Mobile Robots in Laboratory Automation – A Contribution to Total Automation

Dissertation

for

obtaining of the academic title

Doktor-Ingenieur (Dr.-Ing.)

Faculty of Computer Science and Electrical Engineering
University of Rostock

Submitted by:

Jiahao Huang, born on 26, September 1994 in Xiangtan, China

Rostock, Germany, 2024

https://doi.org/10.18453/rosdok_id00005268

Reviewers:

1. Reviewer:

   Prof. Dr. -Ing. habil. Kerstin Thurow

   Institute of Automation, University of Rostock, Germany

2. Reviewer:

   Prof. Dr. -Ing. habil. Hui Liu

   School of Traffic and Transportation Engineering, Central South University, China

3. Reviewer:

   Prof. John T. Wen,

   Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, USA

Date of Submission: 2025

Date of Defense: 2025

Acknowledgment

Three and a half years of doctoral studies have passed in a blink of an eye. Although I am an introvert who usually keeps to myself, I am deeply grateful to everyone who has helped me. Even though I plan to return to China after my graduation, I will greatly miss the life in Rostock.

Firstly, I must express my gratitude to my supervisor, Prof. Dr.-Ing. habil. Kerstin Thurow. Under her diligent supervision, I was able to successfully complete my PhD. She is an outstanding professor who, aside from her hard work, took the time to carefully revise the papers I submitted, providing direction and suggestions for my research. She grounded me in practical work and gave me valuable project research experience. She is a role model in my research career, from whom I learned the German virtues of meticulousness and practicality as an engineer, which will greatly benefit my future career. She also provided financial support for my PhD extension.

Furthermore, I want to thank my group leader, Dr.-Ing. Steffen Junginger. He is a very warm person, whose conversations always reduce my stress, offering care and encouragement like a spring breeze. He has been a significant support in both my work and personal life. Also, Prof. Dr.-Ing. habil. Hui Liu, who guided my scientific career. Without him, I would not have the opportunity come to CELISCA to pursue my doctorate or received funding from the China Scholarship Council. Despite being in China, he frequently cared about my studies and life, offering advice. Dr.-Ing. Thomas Roddelkopf, the lab's programming expert, always answered my difficult programming questions and provided the help I needed.

Moreover, I am grateful to have met such warm and friendly all my colleagues at CELISCA. It is my honor to work with them. In particular, M.Sc. Mohamed Ali Tlili, M.Sc. Simon-Johannes Burgdorf, Prof. Dr. Mohammed Faeik Ruzaij Al-Okby, Mr. Heiko Engelhardt, M.Sc. Anna Bach and Prof. Dr.-Ing. habil. Heidi Fleischer.

I am also thankful for the friends I met in Germany who encouraged me. We cooked, drank, chatted, and traveled together. Thank you for helping me solve various problems, especially since I cannot speak German— Mr. Bingxu Ren, Mr. Yuchen Xue, Mr. Chien Liu Mr. Disheng Su and Dr.-Ing Xiangyu Gu.

Special thanks to the China Scholarship Council and CELISCA for their financial support during my study.

Lastly, I express my deep appreciation to my family for their silent support behind the scenes, shielding me from troubles and encouraging me. And my fiancée, Caini Ouyang, for her willingness to wait for me in China, and for keeping up with our cross-time-zone chats every day.

Content

List of Figures

List of Tables

List of Abbreviations

CELISCA	Center for Life Science Automation
IoT	Internet of Things
AGV	Automated Guided Vehicle
AIC	AGV Interface Controller
MFC	Microsoft Foundation Classes
AMR	Autonomous Mobile Robot
HRI	Human-Robot Interaction
IMU	Inertial Measurement Units
RF	Radio Frequency
RSSI	Received Signal Strength Indication
UKF	Unscented Kalman Filter
DWA	Dynamic Window Approach
WMS	Workflow Management Systems
PCS	Process Control Systems
ICS	Instrument Control Systems
FMS	Fleet Management System
IP	Internet Protocol
TCP	Transport Control Protocol
UDP	User Datagram Protocol
AMCL	Adaptive Monte Carlo Localization
MTP	Microtiter Plate Format
GUI	Graphical User Interface

1 Chapter 1 Introduction

1.1 Background

With the start of a new generation of global industrial revolution and the push by Germany's "Industry 4.0" [1] and the United States' "Industrial Internet," mobile robot technology in smart environments has seen rapid development. In particular, major advances in sensor technology and information control technology have greatly improved automation levels in various fields. These technological advancements have not only led to a wave of changes in the global manufacturing industry but have also enabled the practical application and wide interest in mobile robots in many areas, including industry, laboratories, warehousing and logistics, transportation, shopping, and healthcare [2].

Automation in laboratories has emerged as a critical factor for increasing efficiency, reducing human error, saving energy and improving the overall quality of research. With the continuous advancements in robotics and automation technologies, laboratories are increasingly adopting innovative solutions to streamline their workflows and optimize resource utilization [3]. One area where automation has a significant potential is the transportation of labware and samples between different stations within a laboratory, particularly in multi-floor facilities. The integration of AGVs for labware transportation offers numerous advantages, such as reduced human intervention, decreased risk of contamination, and increased overall productivity [4]. Recently, with the innovation of hardware technology and algorithms, mobile robots have evolved to include advanced features, such as enhanced environmental perception capabilities, spatial definition abilities [5], interaction capabilities with doors/elevators/human [6], and multi-robot collaboration [7]. This further enhances their efficiency in complex laboratory environments.

Center for Life Science Automation (CELISCA), as the central scientific research center of the University of Rostock, is dedicated to its work towards the Future-Lab [8]. CELISCA laboratory conducts various medical experiments and testing, requiring the handling of medical materials and labware across different rooms and floors. The laboratory aims to reduce repetitive tasks for laboratory personnel and achieve 24/7 fully automated processes. Due to the multi-floor layout of the laboratory and the distribution of different experimental materials, equipment, and workbenches in various rooms, using mobile robots as transport vehicles to connect laboratory islands becomes indispensable. The laboratory includes numerous devices, elevators and automatic doors integrated with the Internet of Things (IoT) technology. The development of mobile robots aligns with CELISCA's research needs and work environment.

Compared to traditional stationary industrial robots, mobile robots have the advantage of not being limited by their working range. They can move on their own through controllers and, without human operation, rely on software systems and sensor-based navigation systems to perform tasks along predefined paths in complex environments, bringing new possibilities to laboratory automation.

1.2 Autonomous Mobile Robot

The development history of robotics has exceeded sixty years. It has seen comprehensive progress of computer science, algorithm development, computer hardware, sensors, automation control and mechanical design. The mutual promotion of these fields not only promotes the development of industrial automation, but the demand for automation is also constantly increasing [9]. These factors are promoting the improvement of mobile robots' environmental perception and mobile autonomy, transforming Automated Guided Vehicle (AGV) into AMR. AMR are capable of independently exploring their surroundings and generating detailed 2D or 3D maps of their environment. They can determine the most efficient route to their destination and adeptly navigate around any temporary or moving obstacles they encounter along the way. Moreover, these robots can interact with humans, providing assistance in various tasks, and have reached a level of understanding that allows them to recognize and interpret their environment on a semantic level [10].

Navigation is the most important technical area of mobile robots. With the rise of AMR, current navigation technology integrates various sensors for fusion. AMR can integrate multiple sensors such as LiDAR, depth cameras, and ultrasonic sensor, etc., to achieve in-depth perception of the environment [11]. First, LIDAR and depth camera provide the robot with high-precision environmental scanning capabilities, allowing it to generate detailed maps of the surrounding environment. Secondly, depth camera and LIDAR data can also be fused with other data such as IMU, allowing the robot to better understand the depth information and visual features of its environment. Data from these sensors are integrated through advanced fusion algorithms to ensure the robot has a comprehensive perception of its environment. In the established map, the mobile robot can also accurately position and avoid obstacles based on these two sensors during movement [12].

This perception capability places high demands on data processing algorithms and must be combined with powerful built-in computing resources. The processing performance of onboard computers is sometimes insufficient, but cloud computing services based on the Internet have lifted the limitations of computing power [13]. The deep perception and cloud computing give AMR higher autonomous exploration and decision-making

capabilities, allowing it to flexibly navigate dynamic and complex environments, whether in a laboratory environment where humans and machines coexist or in automated industrial settings. This progress not only marks the transition from basic navigation to advanced intelligent navigation, but also demonstrates AMR's human-robot collaboration capabilities in complex scenarios, which is similar to the application of driverless technology [14].

In addition, in the specific environment of the laboratory, AMRs' autonomous navigation capability can more effectively realize accurate navigation of high-value equipment in small spaces. They can safely interact with people and other robots while ensuring the accurate completion of tasks. This all benefits from AMRs' advanced capabilities in environmental awareness, dynamic obstacle avoidance, and communication with building infrastructure such as automatic doors and elevators. These capabilities make AMR a key technology for improving laboratory automation.

1.3 Challenges

In the pursuit of transforming laboratories into Future-Lab, integrating mobile robots is a vital step towards enhancing automation. Utilizing mobile robots as connectors between various research stations across different rooms and floors in life science laboratories introduces complexity and the following challenges.

The first thing that needs to be considered is safety. Security aspects must be considered from three sides. The first aspect is the safety of handling reagents and materials using mobile systems. The life science laboratory environment contains flammable, explosive, toxic and harmful chemical substances, as well as samples with high biosafety levels. During the process of transporting and picking up labware by mobile robots, it is necessary to ensure that samples do not tip over and cause the leakage of toxic and harmful substances. There are high requirements for the stability of mobile robots and the accuracy during pick and place operations.

The second aspect concerns the safe navigation of the mobile system in complex environments. The laboratory environment is complex and mobile robots need to pass through many narrow areas. Mobile robots and workers are usually required to work in the same environment. Moreover, experimental equipment, workbenches, and other fixed or movable obstacles increase the risks of mobile robot operation. Mobile robots are required to detect static/dynamic obstacles. The high accuracy of navigation can ensure that mobile robots can move safely in crowded spaces.

The third aspect concerns safety when people and robots move in a shared work area. The laboratory is used jointly by staff and mobile robots. During robot movement, the

system should detect human presence and navigate around people appropriately. It's also necessary to set safe speeds and maintain a safe distance to ensure the safety of personnel. Alerting staff during robot operations and having an emergency stop button to halt movement immediately in case of danger are critical measures. Additionally, when robots use elevators or activate automatic doors, the presence of staff must also be considered.

Then the mobility of the mobile robot in the laboratory needs to be considered. In a highly distributed, multi-story laboratory. Mobile robots need to traverse different sections of the laboratory. The laboratory has automatic doors and elevators, controlled by PLC. In case of closed-system elevators, additional methods are required to assist the mobile robot in using the elevator. Moreover, the elevator is public, and there are often people or goods in it. In the context of laboratory automation and intelligence, laboratories are required to have a full-coverage, stable and high-speed internal communication network. In this way, the mobile robot can communicate with the control system and surrounding equipment without losing the connection and causing mission failure.

After ensuring that the mobile robot can move normally in the laboratory, timeliness should also be considered. The purpose of introducing mobile robots is to realize automated laboratories, which can ensure the normal conduct of experiments without people. The environmental conditions of the laboratory may change with experimental needs, such as temperature, air pressure, light, etc. Mobile robots need to be able to work stably 24 hours a day under these changing environmental conditions and not be affected by changes in the external environment. It can charge independently when the battery is low and then continue working. Since many experimental procedures in the laboratory are time-sensitive, sample processing needs to be completed within a limited time. Therefore, mobile robots need to ensure that experimental materials can be delivered in time, which places high requirements on the efficiency of the entire operation process of mobile robots.

Appropriate pick and place operations are a challenge, too. The laboratory primarily needs to transport labware and samples. Although mobile robots are AGVs specialized for indoor transport, the pick and place components still need to be designed specifically for laboratory characteristics. First, integration with experimental workstations must be considered, followed by stability during the pick and place process. Additionally, it is essential to increase the number of items transported per trip while ensuring safety.

For some complex experiments, or when multiple experiments are performed simultaneously. Multiple mobile robots may be required to work together during the

experiment. This requires a good collaboration mechanism between robots to efficiently allocate tasks to improve overall work efficiency. Moreover, in the narrow environment of the laboratory, multiple robots interact to avoid each other and enter the narrow space in an orderly manner. These are no small challenges.

1.4 Contribution

The main goal of this dissertation is the integration of mobile robot systems into complex distributed life science laboratories to achieve total laboratory automation. Mobile robots will be used to combine different manual, semi and full automated automation island. For this purpose, both the mobile robots themselves and their navigation and localization requirements must be considered, as well as their interaction with other automation units and the development of a suitable infrastructure. The presented work does not aim to develop new strategies for localization and navigation. Rather, the possibilities of a commercial system are used and adapted to the specific requirements for use in life science laboratories.

The main contributions will be as follows:

- To integrate the mobile robots, a virtual image of the complex spatial situation, which included several laboratories on different floors, was created. Different areas such as narrow areas, single robot areas or preferred-line were created.The laboratory map was created using LiDAR, and semantic information was added to the 2D map. The behavior of mobile robots was optimized to ensure their normal operation in the complex and narrow environment of the laboratory. Intensive investigations have been carried out regarding the positioning accuracy of the mobile system itself as well as for the grasping and placing of the labware on the transfer station.
- To enable the movement of the robots in a complex environment, a suitable infrastructure concept was developed and realized. This includes the opening of several doors during the robots movement as well as solutions for a safe use of elevators. A GUI was developed in C#, displays the real-time communication status of the Virtual PLC, interaction statuses between the mobile robots and the elevator, and real-time readings from the IoT module. This allows staff to monitor the operational status of the mobile robots and elevators in real-time. Using an IoT-based approach, the status of the elevator doors is detected by ultrasonic sensors. The actual floor information is obtained using a combination of air pressure sensors and beacon based technology. A reasonable elevator interaction logic was designed, involving communication and information

processing between the elevator, the IoT module, and the AGV Interface Controller (AIC).

- For a total automation of process, the creation of a higher-level control system is necessary, which enables the planning and monitoring of the ongoing processes. A suitable multi-level management system should provide comprehensive scheduling for all integrated devices and stations. The mobile robots are integrated and controlled into the management system through integration as devices. The initial control code for supervising the mobile robot was developed in C++ using Microsoft Foundation Classes (MFC). This program tested the communication class's accuracy, focusing on assigning transportation tasks and tracking their status. Later, this work logic code was integrated into a multi-level management system.

1.5 Chapter Overview

The organization of this dissertation is as follows:

Chapter 2 provides an overview of the development of mobile robots and details indoor navigation technology, including mapping, localization and path planning. Indoor transportation systems and laboratory management systems are introduced, as well as elevator processing techniques, including elevator door status detection and floor estimation. This chapter provides an intuitive understanding of mobile robots carrying out transportation work in indoor environments.

Chapter 3 discusses the general automation concept of mobile robotics in life science laboratories, first introducing the automation of CELISCA. Then the hardware composition and related software structure of the mobile robot are introduced. This chapter provides an overview of how mobile robots can be integrated into daily laboratory work. Please include here the requirements for the system.

Chapter 4 introduces the implementation process of the mobile robot system in detail, starting with the basic navigation parts, include mapping, localization and movement.. Then it introduces how to transport labware, charging strategies, compare to another transport robot in CELISCA, and system performance results and analysis.

Chapter 5 describes the integration of mobile robotics technology into laboratory infrastructure, including automatic door handling and closed-system elevator handling, and performs related experiments and test results.

Chapter 6 introduces the laboratory's top-level control system, SAMI, and the integration of the mobile robot system as a transportation tool within SAMI. It then

outlines the three main application processes of mobile robots in life science laboratories. The chapter also verifies the specific implementation effects of the various technologies discussed in the previous chapters.

2 Chapter 2 State-of-the-Art

The field of robotics has always been a hot spot for researchers to explore, giving birth to many subdivided research directions. This chapter focus on the latest progress in the field of mobile robots, especially the technologies for indoor mobile robot navigation. These innovations provide directions for the development of indoor mobile robots and are also beneficial to laboratory transportation systems.

2.1 Definition and Types of Mobile Robots

Mobile robots are sophisticated machines that can operate autonomously or be controlled remotely, designed to move through a variety of environments without direct human intervention. These robots are essential in many industries for performing tasks that are either dangerous, repetitive, or require precision beyond human capability.

The main distinction between AGVs and Autonomous Mobile Robots (AMRs) revolves around their navigation systems and their adaptability to different environments. AGVs are traditionally used in structured environments where the path is either embedded into the floor through magnetic strips or marked with lines or wires. This makes AGVs highly effective in environments like manufacturing plants and warehouses where the layout remains constant and the paths do not need to change frequently. However, the reliance on predefined paths limits their flexibility and the environments in which they can operate effectively.

On the other hand, AMRs are equipped with advanced onboard sensors such as LiDAR, cameras, and complex algorithms that allow them to understand and interpret their environment dynamically. This capability enables AMRs to navigate autonomously without the need for marked paths, adjusting their route in real time to avoid obstacles, adapt to new layouts, and perform tasks in varying environments. This makes AMRs particularly valuable in settings where conditions can change unpredictably, such as in dynamic manufacturing settings, outdoor environments, or places where they need to interact closely with humans.

Mobile robots have various designs and can be divided into the following five main types according to the characteristics of their movement methods:

- Wheeled mobile robots: This is the most common type and includes movement models such as differential, four-wheel drive, four-wheel steering, and Ackermann steering. Due to their simple mechanical structure and efficient adaptability to the ground, wheeled robots can move quickly on flat surfaces, offering good stability and easy control [15].
- Tracked mobile robots: When robots need to move on unstable or rough surfaces, the tracked design provides better traction and stability. They are suitable for complex terrains like forests, sandy areas, and snowy fields, and are often used in military applications, search and rescue missions, and other off-road needs [16] .

- Legged mobile robots: Legged robots are bio-inspired mechanisms actuated by engines or electric motors that use legs to support and move the trunk. Multi-legged robots can adjust their leg configurations to adapt to more complex terrains than tracked robots [17].
- Flying robots: Commonly referred to as Unmanned Aerial Vehicles (UAVs), they can be categorized into fixed-wing, rotary-wing, multirotor such as quadcopters, and Vertical Take-Off and Landing UAVs. UAVs are equipped with intelligent navigation systems that allow for autonomous flying following preset flight paths or can be precisely controlled remotely. UAVs are widely used in military operations, agricultural monitoring, image capture, express delivery, traffic surveillance, and security patrols [18].
- Underwater robots: These robots are designed to perform underwater operations, such as geophysical data collection, deep-sea mining, seabed mapping, marine exploration, and many other related activities ranging from military to scientific applications. They are usually equipped with thrusters and complex control systems to deal with currents and pressures underwater [19].

Given these types, the decision to utilize wheeled AMRs in our life science laboratory settings is driven by several key considerations. Wheeled robots offer distinct advantages in structured environments typical of laboratory settings, where conditions are stable and surfaces are predominantly flat. The inherent efficiency and ease of control of wheeled robots make them ideal for navigating the predictable pathways of a laboratory, ensuring reliable transportation of sensitive materials and labware.

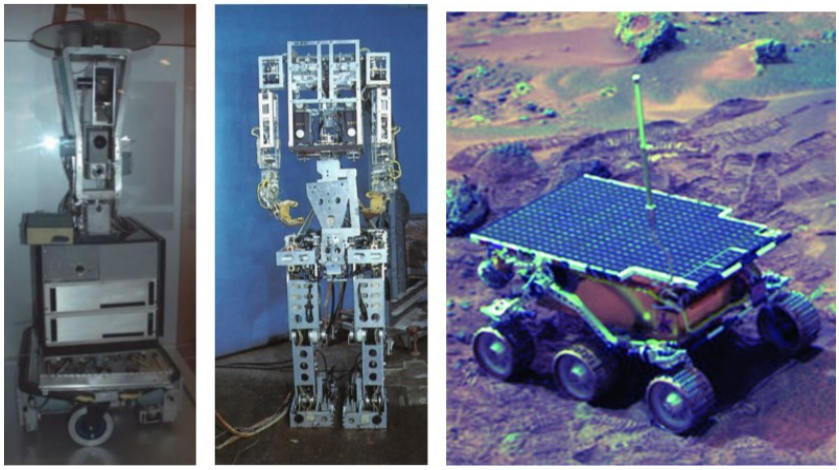
The adaptability of AMRs, equipped with advanced sensors and navigation technologies, enhances their suitability for dynamic laboratory environments. Unlike AGVs, which require modifications to the environment such as the installation of magnetic strips or wires, AMRs can integrate into existing infrastructures without such alterations. This flexibility allows AMRs to autonomously detect and adapt to changes within the laboratory, such as rearranged furniture or temporary obstacles, reducing downtime and increasing efficiency.

Moreover, the ability of AMRs to recalibrate their paths in real-time offers significant advantages in environments where laboratory setups can change frequently or where unexpected interruptions are common. This capability ensures that AMRs can continue to operate effectively without manual intervention, making them a robust choice for laboratory operations that require continual adaptability and precision.

2.2 Short History of Mobile Robotics

In the 1960s, the Stanford Research Institute in the United States developed an intelligent robot named Shakey, which is recognized as the world's first truly mobile robot [20]. Shakey uses TV cameras, laser rangefinders, collision sensors, etc. to sense its surroundings, and uses this

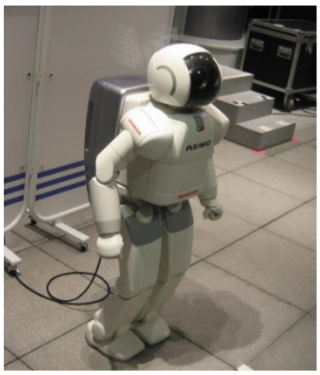
information to build its world model, using algorithms to independently plan its path. At the same time, in the 1970s, WABOT-1, the first bipedal robot developed by Waseda University in Japan, had a complete limb drive device and was equipped with visual and voice systems [21]. In 1997, the American Sojourner Mars rover was carried by the Mars Pathfinder probe and landed on Mars for the first time, as shown in Figure 1. It was the first robotic exploration rover used by humans for scientific investigation on Mars [22], is an autonomous robotic vehicle that can also accept ground remote control. During its working period on Mars, it traveled a total of more than 90 meters, analyzed the composition of rocks on Mars, and took hundreds of high-definition pictures. Sojourner conducted the first roving exploration on the surface of Mars and beyond the Earth system in human history.



(a) (b) (c)

Figure 1 Pioneers of Robots, (a) Shakey robot [20]; (b) WABOT-1 robot [21]; (c) Sojourner rover [22]

Since the beginning of the 21st century, the continuous progress of science and technology has laid a solid foundation for the vigorous development of mobile robot technology. Mobile robot technology has not only made great achievements in the industrial field, but has also begun to be widely used in life services, scientific research and other fields. The ASIMO robot developed by Japan's Honda Company in 2000 is a walking robot [23]. It has various human body movements, can set actions, and make corresponding actions by judging gestures and sounds, and can replace service personnel. It is a representative work of humanoid robots to perform simple service tasks such as serving tea and water, saying hello and so on. In 2001, the portable sweeping robot Roomba was released by the American iRobot company [24]. Because of its portable function, it occupies most of the global market. This mobile robot uses SLAM technology to build indoor maps and is lightweight, efficient and ultra-practical. It has strong functions, but it also has shortcomings, such as high noise pollution and high collision rate. In 2003, Kiva developed a mobile robot suitable for automated production lines and warehousing logistics, which can complete cargo handling tasks between two points. The launch of Kiva has greatly reduced cargo handling time, thus greatly improving production efficiency [25].



(a)

(b)

(c)

Figure 2 a) ASIMO robot [23]; b) Roomba robot [24]; c) Kiva robot [25]

In 2008, the American company Boston Dynamics developed the quadruped robot BigDog for military use. It is equipped with about 50 sensors to measure the body posture of the robot. It uses four low-friction hydraulic cylinder actuators to provide power for the joints. It can carry more than With a weight of 150 kilograms, it can cross complex terrains such as mud, rocks, forests and snow, reaching a running speed of 6.4 kilometers per hour [26].

Figure 3 The BigDog robot [26]

In 2012, the state of Nevada in the United States issued a legal license to Google's self-driving car, marking a new era in autonomous vehicle research. In 2014, Tesla launched its first electric car equipped with autonomous driving features, the Model S. In 2015, Uber launched the

world's first commercial autonomous taxi project. In 2017, Tesla introduced the Model 3, equipped with advanced hardware capable of achieving higher levels of autonomous driving. By 2020, Tesla's autonomous vehicles were capable of performing functions such as highway autopilot, automatic overtaking, and automatic parking, entering the stage of advanced autonomous driving [27]. As more institutions and companies increasingly invest in autonomous driving technology, autonomous vehicles are going to enter the Level 5 autonomy, which represents full automation without any human intervention needed under all driving conditions [28].

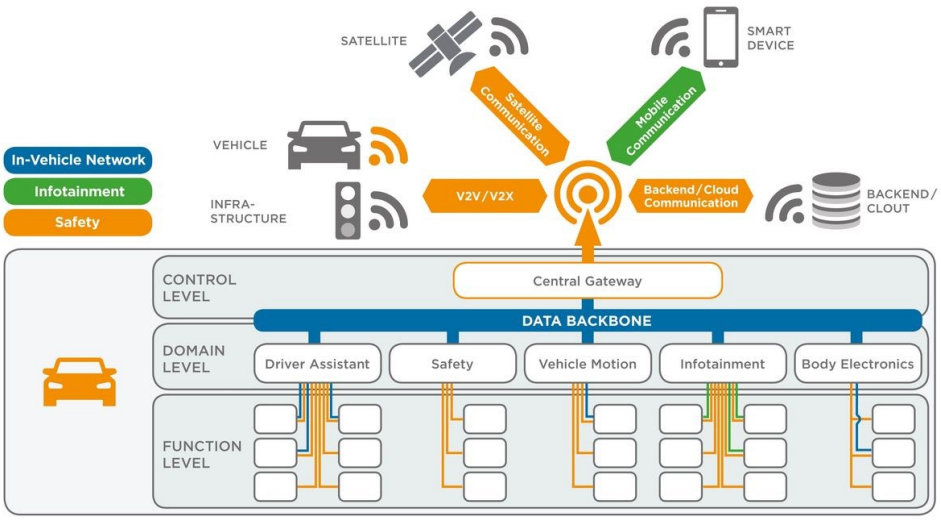


Figure 4 Functional domains for connected, autonomous vehicles with high speed in-vehicle networks [29].

According to the World Robotics 2023 Report by the International Federation of Robotics (IFR) [30], 553,052 industrial robots were installed globally in factories, marking a 5% growth from the previous year. By region, 73% of the newly deployed robots were installed in Asia, 15% in Europe, and 10% in the Americas. China remains the largest market to date, with an annual installation of 290,258 units. Germany ranks fifth with a total of 25,600 units installed. The following chart displays changes in robot installations across five main application areas: The electrical/electronics industry leads in robot installations, with numbers increasing from 112,000 units in 2020 to 157,000 units in 2022. The automotive industry follows, showing a steady number of 136,000 installations each year from 2021 to 2022.

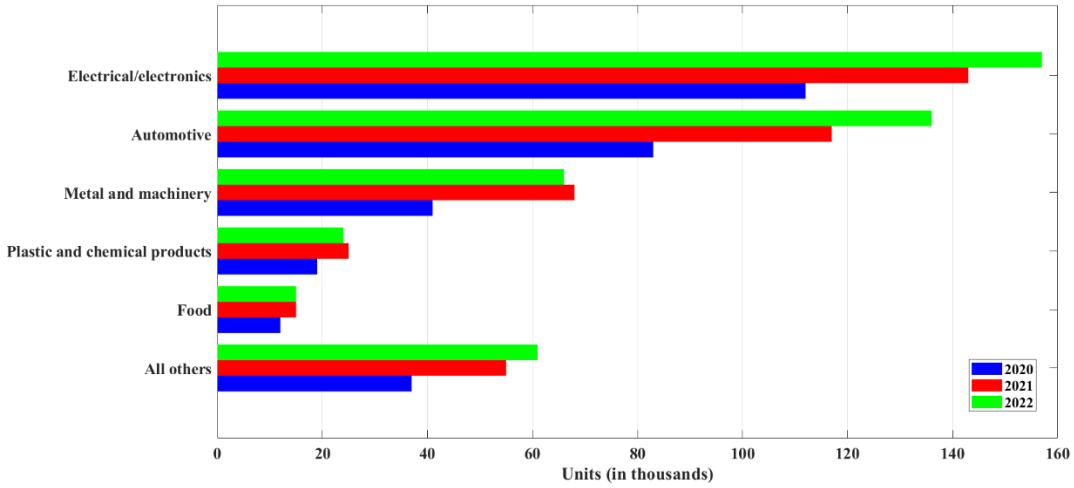


Figure 5 Annual installations of robots by customer industry – World [30]

2.3 Market Situation Mobile Robots

The landscape of mobile robotics is evolving, driven by advancements in both commercial applications and research initiatives. In commercial sectors, mobile robots are deployed across various industries, including healthcare, warehousing, hospitality, retail and factory, to enhance efficiency and automate repetitive tasks. This section will explore some commercially available mobile robots, as well as research projects that are advancing mobile robot technology.

- **Healthcare**

The AETHON TUG robot (Aethon Inc., Pittsburgh, USA) is designed to automate the internal logistics of hospitals by transporting medications, laboratory specimens, and other supplies throughout the facility. This autonomous robot helps reduce the workload on hospital staff and improves operational efficiency by ensuring timely and safe delivery of essential items [31]. The Xenex Disinfection Robot (Xenex Disinfection Services, San Antonio, USA) is employed in healthcare settings to sterilize rooms using pulsed xenon ultraviolet light, which is highly effective at killing harmful pathogens and reducing hospital-acquired infections. This robot is quick, efficient, and can disinfect a room in a matter of minutes, making it an essential tool in maintaining a clean and safe hospital environment [32]. The Da Vinci Surgical Robot (Intuitive Surgical, Sunnyvale, USA) represents a significant advancement in minimally invasive surgery, offering high precision, flexibility, and control beyond the capabilities of the human hand. It allows surgeons to perform complex procedures with enhanced vision, precision, and control, leading to faster patient recovery times and reduced risk of complications [33]. Moxi (Diligent Robotics, Austin, USA) is a socially intelligent robot assistant designed by Diligent Robotics to support clinical staff in hospitals. It can autonomously perform routine tasks such as fetching and delivering supplies, thereby allowing the medical staff to focus more on direct patient care. Moxi's integration into hospital workflows boosts efficiency and helps alleviate the stress on

overburdened healthcare workers [34].

