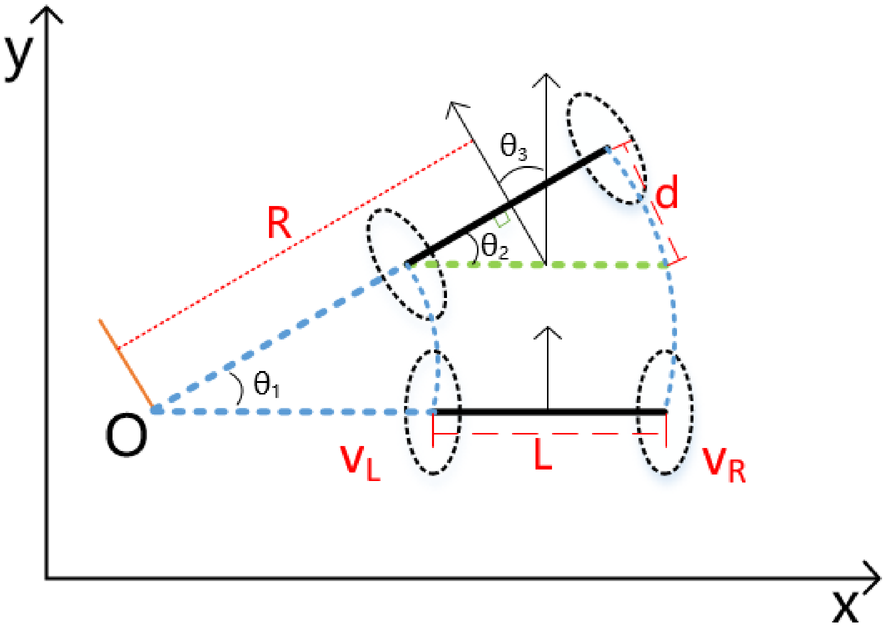


Figure 3.7 - The kinematic model of the motion platform

The moving speed of the platform v is equal to the average of the left and right wheel speeds:

$$v = \frac{\mathrm{v}_R + v_L}{2} \tag{3-1}$$

How to calculate the platform's heading angle and how to calculate the angular velocity. As shown in the figure, by superimposing the positions of the platform at two moments, it is clear that the amount of change in the heading angle of the platform is θ_3. From the geometric relationship in the figure we can get:

$$\theta_1 = \theta_2 = \theta_3 \tag{3-2}$$

When the platform makes a circular motion, it starts from the starting point and returns to the starting point around the centre of the circle. In this process, the cumulative heading angle of the platform is 360 degrees, and it also moves 360 degrees around the centre of the trajectory, indicating the angle of change of the heading angle of the platform equal to the angle at which it rotates around the centre of the motion trajectory. Among these three angles, θ_2 is easy to calculate. Since the platform is in continuous motion, the time interval between two adjacent moments is very short, and the angle change is small, so there is the following approximate formula:

$$\theta_2 \approx \sin(\theta) = \frac{d}{L} = \frac{(\mathrm{v}_R - v_L)\Delta t}{2} \tag{3-3}$$

The angular velocity of the platform around the center of the circle can be calculate, which the speed of the platform’s heading angle is given:

$$\omega = \frac{\theta_1}{\Delta t} = \frac{\mathrm{v}_R - v_L}{L} \tag{3-4}$$

Therefore, the radius of the circular motion of the mobile platform can be introduced.

$$R = \frac{v}{\omega} = \frac{L(\mathrm{v}_R + v_L)}{2(\mathrm{v}_R - v_L)} \tag{3-5}$$

Now the formulas used in the basic motion control of the platform have been obtained.

According to the above formula, on the platform, the speed of the two wheels is *vleft*, *vright*. When *vleft*=*vrigh*t>0, the platform goes straight; when *vleft* is not equal to *vright*, the platform will turn and here has different situations:

When *vleft* is not equal to *vright* and both of them greater than or equal to 0, in this case, the platform will turn to the side with a smaller speed. When the *vleft* and *vright* are not equal and one speed is less than 0, the platform will make a smaller radius steering. When the two speed values are the same but the directions are opposite, the platform will do the spin motion. The above analysis is based on the ideal situation, without considering the friction between the wheel and the ground, the output loss of the motor.

But considering the existence of the universal wheel, the friction of the ground, the final speed of the wheel and the output of the motor are inconsistent, the problem will simplify by adjusting the working time of the motor:setting one wheel to be stationary and the other wheel rotates to achieve turning.

The specific time settings are shown in the code:

Here are 8 different actions according to the distance value from the 7different angles, including go straight, turn left (30 degrees, 60 degrees, 90 degrees), turn right (30 degrees, 60 degrees, 90 degrees), turn 180 degrees.

```
rMotor.setSpeed(255)
lMotor.setSpeed(0)
time.sleep(1)
rMotor.setSpeed(0)
lMotor.setSpeed(0)
```

On the platform, one wheel rotates, the other wheel is stationary to achieve the steering of the platform, the *setSpeed* (x) function sets the motor speed, the maximum value is 255, and the *timesleep(x)* function can set a time delay *t*. During the time *t*, the speed of the motor is always maintained at the value *x* set by the *steSpeed()* function. The above-mentioned platform kinematics model knows the relationship between the angular velocity of the wheel and the heading change. By setting the value of *x* in the *setSpeed(x)* and the value of *t* in *timesleep(t)*, the steering control of the platform at specific angle can be completed. The execution process of the platform's steering control code is as follows

1 Start two motors, set the left motor speed to 255, and the right motor speed to 0;

2 Set the delay time to 1s, which means that the two motors remain in this state for 1s.

3 Set the two motor speeds to 0, meaning that the platform stops moving.

3.3.3 The environmental model obtained by LIDAR-lite-v3

The LIDAR and stepper motor are connected by a rotating platform to form a two-dimensional distance measuring system. The LIDAR and stepper motor are bonded together by tape, when the stepper motor rotates, the LIDAR will also rotate with the stepper motor. In this system, the rotation mode of the stepper motor was designed. the device cannot be rotated 180 degrees, due to the connection of the wires. The rotation mode can be seen in Figure 3.8.

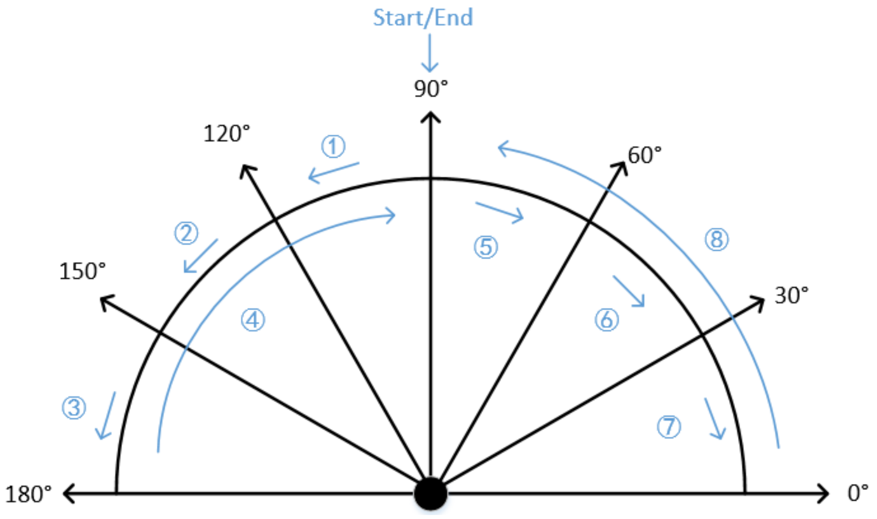


Figure 3.8 - Stepper motor rotation mode.

The starting and ending positions of the rotation are the centre of the platform's orientation. In one cycle, the stepper motor rotates a total of 8 times, mark the number of steps of the stepper motor as 1 to 8 and when the cycle is over, the start position coincides with the end position. According to the stepper motor driver`s function, the stepper motor is characterized by 300 steps/rev, which means the stepper motor rotates 300 times in 360 degree, every step rotates 1.2 degree. The settings of the stepper motor rotation sequence were1-3, 4-7 rotates 30 degree, equal to 25 steps, and sequence 4 and 8 rotates 90 degree, equal to 75 steps. When the 1-7 step is over, the LIDAR will work.

According to the distance measured by LIDAR in 7 directions, the approximate shape of the obstacle can be drawn, the platform as the centre of the surrounding environment can be established. The Surrounding environment geometrical estimation based on LIDAR measurements is shown in Figure 3.9.

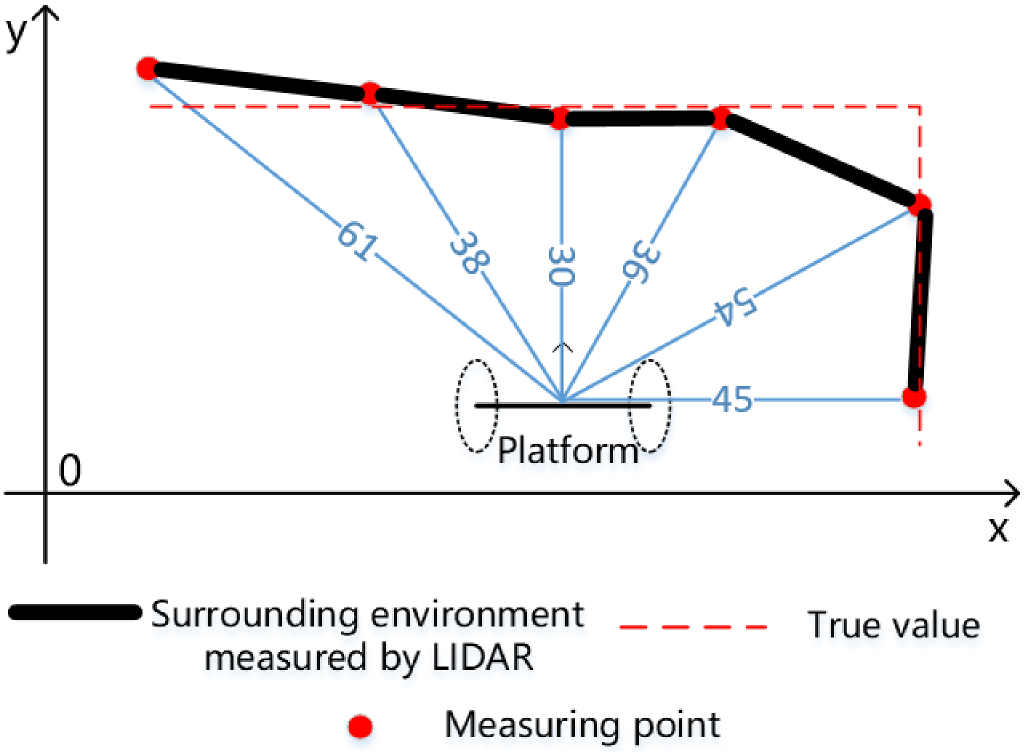


Figure 3.9 - Surrounding environment geometrical estimation based on LIDAR measurements

Distance measuring: Python code description:

The code for measuring the distance is a while loop. In this loop, the angle of each rotation is different. The angle is determined by controlling the output of the motor, the stepping motor is set to 200 steps per revolution, which means that the stepping motor rotates one step equivalent to 1.8 degrees. 15steps equivalent to 30degrees, 45steps equivalent to 90degrees. The way of rotation is shown in Figure 3.6. The distance in 7 directions is measured first, and the angle value and the corresponding distance value are stored in the dictionary **dist**, the dictionary is a kind of variable container model in python and can store any type of object. In the end, the angle value and the distance value are uploaded to the firebase.

Distance measurement python code:

```
    myStepper = mh.getStepper(200, 2)    # 200 steps/rev, motor port #1
    myStepper.setSpeed(200)                  # 30 RPM
    dist={0:0,1:0,2:0,3:0,4:0,5:0,6:0}
while True:
    dist={0:0,1:0,2:0,3:0,4:0,5:0,6:0}
    shortdir={0:0}
    myStepper.step(15,Adafruit_MotorHAT.BACKWARD,Adafruit_MotorHAT.DOUBLE)      #rotate 30 degree
    dist[0]= lidar.getDistance()
    result = firebase.post('lidar-car', {'angle':60,'lidist':str(dist[0])}) #upload it into firebase
    print("The distance in the 60 degree is",dist[0],"cm")
    time.sleep(1)
   …
    myStepper.step(45,Adafruit_MotorHAT.FORWARD,Adafruit_MotorHAT.DOUBLE)      #rotate 90degree
     dist[3]= lidar.getDistance()
     result = firebase.post('lidar-car', {'angle':90,'lidist':str(dist[3])})
     print("The distance in the 90 degree is",dist[3],"cm")
     time.sleep(1)
```

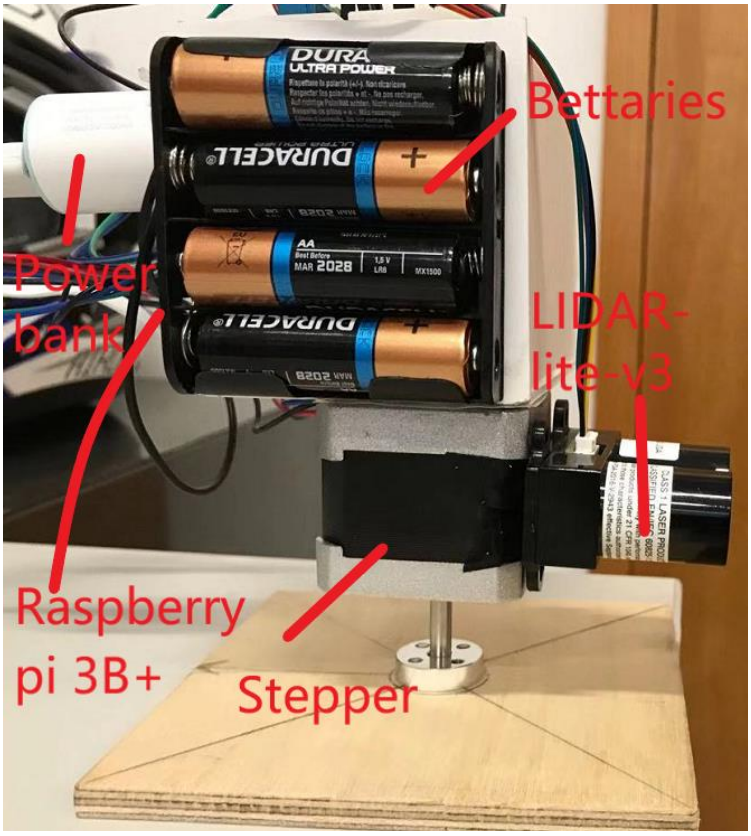


Figure 3.10 – 2D laser scanner hardware implementation

After achieving measure in seven angles at the range of 180 degrees ahead, the system was improved and redesigned to achieve full-angle measurements, enabling 360-degree distance data (point cloud data) acquisition and data-based mapping. The 2d laser scanner was developed based on the LIDAR-lite-v3, which combines the Raspberry Pi and the stepper motor to measure the LIDAR-centered 360-degree distance on the 2d plane of the laser lens. The picture of the 2D laser scanner is shown in Figure 3.10. The specific implementation of the 2D laser scanner system is as follows:

First, based on the previous code, the speed of the stepper motor is set to 100 steps per revolution. Due to the mechanical structure limitation of the stepper motor, the angle of rotation of each step is 3.6 degrees. The dictionary element is still used in the program to store point cloud data. The code: Adafruit_ MotorHAT.BACKWARD means the stepper rotates counterclockwise.

These code: firebase.post ('liadr-car',{'angle':3.6*i,'lidardist':str(mapdata[i])}) representative uploads the data to firebase, angle value = 3.6*i. The program stops after measuring 100 times. The implemented code is following presented:

```
myStepper = mh.getStepper(100, 1)    # 100 steps/rev, motor port #1
mapdata = defaultdict(int)
for i in range(0,101):
    mapdata[i]=lidar.getDistance()
    mystepper.step(1,Adafruit_MotorHAT.BACKWARD,Adafruit_MotorHAT.MICROSTEP)
    result = firebase.post('liadr-car',{'angle':3.6*i,'lidardist':str(mapdata[i])})
    print(mapdata[i]).
```

After acquiring the point cloud data, the data file (json) is exported from the firebase real-time database, the file is converted into an excel file, and the point cloud data is drawn in the form of polar coordinates using the plot library in python. The partially exported data is shown in Table 3.1, and the polar coordinate drawing is shown in Figure 3.11.

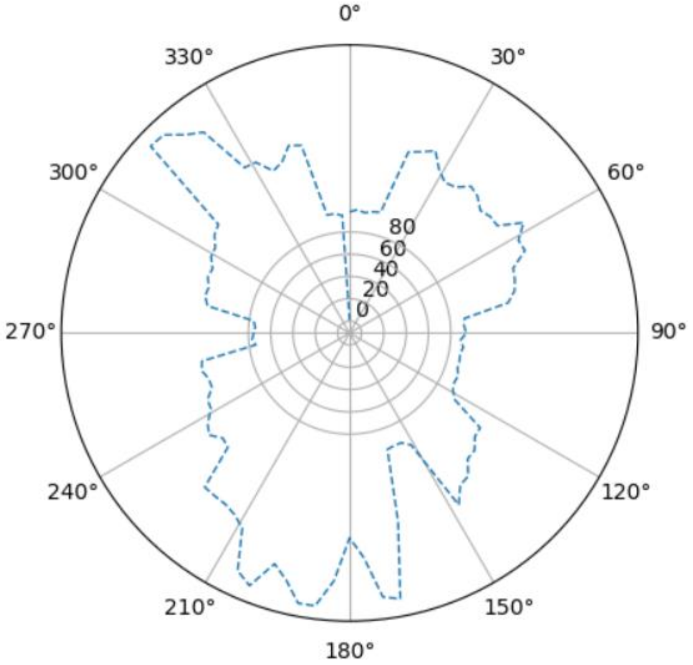


Figure 3.11 - The polar coordinate drawing based on the cloud point data

Table 3.1 - Table of the could point data

Angle	Distance(cm)
0	110
3.6	110
7.2	110
10.8	276
14.4	273

3.3.4 Obstacle avoidance strategy based on the seven orientations distance values

The setting for obstacle avoidance strategy for first platform is performed as following. Based on the distance value from 7 directions, first sorted the value, the largest and smallest data are marked as the longest distance and the shortest distance.

If the maximum distance is less than 35 cm, means that there is not enough space in the seven directions in front of the platform to move, the platform will rotate 180 degrees. When the maximum distance is greater than 35 cm and the minimum distance is less than 100 cm. That is, the safe distance between the platform and the surrounding environment is 35cm-100cm. Then choose the direction of the maximum distance to avoid obstacles, the platform turns to the direction of the maximum distance to avoid obstacles. The flow chart of the obstacle avoidance strategy is shown in Figure 3.8.

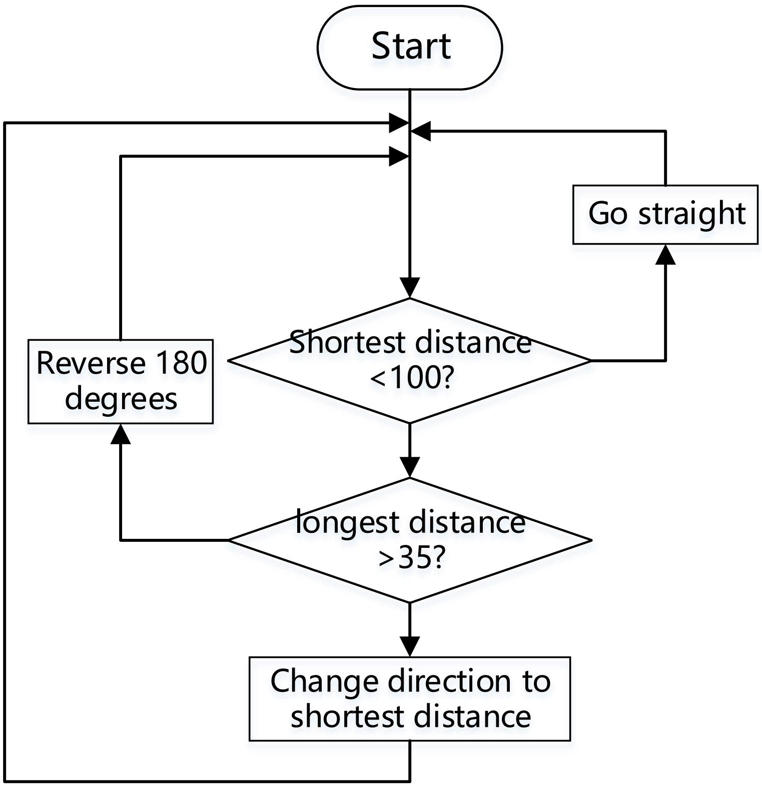


Figure 3.12 - Obstacle avoidance strategy flow chart

Obstacle avoidance strategy code description:

In the previous section, I introduced that the dictionary **dist** is used to store angle and distance values, and has stored the data in **dist**, and then uses the sort function sorted to sort the distance data in **dist** from small to large, and then saved the angle value in the **shortestNo/longestNo** and the distance value in the **shortestdist/longestdist,** the code for platform motion control is omitted and will be described in detail in the next chapter. The obstacle avoidance strategy determines the action by identifying the shortest distance angle and the longest distance angle.

```
dist={0:0,1:0,2:0,3:0,4:0,5:0,6:0}
#sort distance
redist=sorted(dist.items(),key=lambda item:item[1])
#get the shortest distance angle
shortdir=sorted(dist.values())[0]
shortest=redist[0]
shortestNo=shortest[0]
shortestdist=int(shortest[1])
#get the longest distance angle
longdir=sorted(dist.values())[6]
longest=redist[6]
longestNo=longest[0]
longestdist=int(longest[1])
     if shortestdist<100 and longestdist>35:
           if longestNo = =0:
                 turnleft90
             elif longestNo = =1:
                 turnleft60
             elif longestNo = =2:
                 turnleft30
             elif longestNo = =4:
                 turnright30
             elif longestNo = =5:
                 turnright60
             elif longestNo = =6:
                 turnright90
         elif longestdist<35:
             go straight
         elif shortestdist>100:
              go straight
```

3.4 Remarks

All content of the platform has been explained. After the platform is completed, some deficiencies are found in the process of use. These two functions cannot be realized: self-positioning and navigation. Insufficient scanning accuracy for environmental information, the proposed obstacle avoidance strategy can only avoid obstacles in static scenes. Based on the above deficiencies, on the basis of previous platform, the encoder and IMU sensor are added to provide the odometer information, and the RPLIDAR A1 is selected to achieve higher frequency scanning of the environment, which has formed a more accurate environment map. The SLAM work is completed with the odometer information provided by the encoder and the IMU sensor. This new platform was developed to improve the capable of performing environmental scanning, self-positioning, path planning, and obstacle avoidance without external intervention.