- **Warehousing**

Amazon warehouse robots were initially developed by Kiva Systems and later acquired by Amazon; these robots are used extensively in Amazon's fulfillment centers (Amazon Robotics, North Reading, USA). They automate the picking and packing process by moving entire shelves of products to the packing station, significantly reducing the time and physical strain on human workers [35]. Geek+ (Geek+, Beijing, China) offers a range of robotic solutions, including picking, moving, and sorting robots. Their robots use advanced AI to navigate and can be integrated into existing warehouse management systems to improve efficiency and reduce errors [36]. GreyOrange robots (GreyOrange, Georgia, USA) use machine learning to decide the shortest and fastest route to move products, improving overall efficiency. This system includes both goods-to-person and sortation robots that are capable of handling multiple tasks from storage to delivery preparation [37]. LocusBots (Locus Robotics, Wilmington, USA) work alongside human workers to fulfill orders more efficiently. They navigate through the warehouse using Locus Robotics' proprietary technology, helping to pick goods and transport them to the shipping area. These robots are known for their ability to quickly adapt to changes in inventory or demand without requiring reprogramming [38].

- **Hospitality**

Savioke's Relay Robots (Savioke, San Jose, USA) are designed for room service and delivery within hotels. They can navigate through hotel corridors autonomously to deliver items like towels, toiletries, and snacks directly to guest rooms. Hotels such as Aloft have utilized these robots to enhance the guest experience and reduce wait times for deliveries [39]. LG's CLOi ServeBots (LG Electronics, Seoul, South Korea) are deployed in several hotels. They are designed to deliver food and beverages, manage linen and cleaning services, and guide guests around the hotel premises. Their ability to interact with guests and handle multiple service tasks simultaneously helps improve operational efficiency [40]. Henn na Hotel (Japan) is famous for being staffed almost entirely by robots. From receptionist robots that handle check-ins and check-outs to robot porters that transport luggage, the hotel showcases an extensive integration of robotics in whole hospitality management [41].

- **Retail**

Pepper (SoftBank Robotics, Tokyo, Japan) is a humanoid robot used in retail stores to interact with customers, provide assistance, and even gather customer feedback. Pepper can recognize human emotions and adapt its interactions, making it suitable for customer service roles in stores like b8ta and SoftBank's own mobile phone stores [42]. Bossa Nova robot (Bossa Nova Robotics, San Francisco, USA) is used by major retailers like Walmart. These robots roam aisles to scan and check inventory levels, identify misplaced items, and help in price verification. Their use allows employees to focus on customer service by handling routine checks and

updates more efficiently [43]. LoweBot robot (Fellow Robots, San Jose, USA) are deployed in Lowe's home improvement stores to help customers find products in the store by navigating them to the correct aisle while also scanning inventory and helping with price checks. It uses natural language processing to understand customer queries and respond accordingly [44].

- **Factory**

In factory, AGVs such as the KUKA KMR iiwa (KUKA, Augsburg, Germany), Omron LD Series (Omron Corporation, Kyoto, Japan), and MiR200 (Mobile Industrial Robots, Odense, Denmark) are revolutionizing modern manufacturing. Autonomous solutions in factories minimizes the need for human oversight while maximizing productivity and precision in tasks ranging from assembly to logistics.

The KMR iiwa is a flexible, autonomous mobile robot that combines KUKA's LBR iiwa lightweight robot with a mobile platform. It is designed for delicate assembly tasks that require high precision and safety. This AMR can navigate complex environments autonomously, adapting to changes and avoiding obstacles with advanced sensor technology. It is particularly suited for industries where sensitive handling and versatility are crucial [45].

The Omron LD Series comprises autonomous mobile robots designed to transform material handling within industrial environments. Equipped with proprietary navigation software, these robots can identify their environment and choose the best path to their destination, avoiding obstacles along the way. The LD Series robots are widely utilized in manufacturing and logistics to efficiently move items from one location to another without human intervention [46].

The MiR series robot is a more compact, flexible AGV that can automate the transportation of goods and materials across various industries without altering the layout. With a payload capacity of 200 kg and a towing capacity of 500 kg, the MiR200 is ideal for small-space operations needing automated transport solutions. It integrates easily into existing environments and can be programmed to navigate using simple web-based interfaces [47].

The Freight series robot (Fetch Robotics, San Jose, USA) is part of a larger system designed for warehouse and manufacturing environments. This robot is specifically engineered to move large payloads through busy and dynamic workspaces safely. It uses cloud software for navigation and can be integrated with other Fetch robots to create a comprehensive in-facility delivery system, significantly optimizing material transport without the need for facility modifications [48].

OTTO Motors' AMR (OTTO Motors, Kitchener, Canada) is a heavy-duty autonomous mobile robot designed for moving large volumes of materials in manufacturing plants and warehouses. It features dynamic obstacle avoidance technology and high-load capacities, making it an optimal choice for industries looking to improve throughput without compromising safety. OTTO Motors specializes in flexible, scalable solutions that can be customized to specific operational needs [49].

➢ Research

The existing body of scholarly work on this subject emphasizes the importance of development and application in the field of robotics, especially in the context of automation workflow systems.

Kevin, a flexible mobile robot designed by Wechelmann et al. in the Fraunhofer IPA research institute (Stuttgart, Germany), to perform transporting labware activities in life science laboratories. The study aimed to design and assess the robot's interactive capabilities, including sound and light signals, and their effectiveness in communicating system status to laboratory personnel [50].

Figure 6 Robot Kevin Handling Labware in Storage [50]

Figure 7 Robot H20 Transporting Labware [51]

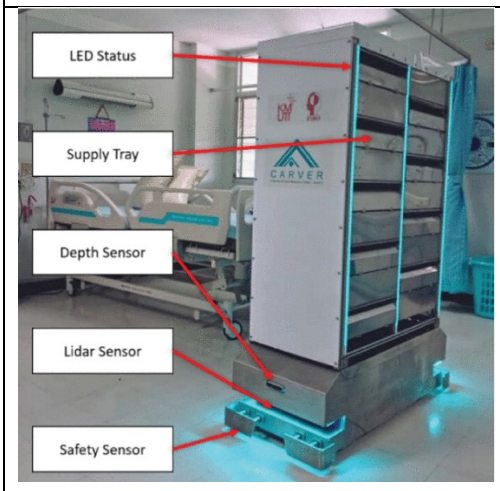


Figure 8 Main Components of Carver-Cap Robot [2]

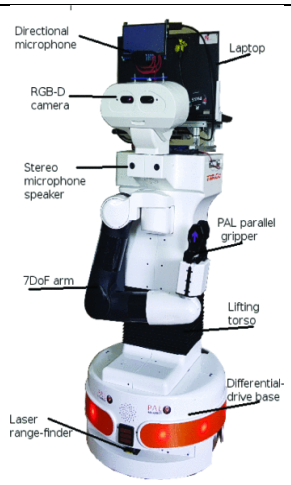
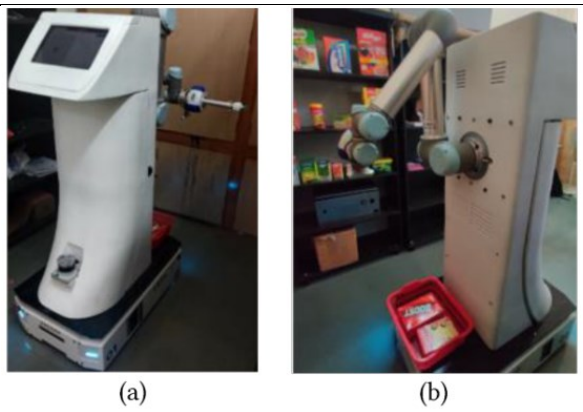


Figure 9 Robot TIAGo hardware specification [52]



(a) (b)

Figure 10 Picture of SAR after fabrication and assembly in (a) partial front view and (b) partial back view [53]

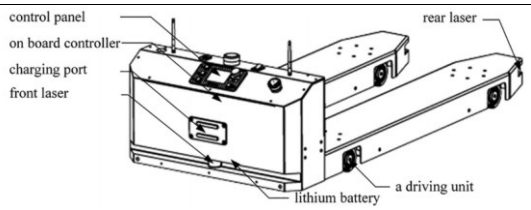


Figure 11 Mechanical diagram of logistics robot [54]

Thurow et al. developed a multi-floor laboratory transportation system based on H20 mobile robots at CELISCA. The system integrates innovative solutions for indoor navigation, intelligent collision avoidance, and labware manipulation. H20 can autonomously navigate through doors and ride elevators in a laboratory environment where personnel movement is frequent. It also has the capability to automatically identify labware and use its robotic arms for grasping and placement [51]. This robot system also includes obstacle avoidance features based on human-robot interaction. It enables the robots to interact with humans via gestures, which are recognized by a Kinect 2.0 sensor. These gestures dictate movements like moving forward, moving backward, or initiating collision avoidance maneuvers. Gesture recognition is performed by classifying human skeletal images using Support Vector Machine (SVM) modeling [52].

Thamrongaphichartkul et al. proposed a framework for an IoT platform for AMRs used in hospital logistics. This research focuses on improving efficiency and safety in hospitals, particularly during the COVID-19 pandemic, by using AMRs for the delivery of food and medical supplies, thereby limiting physical contact between patients and healthcare workers. The AMR, named "CARVER," is equipped with an IoT module that connects to a server-side IoT platform. This allows healthcare workers to effectively monitor and control the robots via a web application [2].

Kramer et al. proposed a comprehensive system to enhance human-robot interaction in a future convenience store scenario. By integrating multiple Human-Robot Interaction (HRI) modalities including gesture recognition, customer reidentification, customer profiling, and a recommendation system, this research outlines a more user-oriented and robust interaction pipeline [53].

Ilampooranan et al. developed a Shopping Assistance Robot (SAR) designed to aid customers

in retail environments. The SAR combines a humanoid structure with a mobile base and is equipped with 3D LiDAR sensors for safe navigation, a UR-5 manipulator arm for tasks like picking up items, and a display for interactive communication with users. The robot works with in-store cameras and a server system that helps it identify and navigate to customers who may need assistance or have pre-ordered goods waiting for pickup. The design and testing process underscore how SAR can perform reliably in real-world scenarios, emphasizing customer interaction and the practicality of service robots in retail [54].

Lin et al. developed an Automated Guided Logistics Robot specifically for pallet transportation within warehouses. This robot is capable of omnidirectional movement without the need to rotate its chassis. It uses lasers mounted at the front and back for self-localization and obstacle avoidance, as well as for pallet positioning, achieving a positioning accuracy of ±5 mm. It can pick up and transport pallets weighing up to 1,000 kilograms [55].

2.4 Indoor Navigation Technologies

Indoor mobile robots are an important branch of mobile robots. As indoor mobile robots gradually transition from fixed industrial settings to unstructured common indoor environments, they must solve four basic problems to achieve efficient autonomous navigation: mapping, localization, path planning, and dynamic obstacle avoidance [56].

2.4.1 Mapping

Mapping is fundamental for achieving autonomous navigation and effectively carrying out tasks. Maps provide robots with a framework of the environment, within which they can understand and remember the geometric features of their location, including fixed structures like walls and doors, as well as common obstacles they may encounter during operation. Various types of maps are also used for different autonomous exploration methods, such as grid maps, topological maps, and feature maps. A current emerging trend is to build upon maps created from general scanning by manually adding specific meanings to the environment to create semantic maps. Below is a detailed introduction to several common types of maps used in mapping.

1. Grid map divides the environment into a series of small cells, each representing an area in the environment. Each cell typically has three states: "occupied" means presence of obstacles, "free" means no obstacles, or "unknown" means unexplored. In a grid map, the robot can determine whether an area is passable based on the occupancy state of the cells, which is fundamental for subsequent path planning and obstacle avoidance. The advantage of a grid map is that it provides a simple and intuitive method to represent and update the robot's understanding of the environment. This type of map is particularly suitable for situations that require precise obstacle avoidance and path planning.

He et al. proposed a dynamic spatio-temporal grid map to improve the obstacle avoidance performance of mobile robots with Mecanum wheels in dynamic environments [57]. This method can identify and predict the trajectories of static and dynamic obstacles and achieve collision-free movement behavior and path planning in dynamic environments. By utilizing the A* algorithm for continuous neighborhood search, it resolves the issue in grid-based path planning algorithms where only the grid centers are chosen as searchable nodes. By adjusting the traditional A* algorithm's searchable neighborhoods from discrete eight directions to continuous infinite directions, the search direction also becomes any continuous direction. Applying the ID3 algorithm simplifies the decision model, significantly enhancing the efficiency and effectiveness of path planning, and experimental results validate the effectiveness of the proposed method.

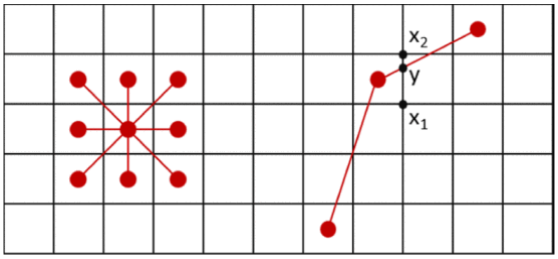


Figure 12 Traditional A* algorithm 8 adjacent edges (left), searchable continuous neighborhood A* algorithm (right) [37]

2. Topological Map focus on significant features in the environment and their relationships, rather than specific geometric shapes. This type of map is structured using key location nodes and edges that connect these nodes, similar to a subway route map. Unlike grid maps, topological maps reduce the need for precise positioning throughout the map, decrease the level of detail, and can be directly used for path planning.

   Kumaar et al. explored how to enhance the navigation capabilities of service-oriented mobile robots in dynamic environments using a path planning method based on reinforcement learning. The authors stored environmental information in topological maps and utilized the Q-Learning algorithm to train the robot to learn paths, demonstrating the method's application in various service settings. Experimental results indicated that the proposed system could learn and generate efficient paths based on topological maps, achieving an accuracy of 95%. This AI-based framework can expand the capabilities of path planning systems to adapt to long-term path changes [58].

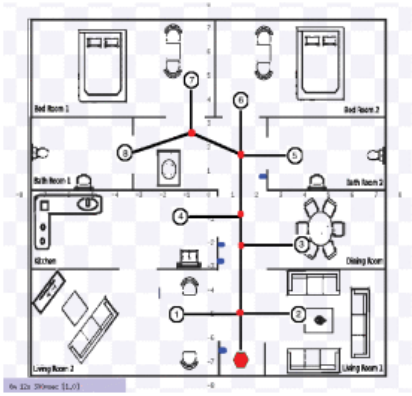


Figure 13 House layout based on Topological map [38]

3. Voronoi diagram is a method of dividing a plane into regions generated by a set of Voronoi points, where each region contains all the points closest to a specific Voronoi point. In mobile robot applications, Voronoi diagrams can be used to generate a special form of topological map, in which the boundaries of the Voronoi diagram define paths where the robot can safely move. These paths typically lie in the "middle" areas between obstacles, thereby minimizing the risk of collisions between the robot and obstacles.

   Liu et al. proposed a path planning method that combines weighted Voronoi diagrams with an improved A* algorithm to enhance the navigation efficiency of mobile robots in complex environments. By adopting the concept of the A* algorithm and integrating ideas of map construction, the method prioritizes the expansion of weighted Voronoi points as preferred expansion nodes. This approach allows for finding shorter paths in complex environments, thereby improving search efficiency, as well as the safety and accuracy of the paths. Through simulation experiments, the algorithm was able to quickly find shorter paths, not only enhancing search efficiency but also improving path safety and accuracy. Moreover, by using Bézier curves to smooth the path's turning points, the quality of the paths was further improved [59].

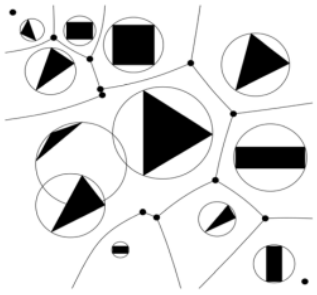


Figure 14 Weighted Voronoi diagram [39]

4. Feature maps utilize key features extracted from the environment, such as corners, edges, or other prominent objects, to describe the environment. They are a type of sparse map, where the map contains only the spatial points that have features. These features are precisely located on the map and are used for robot localization and navigation. Feature maps are

especially common in visual SLAM systems, as these systems rely on visual features to track the movement of the robot.

Gao et al. proposed a novel 2-D laser-based autonomous exploration approach for mobile robots, combining graph-based SLAM with directional endpoint features and polygon map construction [1]. The approach consists of three modules: graph-based SLAM using directional endpoint features, polygon map construction, and exploration. Unlike traditional 2-D SLAM, the proposed 2-D graph-SLAM utilizes 3-D "directional endpoint" features, which provide a more detailed representation of the environment. The polygon map, essential for navigation, is efficiently maintained using a "circular-doubly linked list" data structure, allowing for easy initialization and updates [60].

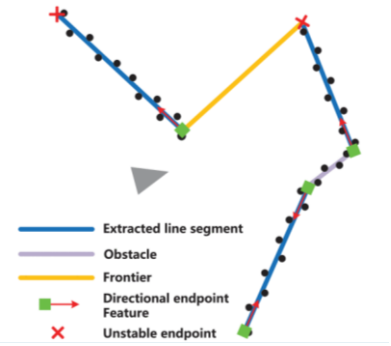


Figure 15 Schematic illustration of the polygon map. The gray triangle denotes the mobile robot, while the red arrow and the green filled square represent the directional endpoint feature, and The unstable endpoint is marked by the red cross. In addition, three line segments of different colors represent three types of edges between two consecutive vertices [40]

5. Semantic mapping is a field that has seen significant advancements in mobile robot navigation recently. In addition to providing geometric information about the physical space, semantic maps also include descriptive information about objects and scenes in the environment. This enables robots to not only "see" the objects around them but also understand their meaning and function. The environmental semantic information typically referred to includes scene information, such as corridors, rooms, elevators, and automatic doors in a laboratory environment. This type of map is beneficial for performing more complex, contextually meaningful tasks.

Research on robot exploration using semantic information primarily focuses on environmental modeling and efficient exploration. Xiang et al. proposed a structure-aware semantic mapping method for autonomous disinfection robots working in subway environments. Prior structural information such as the positions of pillars and seats within subway cars are used to assist in mapping, and extracting different levels of semantic information from point clouds to meet the disinfection requirements of the robots [61]. In some practical applications, imprecise structural diagrams, exterior wall outlines, or emergency maps of environments are available, and some scholars use such prior information to complete

exploration and mapping tasks. Luperto et al. proposed using inaccurate or even incomplete environmental information to facilitate the selection of target exploration areas for robots, and experimentally validated the effectiveness of this algorithm in promoting exploration [62]. Mielle et al. developed a mapping method using outdated, incomplete, or even inaccurate emergency maps combined with graph-based SLAM techniques, enabling the fusion of emergency maps with newly created maps to produce accurate environmental maps [63]. Gomez et al. proposed a method for autonomously creating hybrid indoor three-dimensional maps with mobile robots [64]. This hybrid mapping method uses semantic structural information to segment indoor environments, connecting dense local maps through a global topological map to represent the environment's three-dimensional information with less resource consumption.

2.4.1.1 Discussion

For laboratory mobile robots, where the environment is structured but can present dynamic challenges, combining grid maps with semantic maps is the chosen methodology. Grid maps offer a high resolution of the environment, vital for accurate movement and interaction with delicate labware and samples. They are particularly beneficial in structured lab environments for detailed path planning and precise obstacle avoidance.

On the other hand, semantic maps add contextual understanding, providing robots with higher-level information about their surroundings. This includes identifying specific lab areas such as storage, workstations, and equipment, allowing robots to perform complex, context-sensitive tasks more effectively.

The fusion of grid maps and semantic maps addresses the limitations of other methods by enhancing the robot's environmental perception and decision-making capabilities. It allows for efficient path planning and navigation in the lab, ensuring robots can safely and effectively execute tasks among sensitive equipment and busy personnel. This approach is expected to increase the autonomy and operational efficiency of laboratory mobile robots, making it a robust solution for complex indoor environments. Table 1 shows a comparison of several map methods.

Table 1 Mapping Methods Comparison

Method	Description	Features	Applications
Grid Map	Divides the environment into small cells representing areas. Suitable for precise obstacle avoidance and path planning.	Simple, intuitive representation. Suitable for precise navigation.	Dynamic environments requiring precise obstacle avoidance and path planning.
Topological Map	Focuses on significant features and their relationships, like nodes and edges. Reduces the need for precise positioning.	Decreases detail and precise positioning needs. Useful for path planning.	Service-oriented robots in dynamic environments, efficient path learning.
Voronoi	Divides a plane into regions using Voronoi points. Defines paths in the	Enhances safety and accuracy by avoiding obstacles. Suitable for	Complex environments where safe

Diagram	middle areas between obstacles, minimizing collision risks.	complex environments.	navigation is crucial.
Feature Map	Utilizes key features like corners and edges for localization and navigation. Common in visual SLAM systems.	Precise localization and navigation using features. Common in visual-based navigation.	Visual SLAM systems for tracking and navigation.
Semantic Map	Includes both geometric and descriptive information about objects and scenes. Allows for contextually meaningful tasks.	Combines geometric and semantic understanding. Useful for complex tasks in dynamic environments.	Contexts requiring complex, context-sensitive tasks, like disinfection robots in subway environments.

2.4.2 Localization

Localization is an indispensable part of a robotic navigation system. Localization refers to the process by which a robot identifies its correct position within a known map. Mobile robots can utilize various sensors, such as LiDAR, visual cameras, Inertial Measurement Units (IMU), and Radio Frequency Technology, to collect detailed information about their surrounding environment. By comparing this data with the map, mobile robots can precisely determine their location. Then, they can proceed with subsequent movements.

2.4.2.1 Inertial Measurement Unit

The IMU is the most basic sensor in mobile robots, and all mobile robots are equipped with IMU. The IMU is composed of an accelerometer and a gyroscope, which can measure the acceleration, angular velocity and angle increment of the mobile robot. Real-time positioning of the robot is achieved through integral calculation of the motion trajectory and pose according to the parameters of acceleration, angular velocity and angle increment. This method can only rely on the internal information of the robot for autonomous navigation [65]. Therefore, the calibration accuracy of the IMU directly affects the positioning accuracy of the mobile robot.

The advantages of inertial navigation are strong anti-interference; inertial sensors will not reduce the accuracy due to the interference of external environment signals such as Wi-Fi, sound waves, etc. The working speed of the IMU is fast, and the inertial navigation of the IMU is more suitable for fast-moving objects than other positioning technologies. The main disadvantage is that the positioning error of inertial navigation is cumulative [66]. Thus, the last positioning error will affect the next positioning, which will cause the error to expand. As the robot is moving, the positioning error of inertial navigation will continue to increase if there is no correction process. Therefore, other navigation methods is used in combination with inertial navigation. In this way, the positioning method will retain the advantages of strong anti-interference and high speed of inertial navigation, and will limit the error to a certain range.

2.4.2.2 Visible Light Communication

Localization based on visible light is a new type of localization method. The key element of VLC communication is the LED light [67]. The communication principle is to encode

information, and transmit the encoded information by adjusting the intensity of the LED, and the photosensitive sensor can receive and decode the high-frequency flickering signal. Using photo sensors to locate the position and orientation of LEDs, RSSI, TOA and AOA of radio methods are the main localization techniques [68]. In contrast, also a camera can be used to receive images of LEDs [69]. This approach is similar to computer vision.

LED is a kind of energy saving lighting equipment, which has the characteristics of low energy consumption, long life, environmental protection and anti-electromagnetic interference. With the popularity of LED lights now, the cost of environmental modification using VLC positioning is lower, which is similar to Wi-Fi. LED lights can be installed in large numbers indoors while considering the functions of lighting and communication. Due to its high-frequency nature, it transmits information without disturbing the illumination [70]. A major disadvantage of VLC is that it can only communicate within the line of sight (LOS) range. Moreover, light does not interfere with radio frequency equipment and can be safely used in places where radio frequency signals are prohibited.

2.4.2.3 Infrared Detection Technologies

Infrared is an electromagnetic wave that is invisible to the human eye and has a longer wavelength than visible light. The positioning of indoor mobile robots based on infrared methods relies on artificial landmarks with known positions, which can be divided into active landmarks and passive landmarks according to whether the landmarks require energy or not. The principle of active landmarks is that a mobile robot equipped with an infrared transmitter emits infrared rays, a receiver installed in the environment receives the infrared rays, and then calculates the position of the robot, which can achieve sub-meter accuracy. The principle of passive landmarks is that the arrangement in the environment can reflect infrared landmarks, and the mobile robot receives the reflected infrared information at the same time, and obtains the landmark ID, position and angle and other information [71]. At present, infrared-based positioning technology is relatively mature, but its penetration is poor, and it can only perform line-of-sight measurement and control. It is susceptible to environmental influences such as sunlight and lighting, so the application is limited.

CELISCA has developed a complete automatic transportation system for multi-mobile robots across floors [51]. The positioning method was selected by Hagisonic's StarGazer. It performs indoor positioning based on infrared passive landmarks, and obtains the relative position of the robot and passive landmarks through visual calculation to evaluate the robot's own position and orientation. The average positioning accuracy is about 2 cm, and the maximum error is within 5 cm. Bernardes et al. designed an infrared-based active localization sensor [72]. Infrared LED lights are arranged on the ceiling, and the mobile robot is equipped with an infrared light receiver. Combined with EKF, the pose of the mobile robot is calculated using the emission angle of the LED.

2.4.2.4 Ultrasonic Detection Technologies

Ultrasonic is defined as a sound wave with a vibration frequency higher than 20 kHz, which is higher than the upper limit of human hearing and cannot be heard. At present, ultrasonic ranging has three methods: phase detection, acoustic amplitude detection, and time detection. In practical applications, most of them are time detection methods using the TOF principle. The time detection method obtains the distance by calculating the time when the ultrasonic transmitter transmits the ultrasonic wave and the time difference and sound speed of the ultrasonic wave received by the receiving end [73]. The accuracy can theoretically reach the centimeter level. Among them, the phase detection method has the advantages of better detection accuracy. Its principle is to calculate the measurement distance based on the ultrasonic wavelength by comparing the phase difference when the ultrasonic wave is transmitted and the phase difference when the ultrasonic wave is received. It also brings a defect. The measured phase difference is a multiple solution value with a period of $2n\pi$, so its measurement range needs to be within the wavelength.

2.4.2.5 Geomagnetic Field Detection Technologies

The earth is wrapped in a natural geomagnetic field formed between the north and south poles, and the strength of the magnetic field varies with latitude and longitude. At present, geomagnetic fingerprints are mainly used for indoor positioning based on geomagnetism. Magnetic fingerprints have unique properties in space, with geomagnetic modulus values in a small contiguous region forming a fingerprint sequence [74]. Geomagnetic fingerprint positioning includes two stages: offline training and online positioning. During the offline training stage the geomagnetic sensor is used to collect the geomagnetic information about the sampling points, and the information is stored in the fingerprint information database. In the online positioning stage the geomagnetic information about the current location obtained by the geomagnetic sensor is used to form the fingerprint information, and match the information in the fingerprint database. The geomagnetic field is less affected by external factors, there is no cumulative error, and there is no NLOS problem, but the accuracy is at the meter level. Generally, geomagnetism is used as an auxiliary positioning tool to correct accumulated errors, or used with other sensors for information fusion.

Lv et al. developed a self-tuning Kalman filter that fused the geomagnetic sensor, IMU and encoder, and used the real-time update global position characteristics of the geomagnetic sensor to correct the error accumulation defect of dead reckoning [75]. The positioning method of particle filter-based is presented by Isaku Nagai et al. to fuse data from multiple magnetic sensors and an optical sensor and a gyroscope [76].

2.4.2.6 Classical Methods

Radio frequency technology is one of the most widely used technologies in the field of indoor positioning. This technology is based on radio frequencies and covers a wide range of areas.

Utilizing existing communications infrastructure to implement positioning functionality is convenient and economical because it can be expanded with low energy consumption and low cost. The disadvantage is that the accuracy is not high, and further optimization is required. Usually, these optimization methods are based on filters [77] and machine learning [66]. Radio frequency technologies include Wi-Fi, Bluetooth, ZigBee, Radio Frequency Identification (RFID) and ultra-wideband (UWB).

Radio frequency (RF) technologies are pivotal in mobile robot positioning systems, providing low-energy and cost-effective solutions for navigation. Classic RF positioning algorithms include Received Signal Strength Indication (RSSI), Time of Arrival (TOA), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA), each offering distinct mechanisms for determining a robot's location. RSSI-based methods, such as triangulation and fingerprinting, calculate the robot's proximity to known Access Points (APs) via signal attenuation models [78], they require a stable environment. Fingerprinting, although more labor-intensive due to its two-stage process involving data collection and matching, yields higher accuracy. Channel State Information (CSI) offers a refined alternative to RSSI by providing detailed frequency response data, enhancing stability and positioning accuracy despite its larger data transmission requirements [79]. Non-RSSI-based approaches, TOA and TDOA [80], calculate distances using signal travel times, while AOA determines the robot's location through the angle relative to Aps [81]. These methods typically face challenges with multipath and non-line-of-sight conditions but can deliver effective results with proper synchronization and equipment calibration.