Chapter 4 LIDAR test platform with encoder and IMU sensor

In order to improve the inadequacies of the first platform and realize the function of the laser SLAM, a second platform was developed.In this paper [50], the intelligent robot operation procedures is introduced in detail: 1.Self-positioning; 2.Environment drawing; 3.Path planning; 4.Autonomous obstacle avoidance. In order to realize these five functions, taking into account the overall complexity and operational difficulty of the system, the robotic operating system(ROS) is used to realize the function of the new platform. The encoder and the IMU sensor are added and another 360 degree rotating LIDAR: RPLIDAR A1 is adopted, these three sensors are designed to implement the above five functions.The specific information about the platform with encoder and IMU sensor is as follows:

4.1 Hardware components

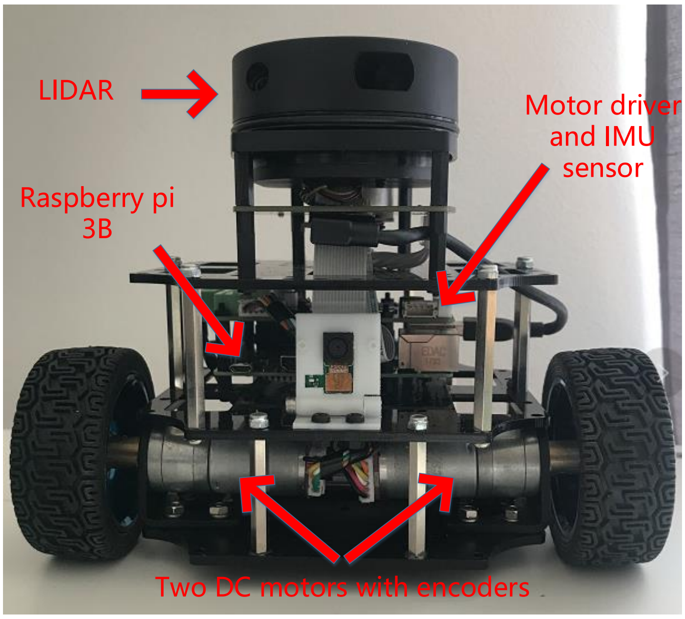


Figure 4.1 – Second platform hardware connection diagram.

Due to the requirements of self-positioning, real-time environment mapping, remote motion control and path planning. Another LIDAR test platform with encoder and IMU sensor is developed, it is similar in structure to the first platform. It is composed of a raspberry pi with a LIDAR and two dc motors. But the new LIDAR, encoder and Inertial measurement unit (IMU) sensor were installed in this platform to achieve these above functions. These components are introduced as follows:

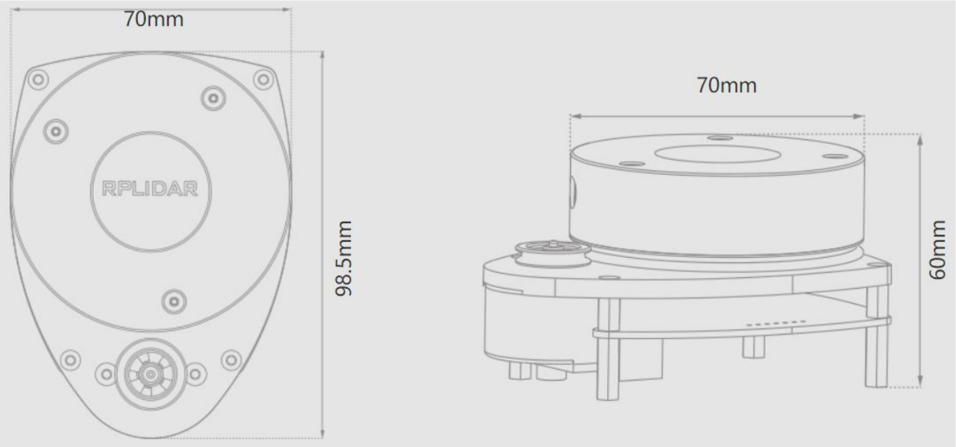


Figure 4.2 - Dimensions of RPLIDAR A1 [51]

The LIDAR used in the paper is a RPLIDAR A1, shown in Figure 4.2, this LIDAR is a low cost 2D LIDAR developed by SLAMTEC. Install it at the top of the test platform. It can perform a 360-degree laser ranging scan within an 8-meter radius of the 2D plane

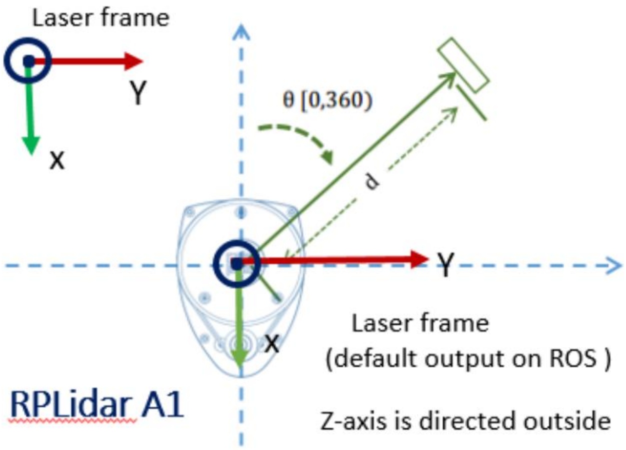


Figure 4.3 - Emission and detection diagram of RPLIDAR A1

In Figure 4.3, the transmission and detection of RPLIDAR A1 is shown, x, y, and are the coordinate axes of the point cloud data detected by the LIDAR, and the laser probe rotates counterclockwise on the xy plane. The measured distance value and the corresponding angle value are plotted with a polar coordinate map centered on the origin of the coordinate axis, called a point cloud data map. The cloud map information can be used in environment mapping and positioning navigation.

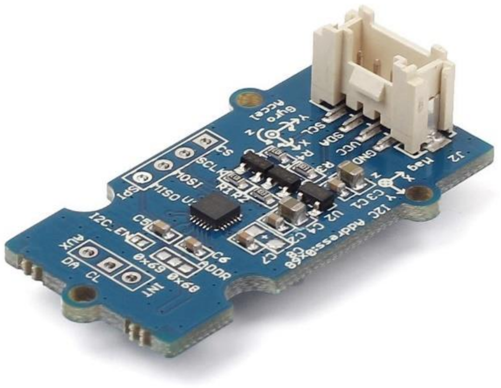


Figure 4.2 - Inertial measurement unit

The MPU-9250 is selected as the IMU sensor component of the platform. The MPU-9250 is an integrated 9-axis motion tracking device designed for the low power, low cost, and high-performance requirements of consumer electronics equipment including smartphones, tablets and wearable sensors. MPU-9250 features three 16-bit ADC for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs and three 16-bit ADCs for digitizing the magnetometer outputs.

Figure 4.3 - DC Motor with encoder

Two DC motors are used in this article and are equipped with two encoders. The motor used has a rated voltage of 5V. The encoder is an electromagnetic Hall encoder that records the number of turns of the wheel based on the electromagnetic effect between the two magnetic probes and the magnetic rotor.

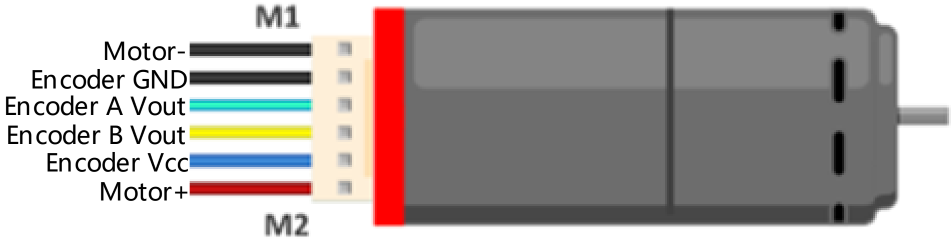


Figure 4.4 - Encoder motor pin out

The encoder is a two-phase magnetic encoder. The rotor of the encoder is connected to the shaft of the motor and rotates with the rotation of the motor. There are two probes on the encoder, which can detect the transformation of the rotor pole and can output two codes. Signal. The pin diagram is shown in Figure 4.4.

4.2 Software components

4.2.1 Platform odometer information acquisition

The odometer information is the amount of change in position and posture of an object on a plane or in space. In a three-dimensional space, the pose of an object can be determined by parameters: (x, y, z, roll, pitch, yaw). These six parameters are the coordinate values in the space coordinate system and the angles changed on the three axes of (x, y, z). The (x, y, z) coordinates are used to determine the position of the object. The three parameters of roll, pitch, and yaw determine the posture of the object. In this project, the experimental platform structure is a two-wheel differential car designed to move on the horizontal ground. For the simplicity of the experiment, ignore the rotation of the y-axis and the z-axis, and set the three parameters of z, pitch and roll to 0. This means that the position and posture of the platform on the horizontal plane can be determined by knowing the values of (x, y, yaw).

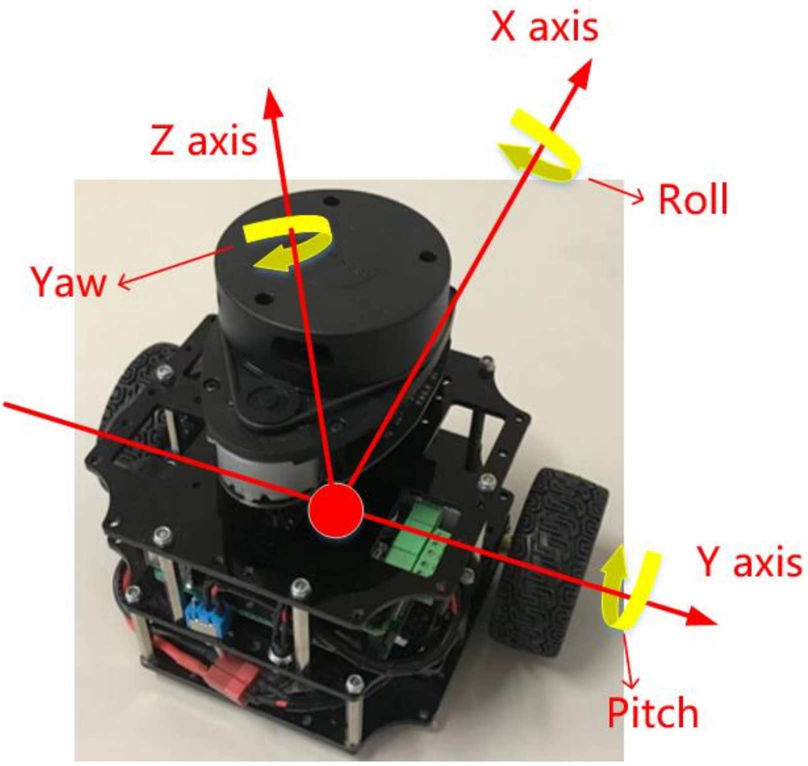


Figure 4.5 - 6 degrees of freedom of the test platform: (x, y, z, roll, pitch, yaw)

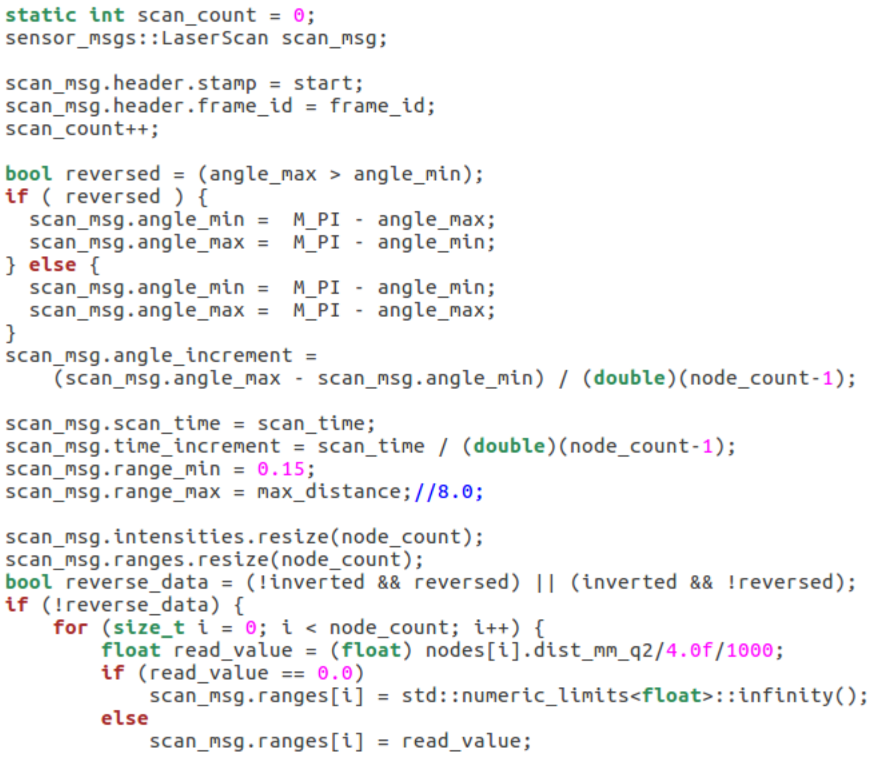
The encoder data of the left and right wheels of the platform is obtained through the serial port. After sending the sensor data request through the serial port, the platform will return the sensor data information in the form of data stream. The data update interval is 15ms, and the value range of the encoder return data is $0-65535$. After the value exceeds the range and restarts counting, the real-time linear velocity of the left and right rounds can be calculated from the encoder data using the following formulas:

$$\text{LEFT: } v_l = \frac{\Delta LECounts*2\pi}{n*\Delta t} * r \qquad \text{RIGHT: } v_r = \frac{\Delta RECounts*2\pi}{\mathrm{n}*\Delta t} * r \qquad (4\text{-}1)$$

In the formula: v_l And v_r are the real-time linear velocity is the real-time linear velocities of the left and right rounds respectively; Δt is the data sampling period, and the values are 15ms; $\Delta LECounts$ and $\Delta RECounts$ are the counts of the left and right encoders increasing in the sampling period respectively; n is the number of counting pulses generated by one turn of the coil; r is the radius of the wheel (in second platform is 0.05m).The heading angle change of the platform is provided by the IMU's gyro sensor. The speed and orientation of the platform are transmitted from the stm32 to the Raspberry Pi through the serial port. The transmission process uses hexadecimal numbers for transmission. In this case, the platform speed and heading angle are obtained, the platform position can be calculated.

4.2.2 Platform cloud point (LIDAR) information acquisition

When using RPLIDAR A1, this LIDAR can scan the surrounding 360 environments by rotating the laser probe. The maximum number of measurements per second can reach 8000 times. The rotation is continuous and uninterrupted, so how to know the scan result with the corresponding angle is an important issue. Based on the performance based on the RPLIDAR A1, the following parameters are set to record the



```
static int scan_count = 0;
sensor_msgs::LaserScan scan_msg;

scan_msg.header.stamp = start;
scan_msg.header.frame_id = frame_id;
scan_count++;

bool reversed = (angle_max > angle_min);
if ( reversed ) {
  scan_msg.angle_min =  M_PI - angle_max;
  scan_msg.angle_max =  M_PI - angle_min;
} else {
  scan_msg.angle_min =  M_PI - angle_min;
  scan_msg.angle_max =  M_PI - angle_max;
}
scan_msg.angle_increment =
    (scan_msg.angle_max - scan_msg.angle_min) / (double)(node_count-1);

scan_msg.scan_time = scan_time;
scan_msg.time_increment = scan_time / (double)(node_count-1);
scan_msg.range_min = 0.15;
scan_msg.range_max = max_distance;//8.0;

scan_msg.intensities.resize(node_count);
scan_msg.ranges.resize(node_count);
bool reverse_data = (!inverted && reversed) || (inverted && !reversed);
if (!reverse_data) {
    for (size_t i = 0; i < node_count; i++) {
        float read_value = (float) nodes[i].dist_mm_q2/4.0f/1000;
        if (read_value == 0.0)
            scan_msg.ranges[i] = std::numeric_limits<float>::infinity();
        else
            scan_msg.ranges[i] = read_value;
```

The above code comes from node.cpp, the official RPLIDAR A1 driver provided by SLAMTEC. The following parameters are set: the baud rate of the information transmission, the number of times of the LIDAR measures per second, the angular increment between the two measurements, the initial measurement angle, the end measurement angle, and the time taken for the measurement. Since the measurement uses a loop, the number of data transmitted per second by the computer needs to be the same as that of the actual LIDAR measurement. Here, based on the performance of the motor, the speed of the motor is set to 1080degree/second, corresponding to the LIDAR per second. The number of measurements is 7,200, the angular increment of each measurement is 0.15 degrees, and the corresponding baud rate is 115200, which means that 7200 data is transmitted per second, and the size of each data is 16 bits. In the ROS system, after running the program, the LIDAR data can be displayed through the Robot Visualization tool (RVIZ) [52]. In Figure 4.6, an image of LIDAR point cloud data displayed using the RVIZ tool is shown.

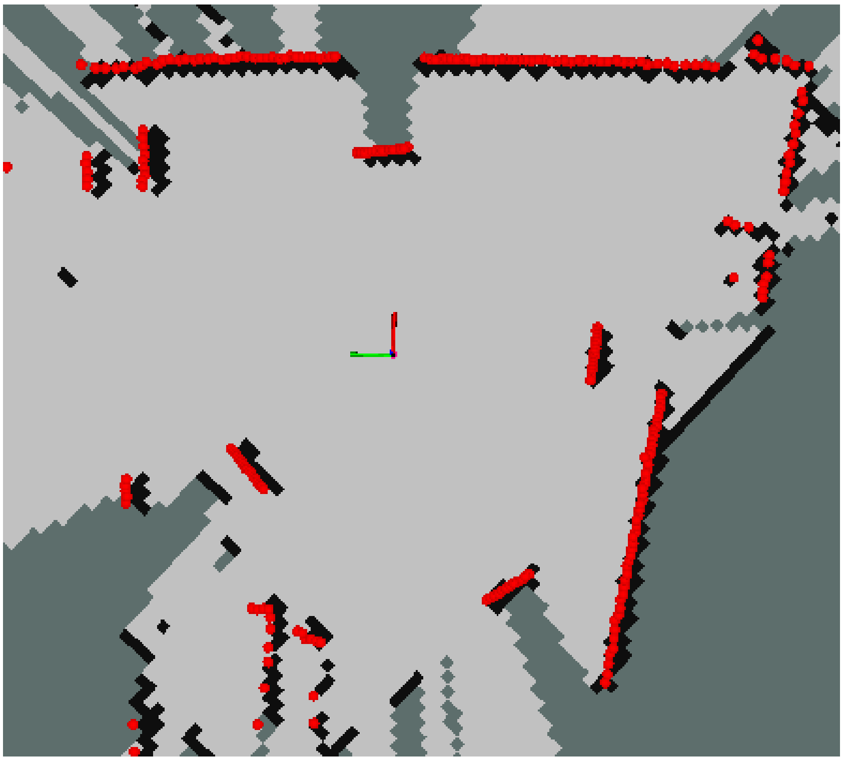


Figure 4.6 – LIDAR point cloud image using RVIZ tool

4.2.3 Simultaneous positioning and mapping (SLAM)

In the second test platform, the platform's own positioning function is supported by RPLIDAR, encoder and IMU sensor. The RPLIDAR provides point cloud data for the surrounding environment, and the encoder and IMU provide the odometer message for the platform. The data processing and function realization of laser SLAM is realized by the *gmapping* package of the ROS system.

GMapping is selected as the SLAM algorithm of this work. The implementation for the *gmapping* algorithm is realized by the *gmapping* package in ROS, which is an open source package, and we explain how it works from its node structure. *Gmapping*'s ROS node *slam_gmapping* provides laser SLAM. It creates a 2D grid map based on point cloud data and odometer message collected by the platform.

4.2.3.1 Data transmission structure

In the ROS system, the relationship between the */gmapping* node and other nodes is shown in Figure 4.7:

In the ROS system, the process of */gmapping* mapping is as follows:

1) The */rplidarNode* node publishes the point cloud data of LIDAR reference system to the */scan* topic, and */gmapping* get the data by subscribing the */scan* topic.

2) */gmapping* topic publishes point cloud data to */tf* topic, */base_footprint_to_laser* publishes conversion parameters to */tf* topic, */tf* topic uses the parameters converts data of LIDAR reference system into data of platform reference system,

3) The */gmapping* node subscribes to the */tf* topic to obtain the converted data, and publishes the data to the */map* topic. The */move_base* node subscribes to the */map* topic to obtain the map data for implementing the navigation function.