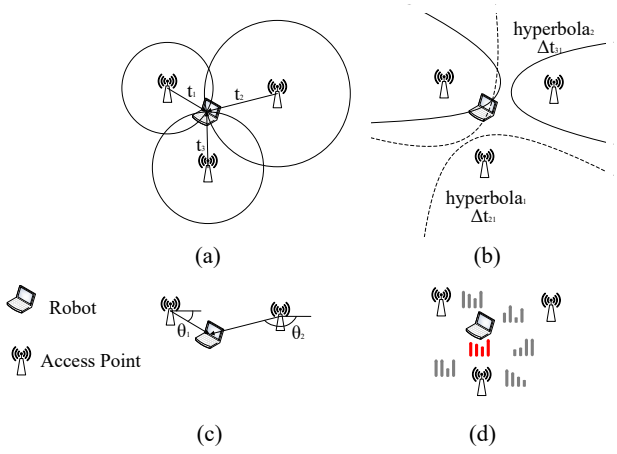


Figure 16 Radio Frequency technologies, (a) TOA; (b) TDOA; (c) AOA; (d) RSSI

Wi-Fi technology, the most commonly used communication method, is a prevalent choice for indoor positioning due to its simplicity in deployment and lack of additional hardware needs. However, challenges like signal fluctuation and multipath fading necessitate advanced processing techniques, including filters [77] and machine learning [82] to enhance accuracy.

Bluetooth, especially its low-energy variant, is favored in IoT applications for its low cost and energy efficiency. Advances in Bluetooth technology, notably the introduction of Bluetooth 5.0,

promise sub-meter level accuracy [83], making it more suitable for mobile robot positioning. The trend of Bluetooth-based indoor positioning is to combine optimization algorithms with positioning technology. Mankotia et al. implemented a Bluetooth localization of a mobile robot based on iterative trilateration and cuckoo search (CS) algorithm based on Monte Carlo simulation [84]. Comparing the cuckoo algorithm and particle filter in the simulation, the positioning accuracy of CS algorithm (mean error 0.265m) is higher than that of particle filter (mean error 0.335m), and the overall error is reduced by 21%.

ZigBee is characterized by its low cost and high reliability but grapples with stability issues and environmental interference, leading to meter-level positioning accuracy. Its positioning principle is similar to that of Wi-Fi and Bluetooth, and it is also mainly based on RSSI to estimate the distance between devices. Methods to improve ZigBee positioning involve integrating algorithms that account for dynamic environmental changes and signal noise [85]. Wang et al. designed a mobile robot positioning system based on the ZigBee positioning technology, and combined the centroid method and the least square method to improve the positioning accuracy of the RSSI algorithm in complex indoor environments [86].

RFID positioning technology uses radio frequency signals to transmit the location of the target, works with passive and active tags, with passive tags being more common in indoor scenarios due to their cost-effectiveness. UHF-RFID tags, in particular, are employed in mobile robot positioning for their simple structure and rapid data transmission capabilities. Filtering techniques are applied to enhance signal accuracy and address RFID's reflection and divergence problems. DiGiampaolo et al. [87] and Wang et al. [88] both chose Rao-Blackwellized particle filters as the algorithm for RFID SLAM. The reflection and divergence problem of RFID tags is serious. Magnago proposed an unscented Kalman filter (UKF) to solve the phase ambiguity of the RF signal backscattered by UHF-RFID tags [89].

Lastly, UWB as a carrierless communication technology, UWB uses non-sinusoidal narrow-band pulses with very short periods to transmit data, which can achieve high-speed, large-bandwidth, and high-time-resolution communication in proximity [90]. Extremely short pulse modulation makes it possible to greatly reduce the effects of multipath problems. Low transmission power avoids interference with Wi-Fi, BLE or similar devices. UWB has stronger wall penetration than Wi-Fi and BLE. Recently, UWB has become the focus in the field of indoor positioning due to its stable performance and centimeter-level positioning accuracy, making it one of the ideal choices for the positioning of mobile robots. High-precision positioning at the centimeter level requires numerous anchor points, resulting in high costs.

Traditional UWB localization systems are mostly based on mathematical methods and filters [91] [92]. Recently, data fusion is the trend of UWB positioning. A Constraint Robust Iterate Extended Kalman filter algorithm has been proposed by Li and Wang to fuse UWB and IMU [93]. Sage-Husa Fuzzy Adaptive Filter is used to fuse UWB and IMU [94]. Liu et al. studied how to arrange anchor points scientifically, and proposed a UWB-based multi-base station

fusion positioning method [95]. In the actual robot positioning process, the arrangement and rationality of the base station directly determine the positioning accuracy and benign space area.

2.4.2.7 LIDAR Detection Technologies

LiDAR systems consist of an optical transmitter, a reflected light detector, and a data processing system. First, the optical transmitter emits discrete laser pulses to the surrounding environment. The laser pulses are reflected after encountering obstacles. The reflected laser light is received by the reflected light detector, and then the data are sent to the data processing system to obtain a two-dimensional image or 3D information of the surrounding environment. LiDAR systems are active systems since they emit laser pulses and detect reflected light. This feature supports the mobile robot to work in a dark environment, does not require any modification of the environment, and is highly universal. In indoor situations, 2D LiDAR is mainstream, and 3D LiDAR is usually used in the field of unmanned driving [96].

The 2D LiDAR SLAM is the mainstream method for the indoor positioning of mobile robots. Without prior information, LiDAR SLAM uses LiDAR as a sensor to obtain surrounding environment data, evaluate its pose, and draw an environment map based on the pose information. Finally, synchronous positioning and map construction are realized. At present, there are two types of 2D LiDAR SLAM methods to solve the indoor positioning problem of mobile robots: filter-based and graph-based [97]. The problem to be solved is the estimation of a posterior probability distribution over the robot's location given sensor measurements and control inputs. This can be stated as follows:

$$P(x_t \mid z_{\{1:t\}}, u_{\{1:t\}}) = \eta * P(z_t \mid x_t) * \int P(x_t \mid x_{t-1}, u_t) * P(x_{t-1} \mid z_{\{1:t-1\}}, u_{\{1:t-1\}}) dx_{t-1} \quad (1)$$

where x_t represents the robot's state (position and orientation) at time t; $z_{\{1:t\}}$ represents the lidar observation data from time 1 to t; $u_{\{1:t\}}$ represents the robot control input from time 1 to t; $P(x_t \mid z_{\{1:t\}}, u_{\{1:t\}})$ represents the posterior probability of the robot's state at time t, given the observation data and control input; $P(z_t \mid x_t)$ represents the likelihood probability of observation data z_t, given the robot state x_t; $P(x_t \mid x_{t-1}, u_t)$ represents the transition probability of the current state x_t, given the previous state x_{t-1} and the current control input u_t; and η is a normalization factor that ensures the sum of probabilities is 1. By iterating this Formula (1), the robot's position and orientation can be calculated based on the LiDAR observation data and robot control input.

The filter-based SLAM method has been proposed since the 1990s and has been widely studied and applied [98]. Its principle is simply to use recursive Bayesian estimation to iterate the

posterior probability distribution of the robot pose to construct an incremental map and achieve localization. The most basic algorithms are EKF SLAM and particle filter SLAM. Gmapping SLAM and Hector SLAM are now commonly used methods.

The methods of 2D LiDAR SLAM based on graph optimization include Karto SLAM [99], published by Konolige et al. in 2010, and the Cartographer algorithm, open-sourced by Google in 2016 [100].

2.4.2.8 Computer Vision Detection Technologies

Computer Vision uses cameras to obtain environmental image information. It is designed to recognize and understand content in images/videos to help mobile robots localize. Computer Vision localization methods are divided into beacon-based absolute localization and visual odometry-based SLAM methods. Among them, visual SLAM is a hot spot for the indoor positioning of mobile robots.

Absolute positioning based on beacons is the most direct positioning method. The most common method is the QR code [101]. QR codes can provide location information directly in the video image. By arranging the distribution of QR codes reasonably, the mobile robot can be positioned in the whole working environment. Avgeris et al. designed a cylinder-shaped beacon [102]. Typically, beacons are placed in the environment, and Song et al. innovatively put QR codes on robots and vision cameras mounted on the ceiling [103]. The location of the mobile robot can be determined by tracking the QR code. Lv et al. utilized grids formed by tile joints to assist mobile robot localization [104].

Visual SLAM is similar to LiDAR SLAM. Multiple cameras or stereo cameras are used to collect feature points with depth information in the surrounding environment, and feature point matching is performed to obtain the pose of the mobile robot. Visual SLAM systems use geometric features such as points, lines, and planes as landmarks to build maps. The visual SLAM flowchart is shown in Figure 17 [105].

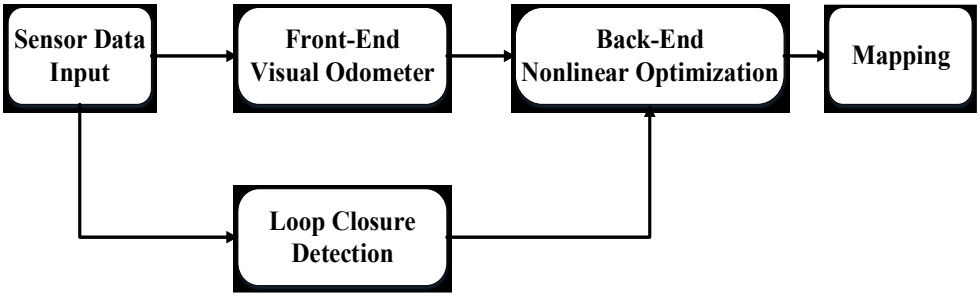


Figure 17 Visual SLAM flowchart

- Sensor data input: Environmental data collected by the camera. Occasionally, the IMU is used as a secondary sensor.
- Front-end visual odometer: Preliminary camera poses estimation based on image

information of adjacent video frames.

- Back-end nonlinear optimization: optimize the camera pose obtained by the front-end to reduce the global error.
- Loop closure detection: According to the image to determine whether to reach the previous position. Form a closed loop.
- Mapping: Build a map of the environment based on continuous pose estimates.

2.4.2.9 Discussion

The review of various robot localization technologies indicates a rich palette of methods, each with its set of advantages and trade-offs, suitable for different aspects of mobile robot navigation in indoor environments. While Radio Frequency and IMU technologies offer cost-effectiveness and strong anti-interference capabilities, they come with limitations such as lower accuracy and cumulative errors, respectively. On the other hand, Visible Light Communication (VLC) and Infrared Detection offer high precision but are constrained by line-of-sight requirements and susceptibility to environmental factors.

Ultrasonic sensors, while inexpensive and effective at measuring distances, are susceptible to angular deviation and can struggle with echo interference in cluttered environments, which reduces their reliability for precise navigation tasks. Infrared technology offers high accuracy in obstacle detection and proximity sensing but is heavily dependent on the absence of ambient light interference and requires a clear line of sight, limiting its effectiveness in complex or densely populated laboratory settings. Geomagnetic sensors, although not affected by cumulative errors, are easily influenced by environmental magnetic fields, leading to significant errors.

Computer vision and LiDAR are both excellent methods that provide high-precision localization for robots. However, computer vision requires substantial computational resources to process images, which can introduce delays in real-time decision-making and increase the power consumption of the system. LiDAR stands out as a method that marries the high-resolution data acquisition capability of LiDAR systems with the robustness of SLAM algorithms for positional accuracy. It is especially suited for environments where precision and reliability are paramount, such as laboratories. LiDAR SLAM's ability to function independently of environmental light conditions and its immunity to many of the interferences that affect RF-based technologies make it a compelling choice for laboratory settings

The decision to use LiDAR SLAM in the laboratory setting is further justified by its flexibility in creating detailed 2D or 3D maps without the need for extensive prior environment information, facilitating real-time updates to maps in dynamic settings. Moreover, when coupled with semantic maps, LiDAR SLAM can provide not just a geometrically accurate

representation of the space but also a context-rich layer that includes the identification and classification of objects and regions within the lab. Table 2 shows a comparison of 12 localization methods.

Table 2 Comparison of 12 indoor localization technologies for mobile robots

Technology	Accuracy level	Cost	Advantages	disadvantage
IMU	0.2m [106]	Low	wide application easy to use Anti-interference	Accumulated error
VLC	<0.05m [107]	Low	easy to deploy No electromagnetic interference	Only line of sight communication
Ultrasonic	0.033m [108]	High	Mature technology High positioning accuracy	Short-distance measurement(< 4m) Signal attenuation
IR	<0.1m [72]	Low	Mature technology	Only line of sight communication Need environmental transformation Affected by sunlight
Geomagnetic	<0.21m [76]	Low	No cumulative error No Need of environmental transformation	Low accuracy (meter-level) Build a geomagnetic database
LiDAR	<0.025m [109]	Medium	strong adaptability Strong stability No Need of environmental transformation	High requirements for algorithms Affected by glass objects
Computer Vision	0.09m [110]	Medium	Collection of rich information strong adaptability No Need of environmental transformation	High requirements for algorithms and computing performance affected by light
Wi-Fi	2.31m [111]	Low	Widely used mature technology easy to deploy	Easy to be interfered Multipath problem
Bluetooth	0.27m [112]	Low	Wide application Low-power consumption	Path loss easy to be interfered
ZigBee	0.71m [113]	Low	Low-power consumption good topology	Poor stability Easy to be interfered
RFID	<0.01m [114]	Low	High positioning accuracy easy to deploy	Need environmental transformation Multipath problem
UWB	<0.1m [115]	High	High positioning accuracy Anti-interference High multipath resolution	High cost

2.4.3 Path Planning

For humans, moving from one place to another is a simple task, almost instantaneously planning the route. For mobile robots, this basic task is a major challenge in the field of robotics. Path planning for robots involves finding a movement trajectory from the starting point of a task to its endpoint, based on environmental information. This trajectory must ensure the safety of the robot, the cargo it carries, and the surrounding environment, while aiming to minimize distance and time, and requiring the entire journey to be collision-free and without obstacles. The length of the path is a performance indicator used to measure the quality of path planning algorithms. Path planning for mobile robots is divided into two types based on the availability of environmental information [116]: global static path planning and local dynamic obstacle avoidance. Global static path optimization occurs when the surrounding environment is known and involves optimizing the path within a global static map model. Local dynamic obstacle avoidance is mainly for suddenly appearing obstacles, requiring modeling of obstacles and planning of detour routes.

2.4.3.1 Global Static Path Planning Methods

1. Dijkstra algorithm is a fundamental algorithm for finding the optimal path [117]. As a point-to-point shortest path algorithm, it calculates the shortest path length from a starting point to every other remaining target point in the graph, with distances represented by weights. Therefore, the Dijkstra algorithm is based on a directed weighted graph containing all target points, with a complexity of $O(n^2)$. As shown in Figure 18, let $G=\{V,E\}$ be a directed weighted graph, where V has a set of vertices A, B, C, D, E, with the special vertex A in V being the source point, and for any edge e in E, the length of edge (e) is non-negative, meaning all weights in the graph should be non-negative. The algorithm ultimately aims to obtain the shortest distance from the source point A to each vertex and save it as a collection. First, the algorithm obtains the direct distance from the source point to each vertex as the initial collection, then calculates the distance between other vertices and their adjacent vertices. If the distance is shorter, it replaces the current one, and iterates continuously to obtain the shortest distances.

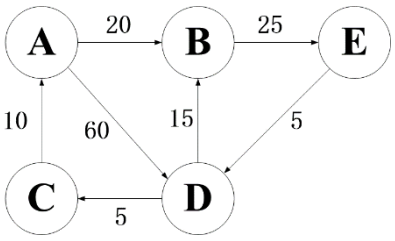


Figure 18 Dijkstra Directed Weighted Graph

2. Floyd algorithm is a classic path optimization algorithm [71]. It is suitable for solving the shortest distances between multiple source points and destination points. The Floyd

algorithm initially requires storing the distances between points in a distance matrix. Then, through three nested for loops, it traverses all point-to-point distances for optimization. This method can yield the optimal path, but the time complexity reaches O(n^3), where n is the number of endpoints. In other words, as the map expands, the distance matrix grows geometrically, and the computational load increases dramatically. Its working process can be explained as follows:

First, initialize two square matrices, where the distance matrix records the distance between each pair of nodes, and the sequence matrix stores the intermediate nodes between the source and destination nodes. Then, update the sequence matrix by considering any nodes as intermediate nodes that exist in the path between the first node and the destination node. The updating process is repeated to find the intermediate nodes between each pair of nodes.

3. Best-First Search Algorithm is a heuristic-driven search method that uses a heuristic function to estimate the cost from the current node to the target node. Compared to the Dijkstra algorithm, it prioritizes nodes that appear to be closer to the goal during the search process, which can often speed up the search, especially in larger maps. However, if the heuristic function is not appropriately chosen, this algorithm does not necessarily guarantee finding the optimal path [118].

4. A* algorithm is the most widely used heuristic pathfinding algorithm for mobile robots at present. It integrates the optimal path-finding feature of the Dijkstra algorithm with the heuristic characteristics of the BFS (Best-First Search) algorithm [119]. While ensuring the optimal path is found, it also guarantees the speed of the search. The cost function of the A* algorithm is given by Formula (2):

$$F(n) = G(n) + H(n) \tag{2}$$

Where $G(n)$ is the actual cost from the start point to the current point, and $H(n)$ is the estimated cost from the current point to the target point. When $H(n)$ is 0, the A* algorithm becomes the Dijkstra algorithm. When $G(n)$ is much smaller than $H(n)$, the A* algorithm becomes the BFS algorithm. Therefore, the A* algorithm has the characteristic of optimal pathfinding and has a significant time advantage, making it currently the most effective direct routing algorithm for solving the shortest path.

Intelligent algorithms are a class of algorithms that imitate biological evolution and the foraging or nest-building behaviors of natural animals, and they have now become the mainstream method for solving complex problems. Currently, commonly used intelligent algorithms include Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Artificial Neural Networks (ANN). Intelligent algorithms are suitable for solving and optimizing complex problems and have the potential for parallelism. However, these methods generally suffer from slow computation speed and the problem of

local optima.

ACO imitates the behavior of ants in nature, searching for food. Ants find the shortest path from their nest to a food source by releasing and sensing pheromones. In the algorithm, this mechanism is used to find the optimal path [120]. PSO is inspired by the flight behavior of flocks of birds. The algorithm optimizes paths by simulating the social behavior of birds or schools of fish. Each "particle" in the swarm represents a potential solution in the problem space, updating their positions by tracking and imitating the current optimal particle [121]. GA are inspired by the biological evolution process and evolve populations through selection, crossover or mating, and mutation operations to find the optimal solution. Individuals in each generation represent potential path solutions [122]. ANN are commonly used for pattern recognition and predictive modeling, they can also be used to find optimized paths, especially in continuous space optimization problems [123].

2.4.3.2 Local Dynamic Obstacle Avoidance Methods

Artificial Potential Field method is widely used in local path planning. It establishes an artificial virtual force field, assuming the robot is a mass point within it, moving under the influence of forces in the entire artificial field. The attractive force field is generated by the destination, and the repulsive force field is generated by obstacles in the environment. The attractive and repulsive fields superimpose on each other, exerting a force on the robot mass point and forming a movement curve. The trajectory of the curve is calculated using the gradient descent algorithm. This concept was originally proposed by Khatib intending the control of the movement of robot arms, avoiding adjacent obstacles, and planning the arm's motion [124]. The artificial potential field method is a simple and very effective path planning strategy with short trajectory planning time. At the same time, this method has some drawbacks; it may encounter trap regions, especially in environments with dense obstacles, resulting in oscillating trajectories. Moreover, the target may become unreachable when the destination and obstacles are very close [125].

Dynamic Window Approach (DWA) samples within a velocity space defined by a dynamic window [126]. In 1997, Fox et al. proposed the DWA for local collision avoidance [127], which utilizes the kinematics of a mobile robot to infer its path. It assesses AGVs using an evaluation function considering constraints, then obtains the optimal linear and angular velocities. Other mobile robot path planning algorithms typically adopt the idea of planning a path and computing velocity commands. In contrast, the DWA algorithm optimizes the search for the best trajectory based on real-time translational and rotational speeds recorded by the odometry, continuously updated with environmental and odometric information through a rolling window. It calculates paths based on the kinematics of the mobile robot. Finally, it rates the paths and selects the optimal one. This effectively ensures the feasibility of the mobile robot's velocity commands and reduces the possibility of local collision avoidance.

The concept of the Rapidly exploring Random Tree (RRT) algorithm is to quickly expand a

tree-like collection of paths to explore most of the space and find a feasible route. It is based on growing a tree composed of random nodes to explore the space and can efficiently discover paths to the target area [128]. The RRT algorithm is a random sampling method for the state space that avoids the computational complexity of precise space modeling by conducting collision checks on the sampled points, effectively solving path planning problems in high-dimensional spaces and with complex constraints. Its main advantage is speed, but it does not guarantee the optimal path and is more suited for local obstacle avoidance.

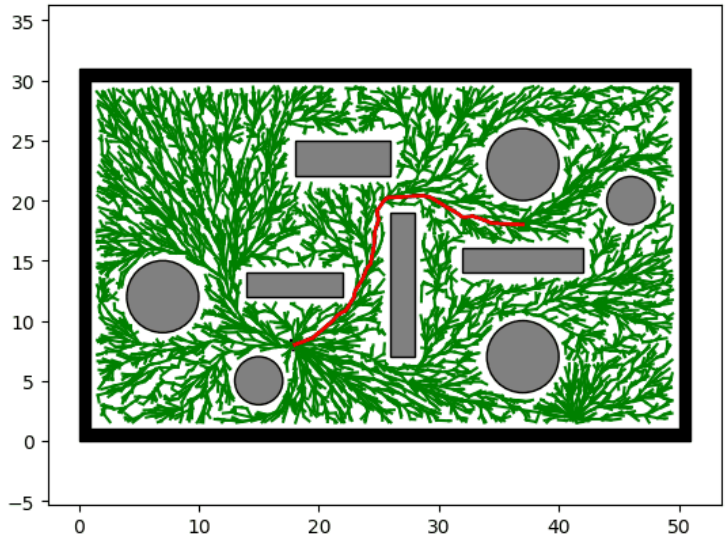


Figure 19 RRT Path Planning, green line is the tree, red line is the path [108]

There are some improvements to the RRT algorithm. The RRT* algorithm continuously optimizes by increasing sampling points after the basic path is generated by RRT, so that RRT has the characteristics of progressive optimization, but the convergence speed of this method is slow [128]. Bi-RRT expands two trees simultaneously from the starting point and the end point. When the two trees meet, the path planning is completed. This method can further improve the search efficiency [129]. Li et al. proposed ARRT-Connect based on the optimization of Rapid Exploration Random Vine and RRT-Connect. This method can handle narrow channel environments well [130].

2.4.3.3 Disscusion

In the field of path planning, the Dijkstra and Floyd algorithms provide traditional solutions for global static path planning, both based on known map information. The Dijkstra algorithm is suitable for point-to-point shortest path finding, while the Floyd algorithm is used for path optimization between multiple sources and destinations. These methods are effective when map information is constant, but their computational load increases dramatically as the problem scale grows. For local dynamic obstacle avoidance, faster and more adaptive algorithms are required. Intelligent algorithms offer robust performance but require extensive time and complex computation processes to find the most accurate values for fitness functions and

various other algorithm parameters.

The A* algorithm, as a heuristic search method, provides an effective mechanism to balance path optimization and search speed. It combines the optimal pathfinding characteristics of the Dijkstra algorithm with the speed advantages of heuristic search, making it the preferred method for global static path planning. The A* algorithm uses a heuristic function to predict the cost from the current node to the target, making it more efficient in large and complex environments than the Dijkstra algorithm.

For local dynamic obstacle avoidance, the DWA provides an effective solution for mobile robots. DWA utilizes robot kinematics to evaluate potential motion paths in real-time and selects the optimal linear and angular velocities. Compared to other planning algorithms, DWA directly optimizes path selection based on the robot's motion capabilities and current environmental information, effectively ensuring the feasibility of mobile robot speed commands and reducing the possibility of collisions. Additionally, the real-time and adaptive nature of DWA enables it to quickly respond to dynamic changes in the environment.

In summary, combining the global path planning of the A* algorithm with the local obstacle avoidance strategy of DWA can ensure optimal pathfinding while enhancing the adaptability and response speed of mobile robots in complex, dynamic laboratory environments. Therefore, using the A* algorithm as the global planning method and DWA as the local obstacle avoidance method is a solid and effective solution for mobile robots in laboratory environments. Table 3 shows a comparison of 8 path planning methods.

Table 3 Comparison of Path Planning Algorithm Disadvantages missing in this tables

Method	Description	Features	Applications
Dijkstra Algorithm	Point-to-point shortest path algorithm using directed weighted graphs.	Simple and ensures optimal path, but computationally intensive.	Global static path planning in known environments.
Floyd Algorithm	Optimizes shortest paths between multiple points using a distance matrix.	Effective for multiple source-destination pairs but has high computational complexity.	Path optimization across multiple points in static maps.
Best-First Search Algorithm	Heuristic-driven search method that prioritizes nodes closer to the goal.	Can speed up search with proper heuristics but may not find the optimal path.	Heuristic-driven pathfinding in large maps.
A* Algorithm	Heuristic pathfinding algorithm combining Dijkstra's optimality and BFS's speed.	Balances optimal pathfinding with speed, highly effective in complex environments.	Global static path planning in complex environments.
Intelligent Algorithms	Algorithms mimicking natural behaviors like ACO, PSO, GA, and ANN.	Handles complex problems with potential for parallelism, but slow computation.	Complex problem-solving in dynamic environments.
Artificial Potential Field	Uses virtual force fields to guide the robot by combining attractive and repulsive forces.	Simple and effective, but may encounter issues in dense obstacle environments.	Local dynamic obstacle avoidance in simple environments.

DWA	Real-time local collision avoidance using velocity and kinematic constraints.	Real-time adaptive response, ensuring feasible paths and reducing collision risk.	Local dynamic obstacle avoidance with real-time kinematic adjustments.
RRT Algorithm	Expands random nodes in a tree-like structure to explore the space and find feasible paths.	Fast exploration of high-dimensional spaces, though not guaranteeing optimal paths.	Local dynamic obstacle avoidance in high-dimensional spaces.

2.5 Infrastructure for Mobile Robotics

The creation of a relevant infrastructure is required to integrate mobile robots into complex laboratory environments. The systems must be able to open doors and identify the status of doors on their path (open, closed). In multi-story buildings, elevators are a key component of mobile robot transportation systems. Since most mobile robots used for transportation are incapable of climbing stairs, ensuring that robots can autonomously navigate within a building and perform transportation tasks requires precise knowledge of elevator status and floor information. Achieving this goal mainly relies on three steps: sending an elevator call signal, identifying the opening and closing states of elevator doors, and determining the current floor of the elevator. This process depends on data input from sensors and accurate recognition by algorithms.

2.5.1 Door Status Detection

Considerable research has been conducted on door detection to execute robotic navigation tasks. Common identification methods for mobile robots include infrared [137], ultrasonic [138], lasers [139], and visual systems [140], or by receiving direct signals from the elevator system [141].

Infrared and ultrasonic sensors function similarly, emitting signals and detecting their reflection off surfaces. The advantage of infrared sensors lies in their low cost and effectiveness in detecting clear boundaries, such as when a door is fully closed or open. However, their reliability may diminish under strong ambient light or if the door's reflective surface interferes with the sensor's ability to accurately detect its status. Ultrasonic sensors (approximately 5m) boast a broader detection range than infrared (less than 1m) and can operate under various lighting conditions, including darkness. While not highly accurate over long distances (the error could be a few centimeters), they are sufficiently precise for detecting the opening and closing of elevator doors. Often, infrared and ultrasonic sensors are used in combination [142].

Laser sensors provide high precision and range, making them suitable for exact measurements. They can determine the status of a door even in well-lit environments and over long distances. However, the higher cost of LIDAR systems and their sensitivity to reflective surface interference may be limiting factors. Jordi et al. proposed an elevator operation method based on 2D LIDAR, focusing on the detection of door states, defining elevator doors as three virtual rectangular areas to check for closed, opening, and fully opened statuses. Nevertheless, this

approach did not include floor detection, leading to incidents in the experiments where mobile robots were directed to incorrect floors. In fact, distinguishing between floors solely with 2D LIDAR is challenging since the structural layout of each floor in multi-floor buildings is often similar [139].

Vision-based systems, utilizing cameras, and advanced image processing, are highly adaptable and can provide detailed environmental information. They are capable of detecting the status and movement of doors, not just whether they are open or closed. The main challenges for visual sensors include their dependence on sufficient lighting conditions and the need for complex algorithms to accurately interpret visual data, particularly given the real-time processing demands of mobile robots [143].

Receiving direct signals from the elevator system could be the most reliable method to determine its status, as it provides real-time data without the need for environmental interpretation. However, this approach requires a compatible and accessible elevator control system, which may not be available or permissible for integration in all buildings. Sun et al. developed an elevator door operation monitoring system, using CAN bus data to determine door status and current floor levels. This method offers a universal application without the need for elevator manufacturers to disclose their protocols [141].

2.5.2 Floor Level Estimation

The mainstream solutions for floor detection primarily involve two methods: air pressure sensors and visual-based systems [144]. Additionally, Beacon technology, which operates on the principle of proximity, is also utilized for this purpose [145].

Air pressure sensors determine the floor level by measuring air pressure, which has enough resolution to distinguish between consecutive floors. However, the absolute air pressure values change daily due to weather conditions, showing significant variation throughout the day. In contrast, the air pressure difference between adjacent floors tends to be relatively stable. Srisura et al. developed a method for floor detection in parking structures using a smartphone barometer based on air pressure differences [146]. Similarly, Zhao et al. used air pressure differences to ascertain floor information [147]. To mitigate the impact of daily air pressure fluctuations, methods like moving averages and Finite Impulse Response smoothing filters are applied to preprocess the pressure data [138].

For visual-based floor detection, cameras and computer vision algorithms are employed to interpret visual cues within the elevator environment. This often involves recognizing numeric indicators on elevator panels or identifying unique visual patterns or landmarks specific to each floor [148]. The most commonly used techniques for reading floor numbers displayed on elevator control panels include Optical Character Recognition (OCR) [149] and the YOLO [150] algorithm. Robots equipped with arms can even press elevator buttons themselves [151]. Advanced implementations utilize the power of deep learning to enhance the robustness and

accuracy of floor detection [152]. Visual systems must adeptly compensate for varying lighting conditions and potential obstructions. The computational resources required for image processing also need to be considered.

Recent advancements in robot-elevator interaction underscore the potential for automating transportation within complex structures like multi-floor laboratories. Efforts to bridge communication gaps between material handling robots and elevator IoT systems are vital for enhancing the efficiency of cross-floor transport. Considering the need for real-time responsiveness and the complexity of varying lighting conditions, the system proposed in this paper utilizes simple and reliable ultrasonic and air pressure sensors for door status detection and floor level estimation, respectively. These sensors are integrated into an IoT module, which communicates in real time over the laboratory's internal network, ensuring efficient and seamless operation.

2.6 Laboratory Workflow Management Systems

The practical implementation and application of mobile robots in the workflows of life science laboratories are emphasized. Most life science labs employ Workflow Management Systems (WMS) [131]. These systems are crucial for automating and managing lab workflows, which include task scheduling, resource allocation, and process sequencing. WMS enhance operational efficiency by ensuring that laboratory tasks are performed in an optimized order, minimizing waiting times and maximizing resource utilization. They also provide tools for real-time monitoring and adjusting workflows, which is critical in dynamic environments where conditions or priorities may change rapidly. Commercially available WMS such as CrelioHealth Laboratory Workflow Management Software, Confience, and Thermo Fisher SampleManager specialize in integrating complex laboratory workflows with high precision and adaptability.

Process Control Systems (PCS) are classic systems used for controlling and monitoring industrial production processes, primarily focusing on the automation, control, and optimization of transportation, workflow, and resource consumption [132]. In a laboratory setting, PCS can be applied to control and monitor the operation of lab equipment and the parameter adjustments in the experimental process, ensuring the experiment's accuracy, stability, and repeatability [133]. Compared to traditional PCS, laboratory-based PCS require more flexibility and adaptability to meet the diverse needs of different labs [134], such as remote labs [135] or virtual labs [136].

Integrating mobile robots into automated laboratories requires more than just experiment record-keeping; it also necessitates a top-level control system for workflow planning and process control, as well as visual monitoring. This involves harmonizing the features of LIMS, PCS, and Instrument Control Systems (ICS). Moreover, effective integration between upper-level and lower-level systems is essential. The upper-level system handles planning, managing,

and monitoring task execution, distributing sub-tasks to various lower-level systems. In mobile robotics, the lower-level systems are responsible for receiving transport tasks and providing real-time feedback on the robot's status, and task progress, and robot navigation detail. Other lower-level systems may include different devices and sensors. This two-tiered relationship ensures task decoupling, allowing each part to operate independently while still being coordinated in a unified manner. This structure is a key form of automation in laboratories.

Gu et al. proposed a novel scheduling method using a Genetic Algorithm (GA) to optimize workflow management in life science laboratories. The Hierarchical Workflow Management System (HWMS) is developed to effectively coordinate both stationary and mobile systems within these labs. Their system aims to integrate traditional PCS with a robust transportation management system involving both automated robots and human assistants. This approach addresses the need for efficient resource allocation and seamless operation among automated workstations, which are vital for improving throughput and operational stability in complex, distributed lab environments [127].

Neubert et al. explored the integration of laboratory execution systems (LESs) to simplify the application and effectiveness of business process management systems (BPMSs) in managing complex laboratory automation processes. The paper highlights how LESs can reduce the number of interfaces between BPMS and various subsystems and simplify complex process modelings, thereby lowering the overall integration efforts required for complex laboratory workflows. This approach is particularly beneficial in settings where high levels of automation and precise process control are required, such as in life science laboratories [130].

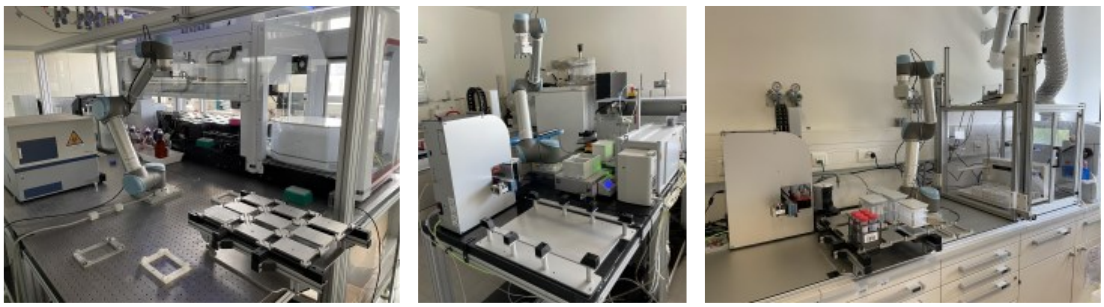
3 Chapter 3 General Concept

This chapter aims to provide a detailed overview of how mobile robots and automation integrate within intelligent laboratory environments. It comprehensively introduces the MOLAR mobile robot hardware and associated software elements. The discussion spans from the top-level multi-layer control systems for overall laboratory processes to multi-mobile robot control systems, and further details the software for elevator interactions.

3.1 Automation in Life Science Laboratories

Similar to many life science laboratories, the CELISCA lab is equipped with a range of advanced scientific devices capable of conducting experiments in areas such as automated cell processing, compound and element specific analysis, cellular and enzymatic screenings, protein processing, diagnostics, or material development. However, like most labs, a significant amount of work in the CELISCA lab remains manual. While the lab has integrated several fully automated workstations, these are often isolated systems with varying degrees of automation. Some stations consist of single automated devices, others are partially automated for specific process steps, including labware and sample storage. Despite these advancements, the transport of materials between different stations often remains manual.

As shown in Figure 20, within the automated workstations, various independent instruments are integrated and scheduled by an automation system, with robotic arms performing the transfer of labware and reagents within these stations. However, when it comes to transporting materials between stations—especially when the stations are distributed across different rooms or floors in multi-story buildings—manual intervention is still required. Transportation can be automated using conveyors when the stations are in close proximity. For more flexibility and transport over longer distances or between floors, mobile robots can be employed, offering a fully automated solution for moving labware and samples across the laboratory.



a) b) c)

d) e)

Figure 20 Automation Stations in CELISCA, a) Bioanalytical Laboratory; b) Biomedical Sample Separation; c) Elemental Analysis; d) Mass Spectrometry Laboratory; e) Crystallisation Laboratory. a), b), and c) are located on the floor 2, while d) and e) are on the floor 3.

The objectives of life science laboratories are to improve processing quality and throughput while minimizing monotonous tasks and reducing the risk of potentially hazardous setups for lab assistants. These improvements also aim to achieve overall cost reduction [153]. Compared to the manufacturing industry, automation in laboratories is generally less developed due to the high cost of implementation, which requires designing a comprehensive solution tailored to each lab's specific characteristics. CELISCA seeks to achieve process automation among all workstations or individual devices in laboratories. To achieve process automation across all workstations and individual devices in laboratories. To address this need, a significant outcome of is the development of a comprehensive workflow management system.

To achieve a higher degree of automation integration, the lab has an array of robotic arms. Dual-arm robot Yaskawa, single-arm robot UR, and humanoid robot H20 are used to assist with the transportation of labware inside workstations. The H20 mobile robot is a humanoid mobile robot independently developed by the laboratory around 2010, used to explore mobile robot technology, more research-oriented, with less efficiency in transportation, and is described in detail in section 2.3.1.

The CELISCA laboratory is located within a multi-story building. The labs are

distributed across different rooms on different floors. The laboratory is equipped with automatic doors and an elevator to connect different spaces, and it has an internal WiFi covering all areas for real-time communication. Since the laboratory's elevator is a closed-system elevator that can only provide the function of being called to the destination floor and does not offer current floor information or elevator door status, this information is crucial for mobile robots taking the elevator. Therefore, the software Virtual PLC was developed in combination with an IoT module to solve this problem.

In the past, CELISCA used the H20 humanoid robot for transporting labware. The H20 robot already had basic functions such as navigation based on StarGazer, obstacle avoidance, and arm-based labware handling. However, this work was more research-oriented, and the efficiency in practical applications was not high. For example, its movement path was a series of straight lines, and it had to stop at each turning point before changing direction. It could only carry two labware items at a time. The arm motors were weak, and if the labware was too heavy, the arm would shake, reducing the success rate of handling. The obstacle avoidance function was not smart enough and could not effectively avoid dynamic obstacles. Therefore, a new ASTI mobile robot, specifically designed for transportation, was introduced.

3.2 Requirements

To build a fully automated laboratory environment, mobile robots have been introduced to replace manual labor for transporting labware and samples. To integrate these mobile robots into the intelligent laboratory environment. This necessitates a set of specific requirements:

- **Transport of Samples/Labware:** Due to the diversity of life science experiments, the laboratory handles various types of labware and samples. The system must accommodate multiple formats, ensuring stability and safety during transport.
- **Delivery of samples / labware:** Multiple workstations in the laboratory have different specifications requiring precise placement and retrieval. This necessitates tailored design parameters for each station's mobile robot telescopic conveyor, including height, extension length, and approach distance.
- **Positioning Accuracy:** High positioning accuracy, with a precision requirement of 0.1 mm, is critical to ensure the accurate placement and retrieval of materials. This prevents spills and ensures the accuracy of subsequent operations by robotic arms at workstations.
- **Movement in Narrow Areas:** Robots must navigate narrow spaces, as laboratories are not as spacious as factories and consist of rooms filled with complex equipment, including areas as narrow as 90 cm.

- **Inter-Laboratory Navigation:** Navigation capabilities must extend across different laboratory rooms and floors, involving the operation of automatic doors and elevators to traverse areas.
- **Swarm Functionality:** The system should support robot swarms to enhance throughput and flexibility in handling samples, including swarm scheduling in narrow areas.
- **Safety Features:** Considering the shared spaces of humans and robots, the system must include safety features to prevent accidents and ensure a secure working environment.
- **Central Control System Integration:** To fully incorporate mobile robots into the laboratory environment, integration must extend beyond physical capabilities to include software systems. The robots must be integrated with the laboratory's top-level workflow management system, making them an integral component of the laboratory's operations and connecting different stations. This allows for real-time monitoring of robot status and facilitates communication between various systems within the laboratory.

3.3 Hardware Concept

In the development of mobile robot for laboratory environments, CELISCA evaluated two configurations: a mobile platform equipped with a robotic arm and a pure transporter mobile robot. The first configuration integrates a classic robot arm on a mobile platform that handles both the transportation between stations and the manipulation of samples and labware. This approach, while consolidating transport and handling functions into a single system, introduces complexities such as decreased positioning accuracy when mobile and limited operational range due to the arm’s reach. This configuration necessitates intricate programming for the robot arm's multiple degrees of freedom and relies heavily on additional technologies such as barcode recognition and image recognition systems to maintain functionality.

On the other hand, mobile robots as pure transporters: This alternative concept uses mobile robots solely for transporting labware between stations, without integrating a robotic arm. Handling tasks, such as transferring labware between the robot and the station, are managed by separate systems at each station. This configuration offers several advantages: mobile platforms without robotic arms are more cost-effective, less complex, and thus more reliable. By decoupling transportation from handling, the mobile robots can cover longer distances and service larger systems without becoming a bottleneck. This approach also allows for greater flexibility in designing transfer

stations and integrating various devices.

Given these considerations, CELISCA decided to implement the second configuration—mobile robots as pure transporters. This decision was driven by the need for flexibility, cost efficiency, and the ability to scale the system without introducing time delays or accuracy issues in handling. By focusing on dedicated transfer stations and separating the transportation and handling functions, the lab can achieve a higher degree of automation integration while maintaining reliability and scalability.

3.3.1 MOLAR Robot Body

The integration of mobile robots into laboratory environments is driven by the need to automate repetitive labware and samples transportation tasks, enhance precision, and increase efficiency. Mobile robots such as the ASTI robot (ABB Ltd., Zurich, Switzerland), designed specifically as an AGV for item transportation, are equipped with advanced navigation and dedicated transport capabilities to meet these demands. They play a crucial role in transporting labware and samples between various stations, reducing the workload on human staff and minimizing the potential for human error. Their design addresses specific laboratory requirements such as maneuvering through narrow spaces, operating in multi-floor settings, and interfacing seamlessly with existing laboratory information systems.

ASTI robot is equipped with the Telescopic Conveyor option is an advanced mobile robot designed for autonomous transportation of laboratory items. It uses laser radar to map the laboratory environment. At the same time, the laser radar is used to scan the environment in real time during operation to locate itself. It also has the ability to avoid static and dynamic obstacles. It communicates with the host computer through WIFI to obtain transportation commands and send real-time status.

It mainly features a telescopic conveyor for lifting loads from frames and racks that the vehicle cannot reach by driving under them. The robot is equipped with sensors like wheel encoders, a gyrometer, a laser scanner, charging cradle sensors, current sensors, and battery cell sensors, ensuring precise navigation and safety. Its mechanical components include a drive train with differential drive and freely rotating wheels, and a lift column with a spindle drive for load lifting. The ProBOT-L is characterized by its adaptability to various tasks and environments, making it a versatile tool in automated logistics and handling operations.

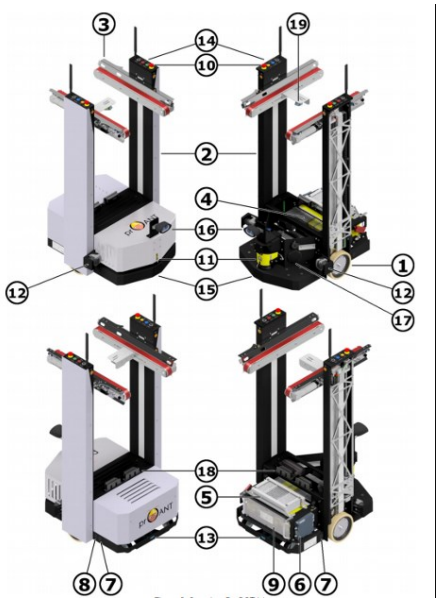


Figure 21 Componets of MOLAR Robot, 1) Drive train 2) Lift columns 3) Load suspension device 4) PLC 5) Core PC 6) WLAN Module 7) LAN Socket 8) USB Socket 9) Battery 10) Emergency stop button 11) Safety laser scanner 12) Side scanner 13) Reverse scanner 14) Illuminated buttons 15) Underbody lighting 16) Bluespot 17) Speaker with amplifier 18) Sensors for load transfer 19) scanner

With the telescopic conveyor, loads can be lifted from frames and racks which the vehicle cannot reach by driving under them. Instead, it stops when it faces the frame and extends the telescoping conveyor after a successful precise positioning. With the telescopic conveyor, the load gets lifted from the frame and pulls the conveyor with the small load carrier back. It drives away only after the conveyor is pulled back and the lifting device is lowered. To avoid tilting while picking and dropping the load, the vehicle is equipped with a counterweight. The telescopic conveyor is equipped with a pressure sensor on the front. If the sensor is activated, the horizontal movement of the conveyor stops to reduce a possible risk of crushing. Furthermore, the telescope is connected with the lifting device of the AMR by a pivot point. If the extending telescope comes across obstacles, it gets lifted. After a few millimeters, the vertical proximity sensor sends a signal and stops the lifting movement.

The MOLAR robot has dimensions of 737 mm x 630 mm x 1,515 mm and weighs 85 kg, with a telescopic conveyor increasing the weight to around 150 kg. It can handle a maximum load of 50 kg, though with the telescopic conveyor, the load capacity is limited to 35 kg. The robot's lift range allows for transfer heights between 440 mm and 1,100 mm, with a lifting speed of 0.035 m/s. It can drive at a maximum speed of 1.2 m/s, with a reverse speed of 0.3 m/s. Navigation accuracy is within ±50 mm, while fine positioning accuracy is within ±10 mm. The MOLAR robot can navigate passable edges up to 5 mm and gaps up to 3 cm, and it can handle floor slopes up to 3°. It is powered

by an 8-cell LiFeYPO4 battery system with balancing boards and temperature control, providing 24VDC and 40Ah.

The main task of the navigation control is the vehicle localization. The vehicle must know its position at any time. Our mobile robot systems use existing contours to localize the vehicles. A SICK S300 Expert (SICK AG, Waldkirch, Germany) is mounted in the front of the AMR and serves as safety laser scanner for personal safety and AMR navigation. It allows detecting obstacles and personal within a viewing field of 250°.

While commissioning, a map of the environment is registered. To this purpose, an AGV is driven manually through the layout. During this scanning phase, the vehicle records the actual laser scanner data for each movement. Doing so, it is able to create an environment map.

During operation, every vehicle is placed once in the map. From this moment, the vehicle uses data from different sensors to recognize its position. Following sensors are used:

- Laser scanner: The laser scanner measurement data provide a feedback of the environment of the AGV. Making comparisons between the scanned map and the current reported environment data, the vehicle can calculate its position.
- Odometry: The vehicle motors are monitored. Through the number of the wheel revolutions, the vehicle can determine its position.

Figure 22 SICK S300 expert safety laser scanner

The differential wheel system is the mobile hardware component. The classical design and functionality of differential wheels allow the robot to perform precise movements and turns by varying the speed of each wheel independently. The mobile robot uses two drive wheels and four caster wheels. The drive wheels are the primary components that control the robot's movement, providing the necessary propulsion and steering by adjusting their speeds. Meanwhile, the caster wheels support the robot's weight and provide stability, allowing it to smoothly navigate and maintain balance without directly influencing its direction or speed. The specific structure is shown in Figures 23 and 24.

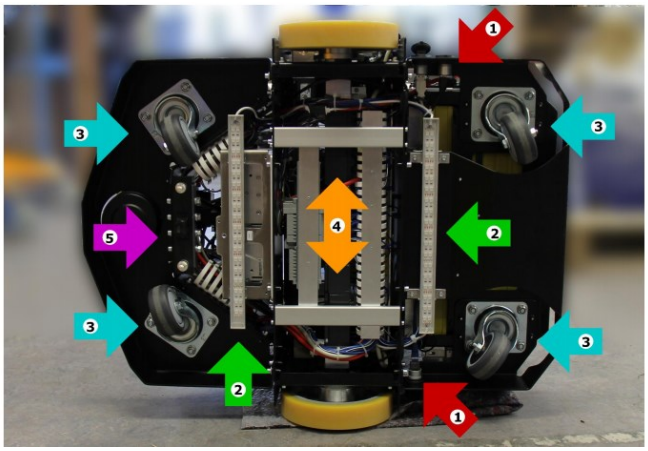


Figure 23 MOLAR robot chassis, 1) USB/LAN socket; 2) LED bar; 3) Support wheels; 4) Motors; 5) Charging contacts and charging cup sensors

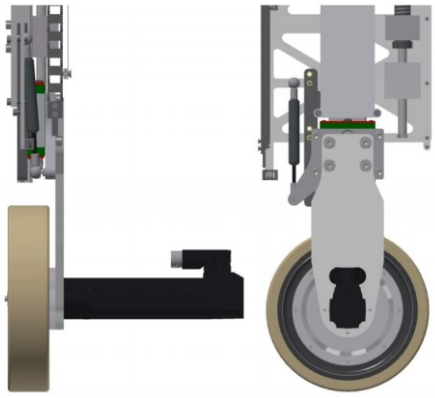


Figure 24 Drive components of MOLAR robot

3.3.2 Interface Robots

Unlike mobile robots, Interface robots are fixed within the workstation, surrounded by all the equipment, tools, laboratory instruments, and reagents. Although many laboratory automation systems now provide fixed conveyors or simple mobile units, these are not sufficient for workstations that integrate multiple devices and perform complex operations. There is a need for robotic arms to carry out tasks such as transferring and pouring. The placement of these arms must be carefully designed to ensure their working radius can cover all operational areas.

Some robotic arms are equipped with cameras or other sensors, allowing them to perform tasks that require precision and adaptability, such as adjusting the position and orientation of lab vessels, measuring the volume of liquids, or identifying labels and barcodes on samples. This sensory feedback enables the robots to autonomously correct their actions, thereby enhancing the accuracy and safety of the laboratory environment. Some robotic arms also have the capability for human-machine interaction, assisting

lab technicians with operations and effectively combining human expertise with robotic precision.

In CELISCA laboratory , mobile robots are used only for transporting labware between different stations. At each station, interface robots, mainly robotic arms, are needed to handle the labware. These robots perform tasks such as picking up, placing, and transferring labware from one process to the next. The interface robots at CELISCA, shown in Figure 25, include different types of robotic arms. These robots are placed at specific workstations to handle the labware that the mobile robots bring. For example, robots like Yaskawa and UR arms are used to move labware accurately and interact with other lab equipment.

These interface robots help ensure that the labware is handled correctly at each step, making the workflow smooth and continuous. Without these interface robots, the automation process would not be as effective.

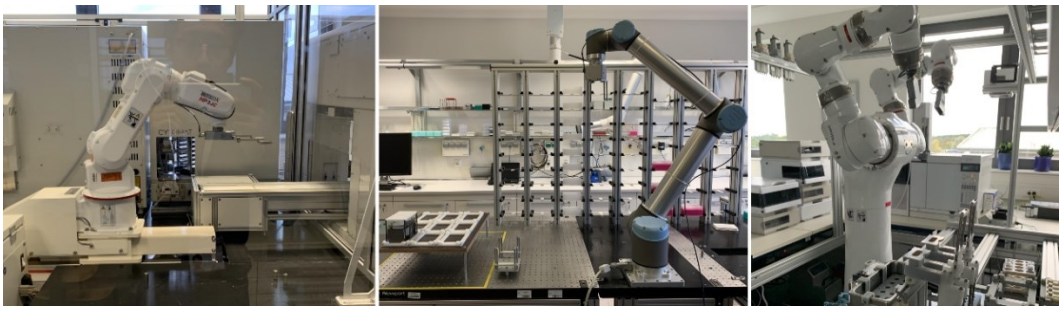


Figure 25 Interface Robots in CELISCA

3.3.3 Infrastructure Module

In CELISCA, the closed-system elevator cannot provide feedback, which poses a problem for mobile robots that need information about elevator doors and floor levels during their operation.To solve the problem of mobile robots using elevators, the IoT module is developed to detect the status of the elevator. The IoT module is installed on the mobile robot to help it recognize floors and detect whether the elevator doors are open or closed. Previously, the laboratory used IoT modules equipped with gas sensors to monitor gas levels in life science laboratories. The module is designed to detect and report hazardous gas levels, providing real-time data to help prevent gas leaks and other potential dangers [154].

To address elevator identification issues, the Infrastructure Module comprises four parts: the Microcontroller Board, which is used for managing control logic and processing sensor data; the Communication Board, which transmits data to the upper control system via WiFi and acquires BLE beacon signals through Bluetooth; the Sensor Board, which collects distance and air pressure data; and the Power Board, which supplies power. Specific parameters and models are as follows [155]:

- Microcontroller Board: This development board is powered by an MKL27Z256VHL4 microcontroller, which features a 32-bit ARM Cortex-M0+ core running at 32 MHz. It supports both the I2C (Inter-Integrated Circuit) and SPI (Serial Peripheral Interface) protocols, known for high performance and low power consumption. These characteristics make it suitable for managing simple sensor connections and functions.
- Communication Board: This board includes the ESP-WROOM-02D WiFi module and the Aconno nrf52840 BLE Module. The WiFi module transmits data to the higher-level systems. The BLE module is capable of receiving signals from BLE Beacons, which may be used for floor positioning in future implementations.
- Sensor Board: It comprises sensors for detecting gases like VOCs, CO_2, and TVOC, as well as sensors for air pressure, humidity, and temperature. An ultrasonic sensor is also included. The IoT module is additionally employed for monitoring gases within life science laboratories to prevent hazardous gas leaks.
- Power Board: The module can be powered through a USB connection from the mobile robot or by batteries.

The IoT Module communicates with the upper control system over the laboratory's internal WiFi network. The upper control system sends UDP Broadcasts to the IoT modules, which include the return address and required sensor information. The IoT modules then send data packets back to the upper control system using the HTTP protocol at intervals of 1000 ms. The laboratory also maintains a specialized database to store data collected from all IoT modules. Although the upper control system can retrieve data from this database, direct communication was chosen for real-time considerations because retrieving data from the database involves first storing data there, and then retrieving it, which introduces a 2-3 seconds delay. Direct communication eliminates this intermediate step, saving time and enhancing real-time response.

3.3.4 BLE Beacon

BLE beacons, which are procured (Moko Smart, Shenzhen, China), the remaining components are independently developed and designed by CELISCA. The BLE beacon uses the iBeacon protocol based on Bluetooth 4.2, with an advertising interval of 0.1s and a Tx power range from +4 to -40 dBm. It is lightweight with only 25.7 g and can be easily attached to laboratory walls using magnets. Its lifespan is over 30 months, as shown in Figure 26.

Figure 26 BLE beacon

BLE beacons are used to assist mobile robots with floor determination inside elevators. A BLE module is installed at the elevator doors on each floor. When a robot equipped with an IoT module reaches one floor, it uses the RSSI principle to identify the beacon with the strongest signal as the current floor. Future plans include equipping trays with IoT modules to track their real-time location. BLE beacons are also installed on various laboratory doors to pinpoint whether a tray is in a particular room or corridor.

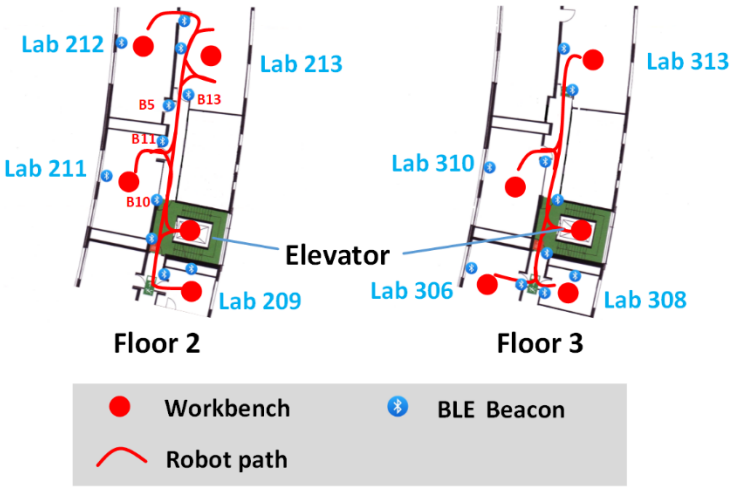


Figure 27 Beacon distribution in the laboratory

3.4 Software Concept

This section primarily discusses the structure and concepts related to the software associated with MOLAR robots. The overall system structure of the laboratory is multi-layered. First, there needs to be a central system that oversees and coordinates the entire automation process of the experiments. Once the specific tasks are planned, the control software for the mobile robots must receive the instructions, assign tasks to the robots, and monitor their status. This software is also responsible for communication with the surrounding equipment that interacts with the robots. During the operation of the mobile robots, the software must control navigation, assist with path planning, and handle

obstacle avoidance. Since the laboratory has multiple floors, a middleware is designed to manage communication between the mobile robots and the elevators. Overall, the software system ensures that the entire automation process runs smoothly and efficiently across different levels and tasks.

3.4.1 System Structure

This section primarily discusses the integration of mobile robots into a multi-level workflow management system designed for life science laboratories. It covers task allocation at the top level and the implementation of distributed devices at the bottom level. Mobile robots act as the link, connecting labware, materials, and various laboratory devices to create a highly automated lab environment. The framework is depicted in Figure 28. To achieve comprehensive automation in the laboratory, the integration of software systems is essential. The embedded mobile robot system architecture is shown in Figure 28. The system is divided into two layers: a top-level workflow management layer and device integration layer.

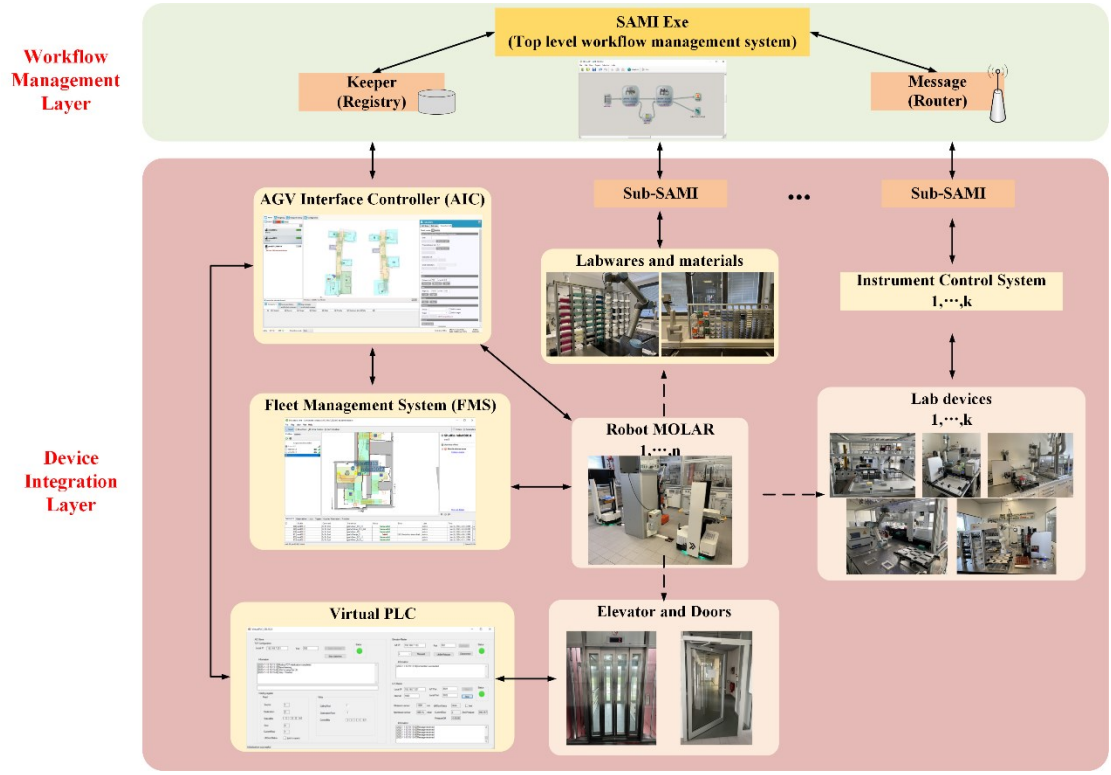


Figure 28 Structure of the multi-level workflow management system in life science laboratory

To create a cohesive, integrated network within highly distributed laboratory environments using mobile robots, a comprehensive workflow management system is essential. Traditional automated production lines are often not agile enough to adapt to the complex and dynamic requirements of a scientific lab. To address this gap, CELISCA has developed a workflow management system built upon the SAMI

platform. This advanced system connects all experimental devices within the laboratory, requiring only that these devices have some form of communication interface. Importantly, there are no stringent specifications or open-source requirements for the equipment. During an actual experiment, lab personnel merely need to prepare materials and labware in designated storage areas, then input the required sequence of devices and actions into the system. The system autonomously generates optimal workflows for each experiment, encompassing the entire process from raw material transport to final sample analysis. Moreover, it provides real-time status monitoring for each step of the experiment and collects feedback from the current devices in use. If a fault occurs, the system can promptly issue an error alert. In Figure 28, AIC and FMS are software developed by the robot company. AIC is responsible for global communication and coordination. FMS handles robot positioning and navigation functions. The Virtual PLC is developed by us as middleware to assist mobile robots in using the elevator. Integrating all automation devices into a comprehensive automation system is our goal to achieve a future-lab.