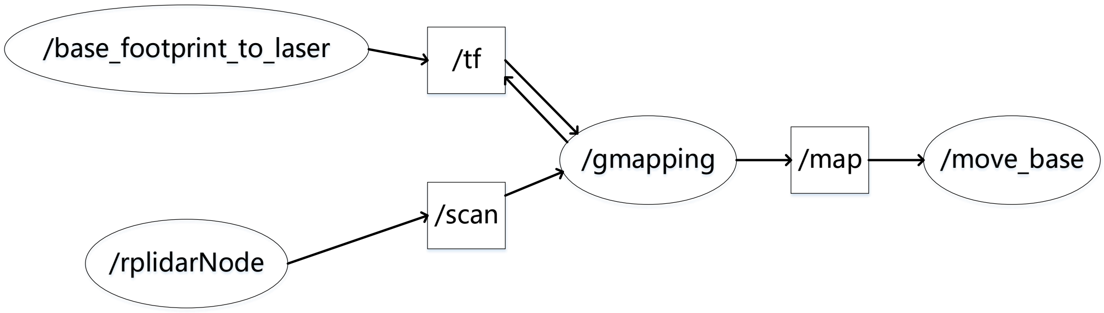


Figure 4.7 – Relationship diagram of /gmapping node

Now, a platform center reference environment map can be obtained, but to complete the complex environment of the laser SLAM also need to control the platform to explore the entire environment, get complete environmental information. Therefore, the commonly used ROS keyboard control program *teleop_twist_keyboard* is used to control the platform. In fact, this program is to issue speed commands to the two motors to control the platform. The kinematics model of the two-wheel differential structure has been introduced in the previous chapters. This program can change the linear speed of the two wheels of the platform separately by publishing the speed information of the two wheels to topic:*/cmd_vel*, this topic is specifically used to pass speed commands. then, the */base_control* node subscribes to the */cmd_vel* topic to get the speed command, and then drives the motor to complete the command to implement the platform's keyboard manipulation. The */base_control* node is used to execute platform motion commands and provide odometer information for other nodes.

4.2.3.2 Coordinate transformation

The acquired point cloud data is two-dimensional data. The data is based on the LIDAR-centric coordinate system. There are two dimensions: angle and distance, the angle value is the angle of clockwise rotation. Combined with the angular distance data, an environmental scan map with the LIDAR center at the current position can be drawn.

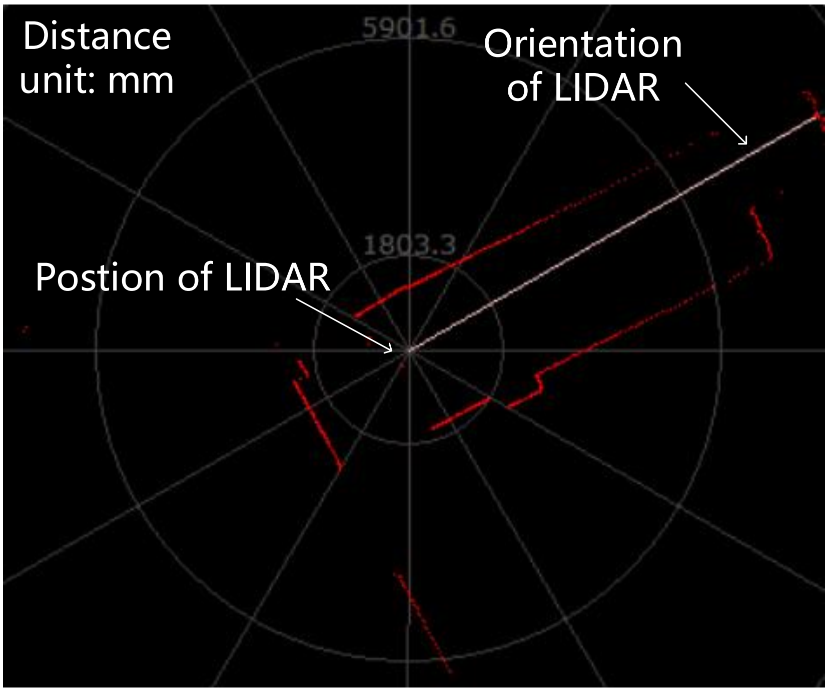


Figure 4.8 – LIDAR Cloud data

The point cloud data is the measured value of the LIDAR as the origin, but since the position of the LIDAR is not at the center of the platform, the point cloud data acquired by the LIDAR will have a deviation value between the LIDAR and the platform center. We use a coordinate transformation to transform the laser data from the *base_laser* (LIDAR reference system) to the platform's reference system *base_link*. I created a node: *base_footprint_to_lazser,* in this node, the parameters of transformation between these two coordinate systems was defined.

Node pkg=”tf” tye=”static_transform_publisher” name=”base_footprint_to_laser”
Args=”0.03 0.0 0.15 3.14159265 0.0 0.0 /base_footprint /laser 50”

In this code, *static_transform_publisher* means this code is for publishing the transform information. The *Args* shows the converted parameters. The parameter structure of *Args* is: *x, y, z, yaw pitch roll, frame_id, period_in_ms*. The *x y z* respectively represents the translation of the corresponding axis, in meters. The *yaw pitch roll* represents the rotation of the three axes around z y x in radians. The platform does not has rotation on the x y axis, so these two values are 0.0. The distance between the center of the LIDAR and the platform is 0.03m on the x-axis, the distance between the y-axis is 0, and the distance between the z-axis is 0.15m. Due to the installation position, the direction of the LIDAR is opposite to the direction of the platform, so the rotation of the z-axis is 360 degrees, arc length is π(3.14159265).The *frame_id* is the parent coordinate system in the coordinate system transformation, the *period_in_ms* is the publishing frequency in milliseconds, which is 20Hz. In Figure 4.9, the positional relationship between the LIDAR coordinate system and the platform coordinate system is shown.

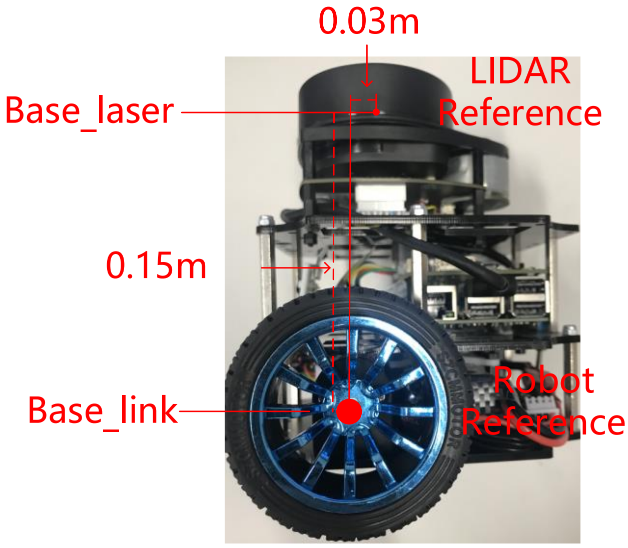


Figure 4.9 – LIDAR reference to platform reference

After obtaining the relationship between LIDAR data and coordinate system conversion, */gmapping* can construct an environment map with the platform center as the coordinate system.

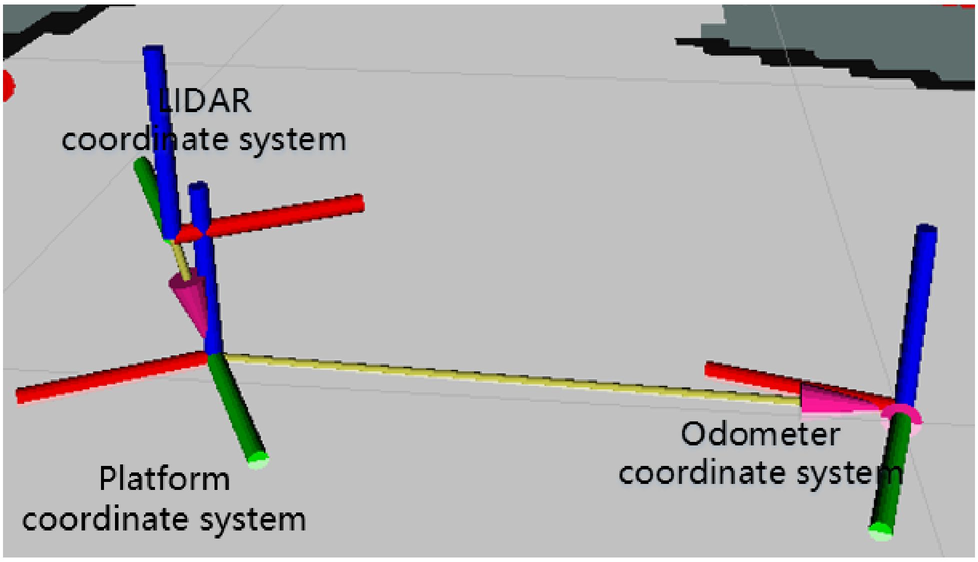


Figure 4.10 – Coordinate conversion in RVIZ

After acquiring the LIDAR point cloud data, the test platform can realize its own positioning in the environment, but the LIDAR cannot scan all the environments in one location, and needs to manipulate the test platform to complete the exploration of the environment, using ROS package: *teleop_twist_keyboard* [53] sends a move command to the test platform, and when the move command is issued, the encoder and the IMU sensor convert the move command into the platform odometer information. During the movement of the platform, the point cloud data of the LIDAR also changes.

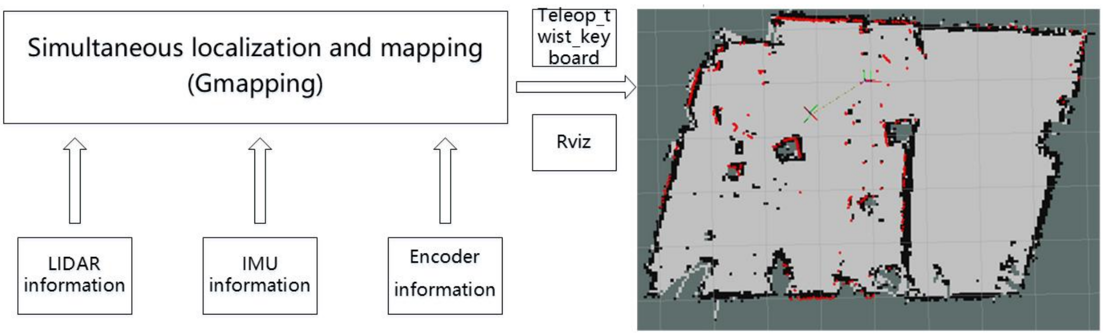


Figure 4.11 – The SLAM flow chart

The SLAM flow chart is shown in Figure 4.11. LIDAR point cloud and platform odometer information data are provided for the *gmapping* algorithm. Use the telescopic keyboard control program to control the platform, explore the modeling of the lab's 2d environment, and determine the location of the platform on the map.

Chapter 5 Result and Discussion

In this chapter, the environmental map measured by the LIDAR-lite-v3 2d environment scanner is displayed, and the measurement error of the system is verified by multiple measurements. The error analysis shows the accuracy and effectiveness of the system. Afterwards, the experiment of the first test platform's obstacle avoidance strategy was demonstrated. The obstacle avoidance effect of the platform was tested in three obstacle avoidance scenarios. The reliability of the obstacle avoidance strategy was verified in these test scenarios.

On the second experimental platform, the SLAM of the laboratory environment was completed and a comparison of the real environment and SLAM results was shown, indicating that the results were accurate. Then based on the measured environmental map, the three scenarios in the simulated port were tested for path planning and obstacle avoidance, and the moving path of the platform was demonstrated, which provided a reference for the application of LIDAR SLAM technology in the port environment.

The second research platform was developed based on the improvement of the previous platform. The second platform is to make up for the previous platform without its own positioning and the ability to continuously observe the environment. In the experimental session, the measurement results of the two platforms were compared, and the first platform was used to simulate the obstacle AGV, the second platform was used to simulate the moving AGV, the two platforms were used to simulate the multiple AGV obstacle avoidance experiment in the static and dynamic scenarios.

5.1 Results of first test platform

5.1.1 2D Environmental mapping by LIDAR-lite-v3

The pre-set scene was scanned and tested using the developed 2D laser scanner. The scene was expressed by an indoor plane, and three target boxes were set as the detection targets. The scene and test results are shown in Figure 5.1.

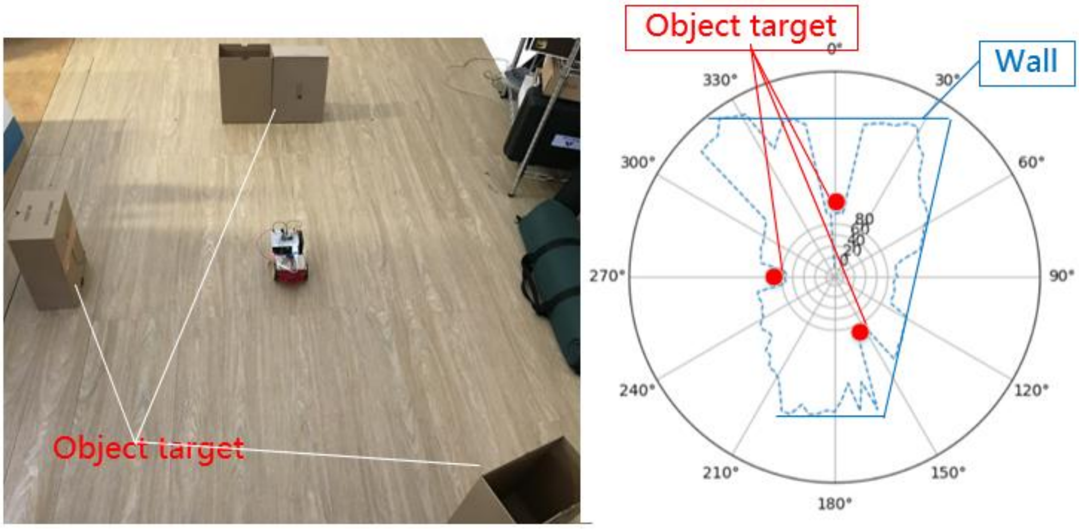


Figure 5.1 - Test scenario and corresponding point cloud map

In the figure, the first picture shows the positional relationship of the obstacle in the scene with the laser scanner, and the second picture is an angle-distance polar map centered on the laser scanner. As can be seen from the polar plot, the first obstacle is located in the range of 352 degrees - 360 degrees, 0 degrees - 7.2 degrees, within this range, the scanner reads 5 data:

352.8 degrees: 100cm;

356.4 degrees: 97 cm;

0 degree: 99 cm; 3.6 degrees,

101 cm; 7.2 degrees: 100 cm.

The second obstacle is in the range of 147 degrees to 158 degrees and reads 4 data:

147.6 degrees: 101cm;

151.2 degrees: 101cm

154.8 degrees: 99cm

158.4 degrees: 98cm

The third obstacle is located in the range of 262 degrees to 277 degrees and records 5 data:

262.8 degrees: 74 cm;

266.4 degrees: 73cm;

270 degrees: 75cm;

273.6 degrees, 72cm;

277.2 degrees, 76cm

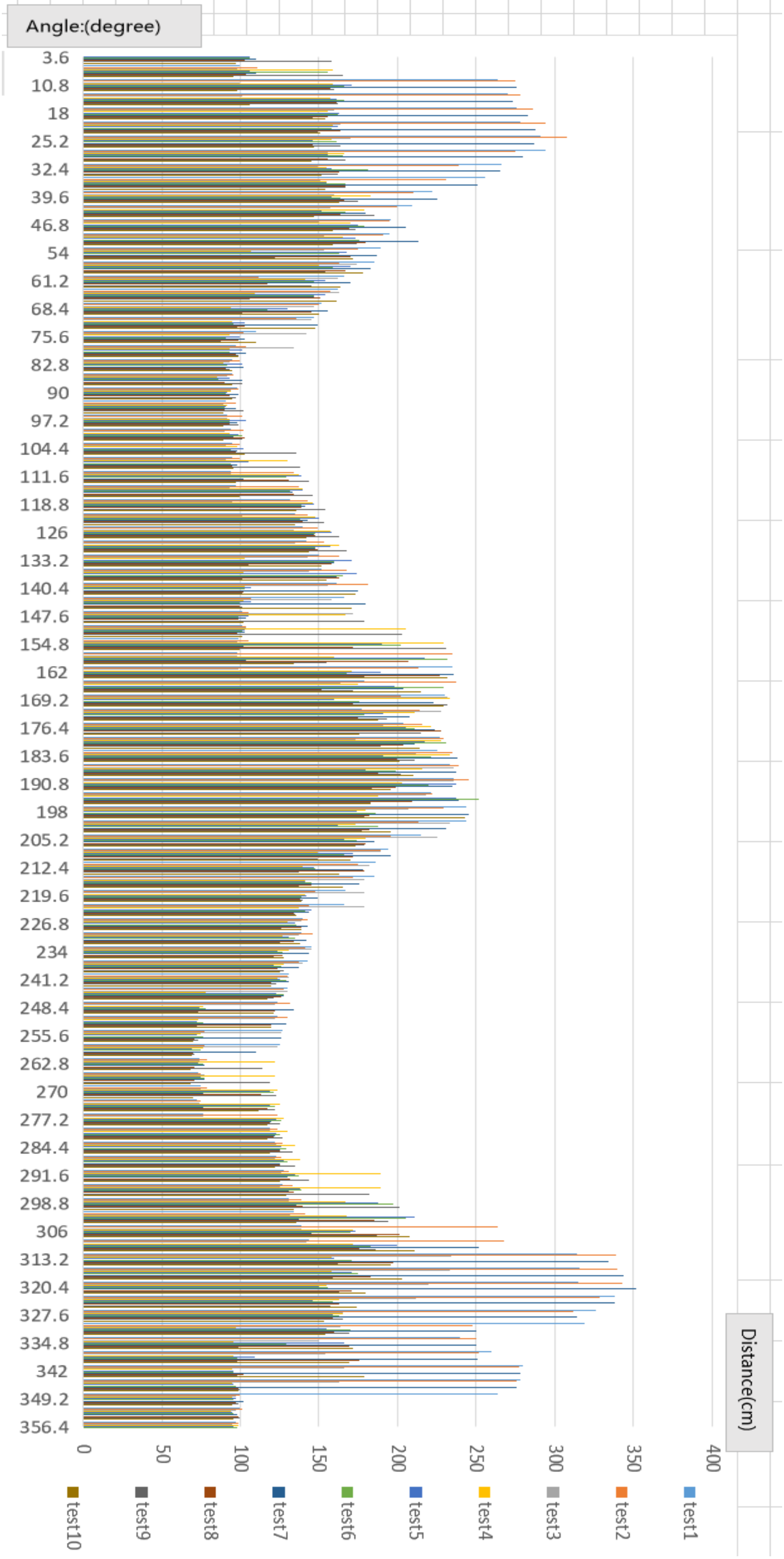


Figure 5.2 - Measurement error of 2D laser scanner

An error comparison diagram of the 10 repeated static measurement results performed in the scene of Figure.5.1 is shown in Figure.5.2. In this static scene, in order to test the accuracy of the laser 2d scanner's testing of the surrounding environment, the scene is set in the indoor plane of the laboratory, three target objects are set in the scene to be used as the detection reference objects. In this scenario, repeated experiments were performed using a laser scanner, and 10 sets of duplicate data were recorded. In Figure 5.2, the horizontal axis is the measured distance in centimeters, and the vertical axis is the angle in degrees. It can be seen from the change of the distance value of the corresponding angle in the figure that in the results of 10 repeated tests, the measured values are basically consistent in the range of 46 degrees - 300 degrees, in the range of 3.6 degrees - 40 degrees, 306 degrees – 349degrees. There is a certain difference in the measured distance values. The overall measurement results tend to be stable, indicating that the 2d laser scanner has certain measurement capabilities and still needs improvement. Error analysis: The cause of the measurement error can be attributed to the abnormal operation of the stepping motor in the interval where the error exists, causing the measurement result not to correspond to the angle value, and displaying the false information.

5.1.2 Obstacle avoidance test based on obstacle avoidance strategy

In order to test the obstacle avoidance strategy proposed in Section 3.3.6, to test the effectiveness of the obstacle avoidance strategy, three scenarios were established: corner; obstacles at various angles; dead ends.

In Figure 5.3, D1 is the maximum value and is greater than the safety distance (35cm – 100cm) determination value, and the remaining D2-D7 are smaller than the safety distance determination value. At this time, the platform turns 90 degrees to the left. The result has shown in Figure 5.4.

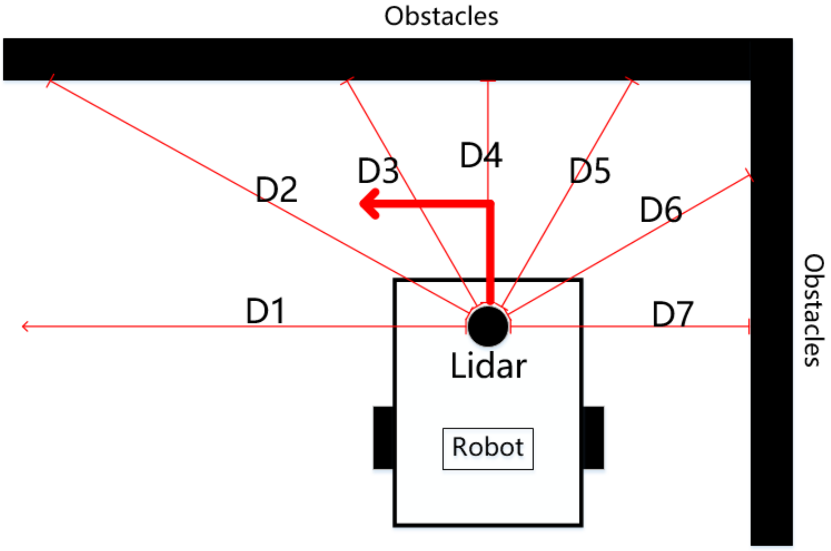


Figure 5.3 - Experimental scenario: corner

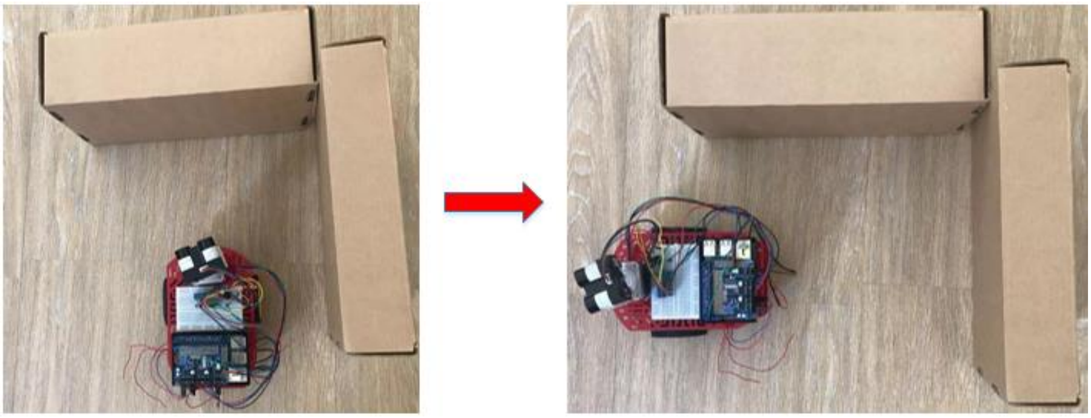


Figure 5.4 - Corner test results

In Figure 5.5, D3 and D4 are both greater than the safety distance determination value, D1-D2, D5-D7 are smaller than the safety determination distance, and the platform will turn to the direction of D3 because D3 is larger than D4. Figure 5.6 shows the test result.

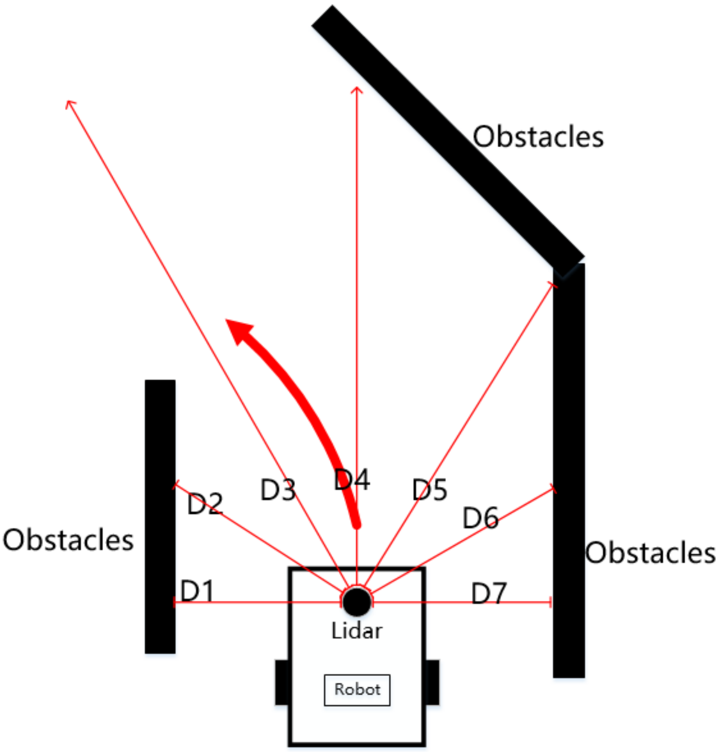


Figure 5.5 - Experimental scenario: obstacles at various angles

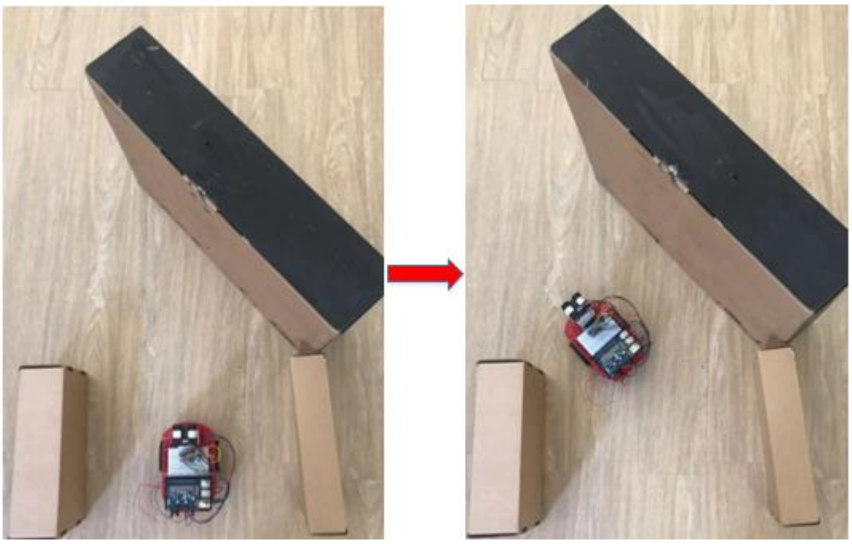


Figure 5.6 - Various angles test result

In Figure 5.7, D1-D7 are both smaller than the safety distance determination value, and the platform will reverse the direction by 180 degrees in this case. The reverse action of the platform can be seen in Figure 5.8.

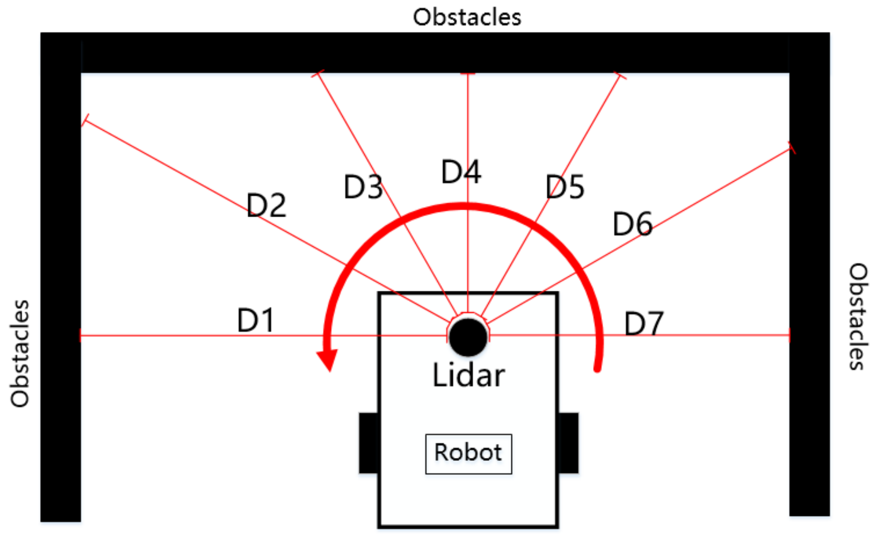


Figure 5.7 - Dead ends

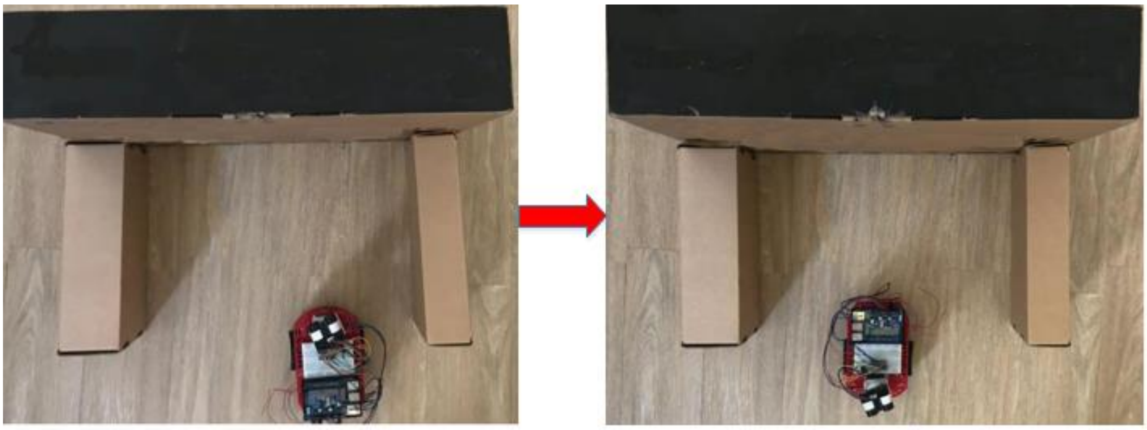


Figure 5.8 - Dead zoom test results

5.2 Results of second test platform

The length of the four walls and the shape of the house are shown in Figure. 5.9. In the lab there are some cabinets and stacked equipment in the laboratory, and the chair table, chairs and tables are scanned in the scanning plane to a shape of four points. As can be seen in the Figure 5.10, obstacles in the laboratory are shown in the picture.

From a top view, the laboratory is a quadrilateral on the 2d plane. The length of the 4 walls is:

Wall with door: 7m;

Wall with windows: 7.2m;

Third wall: 5m;

Fourth wall: 5.47m

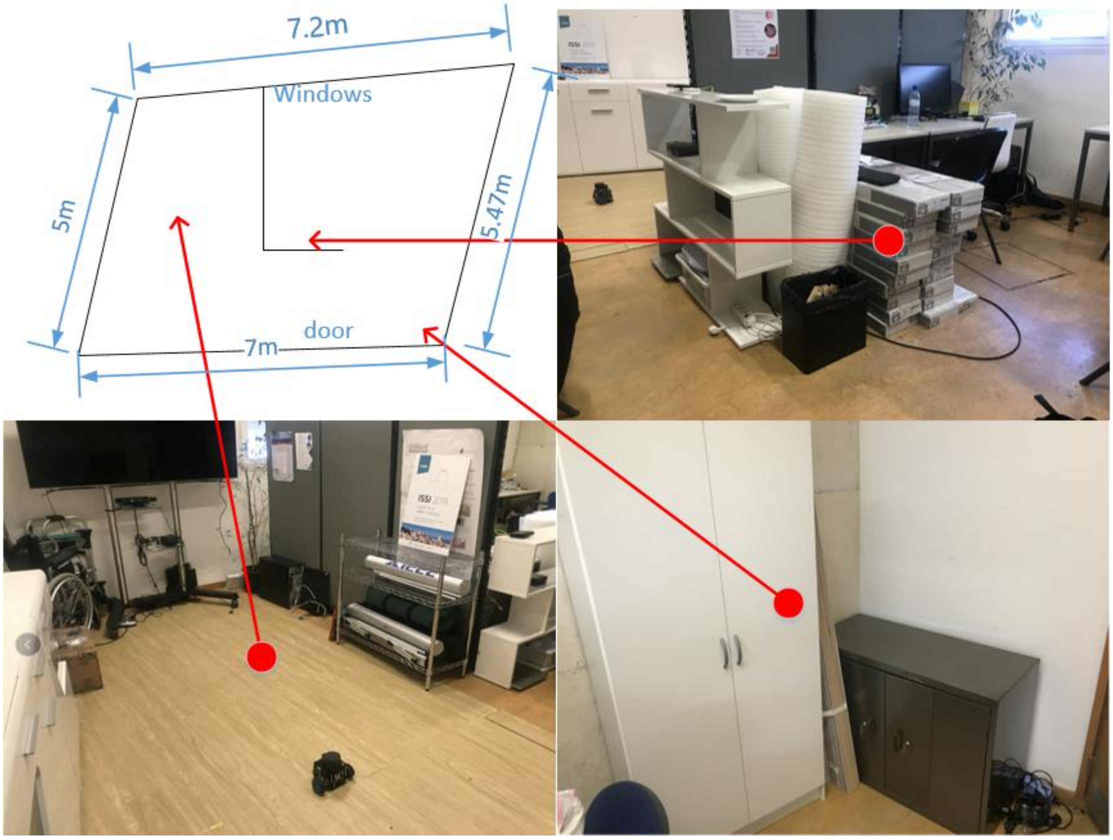


Figure 5.9 - Lab real scene

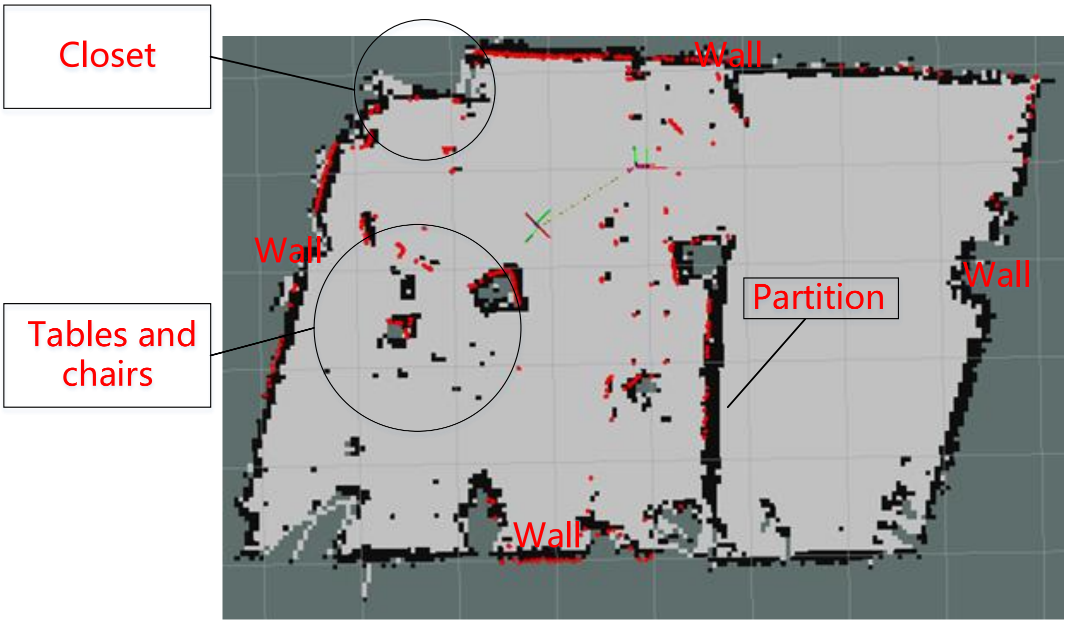


Figure 5.10 – The SLAM result of the lab

Analysis of the results: After comparing the actual image and test results of the laboratory, the experimental results show the position of the laboratory walls, partition, cabinets, chairs, etc., and the angle of the established wall is similar to the actual one. Experimental results show that the SLAM system can get good results.

5.3 Port simulation scenario test results

In this chapter, an actual scenario of the simulated port will be established, and the test platform will be placed in the environment to test the navigation effect and obstacle avoidance effect of the test platform to verify the effect of the lidar in the port environment.

Firstly, the working environment of the automated container port is introduced. Taking the latest built Yangshan port phase 4 of the Shanghai Port as an example, see Figure5.11-5.12, the port environment is a closed, unmanned environment, divided into the shore operation area, the yard and the connection horizontal transport zone. In the shore operating area, it is mainly the crane responsible for loading and unloading containers; the container yard is stacked with containers. In Yangshan Port phase 4, considering the high working efficiency of AGV, the yard is set to be perpendicular to the shore. In the horizontal transportation area, the AGV road is set to travel in one direction. As shown in Figure 5.14.

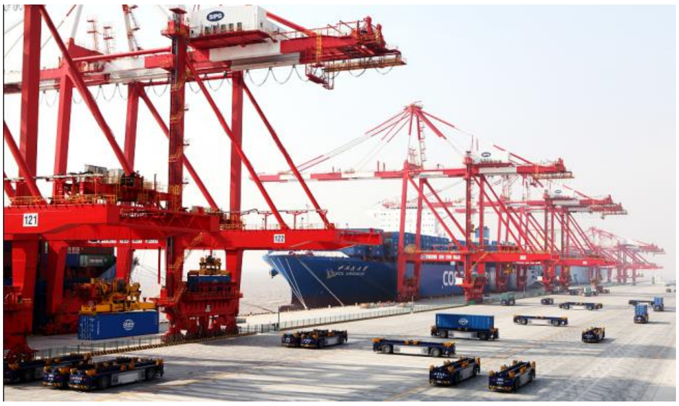


Figure 5.11 - The shore operation area of the Yangshan port phase4

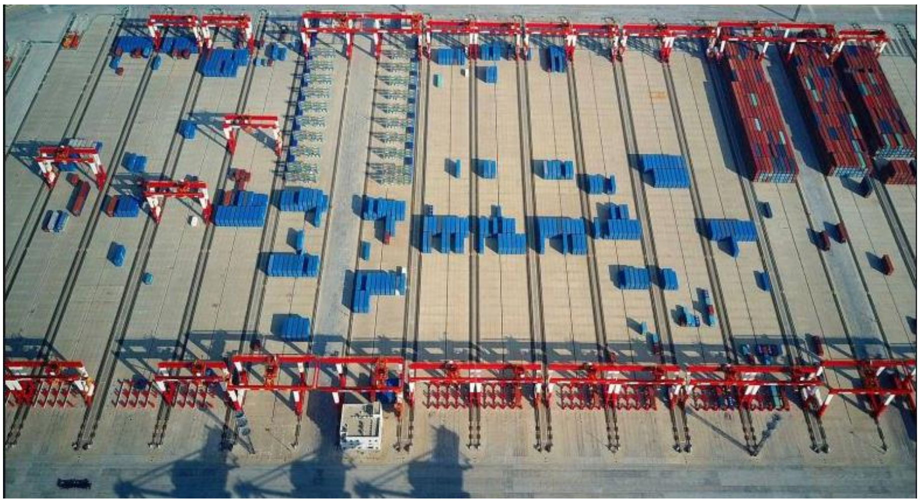


Figure 5.12 - The yard area of the Yangshan port phase4

Based on the description of the automated port in the previous paragraph, we have established a miniature port simulation scenario. The entire port area can be divided into a shore operation area; a horizontal transportation area; a yard area, the main active area of the AGV is in the shore operation area and the horizontal transportation area, and the area shown in Figure 5.12 is the shore operation area. In the area, the shore crane is responsible for the loading and unloading of containers. The AGV is responsible for transporting the unloaded containers to the yard and transporting the loaded containers to the ship in this area.

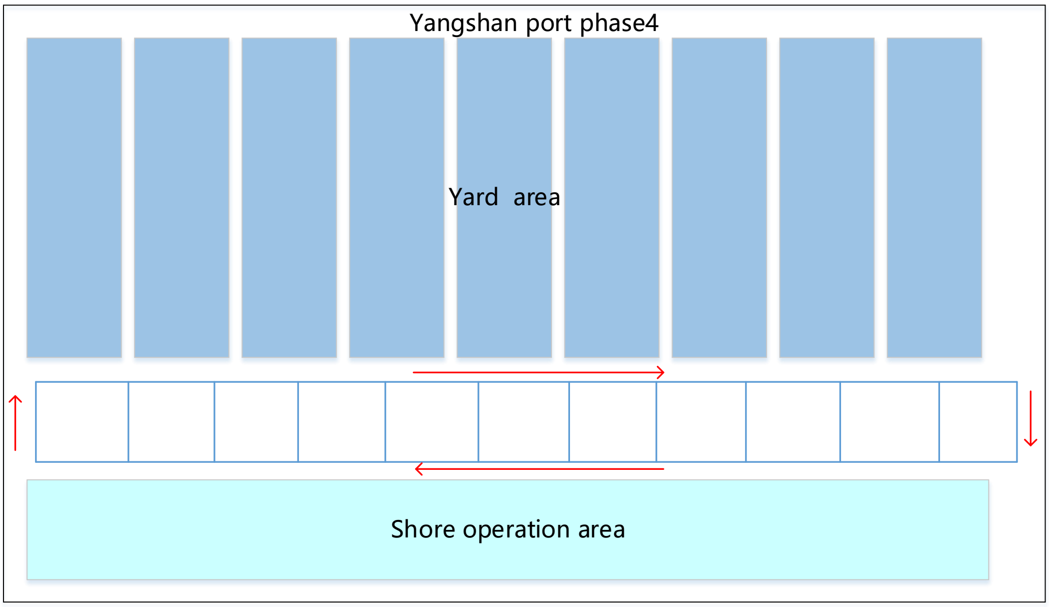


Figure 5.13 - The simulation diagram of the automated port

In the port environment simulation, first platform is used to simulate the obstacle AGV. In the port, other moving or stationary AGVs can block the moving AGV, so the first experimental platform is used to simulate the obstacle AGV. This AGV can be used as a static and dynamic obstacle and can be remotely controlled by a computer. A paper box is installed on the upper part of the AGV to simulate the container. The height of the paper box is enough to be scanned by the LIDAR. The physical picture is shown in Figure 5.14.

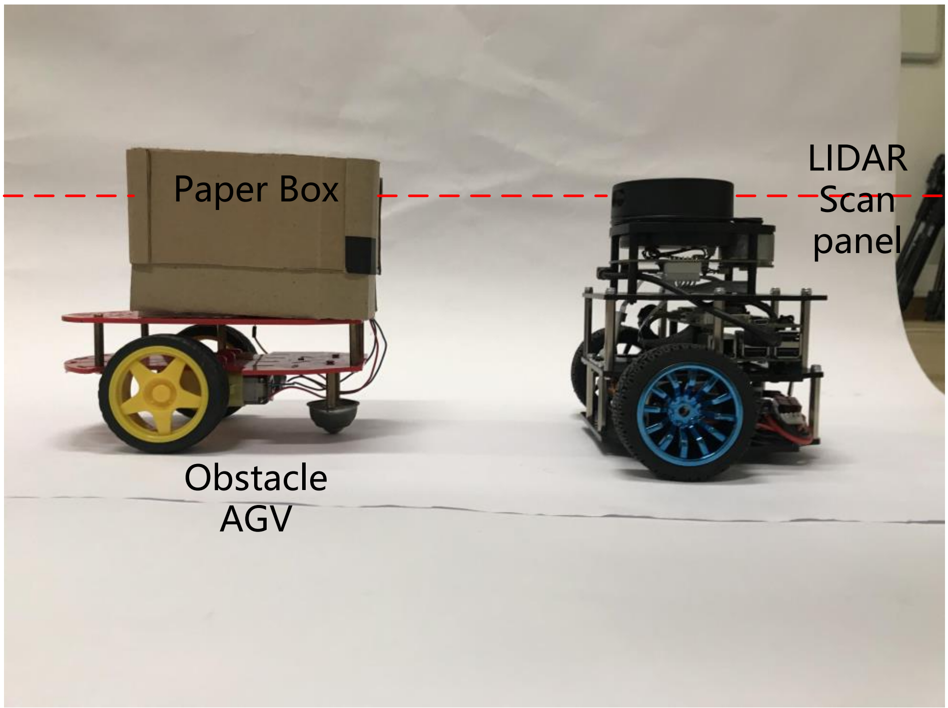


Figure 5.14 - Simulated the AGVs for obstacle avoidance test

5.3.1 Static scene

In the shore operating area, the shore crane is responsible for the loading and unloading operations of the container, and the AGV is responsible for transporting the unloaded containers to the yard and transporting the loaded containers to the ship in this area. In this process, it is a static obstacle avoidance scenario, which is described below:

The AGV will stop at the designated position and wait for the assignment of the mission. At the same time, when other AGVs need to leave the working area after completing the mission, they need to avoid the AGV waiting in front. This is a static obstacle avoidance scenario for the AGV shore operating area. The specific obstacle avoidance process is shown in Figure 5.15.