A hierarchical structure is particularly beneficial in automated laboratories, especially those with complex experimental workflows [134]. This is even more critical when dealing with a multitude of interconnected, heterogeneous automation systems that require frequent status checks for various material conditions (such as temperature, concentration, or sample analysis) [156].

3.4.2 Communication System

To ensure communication between different functional modules, devices, and software levels, integrating mobile robots into laboratory automation requires a unified and building-wide internal network. This network infrastructure is primarily supported by network sockets.

3.4.2.1 Robot WiFi Requirements

The connectivity of each MOLAR robot within the FMS system relies on a dedicated Wi-Fi module, typically a Siemens Scalance W734-1 RJ45. This module ensures the robot can connect to the laboratory's WiFi network and manage the data traffic across various components of the Smart Shuttle. To guarantee optimal performance, specific WiFi standards and conditions must be met:

- WiFi Standards: The system supports IEEE802.11 a/n, which facilitates efficient network communication.
- Security Standards: To safeguard data integrity and network security, the system

adheres to WPA2-PSK security standards.

- WiFi Channel: Utilization of the 5 GHz spectrum is mandatory, with channels dedicated exclusively to Smart Shuttles to avoid cross-traffic and interference from other channels.
- Signal Requirements: A minimal signal-to-noise ratio (SNR) of 25dB is required throughout the Smart Shuttle's operational area to ensure reliable communication.
- Signal Strength: The system requires a minimum signal strength of -75dBm, measured at the WiFi module of the shuttle, to maintain connection stability across the entire driving area.

- Network Configuration: Each shuttle is assigned a visible SSID for easy identification, with WPA2 encryption for security and one static IP per Smart Shuttle to streamline network management.
- These requirements are critical for maintaining a robust and secure wireless network environment, enabling the Smart Shuttles to perform efficiently and reliably in various operational scenarios.

3.4.2.2 TCP/IP Network Framework

Network sockets are endpoints in a communication flow across a computer network and serve as a foundational technology for network and internet communications. Sockets enable communication between two different processes on the same or different machines. To achieve this, sockets utilize Internet Protocol (IP) and Transport Control Protocol (TCP) or User Datagram Protocol (UDP), among others. TCP provides a reliable connection-oriented service, ensuring that data packets arrive in sequence without loss or error, making it ideal for applications where precision is crucial, such as data transfers between critical laboratory systems. On the other hand, UDP offers a connectionless service with lower latency, which is very useful for broadcasting that requires high speed and efficiency but has less reliability. This forms the backbone of the TCP/IP Network Framework, which includes not only the physical hardware connections but also the standardized communications protocols for data exchange across the network.

CELISCA has set up an IEEE 802.11g wireless network for laboratory transportation. This network provides an appropriate wireless communication environment with adequate bandwidth and fast data channel areas for all developed hardware components and systems. Manual testing has confirmed that WiFi signal values in all areas where mobile robots operate are above -60dbm, and all locations have at least two routers

covering them to ensure there are no signal interruptions or blind spots during robot movement. Communication protocols such as Modbus TCP for industrial automation, standard TCP for general communications, HTTP for web services, and UDP for broadcast services have been employed for communication with mobile robots and their surrounding equipment and software. These ensure that different systems within the laboratory environment can effectively communicate, share data, and coordinate operations.

3.4.3 Fleet Management System

IncubedIT serves as the GUI for the Fleet Management System (FMS), primarily responsible for mapping, localization, and navigation of mobile robots. It also facilitates state checks of the mobile robots, logging errors, and adjusting various algorithm parameters. The GUI includes the following main features:

1. Map Visualization and Editing: The FMS-GUI provides a detailed visual representation of the operational environment, allowing users to view the layout of the facility, including locations of charging stations, pick-up and drop-off points, and traffic areas. Users can edit and customize the map to reflect changes in the environment or operational requirements.One of the core functionalities of the IncubedIT map editing is the ability to define and manage traffic areas within the operational map. This feature is crucial for coordinating the movements of multiple AGVs, ensuring they navigate efficiently and safely without collisions or traffic congestions.
2. Safety Precautions: The FMS emphasizes the importance of safety in the operation of AGVs. It provides rules for safe usage, including adherence to safety standards, regular maintenance schedules, and protocols for emergencies. Diagnostics and Error Handling: The FMS-GUI includes diagnostic tools for monitoring the health and performance of the AGVs. It provides real-time data on each vehicle's status, enabling prompt identification and resolution of errors or malfunctions.
3. Traffic Management: Comprehensive overview of order management, including order cancellation, chaining, parallel orders, and assignment algorithms, ensuring efficient task allocation and execution. It could discuss how the system manages shuttle driving behavior, including speed and navigation parameters, to optimize movement within the operational environment. Details on estimating task durations and fine-tuning shuttle positioning for precise operation.
4. System Configuration: Describes the global software stack configuration, system parameters, OPC-UA Server setup, persistence configuration, and more, providing

a foundation for customizing the system to meet specific operational needs.

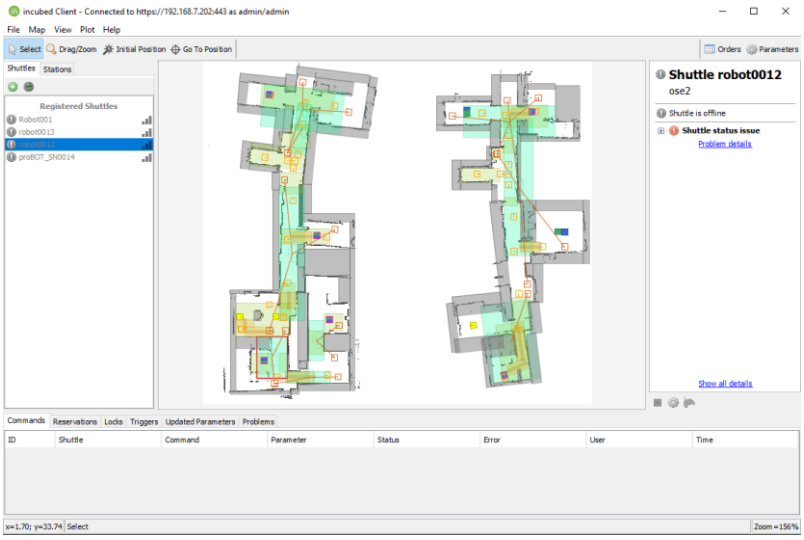


Figure 29 FMS GUI

3.4.4 AGV Interface Controller

The AIC serves as the key middleware connecting SAMI, workstations, MOLAR robots, and peripheral devices such as elevators and doors. The AIC receives transportation requests directly from Global SAMI via HTTP protocol and provides real-time feedback on the current task status of the mobile robots. Additionally, manual intervention through the AIC's software interface can generate corresponding transport tasks as needed. Figure 30 shows the AIC GUI, and Table 4 displays the functions of each area in the GUI.

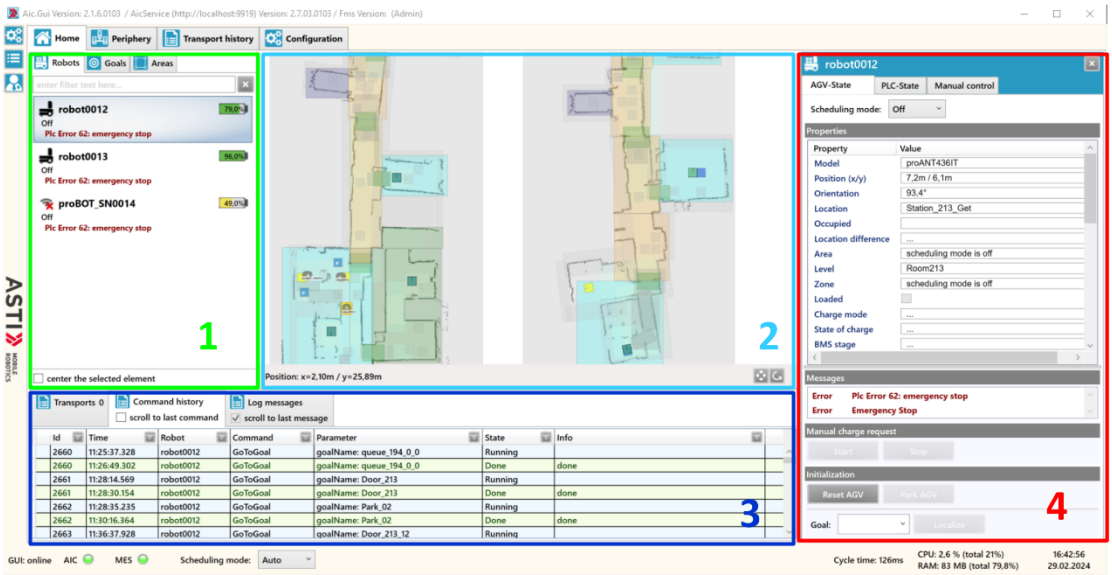


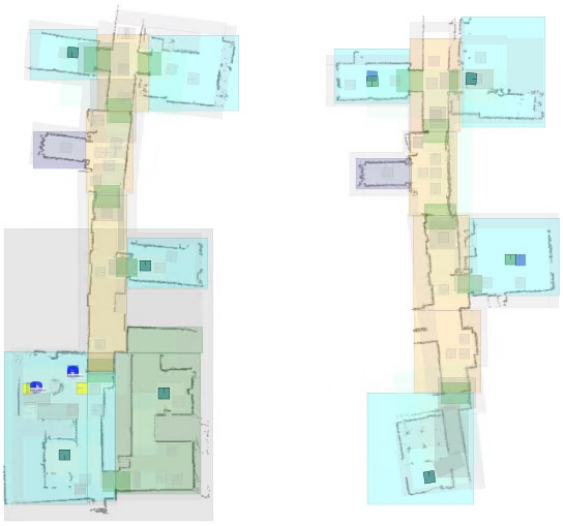
Figure 30 AGV Interface Controller

Table 4 Structure of AGV Interface Controller

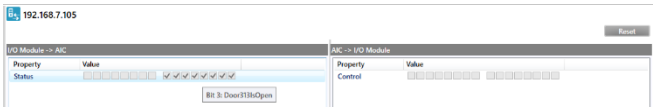
No.	Feature	Description
1	Selection menu	Allows a selection between the tabs
	Robots	List and state of the connected AGVs
	Goals	List and state of all the goals
	Areas	List and state of the created areas
2	Map	Shows the currently scanned map with all goals, areas and AGVs. The configured path is also displayed on the map.
3	Transports	Shows a list of all transport orders not yet completed
	Command history	Shows a list of all commands sent to the AGVs
4	Detail overview	Overview of the selected tab in the selection menu

The AIC intelligently determines which robot should handle a specific task and in what sequence, based on factors like battery level and distance to destination. Comprehensive details about the transport process, including task specifications, timing, and energy consumption, are logged for subsequent data analysis and evaluation. The AIC's capabilities are further enhanced by a real-time map presented in Figure 31, which provides a visual monitoring interface for tracking the current locations and statuses of all mobile robots, as well as the real-time communication statuses of automatic doors and elevator within the system.

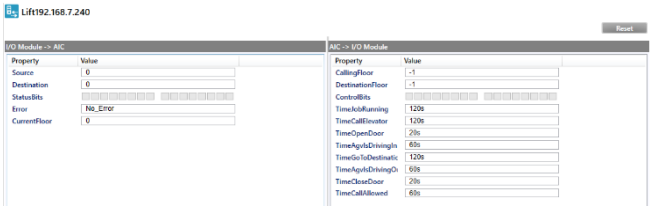
The visual monitoring interface offers a real-time view of all mobile robots' current positions and statuses within the laboratory, as well as the connectivity status of automatic doors and elevators. This includes real-time task statuses and a sortable list of all tasks, which can be manually adjusted via the GUI interface. Communication protocols between automatic doors and elevators with the AIC are all conducted via ModbusTCP. The interface also allows for the monitoring of all automatic doors' open/close statuses, current floor information for elevators, elevator door statuses, and target floors. These pieces of information are essential for facilitating interactions between mobile robots and elevators. As the elevators within CELISCA are not open-source, IoT technologies is employed to acquire current floor and door statuses.



a)



b)



c)

Figure 31 Status visualization in AIC system, a) Visual map of CELISCA, blue parts are robots; b) Communication with automatic doors; c) Communication with the elevator

3.4.5 Virtual PLC

During the elevator usage process by mobile robots, three functions need to be realized: 1. Calling the elevator; 2. Detecting when the elevator doors open to enter; 3. Detecting arrival at the target floor to exit. Unlike humans, mobile robots cannot directly observe the elevator doors opening nor visually confirm the arrival at the target floor through floor indicators. Since the elevators at CELISCA are closed systems, they can only be called to specific floors and provide no additional feedback. Therefore, alternative methods are required to obtain the necessary information for the mobile robots to use the elevators.

Initially, modifying the elevator system was considered, which would require authorization from the elevator company and entail significant costs. A vision-based

method using cameras to detect the numbers on the dashboard representing the current floor was also considered. However, our elevators are semi-outdoor, and sunlight on clear days poses a challenge to the vision solution. Additionally, the elevator dashboard where the floor numbers are displayed is positioned high (2 meters), whereas the mobile robots are 1.5 meters tall, adding further complexity. Another considered approach was to place large numeric labels or QR codes opposite the elevator doors on each floor. The issue with this solution is that the camera's view could be blocked by people or objects inside the elevator, and this method would require alterations at every elevator entrance, affecting the aesthetics.

Ultimately, an IoT module-based solution was developed using ultrasonic sensors to check the state of the elevator doors and an air pressure sensor to determine the floor level. To facilitate the use of elevators by mobile robots, a virtual PLC program was designed. The Virtual PLC is a GUI application developed in C# using WinForms. It acts as middleware between the AIC and the elevator system, allowing interaction between the mobile robots and the elevators. The primary purpose of the Virtual PLC is to address the closed-system elevator issue. It has a modular design that integrates communication with the AIC, Elevator, and IoT module, as well as the logic needed for elevator control and determining the floor level based on air pressure.

The Virtual PLC displays real-time information about the interaction between the mobile robot and the elevator, including the status of the elevator doors and the floor level as detected by the IoT module. It communicates over the laboratory’s internal network using protocols such as TCP/IP, Modbus TCP, HTTP, and UDP broadcast, allowing real-time updates and feedback. This makes it easier for personnel to monitor and operate the system.

This solution does not require any modifications to the existing elevator system, making it adaptable to any building's elevators. Additionally, the IoT module required for this setup is inexpensive, making this approach both cost-effective and widely applicable. More details about Virtual PLC are in Section 5.3.

3.4.6 Workflow Management System

In automated laboratories, the WMS plays a crucial role. Serving as the top layer of the automation architecture, this system manages and controls the automated equipment throughout the laboratory, ensuring that tasks are executed efficiently and in the correct sequence. By integrating various devices and processes, the WMS ensures that experiments are performed accurately, thereby reducing manual errors and increasing throughput.

The primary function of the Workflow Management System is to orchestrate complex workflows involving multiple tasks and stages. It schedules and prioritizes tasks based on predefined algorithms that account for resource availability, task urgency, and optimal usage of laboratory equipment. This centralized management enables coordination across different laboratory devices, facilitating a streamlined workflow that can adapt to changing research needs and experimental conditions. Furthermore, the WMS provides a user-friendly interface for scientists and lab technicians to input experimental protocols and monitor progress. This interface allows users to easily adjust parameters, start or pause experiments, and receive notifications about the status of ongoing processes or maintenance needs.

In CELISCA, SAMI system represents a specialized implementation designed specifically for life sciences environments. SAMI operates on two levels: the global SAMI EX and sub-SAMI modules. The global SAMI EX acts as the central command center, directing overall laboratory operations and integrating data across various platforms. It oversees the entire experimental process from setup to completion, including data collection, analysis, and storage. This level of control ensures that experiments adhere to strict scientific standards and protocols, enhancing reproducibility and reliability.

Each sub-SAMI module corresponds to specific laboratory equipment or function, such as centrifuges, PCR machines, or cell culture systems. These modules are tailored to manage the peculiarities and technical requirements of each device, allowing for precise control over experimental conditions. They communicate with the global SAMI to receive instructions and send back real-time updates on their status and the progress of experimental tasks.

SAMI EX is split into two parts： 1. Method Editor to create methods and schedules; 2. Run Time to run schedules. SAMI EX Editor provides a graphical interface for defining and editing complex workflows within laboratory environments. This interface enables transitions between different laboratory tasks, facilitated by mobile robots which are represented as arrows linking various operational nodes within the workflow. SAMI Run Time provides a real-time visualization of the workflow progress. This interface displays the current status of all tasks, robot positions, and any alerts or notifications related to the process. Users can pause, stop, or modify the workflow in response to unexpected events or changes in the experimental setup. This feature is essential for maintaining flexibility and responsiveness in a high-throughput automated laboratory setting. Additionally, the Run Time component logs all activities, offering detailed reports and analytics that help optimize future workflow designs and improve operational efficiency. This continuous feedback loop between the execution and

planning stages of SAMI ensures that the system adapts and evolves in line with the laboratory's operational needs and scientific goals.

3.5 Robot Navigation

The robot's navigation system mainly concludes map building, localization, and movement within the laboratory environment. Navigation is crucial for mobile robots as it enables them to autonomously move through their environment, avoid obstacles, and reach specific destinations. Effective navigation systems allow robots to perform tasks with precision and reliability in the laboratory. To prevent secondary development, the built-in navigation algorithms of the used robots were used, which are not open-source. The robot core is based on the ROS platform, and the entire navigation system's FMS system operates under Linux Ubuntu 20.04.

3.5.1 Mapping

Map building involves establishing a framework for the specific work environment that mobile robots can understand. It is the foundation for all movements of mobile robots. This environmental framework includes fixed structural features such as the walls, doors, rooms, and elevators of the CELISCA laboratory, and it also adds semantic information based on the point cloud map obtained from 2D LIDAR, which includes divisions of different functional areas.

3.5.1.1 Gmapping Algorithm

Lidar-based SLAM is a common method for map building. The MOLAR robot provides various sensors required for SLAM methods, such as odometer measurements needed by IMU, LIDAR distance measurement data, etc., which can be used to explore unknown environments. In this project, the Gmapping method is used to implement the SLAM algorithm in a laboratory setting.

Gmapping, proposed by Grisetti in 2006, is one of the most refined laser scanner-based SLAM algorithms [157]. This algorithm uses a Rao-Blackwellised Particle Filter (RBPF) as an efficient method for solving the SLAM problem and can be utilized within the ROS system. Gmapping requires odometry data to estimate position and heading. The odometry can be improved by using encoders on the left and right wheels of the mobile robot, and accuracy can be enhanced with a single-axis gyroscope. The Gmapping mapping process is illustrated in Figure 32. It shows how Gmapping takes input from laser scan data and various transformations to generate a map. Here’s a breakdown of the inputs:

- Laser Scan Data (/scan): This is the raw data from a LIDAR sensor that provides distance measurements surrounding the robot.
- Dynamic Transforms (/tf): These are dynamic coordinate transformations that relate the positions and orientations of various components on the robot, which change over time.
- Static Transforms (/tf_static): These are transformations that do not change over time, such as the fixed offset between a sensor and the base of the robot.

These inputs are processed by the Gmapping SLAM algorithm to produce a map of the environment, typically published on the /map topic. This map can be used for navigation and environment interaction tasks.

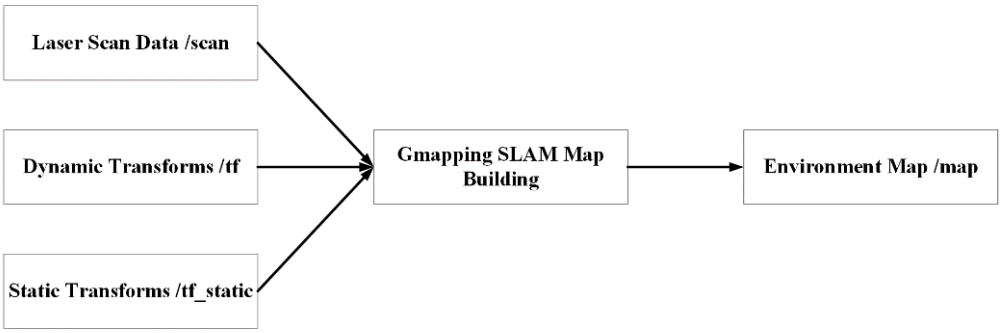


Figure 32 Gmapping Mapping Process

The Rao-Blackwellized method leverages the concept of conditional independence to factor the joint posterior probability of the robot's trajectory $x_{1:t}$ and the map m given the observations $z_{1:t}$ and motion $u_{0:t}$. This results in a factorized posterior distribution, which simplifies the computational process. As shown in Formula (3):

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t}) = p(m \mid x_{1:t}, z_{1:t}) p(x_{1:t} \mid z_{1:t}, u_{0:t}) \tag{3}$$

Where m represents the global map, $z_{1:t}$ are the observations, $u_{0:t}$ represents the motion data, and $x_{1:t}$ are the robot's trajectory at different times, which also includes the robot's pose.

The RBPF commonly uses the SIR (Sampling Importance Resampling) particle filtering algorithm. The process of this algorithm is as follows:

1. Initialization phase: Initially, the mobile robot's starting pose is set, and the LIDAR begins scanning to collect environmental data. Particles are initialized based on the state transition function, and these sampled particles are used to compute the

posterior probability through weighted sums.

2. Update phase: Based on real-time data from within the mobile robot, the weights of the sampled particles are updated. The weight of a sampled particle represents the probability of choosing a corresponding particle that predicts the pose of the mobile robot, i.e., the more likely a sample is obtained, the greater its weight.

3. Resampling phase: Particles are replaced according to the distribution of weights. In the next cycle, the swapped particles are fed into the state transition equation to compute and generate new predicted particles.

4. Map building: The robot's pose and the environmental map are updated using the predicted particles along with the odometer and trajectory data of the mobile robot. The global map is then updated based on the pose and LIDAR scanning data.

5. Repeat steps 2-4: This step ensures that as time progresses and more data is accumulated, the accuracy of the estimation continues to improve.

In the ROS system, Gmapping is a convenient toolkit for parameter adjustments and modifications. The Gmapping package primarily subscribes to the /tf and /scan topics. The /tf topic handles transformations between the sensors, wheel pairs, and the odometer coordinate systems of the mobile robot. The /scan topic is published by the LIDAR and consists of scanning data. The Gmapping algorithm then fuses the data from the /tf topic with the data from the /scan topic to create a SLAM map, which is published on the /map topic. This establishes an environmental map based on a grid. The map is stored as a YAML file and an image file using the map_server package, facilitating the use of the map in subsequent navigation processes.

In the actual Gmapping process, the configuration of parameters significantly affects the outcomes. Here are 10 key parameters selected for adjustment. These parameters need to be tailored according to the specific applications and the operational environment of the MOLAR robot to optimize performance and map quality. Proper tuning of these parameters can significantly impact the output and real-time performance of the SLAM system. These Gmapping parameters provided in Appendix B1.

3.5.1.2 Map Building Processes

For the actual map building process of the MOLAR robot in the CELISCA laboratory, a front-mounted SICK S300 Expert LIDAR (SICK AG, Waldkirch, Germany) for environmental scanning was used. Maps are created using the Gmapping algorithm, which processes the point cloud data gathered by the LIDAR. The software used for this process is IncubedIT. The specific steps are as follows:

1. Launch the IncubedIT client and select a robot to start the map scanning operation.

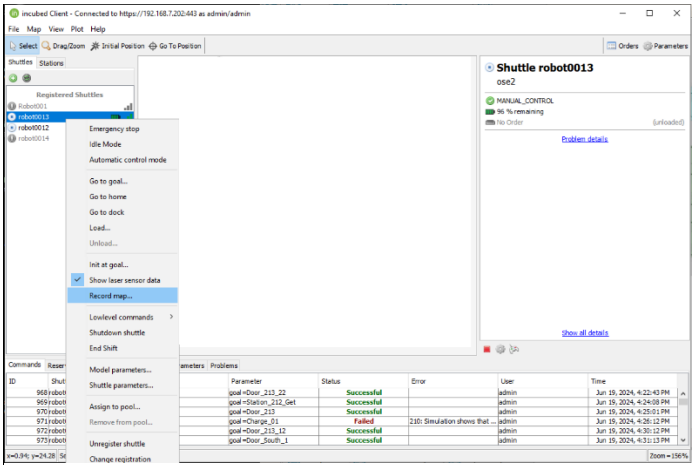


Figure 33 Select a robot to start map scanning

2. The robot is manually moved by personnel to various areas of the laboratory for scanning. Alternatively, the robot can be moved using a joystick or the control functions within the client software. As the robot moves, the point cloud map is simultaneously constructed. Once the map is complete, it is saved.

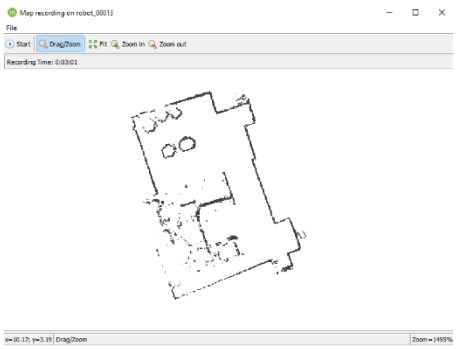


Figure 34 Scan Point Cloud Map of Room 213

3. With the point cloud map established, map editing can begin. This includes adding waypoints, defining regions, and setting functional areas. Detailed methods of map editing will be discussed in Section 4.1.1.3.

4. If certain areas are not properly mapped or changes occur, those areas can be rescanned and the map can be updated accordingly. For example, if new equipment is added to the laboratory and alters the original layout, the old map may no longer be suitable. In such cases, only the affected room needs to be remapped by repeating steps 1-3. The newly scanned data will then be integrated into the existing map to reflect the changes, ensuring that the map remains accurate and up-to-date.

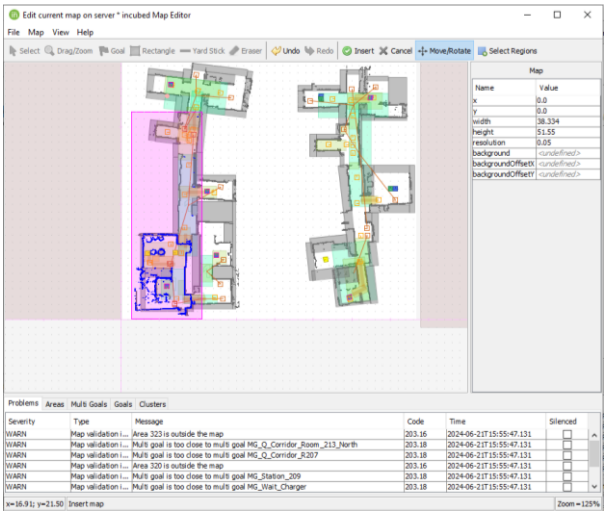


Figure 35 Insert the scan map to adjust the existing map

5. Finally, the edited map is uploaded to the server. The completed map will be updated in the IncubedIT and AIC GUIs. After positioning the mobile robot accurately using IncubedIT's localization feature, normal transportation operations can commence.

Figure 36 Final CELISCA 2D scan map

3.5.1.3 Semantic Map Design

The integration of the MOLAR robot into the CELISCA laboratory layout notably involves the utilization of semantic maps. As described in 4.1.1.2, MOLAR robot utilizes a LiDAR to acquire point cloud information and construct a 2D geometric map of the CELISCA environment, accurately capturing the contours of the indoor space. This geometric map can be applied for robot localization, obstacle avoidance, and path planning functionalities. Many classic mobile robot solutions operate in this manner.

However, to further optimize the behavior patterns of the mobile robot, semantic information was added to the existing geometric map, integrating human perception of the environment into the point cloud map. This enables MOLAR to achieve a higher level of understanding of the CELISCA laboratory. The semantic map provides MOLAR with a semantic understanding of the environment, going beyond the geometric layout and achieving a fusion of semantic layers in the environment. This

enables more intelligent decision-making and interactions within its surroundings. This multi-layered approach to mapping is visually represented in Figure 37, which illustrates the progressive enhancement of the laboratory's environmental model through semantic mapping.