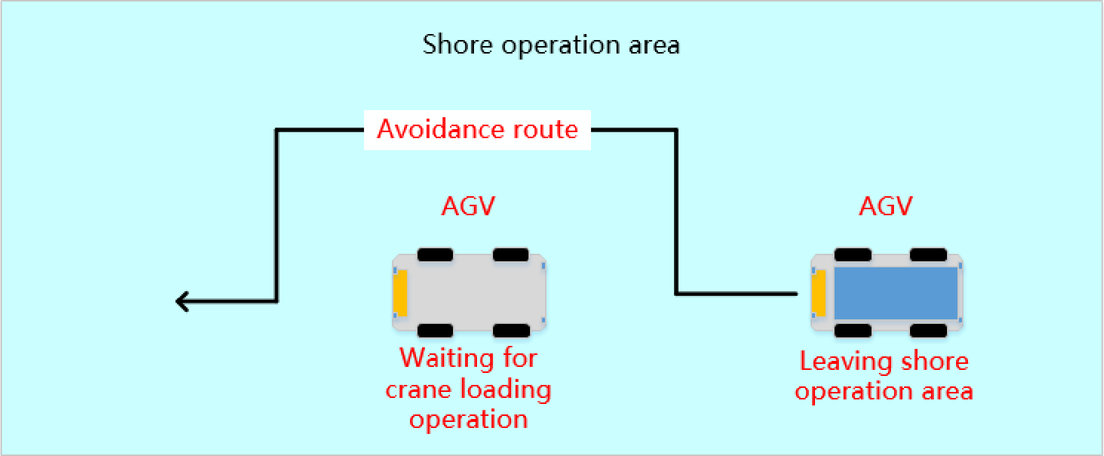

Shore operation area
Avoidance route
AGV
AGV
Waiting for crane loading operation
Leaving shore operation area

Figure 5.15 - Static obstacle avoidance scenario at shore operating area

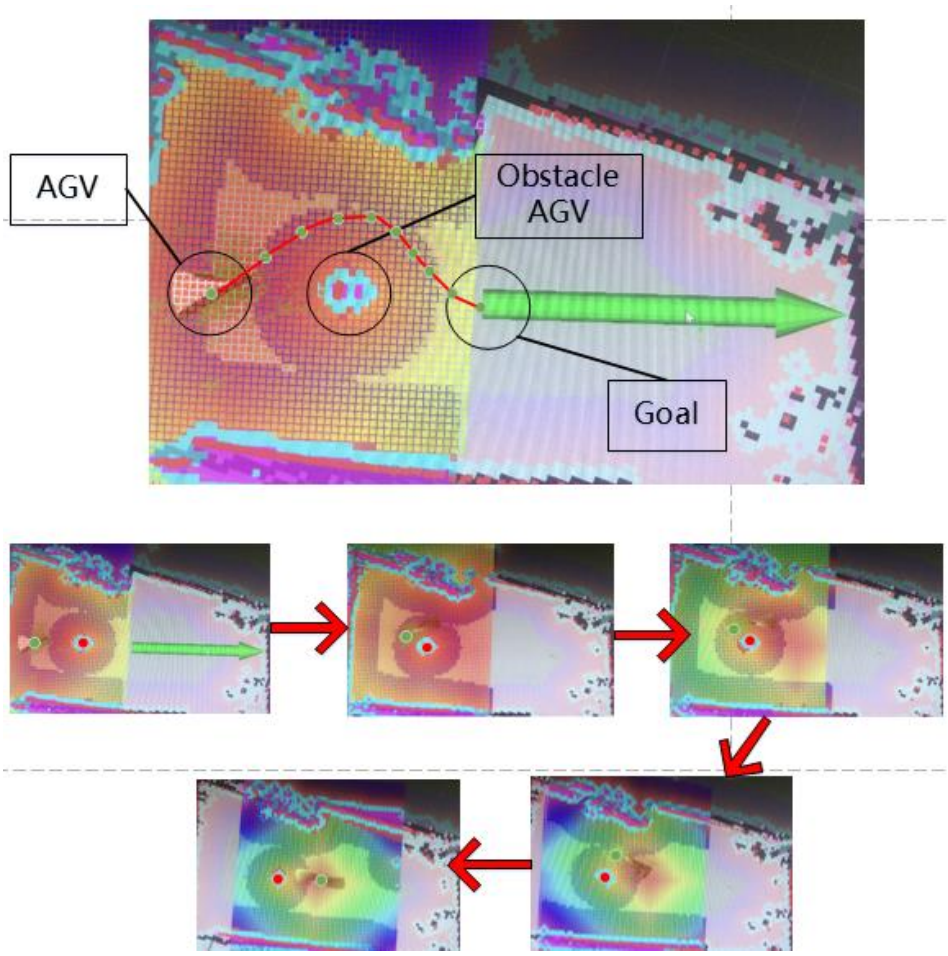


Figure 5.16 - Static obstacle simulation test results

When using the platform to simulate the port static model, first set the target point, which is the starting point of the green arrow in the figure. In the Figure 5.16, the red dot represents the obstacle AGV, the moving path of the moving AGV is marked with a green dot and a red line. In this scenario, the LIDAR detected surrounding environmental information and find no dynamic obstacles. The AGV planned the moving route, avoided the obstacles, and successfully reached the target point.

5.3.2 Dynamic scene

5.3.2.1 Dynamic obstacle avoidance scenario: two AGVs in the same direction

In the previous chapter, a static obstacle avoidance scenario was mentioned, in which an AGV had a stopped AGV in front of it when it left the work area. However, in the seaside operating area, when an AGV is reaching the target location and doing deceleration, at this time, there is another AGV behind the AGV that is about to stop. This AGV needs to leave the area to complete the next task. This scenario is a dynamic obstacle avoidance scenario, the specific scenario is shown in Figure 5.17.

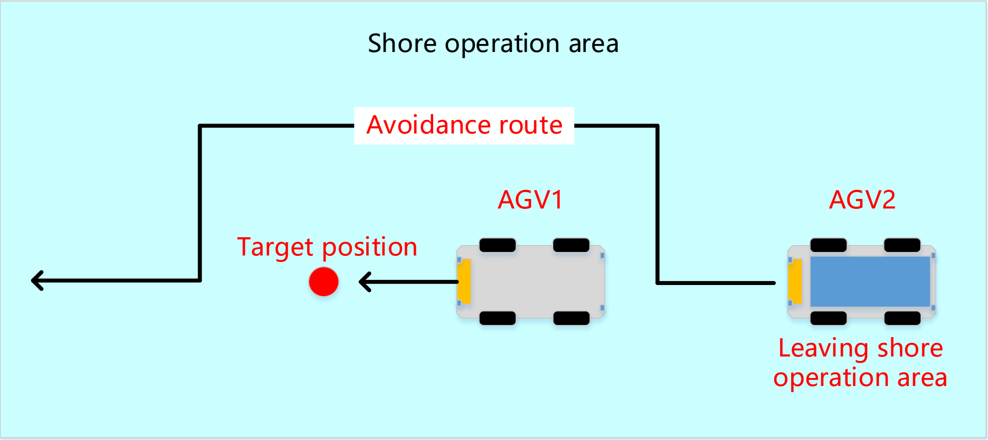


Figure 5.17 - Dynamic obstacle avoidance scenario at shore operating area

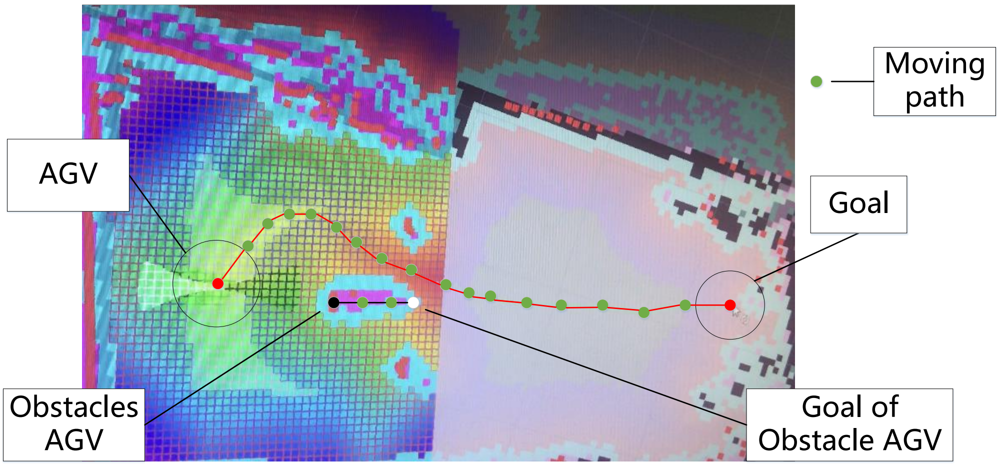


Figure 5.18 - Dynamic obstacle simulation test results of two AGVs in the same direction: Obstacle avoidance scenario description

The obstacle avoidance scenario is described in Figure 5.18. The red dot represents the moving AGV, the moving path of the moving AGV is marked with a green dot and a red line, the black dot represents the obstacle AGV, and the white dot is the end point of its movement. The path of obstacle AGV is marked with a black line and a green dot.

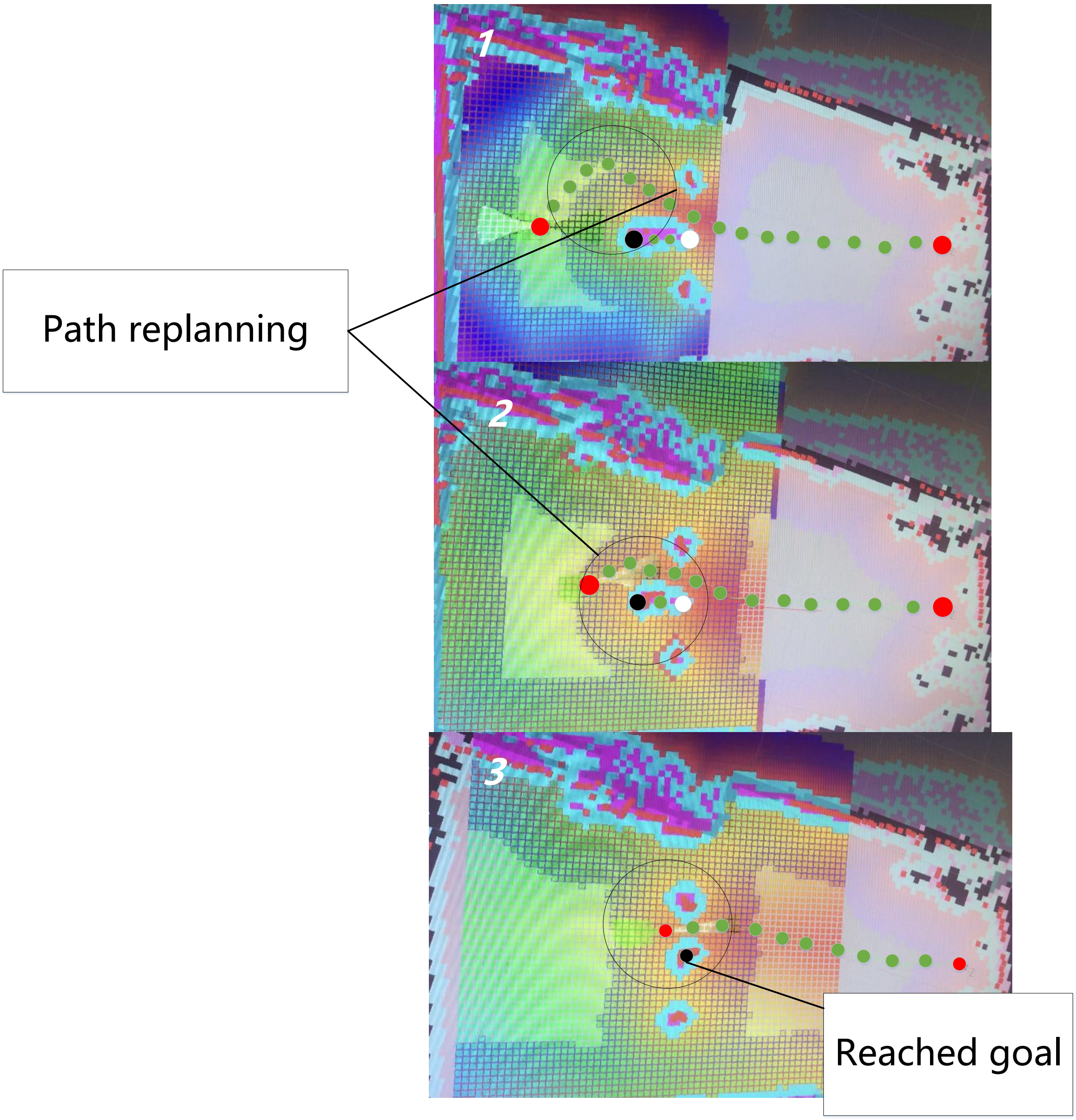


Figure 5.19 - Dynamic obstacle simulation test results of two AGVs in the same direction: Obstacle avoidance process 1

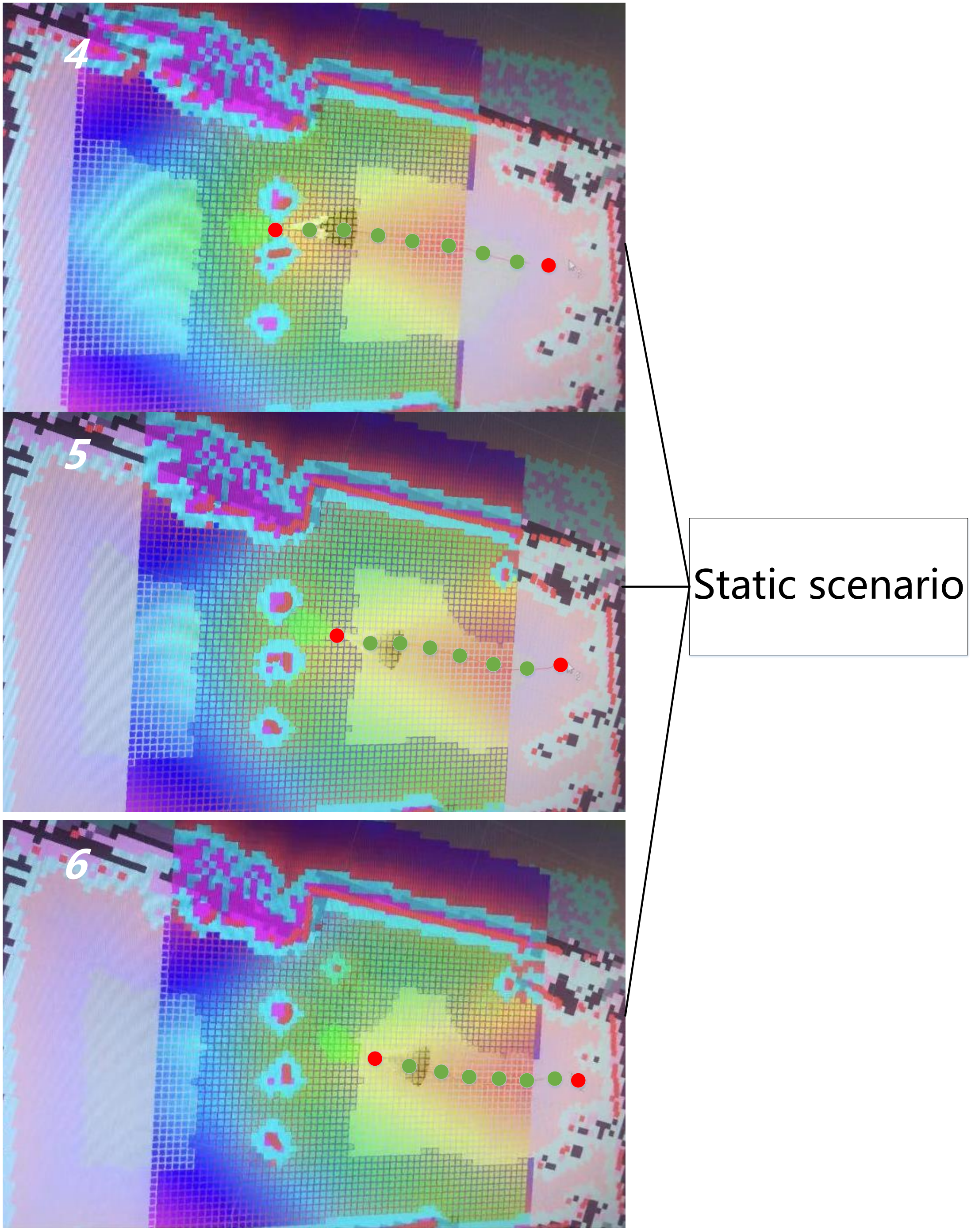


Figure 5.20 - Dynamic obstacle simulation test results of two AGVs in the same direction: Obstacle avoidance process 2

In this scenario, the process of obstacle avoidance is divided into two figures (Figure 5.19, Figure 5.20) to show. In this dynamic scene, in the Figure 5.19, the moving AGV initially detects obstacles in front and plans a path to avoid obstacles. However, when the AGV (obstacle) in front is decelerating, the platform re-plans the path. With the moving of obstacles AGV, the obstacle avoidance path also changes.

In the Figure 5.20, when the obstacle AGV reached the goal point, the dynamic scene also turns into a static scene, there are no obstacles between the moving AVG and the target point and the moving AGV reached the target point.

5.3.2.2 Dynamic obstacle avoidance scenario: two AGVs in the same direction

In the horizontal transportation area, the AGV travels according to a certain path. The specific path is as shown in the Figure 5.21, the path in the outer circle is a single direction path, on the middle path, driving in both sides. On these paths, there is a high probability that two AGVs will meet, so based on this situation, I use two platforms to simulate this scenario, setting two AGVs in one path to face each other.

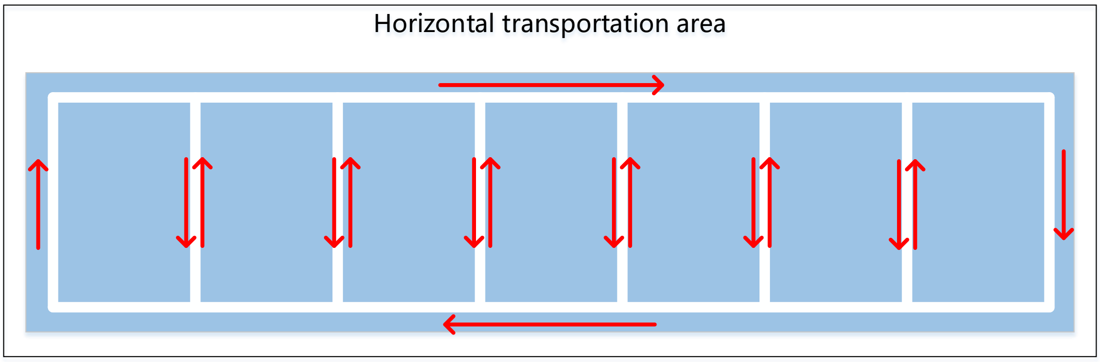


Figure 5.21 - Route path in horizontal transportation area

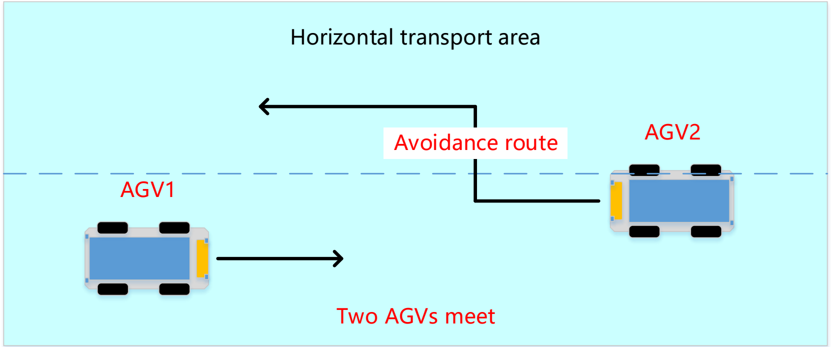


Figure 5.22 - Two AGVs encounter scenarios

When using the platform to simulate the two AGV encounter dynamic model models, the platform detects the oncoming AGV in front, and immediately decelerates to re-plan the path, avoiding obstacles and reaching the target point. The whole process is shown in Figure 5.23. In this picture, the red point is the position of the experimental platform, the green point is the path point, the black point is the position of the obstacle AGV, and the white point is its target point.

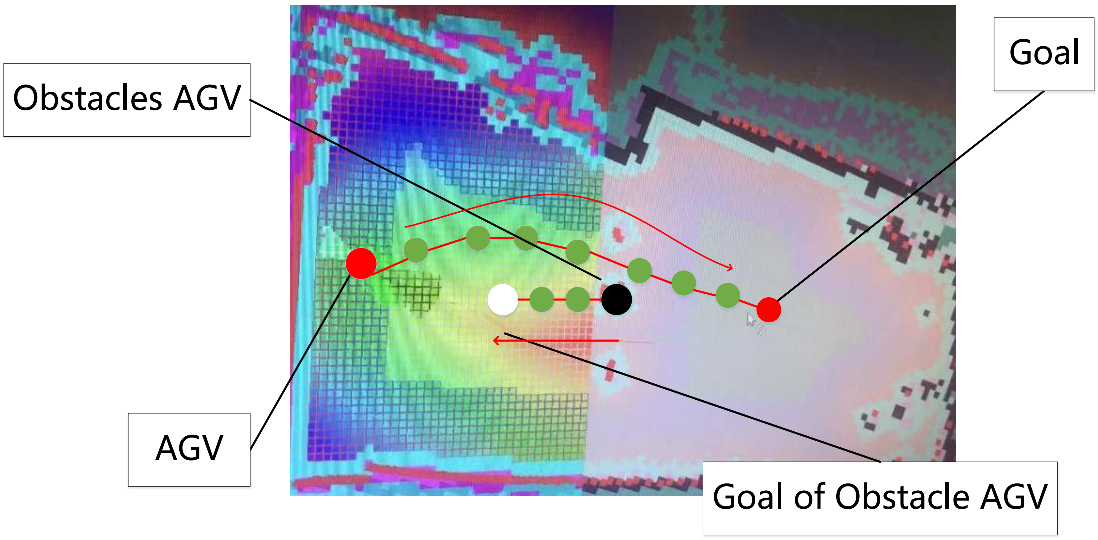


Figure 5.23 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance scenario description

The obstacle avoidance scenario is described in Figure 5.23. The red dot represents the moving AGV, the moving path of the moving AGV is marked with a green dot and a red line, the black dot represents the obstacle AGV, and the white dot is the end point of its movement. The path of obstacle AGV is marked with a black line and a green dot. The moving direction of the two AGVs is marked in the figure

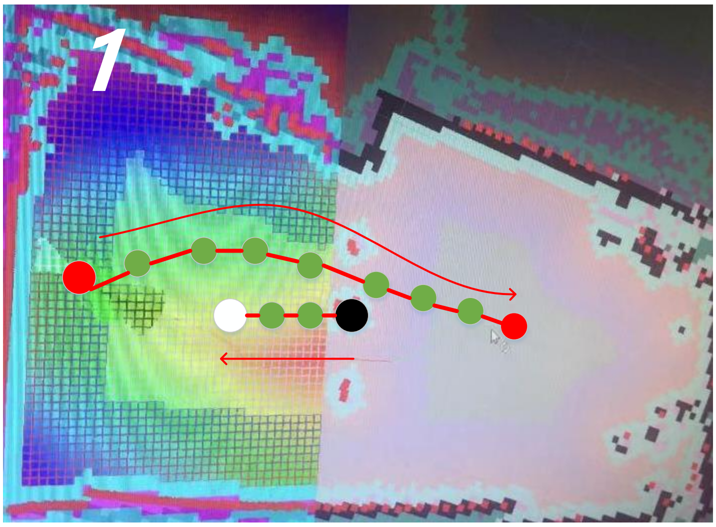


Figure 5.24 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance process 1

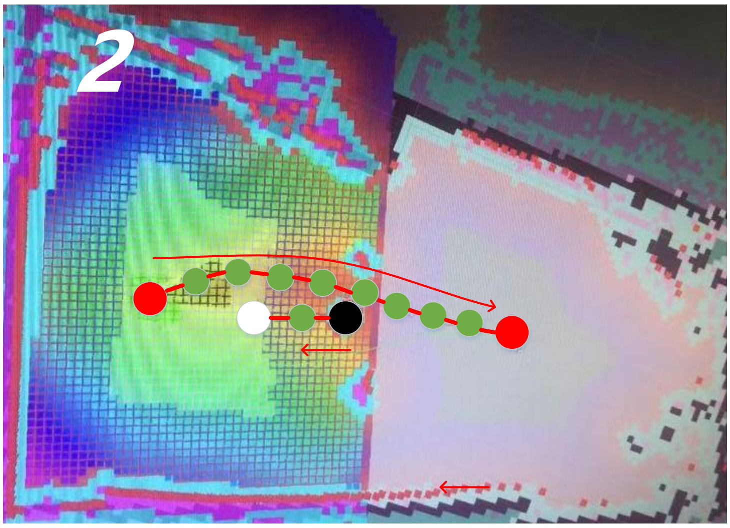


Figure 5.25 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance process 2

In Figure 5.24 and Figure 5.25, two AGVs move on one path, and the moving AGV detects the moving obstacle AGV and plans an escape route.

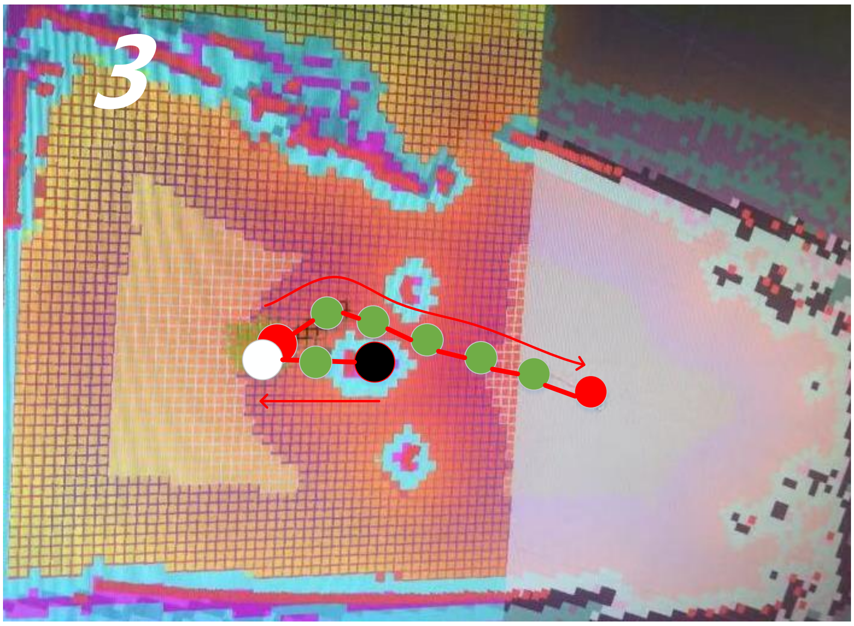


Figure 5.26 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance process 3

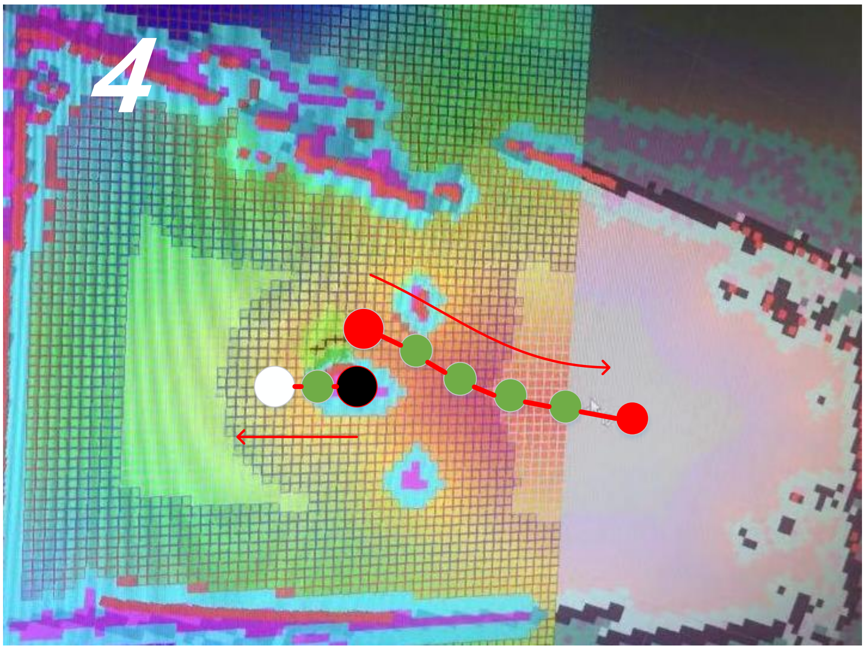


Figure 5.27 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance process 4

As the two AGVs move, the path also changes. In Figures 5.26 and 5.27, the moving AGV avoids the path of the obstacle AGV and successfully avoids it.

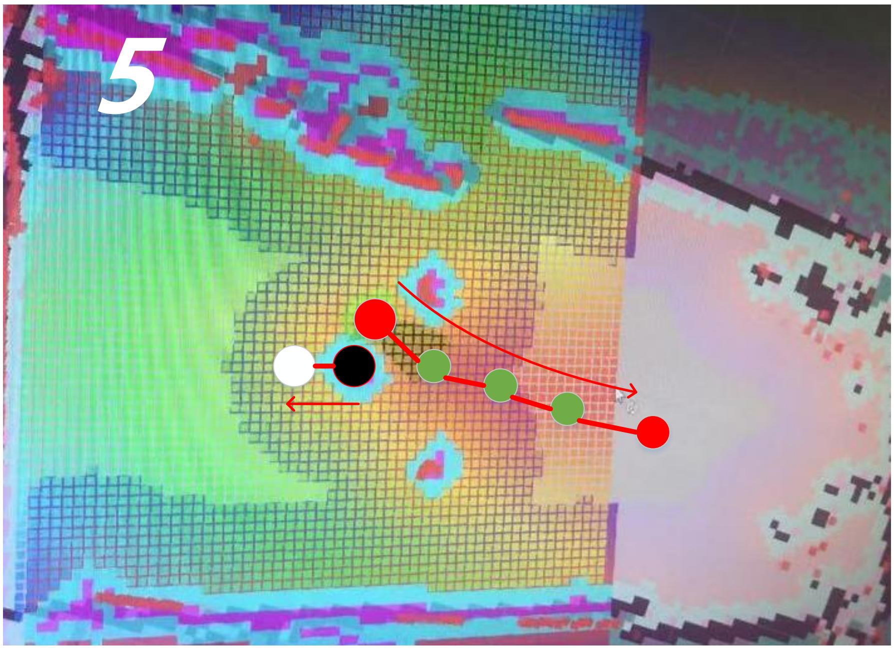


Figure 5.28 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance process 5

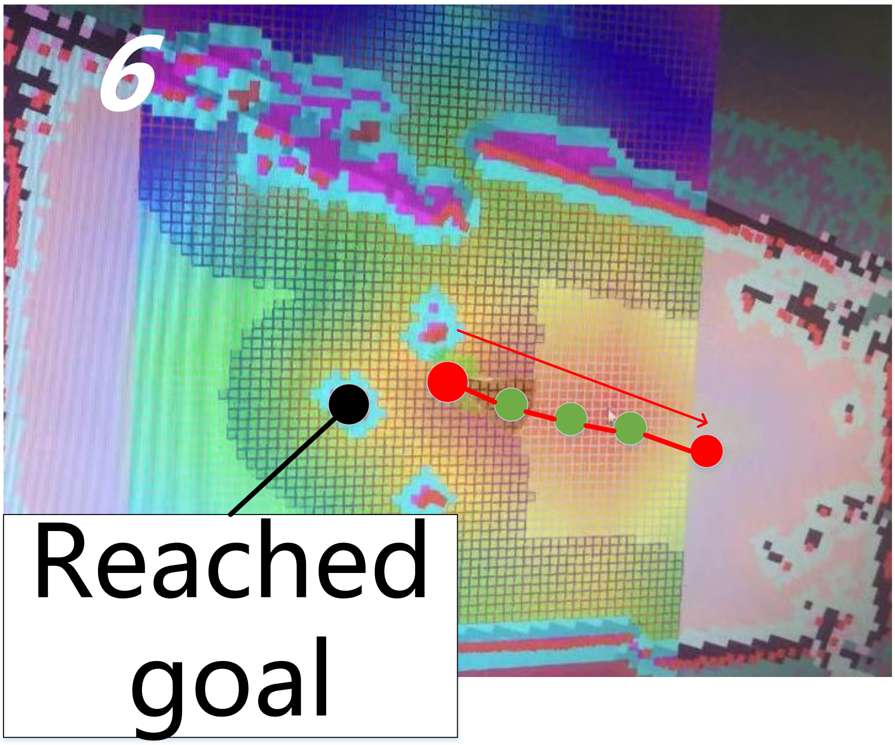


Figure 5.29 - Dynamic obstacle simulation test results of two AGVs meet: Obstacle avoidance process 6

In Figure 5.28, after the AGV avoids the obstacle AGV, it returns to the path and continues to move. In Figure 5.29, the obstacle AGV reaches the target point to perform loading and unloading operations. At this time, the entire dynamic obstacle avoidance process ends.

Chapter 6 Conclusions and Future Work

6.1 Conclusions

Considering the trend and the necessities to use AGV in the automation containers terminals, this MSc thesis attempts to apply the LIDAR sensing to provide environmental scanning capability for AGV positioning and obstacle avoidance. Therefore, the main research focused on SLAM based on LIDAR technology.

This study included two mobile test platforms. The first 2D environment scanner based on LIDAR sensor and Raspberry Pi computation platform was developed as part of the thesis. Using this implementation, the distance information at a specific angle is obtained. Additionally, an obstacle avoidance strategy based on LIDAR point cloud information was implemented and tested. Thus, the acquisition of environmental information in a specific scenario was carried out to be used for testing the obstacle avoidance strategy based on distance measurement. The implemented validation procedure highlights the effectiveness of the implemented strategy.

To improve the capabilities of the AGV obstacle detection a 2D environment scanner based on LIDAR was also developed. This scanning system has a 360 degrees 2D plane scanning capability and accurate environment modeling capability. The system is used to accurately locate the obstacles in the environment and obtain a 2D environment map. For this second experimental platform that is mounted on an AGV emulator a complete SLAM system was developed. The SLAM solution is based on the usage of LIDAR, encoder and IMU. Software development under the framework of ROS system was carried out. The developed software assures the ability to integrate three kinds of sensor information. Using the implemented hardware and software the experimental platform can scan the unknown environment and locate itself. Obstacle avoidance function was implemented and tested with good results.

6.2 Future work

After two types of implementations and performance analysis of the developed system, considering the system functionalities and the real environment of the port, the 2D laser environment scanning can be still considered with limitations. Thus, the laser scanner installed on the AGV, is still characterized by scanning blind spots, so the solution will not provide complete information about the surrounding environment. Considering this we are planning to add new sensors that can compensate the limitations for 2D Laser scanning in 3D environment, such as monocular camera, depth camera, etc. Integrate the information obtained by 3D depth information LIDAR to realize 3D environment modeling. Additionally, the sensor fusion software will be developed.

References

[1] Günther H O, Kim K H. Container terminals and terminal operations[J]. 2006.

[2] Steenken D, Voß S, Stahlbock R. Container terminal operation and operations research-a classification and literature review[J]. OR spectrum, 2004, 26(1): 3-49.

[3] Li Mingsen. Application and development of new automation technology in ports[J]. port science and technology, 2006(6):2-7.

[4] Yang Y, Zhong M, Yao H, et al. Internet of things for smart ports: Technologies and challenges[J]. IEEE Instrumentation & Measurement Magazine, 2018, 21(1): 34-43.

[5] Bostelman R V, Hong T H, Madhavan R. Towards AGV safety and navigation advancement obstacle detection using a TOF range camera[C]//ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005. IEEE, 2005: 460-467.

[6] Wiedemann M, Sauer M, Driewer F, et al. Analysis and characterization of the PMD camera for application in mobile robotics[J]. IFAC Proceedings Volumes, 2008, 41(2): 13689-13694.

[7] Lee S Y, Yang H W. Navigation of automated guided vehicles using magnet spot guidance method[J]. Robotics and Computer-Integrated Manufacturing, 2012, 28(3): 425-436.

[8] Yue-Quan W . Method of AGV transponder installation in automated terminal of Yangshan Deepwater Port[J]. Port & Waterway Engineering, 2018.

[9] Baird R. An autonomous forklift research platform for warehouse operations[D]. Massachusetts Institute of Technology, 2018.

[10] Temeltas H. A REAL-TIME LOCALIZATION METHOD FOR AGVS IN SMART FACTORIES[J]. system, 1(1): 1.

[11] De Simone M, Rivera Z, Guida D. Obstacle avoidance system for unmanned

ground vehicles by using ultrasonic sensors[J]. Machines, 2018, 6(2): 18.

[12] Liu J, Jayakumar P, Stein J L, et al. A multi-stage optimization formulation for MPC-based obstacle avoidance in autonomous vehicles using a LIDAR sensor[C]//ASME 2014 dynamic systems and control conference. American Society of Mechanical Engineers, 2014: V002T30A006-V002T30A006.

[13] Peng Y, Qu D, Zhong Y, et al. The obstacle detection and obstacle avoidance algorithm based on 2-d lidar[C]//2015 IEEE International Conference on Information and Automation. IEEE, 2015: 1648-1653.

[14] Clark S, Dissanayake G. Simultaneous localisation and map building using millimetre wave radar to extract natural features[C]//Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C). IEEE, 1999, 2: 1316-1321.

[15] Wong S. The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards[J]. SANS Institute, 2003: 1-9.

[16] Generational Wi-Fi® User Guide, Wi-Fi Alliance, October 2018

[17] Lee J S, Su Y W, Shen C C. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi[J]. Industrial electronics society, 2007, 5: 46-51.

[18] Miller B A, Bisdikian C. Bluetooth revealed: the insider's guide to an open specification for global wireless communication[M]. Prentice Hall PTR, 2001.

[19] Ylonen T, Lonvick C. The secure shell (SSH) protocol architecture[R]. 2005.

[20] Bradner S. IETF working group guidelines and procedures[R]. 1998.

[21] Forouzan B A, Fegan S C. TCP/IP protocol suite[M]. McGraw-Hill, 2006.

[22] Kozierok C M. The TCP/IP Guide v3. 0[J]. Tcpiguide. com, 2005.

[23] Dean T. Network+ guide to networks[M]. Cengage Learning, 2012.

[24] “Robotic Operation System” [Online]. Availabe: https://en.wikipedia.org/wiki/Robot_Operating_System. [Accessed: 15-Jun-2019]

[25] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I[J]. IEEE robotics & automation magazine, 2006, 13(2): 99-110.

[26] Chen X, Zhang H, Lu H, et al. Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue[C]//2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). IEEE, 2017: 41-47.

[27] Wongsuwan K, Sukvichai K. Generalizing corrective gradient refinement in RBPF for occupancy grid LIDAR SLAM[C]//2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2017: 495-500.

[28] Li J, Zhao J, Kang Y, et al. DL-SLAM: Direct 2.5 D LiDAR SLAM for Autonomous Driving[C]//2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019: 1205-1210.

[29] Denysyuk P, Teslyuk V, Chorna I. Development of mobile robot using LIDAR technology based on Arduino controller[C]//2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH). IEEE, 2018: 240-244.

[30] Ibragimov I Z, Afanasyev I M. Comparison of ros-based visual slam methods in homogeneous indoor environment[C]//2017 14th Workshop on Positioning, Navigation and Communications (WPNC). IEEE, 2017: 1-6.

[31] Shen D, Huang Y, Wang Y, et al. Research and Implementation of SLAM Based on LIDAR for Four-Wheeled Mobile Robot[C]//2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE). IEEE, 2018: 19-23.

[32] Ye W, Fan X, Tang J. Real time UGV positioning based on Reference beacons aided LiDAR scan matching[C]//2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS). IEEE, 2018: 1-5.

[33] Cole D M, Newman P M. Using laser range data for 3D SLAM in outdoor environments[C]//Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006: 1556-1563.

[34] Santos J M, Portugal D, Rocha R P. An evaluation of 2D SLAM techniques available in robot operating system[C]//2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2013: 1-6.

[35] “gmapping.” [Online] Available: http://wiki.ros.org/gmapping [Accessed: 16-Jun-2019].

[36] “hector_slam.” [Online] Available: http://wiki.ros.org/hector_slam [Accessed: 16-Jun-2019].

[37] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007 (6): 1052-1067.

[38] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE transactions on robotics, 2015, 31(5): 1147-1163.

[39] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//ICCV. 2011, 11(1): 2.

[40] Detection L L. Ranging—is a remote sensing method used to examine the surface of the Earth[J]. NOAA. Archived from the original on, 2013, 4.

[41] McManamon P F, Dorschner T A, Corkum D L, et al. Optical phased array technology[J]. Proceedings of the IEEE, 1996, 84(2): 268-298.

[42] “Leddartech Pixell.” [Online] Available: https://leddartech.com/lidar/lidar-modules/ [Accessed: 19-Jun-2019].

[43] “RPLIDAR A3.” [Online] Available: https://www.slamtec.com/en/Lidar/A3 [Accessed: 19-Jun-2019].

[44] Richardson M, Wallace S. Getting started with raspberry PI[M]. " O'Reilly Media, Inc.", 2012.

[45] “Raspberry_Pi.” [Online] Available: https://en.wikipedia.org/wiki/Raspberry_Pi [Accessed: 20-Jun-2019].

[46] “Encoder.” [Online] Available: https://en.wikipedia.org/wiki/Encoder [Accessed: 20-Jun-2019].

[47] Furlani E P, Ghosh S K. Micro-encoder with molded micro-magnet: U.S. Patent 5,982,169[P]. 1999-11-9.

[48] “Inertial_measurement_unit” [Online] Available: https://en.wikipedia.org/wiki/Inertial_measurement_unit [Accessed: 22-Jun-2019].

[49] “Firebase.” [Online] Available: https://en.wikipedia.org/wiki/Firebase [Accessed: 22-Jun-2019].

[50] Sakai A, Ingram D, Dinius J, et al. PythonRobotics: a Python code collection of robotics algorithms[J]. arXiv preprint arXiv:1808.10703, 2018.

[51] “SLAMTEC_rplidar_datasheet_A1M8.” [Online] Available: http://www.slamtec.com/[Accessed: 30-jun-2019]

[52] “Robot Visualization tool.” [Online] Available: http://wiki.ros.org/rviz [Accessed: 30-Jun-2019].

[53] “teleop_twist_keyboard.” [Online] Available: http://wiki.ros.org/teleop_twist_keyboard [Accessed: 30-Jul-2019].

Appendix A - Scientific Article

Article: **Obstacle avoidance for unmanned vehicle based on a 2D LIDAR.**
This article has been accepted and presented at the IEEE ISSI 2019 international conference, August 29-30, Lisbon, Portugal and will be published in IEEEXplorer.