Figure 37 shows the step-by-step semantic design process of the map. First, a 2D map of the environment is created based on LIDAR, as shown in section 4.1.1.2 map building processes. It is a grid-based occupancy map. It shows basic obstacle relationships and is the most commonly used map type for mobile robots in the past. Initially, the 2D geometric layout, captured by a laser scanner, identified the basic structural features of the laboratory, including walls, doors, and open spaces. However, due to architectural elements like glass walls, certain discontinuities appeared in the map. Building upon this, the map editing functionality of IncubedIT is used to segment the 2D map into areas based on their function, distinguishing rooms, corridors, and elevator areas. This classification helps better organize space and plan the robot's movement within the laboratory. In conjunction with modifying the AIC configuration files, connectors such as automatic doors and elevators are set between areas, enabling the mobile robot to automate door opening or elevator use as it moves through different zones. Finally, more detailed functional areas are added: green areas are narrow spaces, numbered areas represent areas restricted to a single robot, and yellow paths are predefined routes. The cyan squares mark pick-up and drop-off points, while the yellow hollow squares are waiting points for single-robot areas.

"Areas" refers to sections designated on the map, each assigned specific functionalities, such as forbidden zones, one-way areas, or narrow spaces. These functional areas can overlap in the same physical space if there are no conflicts. Different functional areas are set up to display various features and functions, as detailed in Table 5.For example, the initial step in the mapping process segments the laboratory based on features such as automatic doors, creating distinct areas for rooms, corridors, and the elevator, each assigned a unique identifier. Movement between these areas relies on automated doors and elevators, necessitating commands for opening and closing doors, and calling elevators for inter-floor transport.

In this foundational segmentation, finer area definitions are imparted based on their specific characteristics. Examples of the most used areas include prohibited zones, which encompass walls, stairs, and non-laboratory spaces such as offices, kitchens, and restrooms. Narrow areas are designated where space is limited, prompting a reduction in the robot's speed to aid navigation through complex terrains. Single-robot zones are established to restrict the number of robots present simultaneously, with designated waiting points for additional robots requiring entry, thus resolving potential multi-robot interaction conflicts. Predefined routes are set within particularly tight and intricate

environments to streamline the robot's movement, minimizing computational load and enhancing transit efficiency.

a) b) c)

Figure 37 CELISCA 3rd floor semantic map, a) 2D geometric layout scanned by the LIDAR; b) marks regions based on functionality—rooms (light blue), corridors (yellow), and elevator areas (purple); c) Functional area segmentation

Table 5 Main areas and stations type used in the CELISCA semantic map

	Type	Description
Area	Forbidden	Area, in which no vehicle is allowed to move at all
	Single_Robot	In this area, only one robot is allowed to stay. Additional mobile systems have to wait outside.
	One_Way	Vehicles are allowed to move only in a predefined direction. The direction is chosen according to the orientation of the area. The vehicles are allowed to cross the area coming from different directions.
	Right_Hand_Traffic Left_Hand_Traffic	In this area, right-hand traffic rules apply. The vehicles must drive on the left or right side of the area
	Prefered_Line	The vehicles in this area try to drive as close as possible to the midline of the area. In case of obstacles on the driveway, the vehicles try to bypass them as close as possible and then to get back to their path.
	Emergency_Escape	This element defines an area, where during an emergency (such as fire) no vehicle is allowed to stop. If a vehicle is in this area when a fire alarm is triggered, it moves as quickly as possible out of this area and stops there.
Station	Get/Put	Labware pick-up and drop-off station with fine-positioning device
	Charging	Charging station with fine-positioning device
	Parking	Stations for idle mobile robots to stand by
	Door_Waiting	The mobile robot arrives at the station and starts sending door opening instructions via WIFI

Below are the criteria and basis for dividing the semantic map areas for the CELISCA

laboratory map.

Forbidden area: Forbidden areas are used to restrict the movement of mobile robots. First, forbidden areas are set up in areas where mobile robots do not need to enter, such as office areas, rooms that do not require entry, kitchens, and non-operational spaces in some laboratory rooms. Next, for the safety of mobile robots, dangerous areas are restricted, including stair areas and glass wall areas (because the navigation uses LiDAR, glass cannot be detected correctly, and some areas of the laboratory are enclosed by glass). Lastly, forbidden areas are used to compensate for defects in the 2D point cloud map scanned by LiDAR, preventing the route planning process from extending beyond the building's boundaries. Adding some forbidden areas can improve the effectiveness of path planning, as mobile robots always tend to move in the middle of the road. Adding some forbidden areas can make the space more regular.

Narrow area: Life science laboratories do not have as much space as factories or warehouses. Many areas are narrow. Mobile robots cannot pass through areas less than 1.2 meters wide due to Safety Fields restrictions. The purpose of setting up a narrow area is two-fold: first, to reduce the speed of the mobile robot to the lowest to ensure safety through narrow areas. Second, to allow the mobile robot to pass through areas wider than itself but less than 1.2 meters, as Safety Fields are temporarily lifted in narrow areas. Therefore, narrow areas are set up in the laboratory's automatic door areas and narrow corridors.

Preferred Line: Preferred Line is used to assist mobile robots in path planning in complex environments. These complex environments are also narrow and are used in conjunction with Narrow areas. The laboratory's Preferred Line is set in the automatic door area to reduce the shaking of the mobile robot when passing through the automatic door area. Secondly, Preferred Line is set in room 313 because the corridor of room 313 is only 0.97 meters. After passing through this narrow corridor, there is a narrow turn immediately, which usually takes a long time to pass without the guidance of Preferred Line. So, the two sections of Preferred Line are combined to allow the mobile robot to smoothly reach the workstation in room 313. Preferred Line is also set in the charging area of room 213 to ensure that the mobile robot does not directly roll over the charger's dock (because the charger's dock is lower than 14 cm, which LiDAR can detect).

Waiting point: Waiting point serves as the docking station for mobile robots in AUTO mode. Life science experiments usually need to be carried out between multiple stations. Mobile robots need to wait while equipment is running at the workstation, and the ideal waiting point is to stay in place. Therefore, a waiting point is set near each workstation. In the robot 'Hotel', two waiting points are set near the charging pile to facilitate mobile robots to wait at the waiting point after charging.

Get/Put station: The setting of the get/put station is based entirely on the position of the workstation's rack. A get/put station is set 1 meter away from the workstation, and this 1-meter distance is used for fine positioning. During the fine positioning process, the mobile robot can use the triangle device to fine-tune the x, y coordinate, and direction, ensuring the accuracy of picking and placing.

Single-Robot area: Likewise, because most areas in the laboratory are small (theoretically more than 2.7 meters wide), they cannot accommodate two robots running at the same time. Therefore, the order of entry for mobile robots is restricted. When dividing a Single-Robot area, it is necessary to note: 1. This area can only accommodate one mobile robot moving. 2. The set waiting points should not interfere with the operation of another mobile robot in the same area. 3. Waiting points should be set at each connection to other areas. Single-Robot areas are mainly set in corridors, small laboratories, and workstation areas in large laboratory spaces.

3.5.2 Localization

Localization refers to the ability of a robot to establish its position and orientation within an absolute or relative reference frame using sensors. As the robot moves, the estimation of its location changes. Therefore, the robot must continuously update its position through active computation. The robot must know its current location to move on to the next point. The MOLAR robot, based on the ROS system, primarily uses the AMCL algorithm for localization, supplemented by Kalman filtering to process odometry and other sensor data, and clustering to handle LIDAR point cloud data. Special triangular devices are also installed at all sites to assist with high-precision positioning, further improving localization accuracy in handling labware.

3.5.2.1 Adaptive Monte Carlo Localization

Adaptive Monte Carlo Localization (AMCL) is a probabilistic localization method suited for two-dimensional environments and is the most commonly used and mature method within the ROS system. It is primarily used by mobile robots equipped with LIDAR sensors in environments where the map is already known. The AMCL localization process consists of four steps:

1. Particle Initialization: Particles are distributed evenly across the map or based on prior knowledge. Each particle starts with an equal weight.

2. Particle Motion Simulation: The robot receives control inputs from the odometer, such as turning and forward speed. The particle filter updates each particle's state based on these inputs and a predefined motion model. When the robot obtains new sensor information (such as from LIDAR scans), AMCL updates each particle's weight.

Weight updates are based on the degree of match between the predicted measurements of particle states and the actual measurements, typically calculated using a likelihood function.

3. Particle Evaluation: The estimated position of the robot is usually the weighted average of all particles, where the weights represent the probabilities of the particles. This step concentrates particles in high-probability areas to enhance the accuracy and efficiency of localization. The area with the highest concentration and highest scoring particles represents the most probable location of the robot at that moment. Adaptive techniques, such as KLD sampling, offer better nonlinear fitting than traditional Gaussian distributions and are more suitable for complex environments.

4. Resampling of Particles: Over time, some particles become very unlikely to represent the actual location of the robot and their weights become minimal. To maintain an effective particle set, AMCL regularly performs a resampling process to eliminate low-weight particles and duplicate those with higher weights.

In the ROS system, the AMCL algorithm allows for localization by using LIDAR to scan the surrounding environment on an existing map. Even after creating a map with Gmapping, that map is used for more stable localization with AMCL. The AMCL node subscribes to the /tf topic, /scan topic, and the /map topic published by Gmapping, as well as the /initialpose topic for initializing the particle filter to estimate the robot's position. The AMCL node publishes the robot's pose estimate on the /amcl_pose topic, the filter's pose estimate on the /particlecloud topic, and the /tf topic. It provides a transformation from the base_frame to the map_frame, enabling the conversion of the robot's pose from odometer-based localization to the map coordinate system. The structure is illustrated in Figure 38. The parameters of AMCL is in Appendix B2.

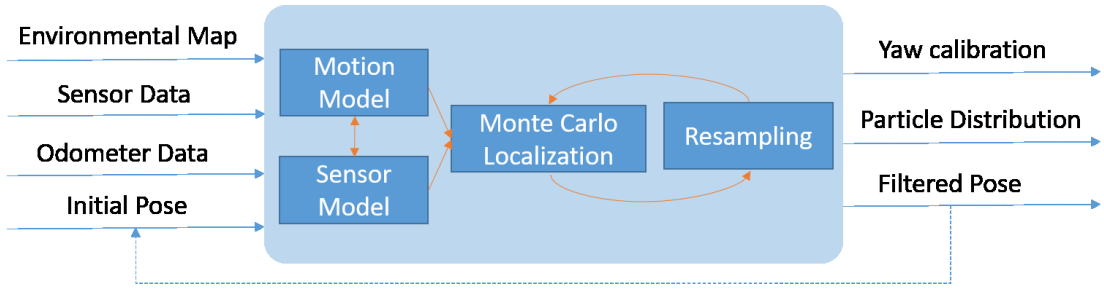


Figure 38 AMCL Process Flow Diagram

3.5.2.2 Fine-Positioning

In the context of mobile robots handling labware and automatic charging, precision is paramount. The required accuracy ensures a 100% success rate, preventing incidents like liquid spills, leaks, or charging mishaps. Due to the tolerances and inaccuracies of the sensors used, the vehicles typically achieve an accuracy of ±5 cm during free

navigation, which is insufficient for the precise material delivery at transfer stations. To address this, fine positioning occurs where a device called "Triangle" is mounted at each station. This setup, based on the AMCL localization algorithm enhanced with a laser scanner detecting the center of the triangle, allows for fine positioning with an increased accuracy of ±1 cm and ±1° towards it, achieving less than 1 cm of positioning accuracy essential for critical operations at these stations. Figure 39 shows the structure of the triangle device.

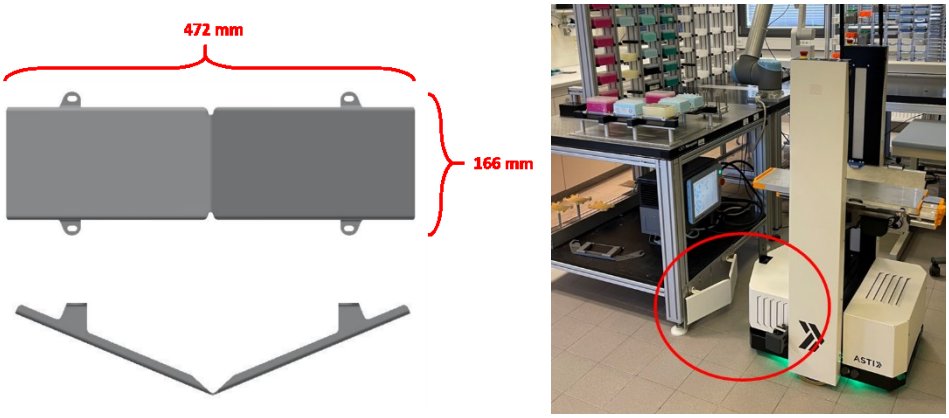


Figure 39 Triangle device for fine positioning

For the Fine-Positioning process using LIDAR and a triangular device the pick-up and charging stations are set one meter away, where the mobile robot enters the area and adjusts its orientation before starting Fine-Positioning. The triangular device is placed at Pick-up and charging station, serving as a prominent landmark. This device is designed to reflect laser signals due to its shape and angles, facilitating easy identification and positioning. Mounted on the mobile robot, the LIDAR device emits lasers and receives the laser signals reflected, which create a 2D image of the triangle device. As the robot moves, the LIDAR continuously scans the surrounding environment. The data captured by the LIDAR includes distances and angles to various objects, with the triangular device often being easily identifiable in the data due to its unique shape and reflective properties. The raw point clouds data obtained from the LIDAR is simplified using the Ramer-Douglas-Peucker Algorithm algorithm (RDP). This algorithm removes points that are not crucial for describing the shape, retaining only key turning points, which helps improve the efficiency and accuracy of subsequent processing. During this step, the edges, and corners of the triangular device captured by the LIDAR are identified as key features. By comparing the simplified point cloud with the known model of the triangular device, the system can accurately determine the position and orientation of the triangular device. Once the triangular device is accurately identified and positioned, the algorithm uses this fixed reference point to calculate the precise position of the robot. This involves calculating the relative position and orientation of the robot to the triangular device. The system continuously updates

the robot's positional information, adjusting the movement strategy based on changes in the environment. The positioning information is used for navigation decisions, ensuring the robot efficiently avoids obstacles and plans routes in complex environments.

3.5.3 Movement

One of the core functionalities of mobile robots is their ability to move within their environment. The MOLAR robot's movement is based on a differential wheel, and the algorithm controlling its movement is move_base in the ROS system.

3.5.3.1 Differential Wheel Control

The MOLAR robot uses a differential wheel structure, which includes two drive wheels and four caster wheels. The drive wheels are responsible for controlling the robot's movement by adjusting their speeds independently, allowing the robot to make precise turns and movements. The caster wheels, on the other hand, provide additional support and stability without directly influencing the direction or speed of the robot.

Since it can happen that the robot's navigation system requests the hardware to perform actions that are beyond its physical limitations, (accelerate/decelerate faster than possible, etc.) it is important to have a strategy implemented by the host that is consistent with the expectations of the navigation system. The motivation behind using a Drive Control System（DCS）is to align the real-world behavior of the robot with the theoretical motion planned by the navigation software. This is especially critical during acceleration and deceleration along curves, where without DCS, the robot's actual path can significantly deviate from the intended trajectory.

The DCS includes three primary components:

- Desired Wheel Accelerations Calculation - This part calculates the accelerations needed for each wheel to follow the desired path or curve accurately.
- Acceleration Limiter - This component ensures that the acceleration commands do not exceed the motor's physical capabilities, adjusting commands proportionally if needed.
- Wheel Speed Limiter - It adjusts the commanded wheel speeds to ensure they do not surpass the maximum capability of the motors, maintaining control over the robot's turn radius.

3.5.3.2 Navigation Control

Following the construction of the SLAM environment map using the Gmapping

algorithm, the mobile robot utilizes the AMCL algorithm to determine its current location within the map, enabling movement initiation. The MOLAR navigation system primarily relies on move_base within the ROS system, incorporating both global and local path planning.

In the navigation process, global path planning is initially conducted by creating a route from the start point to the destination point using the global environmental cost map derived from SLAM, employing the A* algorithm. Local path planning addresses dynamic obstacles that emerge in the local map during the robot's movement, such as people, tables, chairs, and boxes. It executes real-time obstacle avoidance operations and swiftly plans a safe, locally optimal avoidance path using DWA algorithm. Both types of obstacle avoidance planning are based on their respective cost maps: *global_costmap* for global static path planning and *local_costmap* for local dynamic obstacle avoidance. The global_costmap is generated by the Gmapping algorithm, while the local_costmap is produced from the current environmental map scanned by LIDAR. The cost map includes a grid map, inflation factors, motion trajectories, the cost associated with each cell, and the robot's URDF motion model. The cost maps are managed and invoked using the map_server package. During operation, the robot performs real-time updates to the cost map based on LIDAR scan data. The grid costs within the cost map are computed from LIDAR scan data, with each cell's cost probabilistically defined as one of three types: Occupied (obstacle area), Free (navigable area), or Unknown Space. The navigation system's framework is depicted in Figure 40.

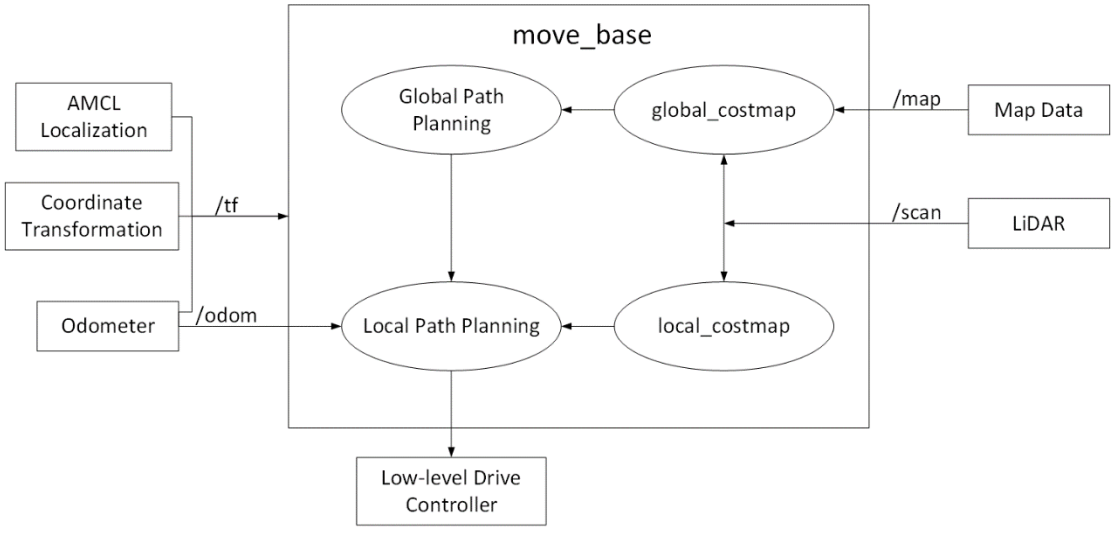


Figure 40 Move_base structure

In the ROS system, the DWA algorithm, based on the global path planning using the A* algorithm and the cost maps, plans local obstacle avoidance paths, including data

on pose and velocity. It directly commands the differential wheel servo motor controllers. Specifically, the DWA algorithm continuously collects real-time data of the mobile robot's x and y coordinates and pose throughout operation. Based on this data, as well as the robot's maximum/minimum speeds, maximum/minimum angular velocities, and acceleration, it simulates future movement states. Over a specified timeframe, it evaluates potential trajectories within a "window" defined in the DWA algorithm. Evaluation metrics primarily include the distance to obstacles, distance to the target point, adherence to the global path, and movement speed, ensuring no trajectory results in contact with obstacles. The optimal path determined by the evaluation is then sent to the differential wheel servo motors. The local cost map, unlike the global cost map, represents a local map that detects obstacles within the environment. The key parameters of move_base in the MOLAR robot are shown in in Appendix B3.

4 Chapter 4 Robot System Implementation

The integration of labware handling methods is discussed, emphasizing the design of trays and racks, as well as the get/put processes essential for successful labware transfer. The design of automated charging strategies is essential to maintain 24-hour operations. A well-planned distribution of charging stations facilitates the efficient movement of mobile robots in complex multi-floor environments. Finally, the chapter concludes with a detailed comparison between the MOLAR robot and the previously used H20 robot, highlighting performance differences in positioning accuracy, transportation efficiency, and charging capabilities. The obstacle avoidance and special area performance of the MOLAR robot were tested. This analysis aims to provide a comprehensive understanding of the practical applications and improvements achieved with the MOLAR robot system.

4.1 Labware Integration

All labware in the CELISCA laboratory adheres to a standardized microtiter plate format (MTP). This uniform format facilitates laboratory automation. Various types of tubes, which hold all kinds of experimental reagents, can be inserted into this standardized base. Typically, each station that stores labware and reagents or conducts experiments is equipped with a robotic arm used to transfer labware between trays and workstations. After experiments are completed, a workflow control system calls for the mobile robot to transport the tray, which contains the labware, to the next workstation. The design of the trays and the racks that support the trays at the workstations is crucial for ensuring safe transportation, considering the mobile robot's positioning accuracy in x, y coordinates and angular precision. A suitable get/put strategy is also a key factor in ensuring a 100% success rate. Figure 41 illustrates the integration results of the mobile robot at the workstation, where it loads the tray filled with labware.

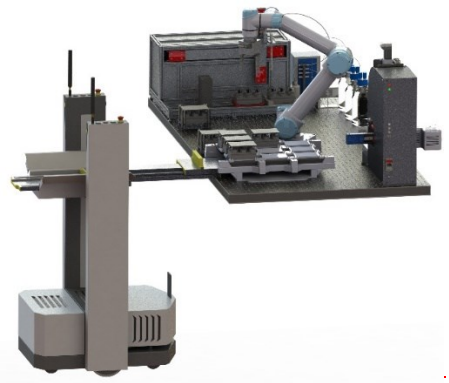


Figure 41 Automated Labware Handling System at CELISCA Laboratory

4.1.1 The Design of Tray and Rack

The transfer station serves as the critical link between mobile robots and workstations. Workstations utilize robotic arms to place processed labware and samples onto the trays located at the transfer station. The station comprises two main elements: a tray for holding labware and samples, and a rack that supports and secures the tray's position. Both the tray and the rack are specifically designed to facilitate the storage and transfer of labware and samples. The tray is mobile, allowing for easy transport, while the rack is fixed at the workstation, ensuring precise alignment for the mobile robot's engagement. During operations, the mobile robot navigates to the designated position at the workstation, extends its telescopic conveyor, and retrieves the tray, as shown in Figure 42.

Ensuring the safety of the labware and samples during the get/put process is paramount. Therefore, the positioning accuracy of the mobile robot in front of the workstation and the tolerance between the tray and rack are critical. The mobile robot's positioning accuracy must be within 1 cm, and the angular precision less than 1°, allowing the robot to center its telescopic conveyor between the tray and rack and parallel to the sides. Additionally, a tolerance is designed between the tray and rack to accommodate the robot's positional variances, allowing for smooth placement of the tray into the rack. This tolerance is kept within ±1 cm on all sides. It is recommended that all supporting racks at workstations be installed at the platform's edge to remain within the reach of the telescopic conveyor and within the working radius of the robotic arm.

To ensure precision throughout the process, a triangle device, as introduced in section 3.5.2.2, is used to enhance the positioning accuracy of mobile robots during the get/put process at all workstations. The sloped design of the tray and rack also accommodates tolerance. The following section will provide a detailed description of the design aspects of the tray and rack.

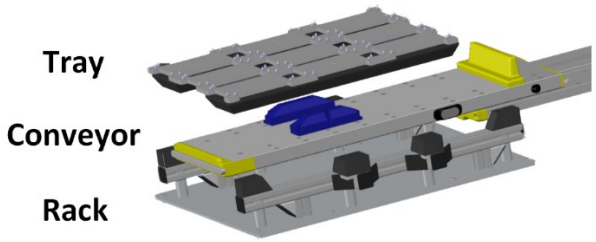
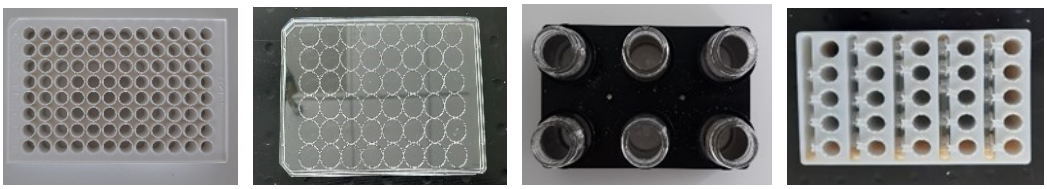


Figure 42 Tray, Rack, and Mobile Robot Conveyor

The laboratory utilizes many types of labware for life science experiments, all sharing a common base format (length: 125 mm, width: 82 mm). Figure 43 are four typical pieces of labware.



a) b) c) d)

Figure 43 Typical Types of Labware, a) Microtiter plate Standard 96 Costar; b) Microtiter plate Standard 48 F bottom light (lid); c) GefäßeKristallisation ; d) Eppendorfgefäße 1.5 ml

The tray is designed to serve as a platform for transporting labware within the laboratory environment. Its primary purpose is to enable the simultaneous and secure transport of multiple labware items between different stations using the mobile robot. IoT module or BLE beacon with tray is put on tray to track real-time position of tray. This unified transport system enhances efficiency and reduces the risk of handling errors during the transfer process. Additionally, the tray facilitates the smooth transfer of labware from the mobile robot to the workstation, ensuring that all items are positioned correctly and securely.

Tray measures 47 cm in length, 35 cm in width, and 3.41 cm in height. It can hold up to 9 labware simultaneously, with each piece of labware secured using eight pegs. The base of the tray consists of sloped sides, each at a 60° angle. The spaces between the labware are designed to facilitate the station arm robot's gripper for the placement and retrieval of labware. Additionally, a BLE beacon can be placed in the small square area in the middle. The gaps also allow the mobile robot's telescopic conveyor to engage with the tray securely, preventing it from sliding during transportation, which helps avoid potential spillage of samples.

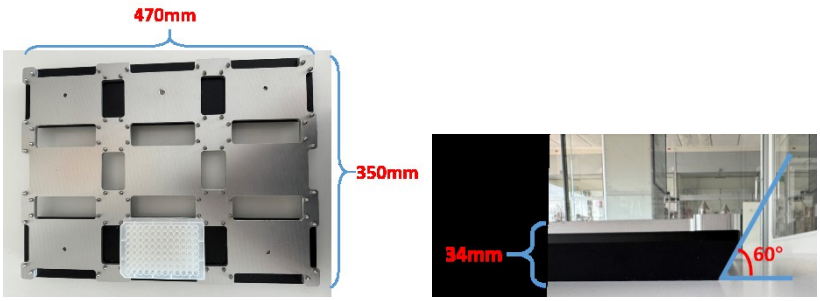


Figure 44 Specifications of the Standardized Labware Tray

The rack is fixed to the station and is used to hold the tray in workstations. It ensures that the tray is always positioned in the same spot on the station, making it easier for the workstation arm robot to transfer labware. Since the arm robot follows a fixed path, it is crucial that the labware is in the same location every time to ensure accurate movement. The rack consists of eight cylindrical columns with right-angle trapezoidal cross-sections (bottom width 6 cm, top width 4.4 cm, slanted side 3.3 cm, and right-

angle side 3 cm), it also has a 60° inclination angle. This angle facilitates a smooth sliding motion for the trays, reducing the friction and force required to place or retrieve the tray, while ensuring that the tray remains stable and securely held in place. The overall dimensions of the rack are larger than the tray, with ±1 cm of redundancy on the sides and front/back. This ensures that the tray can enter and be securely fixed in the rack even with an offset of ±1 cm in any direction.

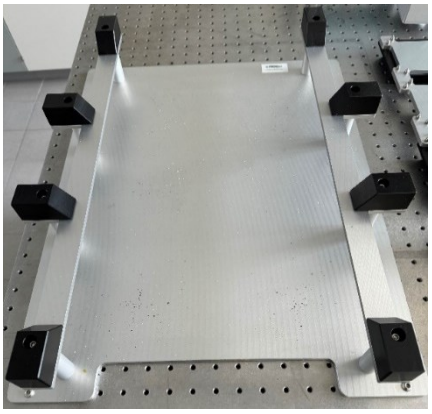


Figure 45 Rack installed in the station

Figure 46 illustrates the integration of the tray and rack, showing both a real-life implementation and a 3D design model. This setup ensures the stability of the tray and its precise placement in a fixed position on the station. By maintaining precise alignment and allowing for placement tolerance, the system facilitates accurate and efficient experimental processes, ensuring that labware is consistently positioned and transferred reliably.

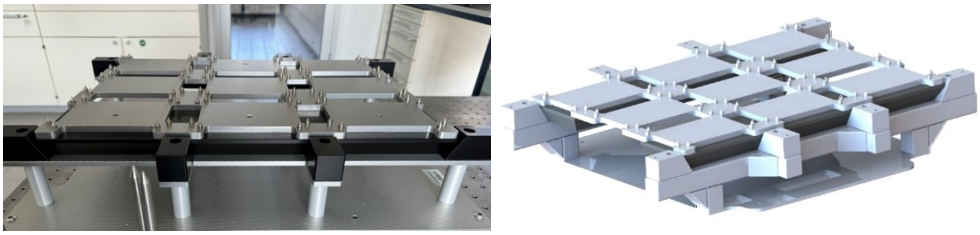


Figure 46 Integration of Tray and Rack in the station

4.1.2 Tray Transfer System

MOLAR robot is equipped with a telescopic conveyor that can move vertically and extend forward, reaching up to 955 mm, with a maximum load capacity of 35 kg. This conveyor facilitates the lifting of tray from station racks that are inaccessible by driving underneath. When the robot aligns with the rack, it precisely positions itself, extends the telescoping conveyor, and then lifts the tray from the rack, retracting the conveyor with the small load carrier attached. The robot departs only after retracting the conveyor and lowering the lifting device.

To prevent tilting during the get and put of tray, the robot includes a counterweight. The telescopic conveyor features a pressure sensor at the front to halt horizontal movements and mitigate the risk of crushing upon activation. Additionally, the telescope is connected to the robot's lifting device via a pivot point, allowing it to lift if it encounters obstacles. If the extending telescope encounters resistance, it is raised slightly until a vertical proximity sensor signals to stop further lifting movement. The specific get process is illustrated in Figure 47.