IEEE
ISSI 2019
AUGUST 29-30
LISBON - PORTUGAL
2nd International Symposium on Sensing and Instrumentation in IoT Era
ISCTE IUL

Organized by the ISCTE-Lisbon University and Telecommunications Institute in Portugal, Shanghai Maritime University in China, the IEEE International Symposium on Sensing and Instrumentation in IoT Era is an important international event in the fields of sensing, instrumentation and measurement.

Obstacle avoidance for unmanned vehicle based on a 2D LIDAR

Zeyu Ma
ISCTE – Instituto Universitário de Lisboa
Shanghai Martime University
Lisbon, Portugal
zmaue@iscte-iul.pt

Octavian Postolache
Instituto de Telecomunicações Instituto Universitário de Lisboa (ISCTE – IUL)
Lisbon, Portugal
opostolache@lx.it.pt

Yongsheng Yang
Institute of Logistics Science & Engineering
Shanghai Maritime University
Shanghai, China
yangys_smu@126.com

***Abstract*—In recent years, international trade ports have become increasingly busy. As the most important form of transportation in international trade, container shipping plays an important role in reducing the labor cost of terminals, improving port capacity and reducing the energy consumption of loading and unloading operations. AGV is gradually becoming the main tool for container handling in large international ports. It can work continuously and continuously, improving the efficiency of container operation. Due to the complex port environment, the technical requirements for unmanned vehicles are highly demanded. Unmanned vehicle obstacle avoidance technology is an important part of many related technologies. The detection and avoidance of surrounding obstacles is to be achieved by unmanned vehicle in safety condition. How to get environmental information and plan the route is very important. In this paper, we design and develop a laser-based avoidance system for autonomous guided vehicle materialized in this work by two-wheeled robot. This system can quickly collect obstacle distance information and effectively avoid the obstacles and determine the new direction and a new path after information processing on the level of computation platform. The local processing on the edge platform expressed by a Raspberry Pi is followed by cloud processing. Experimental results are presented that validate the capability of the system. on navigation based on 2D LIDAR.**

Keywords—LIDAR, obstacle avoidance, Autonomous Robot

I. INTRODUCTION

In recent years, new generations of unmanned vehicles such as AGVs, unmanned vehicles have gradually replaced inefficient container trailers as the main tool for container handling in large international ports that improves the efficiency of container operations. Due to the complex port environment, the technical requirements for unmanned vehicles are highly demanded. The obstacle avoidance technologies are developing every day to realize the safety operation of unmanned vehicles. Obstacle avoidance is a very important function of an unmanned vehicle. In this article, are discussed how robots can avoid obstacles distributed in their operation environment.

In order to acquire information of the surroundings and build a 2D/3D model of the environment, it is used accurate optical measuring instrument, such as a LIDAR. LIDAR is an optical measurement instrument with high that operates free from electromagnetic waves and light interference. The LIDAR permits wide measurement range, and it is a common solution for mobile robots. The basic working principle of LIDAR is to measure the time interval from the emission of the laser beam to the reception of the reflected light from the surface of the obstacle (target object). Raspberry Pi is used to control through a step motor the LIDAR orientation to avoid obstacles autonomously. The Raspberry Pi is a microcomputer that can be used in conjunction with a variety of sensors is part of Internet of Things implemented architecture.

The goal of the work is to develop a system for indoor/outdoor distance measurement to obstacles and surrounding environment modelling for unmanned vehicle. This can improve the operational safety and to reduce collision rate of unmanned vehicles. A study obstacle avoidance strategies and robot motion control in multiple scenarios it is also considered.

II. RELATED WORK

In order to solve this navigation and obstacle avoidance related problems, a variety of sensors are applied. In [1], the author developed an alternative position estimation method and implementation of LIDAR sensor and Raspberry PI for 2D space mapping. In [2], they have developed a new emergency obstacle avoidance module for moving robots that uses Laser Imaging Detection and Ranging (LIDAR) to detect static and moving obstacles. They developed a LIDAR rotation platform and detects low height objects. In [3,4] they describe the performance evaluation of obstacle detection and segmentation algorithms for automatic guided vehicle (AGV) navigation using a 3D real-time ranging camera. In [3] the author verified measurements obtained using a partially accurate 2D scanning laser range finder. In [4] they use PMD camera to measure the obstacle, Pixel-Mixed-Device Technology (PMD) offers a small, light-weight camera generating 3-D images, based on time-of-flight The particular characteristics of the sensor characteristics and application potential for mobile robots. In [5], This paper describes a navigation and control system that uses a differentially driven magnetic point to guide the AGV. In addition, Hall effect sensors, encoders and counters are used for control and continuous guidance. The specific implementation of the magnetic spot navigation method in Shanghai Yangshan Port is introduced in this article[6]. As an inexpensive and simple distance sensor, ultrasonic sensors are widely used in unmanned vehicles. In [7,8] In these two papers, the fuzzy logic obstacle avoidance method based on ultrasonic sensor is introduced. In [7] the author introduces an approximate fuzzy inference method based on KH interpolation in the fuzzy environment of fuzzy rule base. This method can be realized as practical direct fuzzy logic. Control the application. In [9] are described algorithms applied on single 2D laser imaging detection and ranging (LIDAR) sensor to perform autonomous 3D reconstruction of the environment and a robotic operating system (ROS) to implement it on a robot. This paper [10] describes an algorithm that performs an autonomous 3D reconstruction of an environment with a single 2D Laser Imaging Detection and Ranging (LIDAR) which can create a 3D model of the robot's surroundings without prior information or human intervention. In [11], the authors combine LIDAR with the Mecanum drive system. The Mecanum wheel [18] is a design for a wheel that can move a vehicle in any direction. Set the robot to protrude and protrude from the wall along the wall (doors, posts, and other features). The robot scans at an angle to clear the object, pass the object, and then scan the wall to restore its original path. In this paper[12], an efficient obstacle detection and obstacle avoidance algorithm based on two-dimensional lidar is proposed. The algorithm provides obstacle information by filtering and clustering the laser point cloud data. Also, this method generates the forward angle and velocity of robot based on the principle of minimum cost function. In [13] they investigated the trajectory tracking control of a wheeled mobile robot using a laser rangefinder sensor to estimate the robot position and orientation. In [14], the authors completed a real-time automated robotic system for monitoring, enabling simultaneous robot positioning, mapping and navigation with virtually no human intervention. In [15] the authors present an inexpensive 2D LIDAR system using LIDAR-Lite v1 for obstacle detection in self-driving vehicles with a scan angle of 360 degrees. The acquired data is filtered by median and detected by obstacle clustering of point cloud data. A point cloud [19] is a set of data points in space. Point clouds are generally produced by 3D scanners, which measure many points on the external surfaces of objects around them. The results show that near obstacles can be detected at close range. In [16], aiming at the expensive solution for simultaneous positioning and mapping (SLAM) of mobile robots. The authors designed a low-cost mobile robot solution, which uses laser imaging to acquire 2D laser scanning matching data and SLAM's open source GMapping software package, using RVIZ (ROS visualization tool). Realizing the indoor mapping it can build high-precision maps. In [17] the author uses an autonomous mobile robot with a robotic operating system (ROS). The system uses 2D LiDAR and RGB-D cameras with a ROS 2D navigation stack and is implemented on the Raspberry Pi 3 and intel NUC respectively. Comparative usability testing was performed on two systems in multiple experiments. The results show that the robot can avoid objects in the path or stop in the unavoidable situation.

III. System Description

The obstacle avoidance system includes an autonomous mobile robot. The prototype acrylic plate as the chassis of the robot and two-wheel differential structure. The front half has a drive caster with support that combines lidar and Raspberry Pi, controllers and power modules. The main components of the implemented prototype are described in detail.

A. *Localizaton sensor - Lidar*

The localization of obstacles is performed using LIDAR-lite v3, that is a compact optical distance measurement sensor from Garmin. This device measures the distance to object by calculating the time delay between the transmission of a Near-infrared laser signal and its reception after reflecting from the target. Effective range is 40 meters, the measurement accuracy within 5 meters is 2.5 cm. The LIDAR transmits data to the Raspberry Pi through the I2C (Inter-Integrated Circuit) communication protocol and operates at 5V DC, which means it can work well with raspberry pi computation platform.

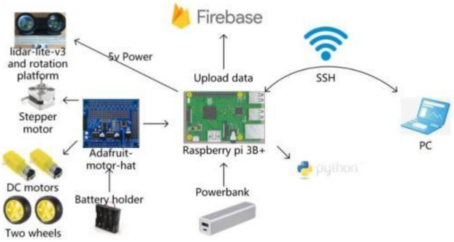
A two-dimensional rotating platform for LIDAR was designed that allows LIDAR to measure obstacle position in multiple directions. The rotating platform consists of a stepper motor and a link plate. The stepper motor is controlled by the Raspberry Pi, and its rotation frequency and rotation mode can be set by code.

B. *Computation platform - Raspberry Pi*

To operate the robot, a Raspberry Pi 3B+ was used. The Raspberry Pi is a series of small single-board computers, it can read multiple sensor data at the same time. With a network interface, users can remotely control raspberry using Secure Shell (SSH) however it is also considered the autonomous navigation algorithms.

C. *Actuators - Motors and controller*

In this robot, a set of two DC motors and an Adafruit-motor-hat as a controller were used. This Adafruit-motor-hat can control two DC motors and one stepper motor simultaneously. The power supply is based on 6V batteries and a power bank for the raspberry. A block diagram of the hardware connections is presented in Figure 1, where the fully assembled robot is presented in Figure 2.



1. Fig. 1. Hardware connections diagram.

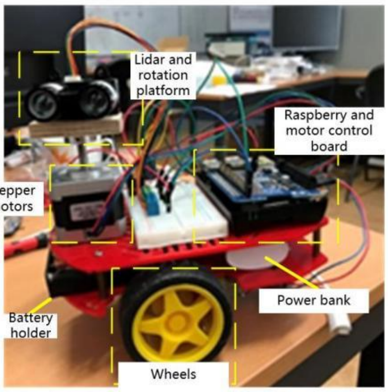

Lidar and rotation platform
Raspberry and motor control board
Stepper motors
Power bank
Battery holder
Wheels

Fig. 2. The implemented Robot prototype

D. *Firebase real-time database*

Firebase is a mobile and web application development platform developed by Firebase, Inc[20]. The firebase database was used to store the distance to obstacle data. Connect the Raspberry Pi on the robot to the Internet via WI-FI. two properties for each piece of data: the measurement angle and the distance value. When a measurement cycle ends, the distance values for all angles are uploaded.

IV. Obstacle avoidance method

In this chapter, the kinematics model of the robot is introduced to illustrate how it moves forward and turns. How the robot moves next is based on the distance measured by the lidar at different angles. The obstacle avoidance method outputs a DC motor command based on these distance values.

A. *Robot kinematics model*

The motion control of the robot is closely related to the motion model of the robot itself. The kinematics model of the robot is introduced to illustrate how it moves forward and turns. The robot used in the study is a two-wheeled robot with two drive wheels and a universal wheel. The following figure shows the pose of the mobile robot at two adjacent moments, θ_1 is the angle at which the mobile robot moves around the arc at two adjacent moments, and θ_3 is the amount of change in the heading angle (toward the head head) of the moving machine at two adjacent moments. L is the distance between the left and right wheels, which is the distance between the right wheel and the left wheel. d is the distance that the right wheel travels more than the left wheel. R is the radius of the circular motion of the mobile robot.

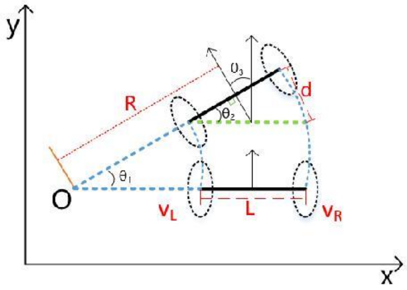


Fig.3. The kinematic model of two wheels robot.

The moving speed of the mobile robot v is equal to the average of the left and right wheel speeds :

$$v = \frac{v_R + v_L}{2} \tag{1}$$

How to calculate the robot's heading angle and how to calculate the angular velocity. As shown in the figure, by superimposing the positions of the robots at two moments, it is clear that the amount of change in the heading angle of the mobile robot is. From the geometric relationship in the figure we can get:

$$\theta_1 = \theta_2 = \theta_3 \tag{2}$$

When the robot makes a circular motion, it starts from the starting point and returns to the starting point around the center of the circle. In this process, the cumulative heading angle of the robot is 360 degrees, and it also moves 360 degrees around the center of the trajectory, indicating the angle of change of the heading angle of the robot equal to the angle at which it rotates around the center of the motion trajectory. Among these three angles, θ_2 is easy to calculate. Since the robot is in continuous motion, the time interval between two adjacent moments is very short, and the angle change is small, so there is the following approximate formula:

$$\theta_2 \approx \sin(\theta) = \frac{d}{L} = \frac{(v_R - v_L)\Delta t}{L} \tag{3}$$

The angular velocity of the robot around the center of the circle can be calculate, which is also the speed of the robot's heading angle:

$$\omega = \frac{\theta_1}{\Delta t} = \frac{v_R - v_L}{L} \tag{4}$$

Therefore, the radius of the circular motion of the mobile robot can be introduced.

$$R = \frac{v}{\omega} = \frac{L(v_R + v_L)}{2(v_R - v_L)} \tag{5}$$

B. *Safety operation distance*

The settings for safe distance for UAV is performed as following. First the array to store the data of each LIDAR measurement cycle was considered.

After the data is sorted, the largest and smallest data are marked as the maximum distance and the minimum distance. The next action of the unmanned robot depends on the values of the maximum distance and the minimum distance. If the maximum distance is less than 35 cm, the robot will rotate 180 degrees. When the maximum distance is greater than 35 cm and the minimum distance is less than 100 cm, the robot turns to the direction of the maximum distance to avoid obstacles.

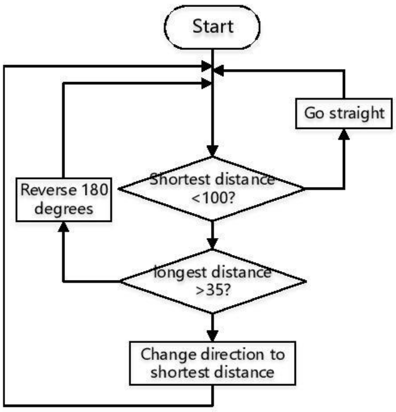


Fig. 4. Safety distance block diagram.

The data is sorted, and the largest and smallest data is marked as the maximum distance and the minimum distance. Whether the unmanned robot is in safe operating mode depends on the maximum distance and the minimum distance. When the maximum distance is less than 35 cm, the unmanned robot will rotate 180 degrees. When the maximum distance is greater than 35 cm and the minimum distance is less than 100

Identify applicable funding agency here. If none, delete this text box.

cm, the unmanned robot can turn to the direction of the maximum distance to avoid obstacles.

C. Stepper motor rotation mode

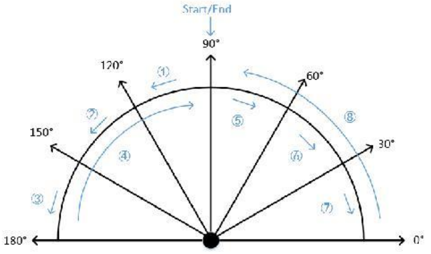
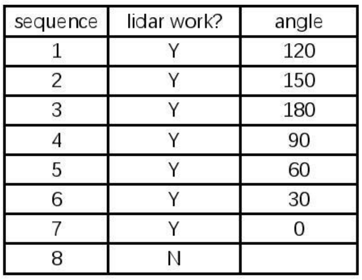


Fig. 5. Stepper motor rotation mode.

The LIDAR and stepper motor are connected by a rotating platform to form a two-dimensional distance measuring system. When the stepper motor rotates, the lidar will also rotate with the stepper motor. In this system, how to make the lidar measure the obstacles in the 180 range in front of the robot is an important issue. First I set the rotation mode of the stepper motor. The rotation mode can be seen in Figure 3. The starting and ending positions of the rotation are the center of the robot's orientation. In one cycle, the stepper motor rotates a total of 8 times, mark the number of steps of the stepper motor as 1 to 8 and when the cycle is over, the start position coincides with the end position. According to the stepper motor driver`s function, The stepper motor is characterized by 300 steps/rev, which means the stepper motor rotates 300 times in 360 degree, every step rotates 1.2 degree. The LIDAR work sequence can be seen in Figure 4. The settings of the stepper motor rotation sequence were1-3, 4-7 rotates 30 degree, equal to 25 steps, sequence 4 and 8 rotates 90 degree, equal to 75 steps. When the 1-7 step is over, the lidar will work.

Table. 1. Lidar work sequence.



sequence	lidar work?	angle
1	Y	120
2	Y	150
3	Y	180
4	Y	90
5	Y	60
6	Y	30
7	Y	0
8	N	

V. EXPERIMENTAL RESULTS ABOUT DISTANCE AND OBSTACLE AVOIDANCE

First, considering the time required for each measurement cycle and the measurement accuracy, I tested the pause time after each rotation of the rotating platform. The length of the pause will affect the travel efficiency of the car and the measurement accuracy of the LIDAR.

In order to test this pause time, I set up a scene: set the car to stand still, set different pause time, measure multiple cycles, compare the accuracy of the measurement results with the measurement time. Measuring distance values at different angles using an angle ruler and a laser range finder. The distance value of each angle has shown in Table 1. The distance measured by the laser range finder was accurate as a benchmark for monitoring comparisons.

Table 2. Distance measured by laser range finder and angle ruler.

angle	Distance(cm)
0	110
30	70
60	45
90	39
120	45
150	32
180	30

The distances in different directions were measured in sequence at different pulse times and the result can be seen at Table 2

Table 3. Distance values of different pulse time.

Pulse time(s) Angle	1.5	1.25	1	0.75	0.5	0.25	0.1	0.05
0	90	87	92	91	85	100	67	101
30	4	69	69	72	68	55	56	80
60	45	47	43	47	44	46	38	42
90	40	41	39	37	38	50	34	32
120	44	45	44	47	38	55	56	34
150	33	31	34	32	38	29	29	35
180	31	31	32	29	30	34	21	42

Different measurement results corresponds to different pause times. Considering the two factors of measurement accuracy and pause time, it is best to set the pause time to 0.5 seconds. The comparison of measurement results is shown in Figure 7.

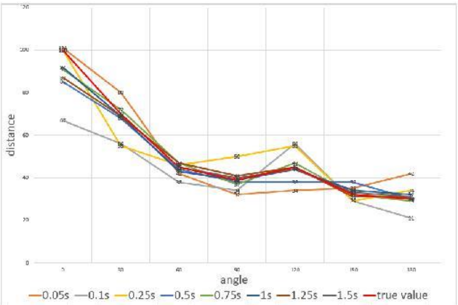


Fig. 7. Distance measurement of different pulse time accuracy comparison chart.

In order to test the obstacle avoidance performance of the robot, various test scenarios were established. First, how the lidar detects obstacles and the specific error comparisons are presented.

Table 4. Measured value of 7 direction(0 ° - 180 °)

position	P1(0 °)	P2(30 °)	P3(60 °)	P4(90 °)	P5(120 °)	P6(150 °)	P7(180 °)
true value	100	63	36.4	31.5	36.7	52.6	45.5
measured value	100	61	38	30	36	54	45

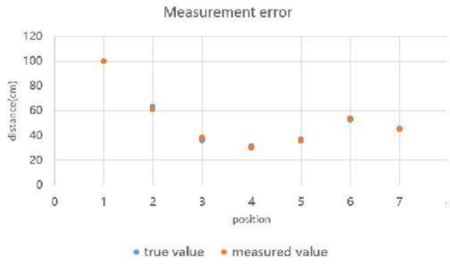


Fig. 8. Measurement error

According to the distance measured by lidar in 7 directions, the approximate shape of the obstacle can be drawn, the robot as the center of the surrounding environment can be established.

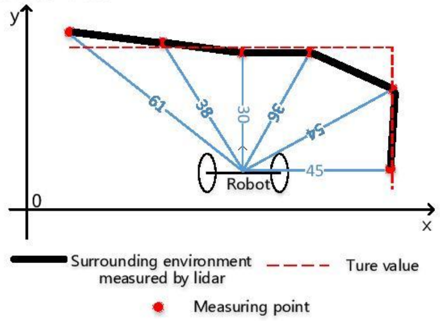
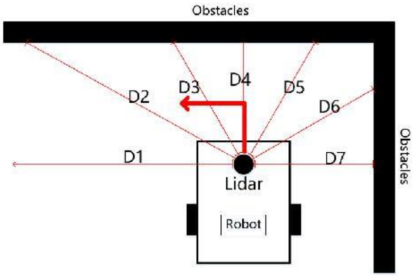


Fig. 9. Surrounding environment established by measurement value

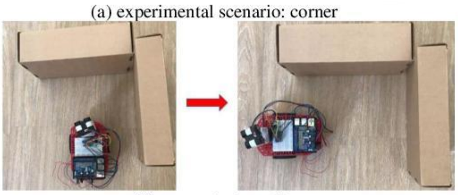
Then, three scenarios were established: (a) corner; (b) obstacles at various angles; (c) dead ends. In (a), D1 is the maximum value and is greater than the safety distance determination value, and the remaining D2-D7 are smaller than the safety distance determination value. At this time, the robot turns 90 degrees to the left.

In (b), D3 and D4 are both greater than the safety distance determination value, D1-D2, D5-D7 are smaller than the safety determination distance, and the robot will turn to the direction of D3 because D3 is larger than D4.

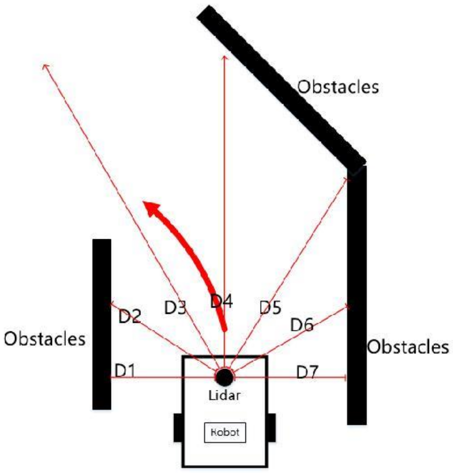
In (c), D1-D7 are both smaller than the safety distance determination value, and the robot will reverse the direction by 180 degrees in this case.