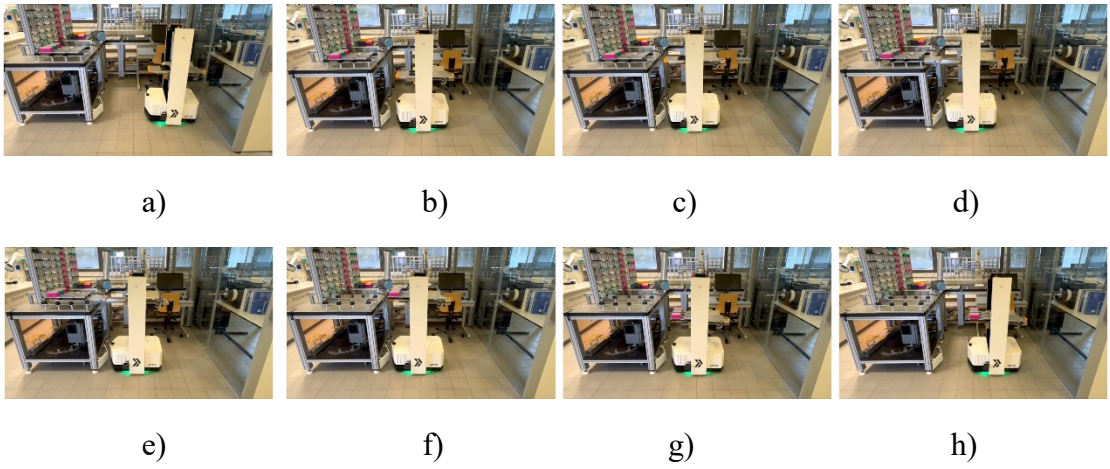


Figure 47 Labware get process, a) The robot arrives at the source point; b) It begins fine-positioning near the station base on the triangle device; c) The conveyor is raised to the corresponding rack height; d) The conveyor extends into the gap between the tray and the rack; e) The tray is lifted from the rack; f) The conveyor carrying the tray retracts back into the robot; g) The conveyor height is lowered to reduce the center of gravity; h) The robot moves back to the get point, ready to proceed to the destination station. This figure is difficult to see since the pictures are very small. You can either only use 2 colums or limit the number of pictures you use here (and show it in detail in the appendix).

Due to the different platform heights at various stations, the rack is installed at different positions. Therefore, different stations require the mobile robot to use specific advance distances, lifting heights, and extension lengths. A general XML get/put configuration is used to instruct the MOLAR robot on its get/put movements. Because each station has unique characteristics, the parameters are finely adjusted for each one, as shown in the Appendix C. All six stations in the laboratory were tested, with each station undergoing 50 trials of labware handling, achieving a 100% success rate.

Due to the uniform design of the three mobile robots, the get/put parameters were initially set as global parameters. However, unavoidable mechanical discrepancies among the robots meant that these global parameters did not suit all robots equally. The parameters were initially set based on Robot 1's performance across all stations and later fine-tuned based on the performance of Robot 2. While Robot 1 achieved a 100% success rate in its get/put operations, Robots 2 and 3 did not achieve the same level of success.

4.2 Charging Strategy

A crucial precondition for implementing 24-hour unmanned automatic transport is the capability of mobile robots to charge autonomously, ensuring a sustainable transport process. Once activated, MOLAR robots continuously relay their data to the FMS system, which, along with the AIC system, displays the real-time status of each robot on the GUI for easy monitoring by staff. Currently, there are three charging stations within CELISCA; two on the second floor and one on the third floor, strategically placed to allow robots to charge conveniently while executing tasks on different floors.

The selection of charging station locations primarily considered the following factors:

- High Traffic Areas: The mobile robot room on the second floor serves as the home for mobile robots and a storage area for labware, making it a frequent starting point for many tasks. This is why two charging stations were installed there.
- Task Start and End Points: On the third floor, the charging station is located in the farthest room accessible to the mobile robots. This room often requires the robots to wait for the completion of experiments before they can return labware and samples, making it a location where the robots spend a significant amount of time.
- Non-disruptive to Normal Operations: All three charging stations are positioned to ensure they do not interfere with the movement of other mobile robots and provide ample working space for staff. Additionally, parking stations are designated next to each charging station, allowing the robots to stand by after completing tasks and return to the area near the charging stations. This arrangement facilitates charging and standby after charging, saving energy spent traveling to and from charging stations. Strategic locations help conserve battery power, which is critical for maintaining 24-hour operations.

When a robot's battery level falls below 40%, the FMS system sends a command to the robot that is not currently engaged in transport tasks to proceed to the nearest charging station. The station is immediately marked as occupied, even before the robot arrives. The mobile robot autonomously navigates to and precisely positions itself on the charging station using the built-in triangle device, ensuring correct alignment of the robot’s charging contacts with the station's charging pins.

During charging, the contact pins at the bottom of the vehicle engage with the charging contacts at the station. A sensor located under the vehicle identifies the charging shoe and confirms correct positioning. The charging process only begins once the vehicle's PLC confirms that it is correctly positioned, activating the internal charging relay to allow the flow of the charging current. Throughout this process, the vehicle blocks direct access to the live charging contacts for safety.

The charging process is terminated either when the battery is fully charged or when multiple transport tasks are queued in the AIC system. The contacts are disconnected from the power supply before the vehicle departs to its next destination or to a parking station. The specific automatic charging process is illustrated in Figure 48.

a) b) c)

Figure 48 Autonomous Charging Process in 'Auto' Mode, a) When the battery level falls below 40% and there are no transport tasks, the robot automatically proceeds to the charging station; b) Charging begins and continues until the battery is full. During charging, the chassis displays a blue light; c) Once fully charged, the robot automatically moves to a waiting point to await tasks, with the chassis showing a green light for normal status.

In "Manual" mode, the mobile robot can be charged manually or through charging commands sent via the AIC. Charging automatically ceases when the battery is full, and the robot shuts down to stand by. This ensures there is no risk of overcharging.

4.3 Tests and Results

4.3.1 Comparison of MOLAR and H20 mobile robot

The H20 mobile robot has been operational in the CELISCA laboratory for over 10 years. Detailed information about this robot is provided in Section 2.3. The H20 is a humanoid robot that can carry only two labware at one time, resulting in lower transportation efficiency. Its higher center of gravity limits its speed. It is more of an academic-oriented robot. Considering practicality, CELISCA choosed the specially designed AGV for transportation, thus establishing the MOLAR robot system.

For the evaluation of the new MOLAR robots compared to the H20 robots a comprehensive comparison of various navigation and transportation metrics in practical scenarios was performed shown as Figure 49. Table 6 presents a comparison of the navigation technologies of the two robots.

Table 6 Comparison of robotic Technologies

	MOLAR robot	H20 robot
Localization	LiDAR + Fine-Positioning	StarGazer
Path Planning	A*(curve)	Floyd(Broken line)
Get and Put	Telescopic Conveyor	Mechanical Arm
Obstacle Avoidance	LiDAR	Camera(HRI) + Ultrasonic

Figure 49 MOLAR robot (right) and H20 robot (left)

4.3.1.1 Localization Performance

The localization accuracy of mobile robots directly determines the success rate of picking and placing labware. The MOLAR robot relies on LiDAR for positioning during the grabbing process, supplemented by fine-positioning with a triangular device.

In contrast, the H20 robot uses StarGazer positioning system, which developed by Hagisonic Company, is designed for indoor localization of intelligent mobile robots. This positioning sensor system utilizes a CMOS camera to capture infrared images reflected from passive landmarks that are typically mounted on the ceiling, as shown in Figure 50. According to the unique ID, X coordinate, Y coordinate, and angle information on each localization landmark, the relative position of the robot and the localization landmark in the image is calculated, and the pose information of the robot is calculated. By installing additional landmarks, the StarGazer navigation maps can be extended to match any size of a laboratory environment.

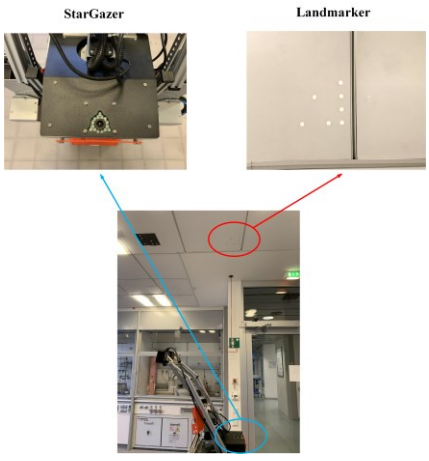


Figure 50 StarGazer-based localization system

In the localization experiment, both the MOLAR robot and the H20 robot were tested by moving to different stations. The get/put labware operation was repeated 50 times for all six stations. A laser rangefinder was used to measure the x and y coordinates, as shown in Figure 51. For the MOLAR robot data were recorded for six stations, while the H20 robot was tested for three stations. H20 robot only can go to 3 stations. Some stations are not available. The results of the positioning experiment are shown in Tables 7 and 8. The success rate of Get/Put here not only includes the picking and placing process but also the process of moving to the station. Max error refers to the difference between the maximum and minimum values in the x-axis direction, and similarly for the y-axis direction.

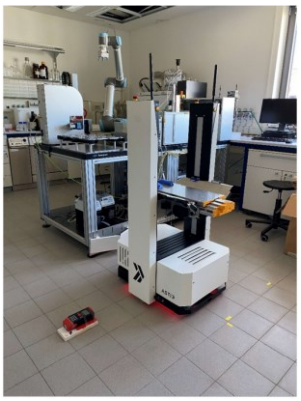


Figure 51 X and Y Coordinate Measurement

Table 7 Positioning Accuracy and Get/Put Success Rates of MOLAR robot 1

Stations	STD		Max		Get/Put
	X (mm)	Y (mm)	X (mm)	Y (mm)	Success Rate (%)
Station 209	5.00	2.69	29	10	100
Station 211	5.50	2.34	30	13	100
Station 212	2.03	1.60	9	8	100
Station 213	4.54	2.03	17	9	100
Station 306	1.65	2.28	7	8	100
Station 310	5.35	2.21	25	10	98

Table 8 Positioning Accuracy and Get/Put Success Rates of MOLAR robot 2

Stations	STD		Max		Get/Put
	X (mm)	Y (mm)	X (mm)	Y (mm)	Success Rate (%)
Station 209	6.79	2.30	29	11	98
Station 211	2.88	2.10	16	9	100
Station 212	2.80	1.95	15	8	100
Station 213	3.79	1.99	19	10	100
Station 306	2.27	1.95	11	8	98
Station 310	4.49	2.02	21	9	96

Table 9 Positioning Accuracy of H20

Stations	STD		Max		Get/Put
	X (mm)	Y (mm)	X (mm)	Y (mm)	Success Rate (%)
Station 213	10.34	7.88	51	47	90
Station 209	8.87	5.23	46	23	90
Station 306	8.08	12.94	37	55	72

The failures of the robot were primarily due to the narrow aisle in room 310, especially around the door area. The mobile robot struggled to pass through smoothly. Observations showed that when there was human interference in that area, the failure rate increased. The reasons for H20 robot failures include:

- Issues with the control software "MultiFloorSystem," causing the H20 to suddenly stop or continue moving forward at the station, leading to collisions.
- The positioning device StarGazer lost its current location due to interference from infrared light on sunny days.
- When the H20 robot traveled to the third floor, it encountered network coverage and fluctuation issues inside the elevator, resulting in a loss of signal and a disconnection from the system. There were also minor issues during elevator use, leading to the H20 robot hitting the elevator door.

From Tables 7 to 9, it can be seen that the overall stability and positioning accuracy of the ASTI robot were higher than those of the H20 robot. The ASTI robot's average standard deviation in positioning accuracy was 3.92 mm (x-direction) and 2.12 mm (y-direction). In contrast, the H20 robot had an average deviation of 9.01 mm (x-direction) and 8.68 mm (y-direction). The ASTI robot's maximum error averaged 19.00 mm (x-direction) and 9.42 mm (y-direction), while the H20 robot's maximum error was 44.67 mm (x-direction) and 41.67 mm (y-direction). These improvements in accuracy help further ensure the success of the subsequent get/put operations.

Compared to earlier H20 robot system, the ASTI robot system has shown a remarkable improvement in accuracy by approximately 65.9% in standard deviations and about 67.1% in maximum error metrics in positioning. It’s significantly enhancing the efficiency and reliability of robotic operations in laboratory settings.

4.3.1.2 Transportation Efficiency

In addition to positioning accuracy, factors such as moving speed, transportation capacity, battery life, and charging speed influence the ability of mobile robots to transport labware efficiently.

Speed: Since the speeds of the two mobile robots are not constant during movement, ASTI robot generates a polyline based on the A* algorithm and then smooths it using a spline curve, resulting in smoother robot operation. In contrast, the H20 robot's path planning is based on the Floyd algorithm, generating polylines without smoothing. This causes the H20 robot to pause and reorient at each polyline point before proceeding to the next segment. Therefore, the comparison method involves calculating the total time taken from the charging station to the get station.

All test runs started from the charging station. The times recorded in Table 10 are under unobstructed conditions, with no elevator occupancy. Each run time varies slightly, representing average time.

Table 10 Average Running Time from Park Station

Target Station	MOLAR	H20
Station 213	40 s	50 s
Station 209	2 min 45 s	4 min
Station 306	4 min 50 s	10 min

The get/put speed is also included in running time. MOLAR takes 1 minute 15 seconds, while H20 takes 1 minute 10 seconds. However, the H20 process is less stable due to a weak motor, causing the arm to shake during the get operation. The time for vision-based labware position recognition is also inconsistent. The H20's get process is more challenging, embedding a trained neural network algorithm, whereas the MOLAR robot performs fixed mechanical actions, offering better robustness.

Transportation Capacity: The MOLAR robot can carry nine labware at a time, whereas the H20 can carry two, distributed on both sides, with one arm controlling each labware outside the robot's body. The ASIT robot has a payload capacity of 35 kg, whereas each arm of the H20 robot has a payload capacity of 800 g.

Charging Comparison: Charging ensures the sustainability of mobile robots. To achieve 24-hour autonomous transportation, automatic charging is essential. Robots that cannot charge automatically will stop operating when the battery depletes. However, merely having an automatic charging function is not enough to guarantee uninterrupted 24-hour operation; other factors like battery management and fault recovery capabilities are also necessary.

The comparison includes the number of charging stations, charging speed, charging safety, and duration of a full charge, as shown in Table 11.

Table 11 Charging Comparision

	MOLAR	H20
Number of Robots	3	5
Number of Stations	3	4
Charging Time	2 h	50 min
Battery Duration	4 h	40 min
Battery Volume	40 Ah/24V	16 Ah/16V
Charging Safety	with charging protection setting	overcharging risk

4.3.1.3 Comparison and Conclusions

Based on the recent analysis and comparisons, here are the summarized findings:

1. Positioning Accuracy: The MOLAR robot uses LiDAR combined with fine-positioning for accurate localization, whereas the H20 robot uses the StarGazer system, relying on ceiling-mounted markers and a CMOS camera. MOLAR robot average standard deviation of x-position is 4.01 mm, y-position is 2.19 mm. H20 robot average standard deviation of x-position is 9.10 mm, y-position is 8.68 mm. The positioning accuracy directly affects the success rate of picking and placing labware, with MOLAR showing a more stable performance due to its robust mechanical actions. MOLAR robot uses advanced LiDAR-based SLAM mapping and positioning. In addition, fine-positioning auxiliary devices are added to each station. H20 robot is based on traditional landmarks. MOLAR robot's positioning accuracy is better than H20 robot overall.

2. Transportation Efficiency: Speed: The MOLAR robot demonstrates smoother and faster navigation due to its use of spline curves for path planning, in contrast to the H20 robot’s unsmoothed polyline paths which cause pauses at each segment. Get/Put Operations: MOLAR is slightly slower but more stable in picking and placing labware compared to H20, which suffers from inconsistencies due to weaker motor performance and variable vision-based recognition times.

3. Transportation Capacity: The MOLAR robot can carry up to nine labware simultaneously, significantly more than the H20 robot, which can carry only two. This makes MOLAR more efficient for tasks requiring the transport of multiple labware items.

4. Charging and Battery Performance: Charging Speed: The H20 robot charges faster (50 minutes) compared to the MOLAR robot (2 hours). Battery Duration: Despite the faster charging, the H20 robot’s battery lasts only 40 minutes, while the MOLAR robot’s battery lasts 4 hours, indicating a better overall endurance. Charging Safety: MOLAR has a charging protection setting, reducing the risk of overcharging, a

potential issue with the H20 robot.

The MOLAR robot outperforms the H20 robot in several critical aspects, including positioning accuracy, transportation capacity, and overall robustness in get/put operations. While the H20 robot charges faster, its shorter battery life and less stable performance make it less suitable for continuous, long-duration tasks. The MOLAR robot's efficient path planning, higher transportation capacity, and longer battery life make it a more reliable choice for labware transportation in practical scenarios.

4.3.2 Special Area Test

Real laboratories have some limitations to the use of mobile robots. This is particularly due to the space available. The practical effectiveness of the semantic map was therefore evaluated as part of extensive research, including Narrow Area, Single-robot Area, Preferred Line, and right-hand Traffic Area. Figure 52 shows the semantic map of the CELISCA laboratory, with the tested areas marked as legends.

Figure 52 Semantic Map of CELISCA Laboratory

4.3.2.1 Narrow Area

Before discussing narrow areas, it is essential to introduce the concept of safety areas to ensure the MOLAR robot operate securely. According to ISO 3691-4, different safety fields are set in the scanner and implemented in the PLC along with navigation

parameters, as shown in Figure 53. These safety fields are positioned 12 cm above the floor level and categorized into warning and protective fields, triggering distinct responses:

1. Warning fields: If an obstacle appears within 120 cm in the driving direction or 60 cm laterally in a warning field, the vehicle reduces its speed to a minimum of 0.3 m/s and bypass the obstacle.
2. Protective fields: If an obstacle enters a protective field where the space is less than 130cm (width of robot 1 m+ 2 *flation distance 0.15 m), the vehicle halts and waits until the obstacle exits the field.

The semantic map's narrow area can alleviate the issue with the 130 cm width restriction, allowing movement at a minimum speed of 0.3 m/s. Some narrow spaces in the laboratory, such as automatic doors and tight corridors, are less than 130 cm wide and require the mobile robot to pass through. Therefore, narrow areas are set in these locations. Without safety fields, the narrowest distance a mobile robot can pass is: robot width + 0.22 m = 0.97 m. The optimal width for a mobile robot's movement is 2.13 m.

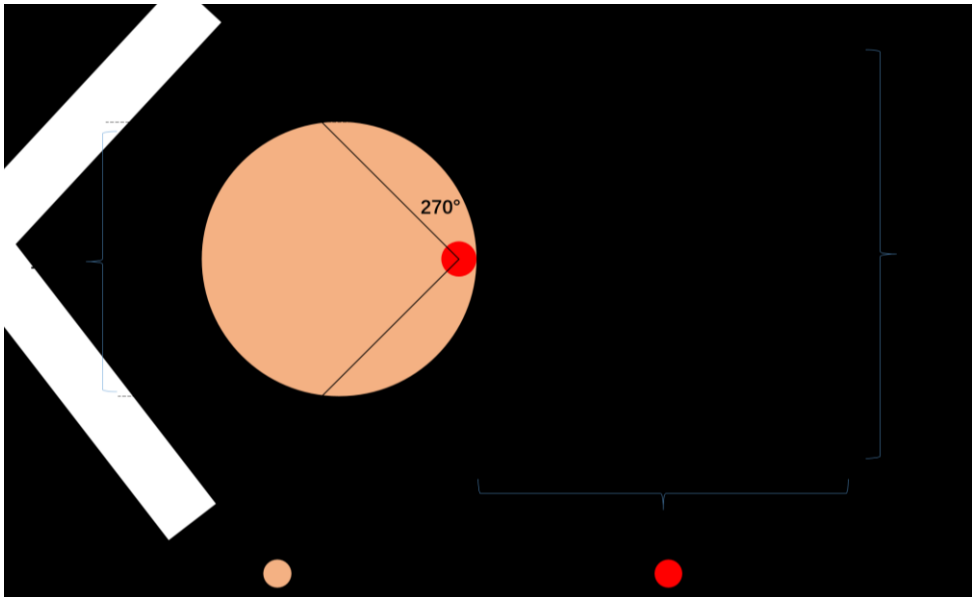


Figure 53 Safety Fields

The navigation system is configured to adjust the reaction based on scanner data and the dynamic footprint established in the software. The size of this footprint is adapted to the speed and is always larger than the safety fields to ideally prevent their activation. The integration of a LiDAR scanner with motor encoders allows for the adjustment of safety field sizes according to the actual speed of the robot.

Transitioning to narrow areas, the CELISCA laboratory environment includes many confined spaces, such as automatic door zones and some narrow corridors. In these narrow areas, mobile robots deactivate their safety area and proceed at the lowest speed

to ensure safe passage. Once the robot exits a narrow area, its speed returns to normal. This strategy is crucial for navigating through tight spaces safely, as depicted in Figure 54, which illustrates the robot successfully moving through various narrow zones.

Figure 54 Robot Cross the Narrow Areas

4.3.2.2 Single-Robot Area

Single-robot areas are implemented to address the issue of a swarm of robots in narrow spaces. In areas with sufficient space, right-hand traffic can be employed to solve the problem of intersecting paths. Due to laboratory limited space, some areas cannot accommodate more than one mobile robot at the same time. If two robots are present simultaneously, it can lead to potential motion planning issues, such as in door areas, corridors, and pick-and-place stations. If two robots confront each other at the ends of a narrow area and cannot avoid each other, it can result in a deadlock.

Currently, the only place in the laboratory where right-hand traffic is used is in front of the elevator, where there is enough width (approximately 2.3 m). However, the results are not ideal. Tests have shown that while it is possible to have opposing traffic without deadlock, it requires 1 minute. For more ideal results, a width of at least 2.7 meters is necessary.

Figure 55 shows station 213 designated as a single-robot area. One robot completes the get tray task and leaves the single-robot area. Another robot waits in waiting point outside the area until it is clear before entering.

a) b) c)

d) e) f)

Figure 55 Two robots interacting in the single-robot area. In image a), Robot 1 is stationed within the single-robot area designated for station 213, while Robot 2 waits at the waiting point, delineated by the solid lines; the single-robot area is outlined with dashed lines; b) and c) Robot 1 leaves the single-robot area; d) Robot 1 leaves the single-robot area, Robot 2 starts up and begins to enter. e) and f) show Robot 2 move into the single-robot area of station 213.

4.3.2.3 Preferred Line

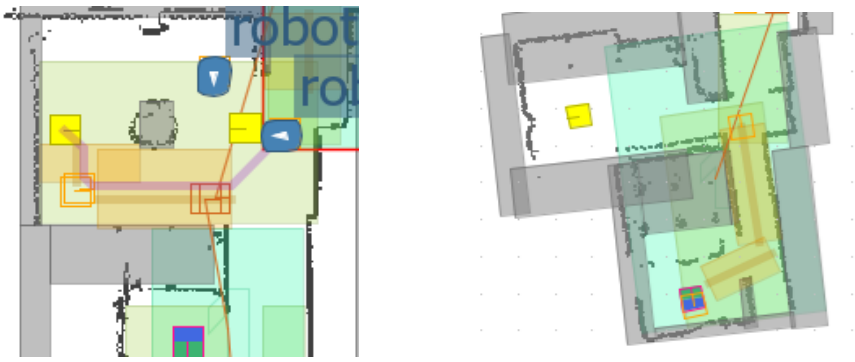
Preferred-Line refers to a predefined path set within the FMS. It could guide robot movement through particularly constrained or complex spaces within a laboratory. This directive is crucial in areas where space limitations or complex layouts might otherwise complicate autonomous navigation. Narrow areas are typically only slightly larger than the robot itself, often with widths less than 1.2 meters, restricting the robot's maneuverability. Complex areas may include spaces with intersecting paths, irregular layouts, or high-traffic zones that require sophisticated navigation strategies to avoid collisions and ensure efficient movement.

The Preferred-Line is established in the FMS's map editing mode by placing it directly on feasible areas of the map. When the mobile robot enters such an area, it will strictly follow the Preferred-Line path if it is safe to do so. Without a set Preferred-Line, the robot's operational efficiency could decrease, potentially causing frequent stops within the area as it spends additional time calculating feasible routes or, in some cases, remaining stationary if no viable path is found.

In Figure 56 a), the purple path represents the real-time path planning result. It shows that during the path planning process, the mobile robot strictly follows the direction of the preferred line in the preferred line area.

Figure 56 b) depicts the Room 313 area, the narrowest part of the laboratory, with an aisle width of 0.97m and a robot width of 0.75m, leaving very little free space. Figure 57 demonstrates how the mobile robot successfully uses the preferred line to navigate through the narrow area of Room 313 to reach the station. Before the implementation of the Preferred Line in this area, the mobile robot was unable to navigate to its destination using standard navigation methods. Now, the success rate of the mobile

robot reaching the workstation inside Room 313 has increased to 98%. There is only a small chance that the robot may get stuck at the turns.



a) b)

Figure 56 Preferred Line Area

Figure 57 Robot Cross Preferred Line Area in Room 313

4.3.2.4 Right-Hand Traffic

In the laboratory, smooth transit through narrow spaces is managed by designating them as single-robot areas. In contrast, wider areas (width > 2.5 m) employ Right-Hand Traffic to facilitate multi-robot operations. Without Right-Hand Traffic, two robots require a width of 3.3 meters to pass each other. Additionally, robots are set to prioritize traveling in the center of pathways. Deadlock will happen when encountering each

other head-on, as they are unable to recognize each other as robots and thus fail to yield. To prevent this, the area is divided into two lanes for opposite-direction travel. This effectively resolves the issue of multi-robot traffic at intersecting pathways. The implementation of Right-Hand Traffic is achieved by setting up unidirectional travel for each lane, placing two lanes of different directions next to each other.

Figure 58 shows the area in front of the elevator, which connects corridors at both ends and serves as a good location for robot crossings. Figure 59 demonstrates two robots coming from different directions, meeting in front of the elevator, successfully passing each other, and reaching the opposite corridors. The total time taken is 50 s, and there is some stop in the robots' movements, primarily due to the limited space in front of the elevator, which is only 2.4 meters wide. If the space were wider, the performance would be improved.

Figure 58 Right-hand Traffic Area in front of the elevator

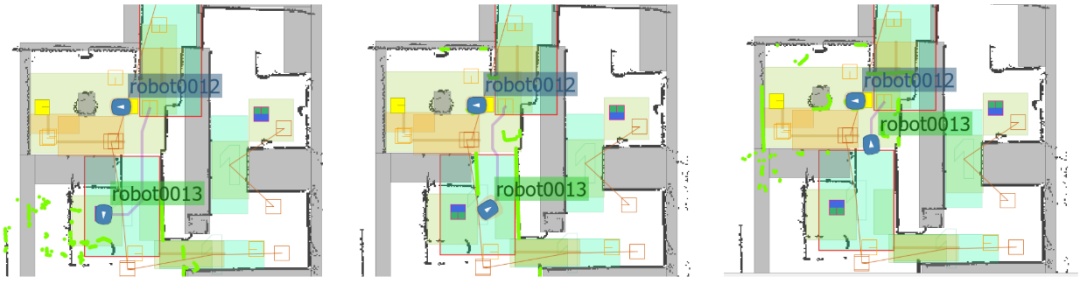
Figure 59 Interaction between Two Robots in the Left-Hand Traffic Area see remakrs above

4.3.3 Obstacle Avoidance Test

Obstacle avoidance is crucial in the laboratory because staff and mobile robots may share the same environment. Movable chairs, boxes, and other items can also become obstacles. Mobile robots need to avoid both static and dynamic obstacles. Tests were conducted separately for static and dynamic obstacles.

4.3.3.1 Static Obstacles

The charging MOLAR robot and the H20 robot parked at the aisle edge were used as static obstacles to test the mobile robot's avoidance capabilities. Figure 60 shows the path planning process in the FMS when a mobile robot encounters obstacles. Figure 61 illustrates the actual obstacle avoidance process. The mobile robot successfully bypassed the obstacle and opened the automatic door to leave the room.



a) b) c)

Figure 60 Obstacle Avoidance FMS Path Planning Process. a) Global static path planning by the mobile robot before encountering obstacles; b) Local dynamic path planning when the mobile robot detects two obstacles; c) Updated grid positions of obstacles during the mobile robot's movement

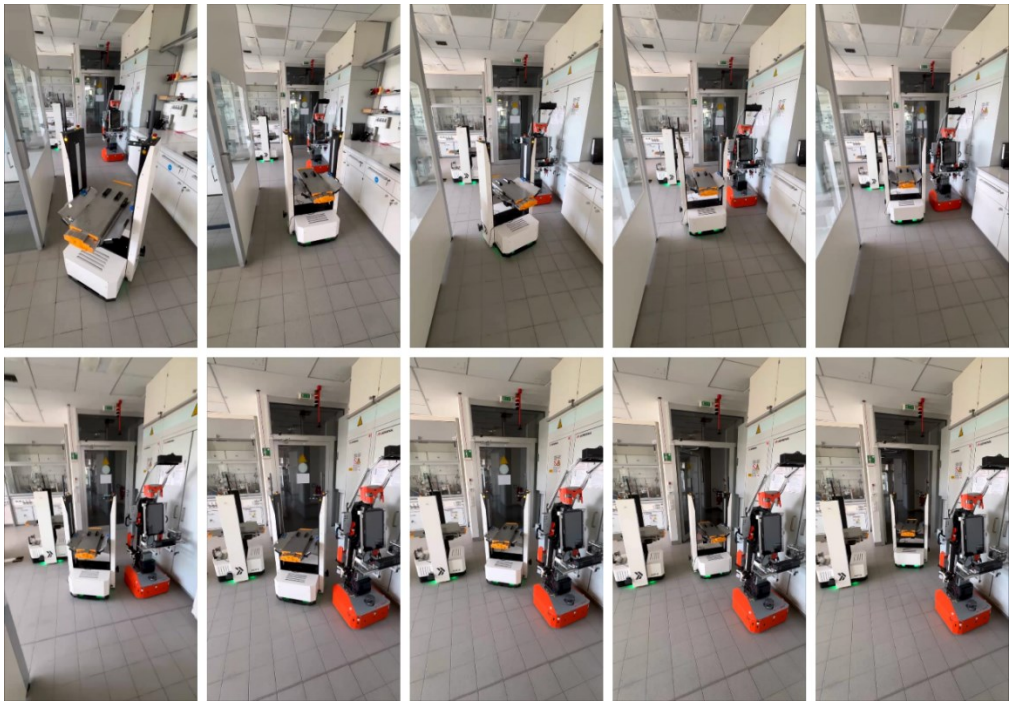


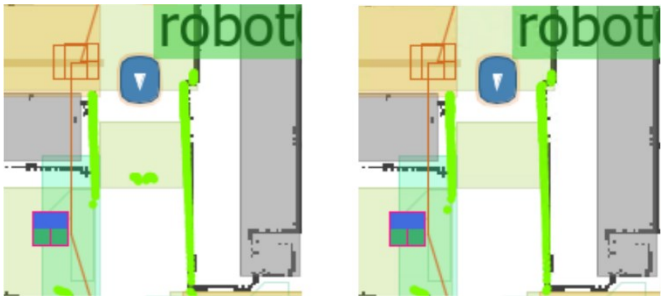
Figure 61 Actual Process of Static Obstacle Avoidance

4.3.3.2 Dynamic Obstacles

The LiDAR on mobile robots is not only used for mapping, but also for real-time detection of surrounding obstacles during movement. In the laboratory, static obstacles refer to fixed, long-term elements such as walls and lab benches. Dynamic obstacles can typically be lab personnel, movable chairs, and temporarily placed tables or goods. Mobile robots must have the ability to avoid dynamic obstacles to operate in environments shared with humans and objects. The obstacle avoidance method used by the mobile robot is the DWA in the move base. Figure 62 shows the environmental scanning map of LiDAR during the robot's movement.