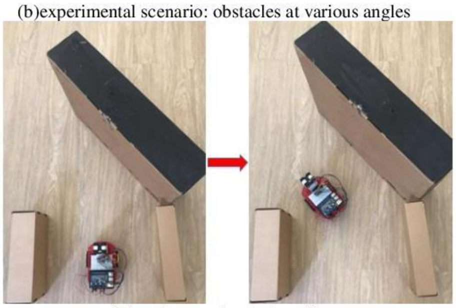
(a) experimental scenario: corner



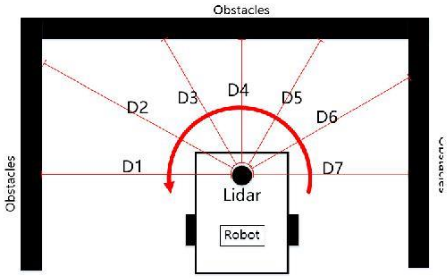
(a) corner test results



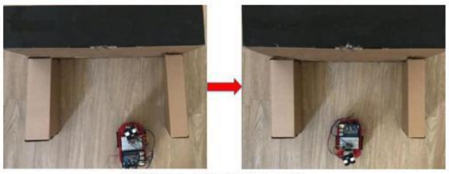
(b)experimental scenario: obstacles at various angles



(b) various angles test result



(c) dead ends



(c) dead ends test results

Fig. 10. (a)(b)(c)Avoiding obstacles experimental scenario.

VI. Conclusion and Future Work

This work uses a LIDAR and Raspberry Pi to build a small robot with obstacle avoidance capability. It is designed to improve the accuracy and safety of robot during autonomous operations. The robot can obtain the distance information of the surrounding environment, and the information can specify the obstacle avoidance strategy and upload it to the real-time database. Some obstacle avoiding scenarios are set and tested, In the next work, the improvement of the obstacle avoidance algorithm will be carried out, and the ability of the algorithm to respond to various scenarios will be improved.

Acknowledgement

This research is supported by Instituto de Telecomunicações (IT-IUL) at ISCTE-IUL, Lisbon, Portugal and Fundação para Ciência e Tecnologia (FCT). This work is also supported by the Shanghai Science and Technology Commission (No. 19595810700, 18295801100, 17595810300) and Yunnan Science and Technology Program (No.2018IB022)

References

[1] Lidar: range-resolved optical remote sensing of the atmosphere[M]. Springer Science & Business, 2006.

[2] Vanicek P, Beran L. Navigation of robotics platform in unknown spaces using LIDAR, Raspberry PI and hector slam[J]. Journal of Fundamental and Applied Sciences, 2018, 10(3S): 494-506.

[3] Bostelman R V, Hong T H, Madhavan R. Towards AGV safety and navigation advancement obstacle detection using a TOF range camera[C]//ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005. IEEE, 2005: 460-467.

[4] Wiedemann M, Sauer M, Driewer F, et al. Analysis and characterization of the PMD camera for application in mobile robotics[J]. IFAC Proceedings Volumes, 2008, 41(2): 13689-13694.

[5] Lee S Y, Yang H W. Navigation of automated guided vehicles using magnet spot guidance method[J]. Robotics and Computer-Integrated Manufacturing, 2012, 28(3): 425-436.

[6] Yue-Quan W . Method of AGV transponder installation in automated terminal of Yangshan Deepwater Port[J]. Port & Waterway Engineering, 2018.

[7] Dong-he Y, Xi-ang L I U. Research of intelligent mobile robot's obstacle avoidance based on ultrasonic sensor[J]. Computer Engineering and Design, 2007, 28(15): 3659-3661.

[8] Kovàcs S, Kóczy L T. Interpolation-based fuzzy logic controller, as a simplified way for constructing the fuzzy rulebase of the path tracking and collision avoidance strategy of an AGV[C]//SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218). IEEE, 1998, 2: 1317-1322.

[9] Takahashi M , Kobayashi K , Watanabe K , et al. Development of prediction based emergency obstacle avoidance module by using LIDAR for mobile robot[C]// International Symposium on Soft Computing & Intelligent Systems. IEEE, 2015.

[10] Ocando M G, Certad N, Alvarado S, et al. Autonomous 2D SLAM and 3D mapping of an environment using a single 2D LIDAR and ROS[C]//2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR). IEEE, 2017: 1-6.

[11] Padgett S T , Browne A F . Vector-based robot obstacle avoidance using LIDAR and mecanum drive[C]// SoutheastCon 2017. IEEE, 2017.

[12] Peng Y, Qu D, Zhong Y, et al. The obstacle detection and obstacle avoidance algorithm based on 2-d lidar[C]//2015 IEEE International Conference on Information and Automation. IEEE, 2015: 1648-1653.

[13] Lima T A, do Nascimento Forte M D, Nogueira F G, et al. Trajectory tracking control of a mobile robot using lidar sensor for position and orientation estimation[C]//2016 12th IEEE International Conference on Industry Applications (INDUSCON). IEEE, 2016: 1-6.

[14] Asadi K, Ramshankar H, Pullagurla H, et al. Building an integrated mobile robotic system for real-time applications in construction[J]. arXiv preprint arXiv:1803.01745, 2018.

[15] Catapang A N, Ramos M. Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle[C]//2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE). IEEE, 2016: 441-445.

[16] Shen D, Huang Y, Wang Y, et al. Research and Implementation of SLAM Based on LIDAR for Four-Wheeled Mobile Robot[C]//2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE). IEEE, 2018: 19-23.

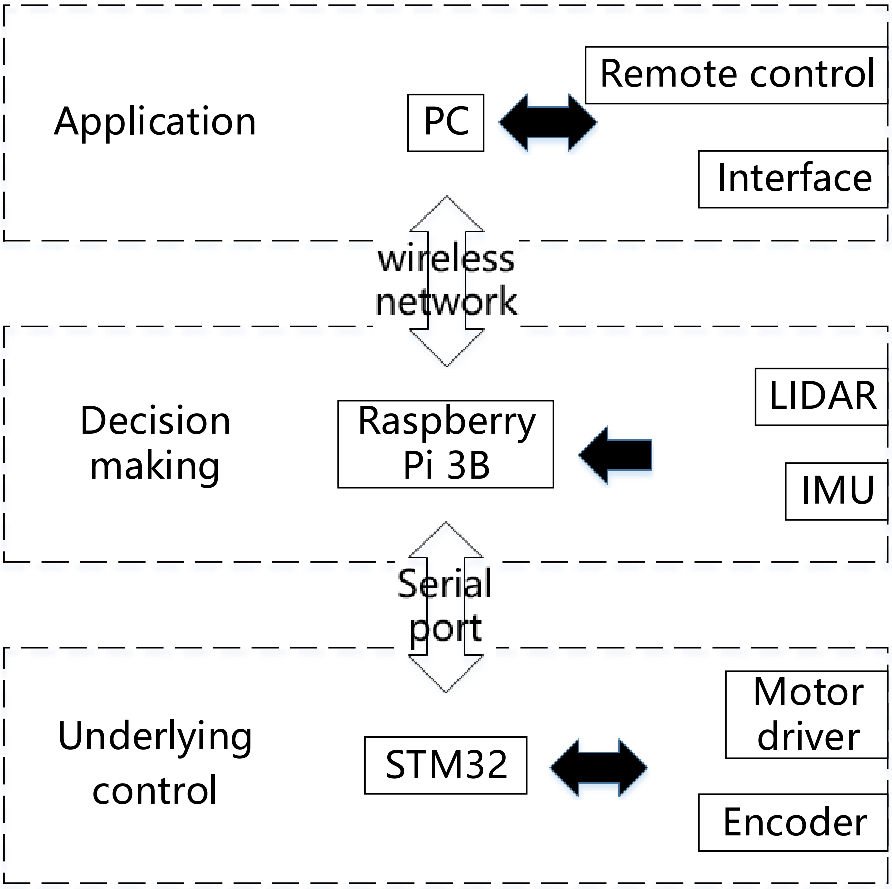
[17] Gatesichapakorn S, Takamatsu J, Ruchanurucks M. ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera[C]//2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP). IEEE, 2019: 151-154.

[18] https://en.wikipedia.org/wiki/Mecanum_wheel

[19] https://en.wikipedia.org/wiki/Point_cloud

[20] https://en.wikipedia.org/wiki/Firebase

Appendix B - The structure of developed ROS system



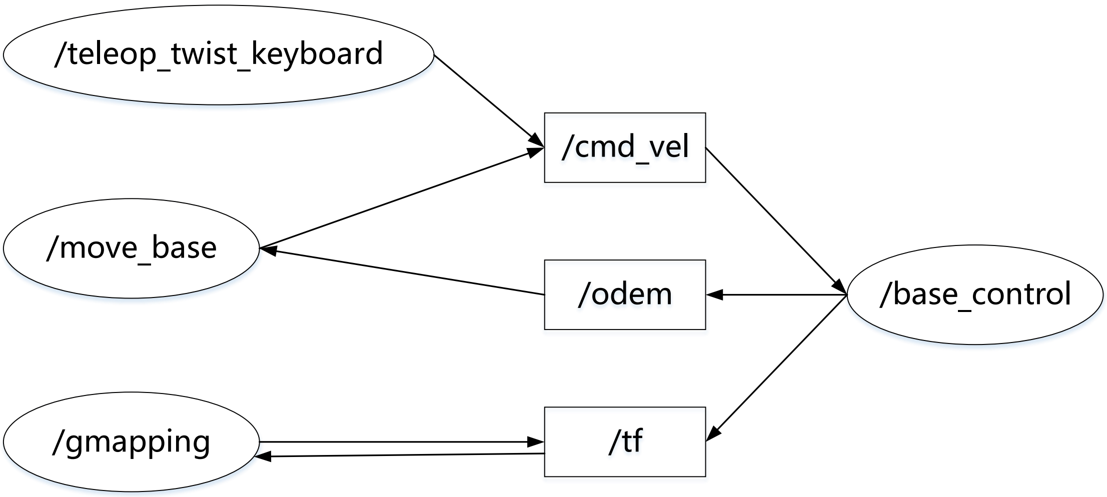
The software part of the robot is mainly the construction of nodes in ROS and the setting of parameters. The software system design of the robot platform needs to combine the distributed processing characteristics of the ROS system to achieve certain modularization and network design, and has certain versatility and portability. The main part of the software consists of the navigation framework of the mobile robot and the human-computer interaction system, including (map construction and real-time positioning), path planning, obstacle avoidance and remote control, etc.

ROS's framework is a processing architecture that uses ROS communication modules to implement loosely coupled network connections between modules. It performs several types of communication, including service-based synchronous RPC (remote procedure call) communication, Topic-based asynchronous Data stream communication, as well as data storage on the parameter server.

The nodes established in the ROS system of the robot have**:**

1) /base_control

In the ROS control model, we drive the ROS chassis, the proportional–integral–derivative(PID) controller adjustment for each wheel, the chassis return speed and the odom data processing module are collectively called base_controller. base_control is a Node. This node processes and executes the cmd_vel data and reports it to the /odom topic.

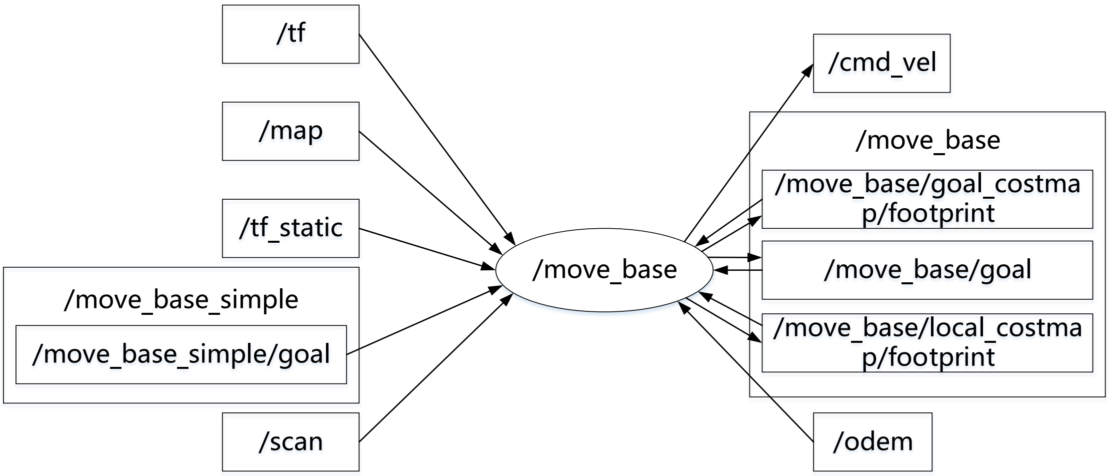


/base_control node diagram

2) /move_base

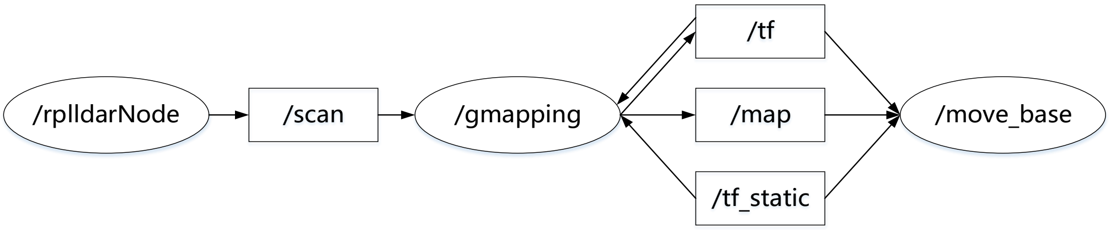
The move_base package provides an implementation of an action, given a goal in the world, will attempt to reach it with a mobile base. The move_base node links together a global and local planner to accomplish its global navigation task.

It supports any global planner adhering to the nav_core::BaseGlobalPlanner interface specified in the nav_core package and any local planner adhering to the nav_core::BaseLocalPlanner interface specified in the nav_core package. The move_base node also maintains two costmaps, one for the global planner, and one for a local planner that are used to accomplish navigation tasks.



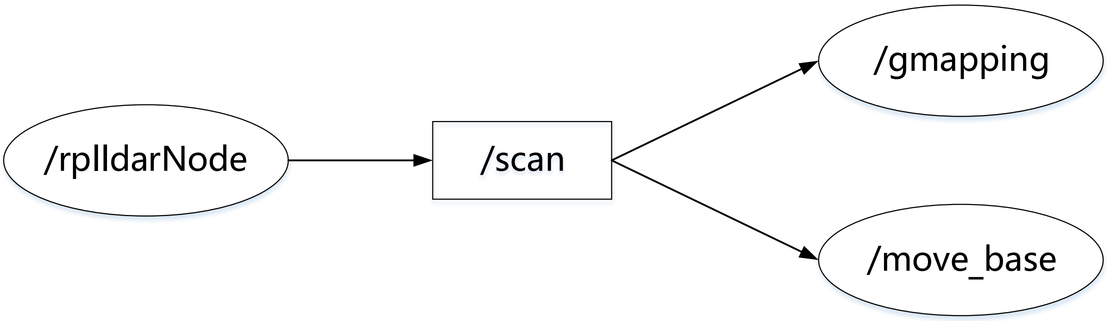
3) /gmapping

The gmapping package provides laser-based SLAM (Simultaneous Localization and Mapping), as a ROS node called /gmapping. Using /gmapping, can create a 2-D occupancy grid map (like a building floorplan) from laser and pose data collected by a mobile robot.



4) /rplldarNode

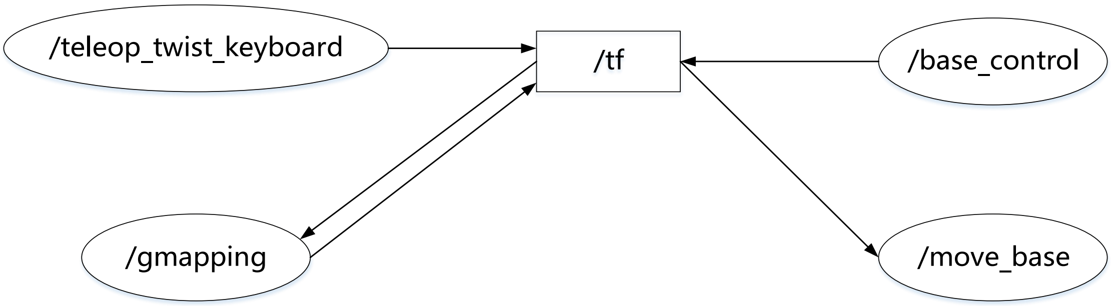
/rplldarNode is a official ROS package provided by RPLIDAR and provides basic device processing for 2D laser scanners RPLIDAR A1 / A2 and A3. rplldarNode is a driver for RPLIDAR. It reads RPLIDAR raw scan result using RPLIDAR's SDK and convert to ROS LaserScan message.



The topics established in the ROS system of the robot have:

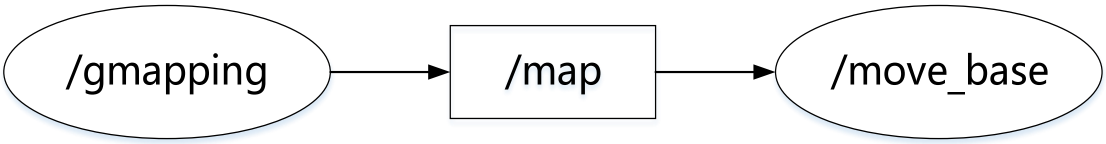
1) /tf

/tf is a package that lets the user keep track of multiple coordinate frames over time. tf maintains the relationship between coordinate frames in a tree structure buffered in time, and lets the user transform points, vectors, etc between any two coordinate frames at any desired point in time.



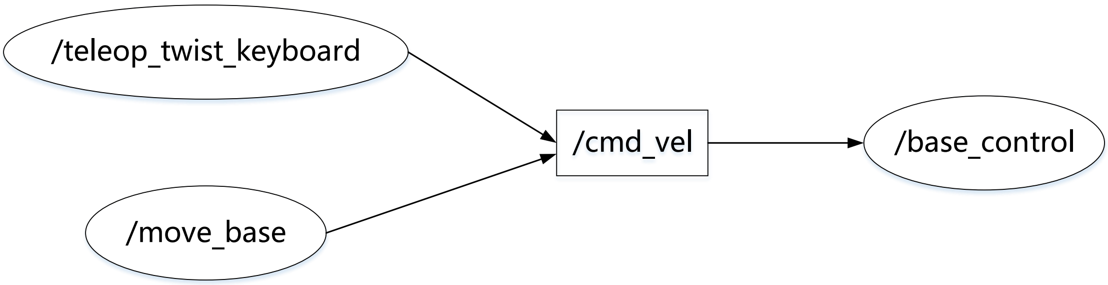
2) /map

The /gmapping node publishes the /map topic, contains the /OccupancyGrid information, which is a two-dimensional grid map, and the /move_base node subscribes to the /map topic from the /gmapping node to get this information.



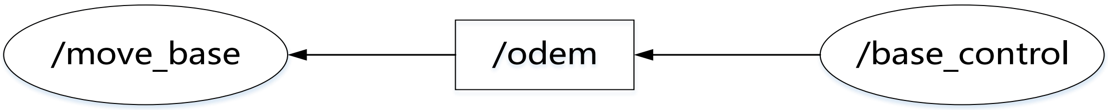
3)/cmd_vel

The /teleop_twist_keyboard node and the /move_base node change and provide the line speed and angular velocity of the car by publishing the /cmd_vel topic. /base_control controls the car by subscribing to the topic by converting the angular velocity and line speed of the car to the line speed of the car wheel.



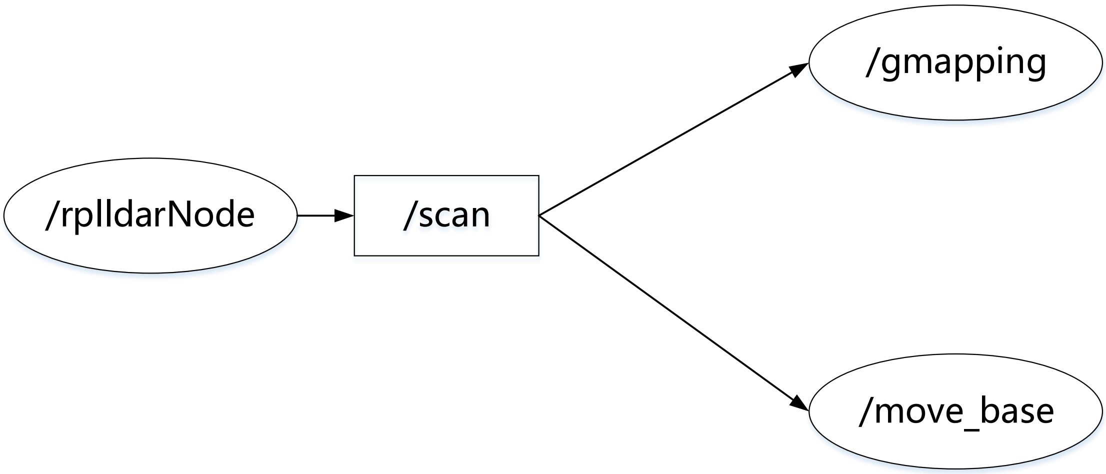
4)/odem

The /base_control node publishes the /odem topic to provide an estimate of the position and speed of the robot in free space. /move_base obtains the odem information by subscribing to the topic to determine the robot's position in the world and associates the sensor data with the static map.



5)/scan

The /rplldarNode node publishes the /scan topic to provide LIDAR scan data, and the /move_base node and the /gmapping node subscribe to the topic to obtain LIDAR data.



In addition to these node topics, there is also a rviz tool. The rviz is a visualization tool for ROS to visualize sensor data and status information. Rviz supports lots of data types, visualized by loading different Display types, and each Display has a unique name.