A scenario was simulated where a staff suddenly moved into the path of the mobile robot, blocking its way. The mobile robot successfully stopped and maintained a safe distance 1.2 m. The robot start to give a voice. Once the staff member moved aside, the mobile robot got enough space and successfully passed through. Figure 63 illustrates the entire process.

The mobile robot underwent 20 tests for avoiding both static and dynamic obstacles while maintaining a forward safety distance of 1.2 meters and a lateral safety distance of 0.15 meters. It successfully avoided the obstacles in 100% of the cases and reached its destination without issues.



a) b)

Figure 62 LiDAR Detection during Dynamic Obstacles Avoidance. The green point cloud represents obstacles detected by the mobile robot; the green points on either side are walls and cabinets, a) A person standing in front of the robot, with LiDAR detecting an obstacle blocking the robot's forward path; b) LiDAR scan of the area when unobstructed by people.

a)

b)

Figure 63 Actual Process of Dynamic Obstacle Avoidance

5 Chapter 5 Infrastructure Integration

Automatic doors and elevators are essential in a multi-floor intelligent environment for spatial division and connectivity. For mobile robots executing transport tasks in the CELISCA laboratory, the autonomous operation of doors and elevators is crucial. This chapter details how the mobile robot system integrates with communications to control automatic doors and elevators.

5.1 Introduction

MOLAR robot systems are specifically designed for laboratory transportation. Currently, MOLAR robots connect a total of six laboratory workstations across two floors, Figure 64 shows the movement trajectory of the mobile robot in the laboratory. Four stations are located on the second floor and two on the third floor, each station equipped with different instruments for various stages of life science experiments. Currently, CELISCA has 3 mobile robots, aiming to achieve 24h automated Point-to-point transportation between all workstations, including continuous transport and multi-robot collaboration. The MOLAR robot is not equipped with a mechanical arm and cannot manually open doors or press elevator buttons to travel between floors like humans. Instead, it relies on remote communication to control the elevator and automatic doors. The elevator used at CELISCA is semi-outdoor with the capability to open at both ends and have metal wall cabins. The doors combine glass and metal and are equipped with a PLC control system that can provide the current status of the doors and operate their opening and closing. The mobile robot can send commands within the inner network to operate these doors. However, managing a closed-system elevator is more complex since the elevator PLC can only call the elevator to a specific floor without providing any additional feedback.

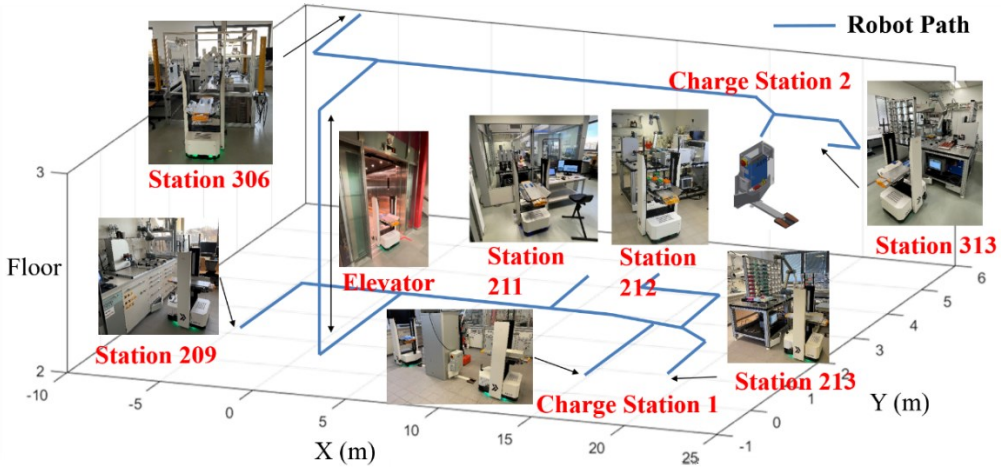


Figure 64 Laboratory overview with MOLAR robot navigation path across two floors

The common logic for mobile robots using elevators includes: 1) navigating to the elevator entrance; 2) calling the elevator to the departure floor; 3) entering when the elevator doors open; 4) calling the elevator to the target floor; 5) exiting when the elevator reaches the target floor and the doors open. There are three main challenges to address: 1) calling the elevator, 2) verifying that the elevator doors have opened, and 3) confirming arrival at the target floor.

Given the excessive natural light inside the elevator, particularly on sunny days, visual-based solutions are challenging and require additional cameras and onboard computers for visual processing, thus were not considered. Instead, ultrasonic sensors were employed to detect the status of the elevator doors, and air pressure sensors to determine the floor. Due to the instability of barometric sensors in extreme weather conditions, a beacon-based floor detection system was also implemented. This solution is more cost-effective and practical, requiring minimal computing power and offering good real-time performance.

5.2 Automatic Door Handling

To achieve a higher level of automation, automatic doors were installed in all laboratory rooms and corridor areas. These doors are equipped with PLC controls and collision detection devices, ensuring safety and efficiency in operation. Communication is facilitated through ADAM sockets (Advantech, Taipei, Taiwan), which are Digital I/O Modules that can be installed in a distributed manner throughout the laboratory. They operate over the network, allowing for remote real-time management and monitoring. These modules are typically used to control simple devices in laboratories and factories.

In the laboratory, there are 14 automatic doors ADAM modules installed. To centrally manage these, eight sockets have been established based on regional divisions. A socket in this context refers to a communication interface that allows multiple doors to be controlled through a single connection, reducing the need for excess cabling. Neighboring doors are grouped and controlled via the same socket to optimize the setup. The socket serves as a centralized point that communicates with the ADAM modules to manage the doors' open/close states. Control Bits, as shown in Table 12, are used to open or close the automatic doors. The Read Bit uses the same server and includes two states: the door is opening/closing and the door is fully open/closed.

Table 12 PLC Control Bit Mapping for Laboratory Automation

Server	Control Bit				Floor
	Bit 3	Bit 2	Bit 1	Bit 0	
Door Server 1			Lab212	Lab213	2nd
Door Server 2			Lab211	Floor2South	
Door Server 3		Lab209	Lab207	Floor2North	
Door Server 4			Lab310	Floor3South	3rd
Door Server 5				Lab 306	
Door Server 6		Lab305	Lab308	Floor3North	
Door Server 7				Lab313	
Lift Server	Floor3	Floor2	FloorE	FloorA	

Based on the Write/Read bit and protocol information, a simple graphical user interface (GUI) was developed in C# for monitoring and remotely controlling the status of doors and elevators. The mini-program, built on the .NET Framework, is a Windows Forms application designed for monitoring and controlling door statuses and scheduling elevators. It consists of two main components: 1. Communication Interface: Responsible for communicating with the PLC (programmable logic controller), fetching door statuses, and sending control commands. 2. User Interface (UI): Provides a graphical display for real-time monitoring of doors and elevators.

Communication details include:

- Door Controller Communication: The mini-program establishes TCP connections with various door controllers using the AdamSocket class. Each door controller is responsible for controlling a set of doors. Connections utilize the Modbus TCP protocol with the standard port 502. The system regularly checks the connection status and attempts to reconnect if the connection is lost.

  Each controller's status is updated through bound events updateDoorStatusdelegate, which are triggered when door statuses change, providing the latest status updates through the UI to the user.

- Elevator Controller Communication: Elevator scheduling is similar to door control,

involving TCP connections with elevator controllers via AdamSocket and using the Modbus TCP protocol. Elevator requests are initiated through specific UI components, sending open/close signals to the elevator controller based on the floor selected by the user.

- Exception Handling: The program includes an exception handling mechanism to ensure continuous operation during network or device failures. It also logs error information for troubleshooting and system maintenance.

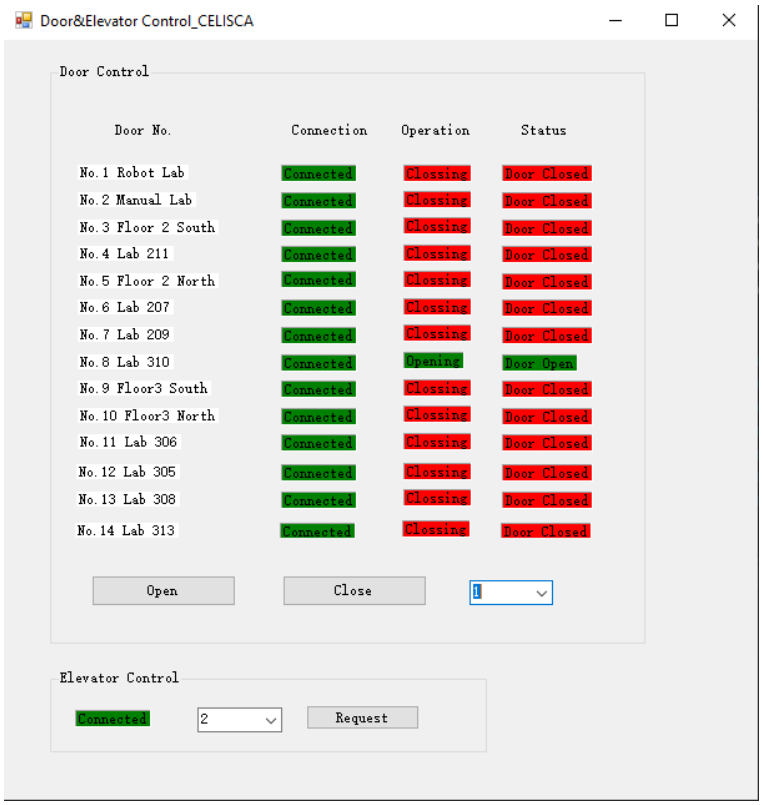


Figure 65 Door and Elevator Control Tool

This tool includes two areas: the door control area and the elevator control area.

1. In the door control area, the system displays several pieces of important information. First, the door number field shows the unique identifier or location of each door, such as “Lab 211” or “Floor3 east.” The Connection Status field indicates whether the system has successfully established communication with the door, using labels like “Connected” to confirm a successful connection. The Operation Status provides the current state of the door's operation, such as “Closing” if the door is in the process of closing, or it will display “Door Open” or “Door Closed” when the door has fully opened or closed, respectively. Finally, the Status field reflects the overall condition of the door, like “Door Closed” when the door is fully shut. For remote control, the

tool includes Door Operation Buttons to open or close the selected door, providing enhanced user control over the door's functionality.

2. In elevator control area, only displays a connection status, indicating whether communication with the elevator has been successfully established. It includes a dropdown menu for selecting different floors and a "Request" button for sending elevator call requests.

The C# GUI successfully verified the correctness and feasibility of the PLC configurations for the laboratory's automatic doors and elevators. This system ensures that mobile robots can access different areas of the laboratory through automatically controlled doors. The logic used by the mobile robot to operate the automatic doors is as follows:

The mobile robot moves to the waiting point of the door using its navigation system, and the AIC continuously monitors the robot's status. The AIC sends an open-door command to the corresponding door socket via WIFI using the write bit. The socket then transmits the signal to the ADAM PLC, executing the open-door instruction. Simultaneously, the door's status is monitored using the read bit. Once the door status indicates it is open and the LIDAR detects no obstacles ahead, the robot proceeds through the automatic door. After passing the waiting point on the opposite side, it sends a close-door command to conclude the process, as illustrated in Figure 66.

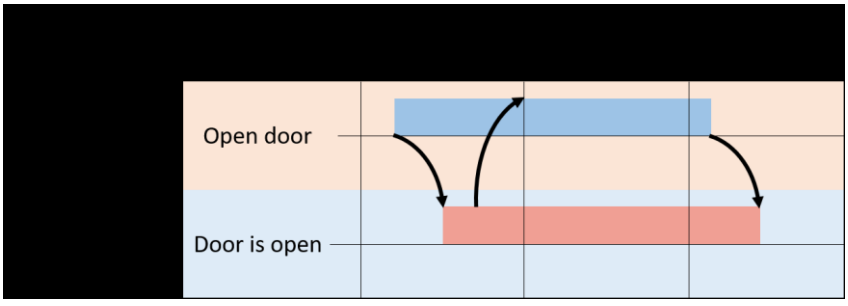


Figure 66 Robot Door Passing Timeline

5.2.1 Automatic Door Interaction Test

Based on AIC, the mobile robot is controlled to pass through the automatic door. The AIC visualizes the peripheral holding register, allowing for an intuitive view of the logic process as the mobile robot crosses the automatic door, as shown in Figure 67.

In Figure 67, under the server, the control and status of two doors are included, but only one door's interaction process is shown here. The holding register on the left represents

the door status (opening/closing, fully open/closed). The right side shows the door command issued by the mobile robot. The specific process is as follows: 1) The mobile robot arrives at the door waiting point and begins to open the door; 2) The door is fully opened; 3) The robot is crossing the door; 4) The mobile robot arrives at the opposite door waiting point and begins to close the door; 5) The door is fully closed. Figure 68 shows the actual process of the MOLAR robot cross automatic door.

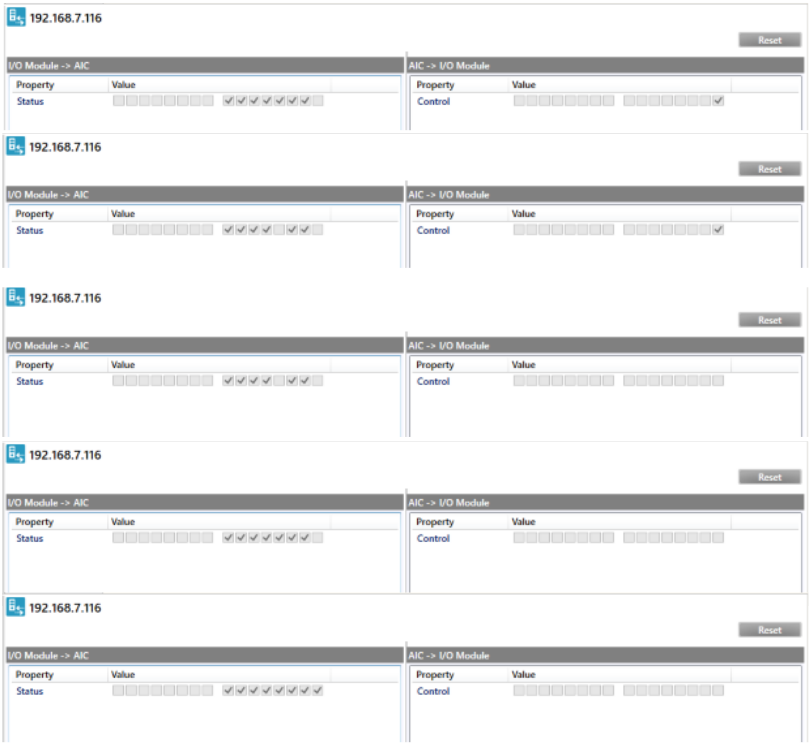


Figure 67 Visualization of Mobile Robot Crossing Automatic Door Logic

Figure 68 MOLAR Robot Cross Automatic Door

5.3 Closed-System Elevator Handling

5.3.1 System Framework

Due to the closed-system nature of the elevators, they can only be called to designated floors and cannot provide the mobile robots with necessary information about elevator door status and current floor. An IoT solution was proposed, utilizing an IoT module combined with ultrasonic and air pressure sensors to acquire elevator-related information. A Virtual PLC software was designed to replace the elevator PLC, acting as a subordinate device to the AIC, to facilitate complete robot-elevator interaction.

The core of the closed-system elevator handling is the Virtual PLC. It functions as middleware between the AIC and the elevator PLC, transmitting real-time elevator status to the AIC, providing the mobile robot with the necessary instant elevator interaction information. The system framework is illustrated in Figure 69. The Virtual PLC facilitates three-directional communication:

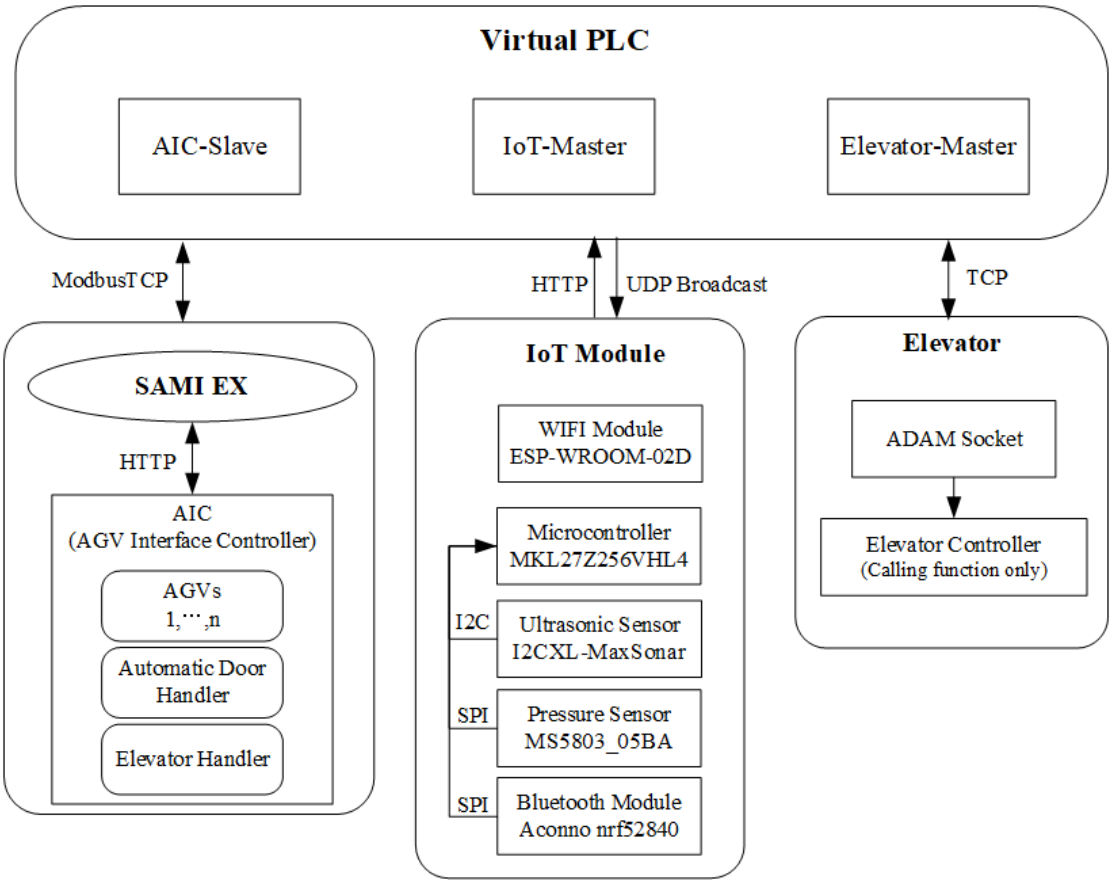


Figure 69 Robot-Elevator Interaction System Framework

1. As a slave device to the AIC, it provides essential elevator status; it receives elevator call commands from the AIC; the communication protocol is based on ModbusTCP.

2. After internal logic processing, as a master device to the elevator, it sends call commands; the communication protocol is based on ADAM TCP.

3. As a master device to the IoT modules, it gathers data from ultrasonic and air pressure sensors and processes this data within the Virtual PLC to convert it into elevator door status and current floor information. It uses UDP broadcast to issue data retrieval commands to all IoT modules, which then send the data back via the HTTP protocol.

Virtual PLC, as the software component handling robot-elevator interactions, acquires real-time elevator statuses through an IoT solution. The IoT modules are installed at the front of each mobile robot. As the superior machine to the IoT modules, the Virtual PLC, within a local network, retrieves readings from all IoT modules installed on mobile robots every second via UDP broadcast, including distance, air pressure, and Bluetooth RSSI.

The IoT modules developed by CELISCA are equipped with a microcontroller, ultrasonic sensors, air pressure sensors, and communication modules including WIFI and Bluetooth, installed on the mobile robots [158]. The ultrasonic sensors acquire distance information to update the status of the elevator doors. The air pressure sensor obtains the current air pressure to update the current floor status. Bluetooth receives signals from beacons to estimate the current floor. These three sensors send their data to the microcontroller, which then transmits all data packets to the Virtual PLC via the WIFI module.

The air pressure sensor, MS5803_05BA (TE Connectivity, Switzerland), communicates via the SPI protocol and has an altitude resolution of 30 cm. The ultrasonic sensor, I2CXL-MaxSonar-EZ (MaxBotix, USA) uses the I2C protocol with a detection range of 20 cm to 765 cm and a resolution of 1 cm. The resolution and range of these sensors are ideal for determining the height differences between floors and the status of elevator doors.

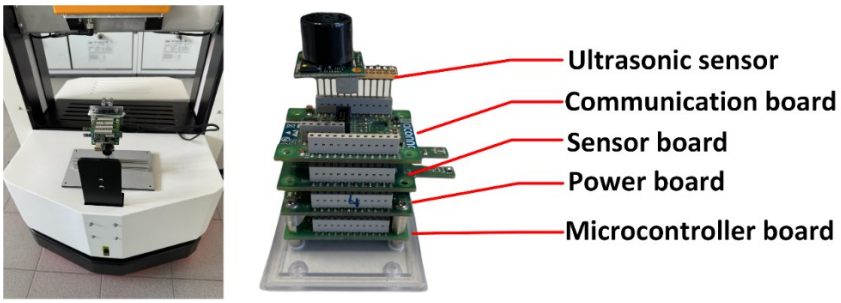


Figure 70 a) MOLAR with IoT Module; b)Components of IoT Module

The communication method between the IoT module and the client involves the IoT cloud sending a UDP broadcast to all IoT modules, which includes the necessary sensor information and the message return path. Upon receiving the message, the IoT module

transmits the sensor data based on the HTTP protocol. The advantages of this design are: 1) Using UDP broadcast can rapidly broadcast messages to all devices, ensuring that all IoT modules re-ceive information almost simultaneously. 2) Transmitting sensor data only when needed can reduce unnecessary network traffic and enhance overall communication effi-ciency. 3) By using the broadcasting method, even if indi-vidual IoT modules temporarily fail to respond, the rest of the system can still operate. Even in areas with dense inter-ference signals, brief signal loss won't affect subsequent normal operations. Moreover, it facilitates the addition of more IoT modules.

The elevator is operated by the ADAM-4060 module, a digital I/O device with relay outputs that assigns the required target floors to specific hardware ports and pin numbers. This module is limited to initiating calls to the current and target floors and does not provide any other information.

5.3.2 Interaction Logic

The interaction between mobile robots and elevators is executed through the communication between the AIC and the Virtual PLC. The AIC acts as the master, and the Virtual PLC operates as the slave, using the Modbus TCP protocol for communication. The interaction process is divided into one read area and one write area based on the state of the mobile robot and the elevator. Various states are represented in the form of holding register bits, with the read area reflecting the current state of the elevator, and the write area representing the requests or actions of the mobile robot. This setup facilitates the complete interaction process between the mobile robot and the elevator. The specific information is presented in Table 13.

Table 13 Modbus Register Map for Elevator Control Interface

Read/Write	Bit	Name	Detail
Read	0	Job Active	Elevator is being called
Read	1	Entry Allowed	The door is open and the AGV can enter the elevator
Read	2	Exit Allowed	The door is open and the AGV can leave the elevator
Write	0	Call Elevator	Call the elevator to the source floor
Write	1	AGV Enter	The AGV is driving into the elevator
Write	2	AGV Inside	The AGV is inside the elevator
Write	3	Go Destination	Call the elevator to the destination floor
Write	4	AGV Leave	The AGV leaves the elevator
Write	5	Reset	Reset the elevator sequence

Figure 71 sequentially illustrates the complete interaction process, from the AGV

calling the elevator to the AGV leaves the elevator. Typically, the elevator side displays three bits indicating that the elevator has been called, the door is open, and entry/exit for the robot is permitted. The AGV side contains six bits showing the elevator being called to the source floor, the AGV entering the elevator, the AGV inside the elevator, the AGV calling the elevator to the target floor, and the AGV leaving the elevator. The entire interaction process involves 11 steps, corresponding to 11 changes in the bits, with the specific procedures described in the text within the Figure 71.

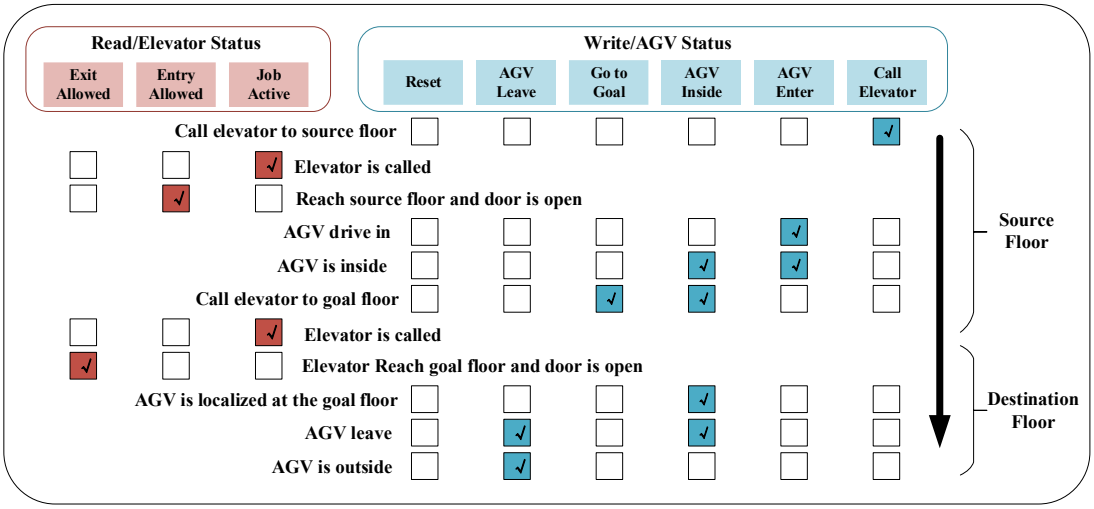


Figure 71 Interaction Process Flow

5.3.3 Design of Virtual PLC

Virtual PLC was briefly introduced in Section 3.3.2. It communicates with the AIC, Elevator, and IoT module. These three modules are loosely coupled, only invoking key functions and current states as needed. Figure 72 shows the Virtual PLC GUI. The specific connections between the three modules are explained as follows:

AIC Slave: Provides the mobile robot with the necessary elevator status, including the current floor and elevator door status. It guides the robot into the elevator and to the destination floor. The blue box in Figure 72 represents the section connected to the AIC software, acting as a slave to the AIC end. The IP and port are the local address of the virtual PLC, which listens for real-time messages from the AIC. The AIC treats the virtual PLC as the actual PLC for the elevator. It acquires the elevator status necessary for the mobile robots to use the elevator, primarily including the current floor of the elevator, the elevator door status, and feedback from elevator calls. The status changes are read from the holding register data. The refresh rate is 5 times per second. The information text box displays the current real-time interaction steps of the mobile robot, starting from calling the elevator before the robot enters it to when the robot completely exits the elevator, divided into six steps.

Elevator Master: Handles calling the elevator to the destination floor. The green box in Figure 72 acts as the master for the elevator, sending elevator call commands. The IP and port are the elevator's address on the laboratory's internal network. Since the elevator does not provide any information, this function area is minimal. Additionally, there are two function buttons: one for manually calling the elevator to a specified floor and another to release the elevator button. This release button can be used in rare conditions where the elevator button may stick. The information text box displays the floor to which the virtual PLC is calling the elevator.

IoT Module Master: Collects data from ultrasonic sensors, air pressure sensors, and Bluetooth Beacons to determine elevator door status and current floor. The red box in Figure 72 acts as the master for the IoT module, collecting real-time information from the IoT modules installed on three mobile robots. This includes distance data from ultrasonic sensors and air pressure data from air pressure sensors. Distance data is used directly to determine the status of the elevator doors. After complex calculations, the air pressure data determines the current floor of the elevator. The IP and port are local addresses. The sampling interval is 1s; too short an interval would overburden the IoT module and laboratory network, while too long an interval could cause high delays in robot movements. The data displayed in the box are key during the use of the elevator by the mobile robots. The information text box shows data being received from the IoT module.

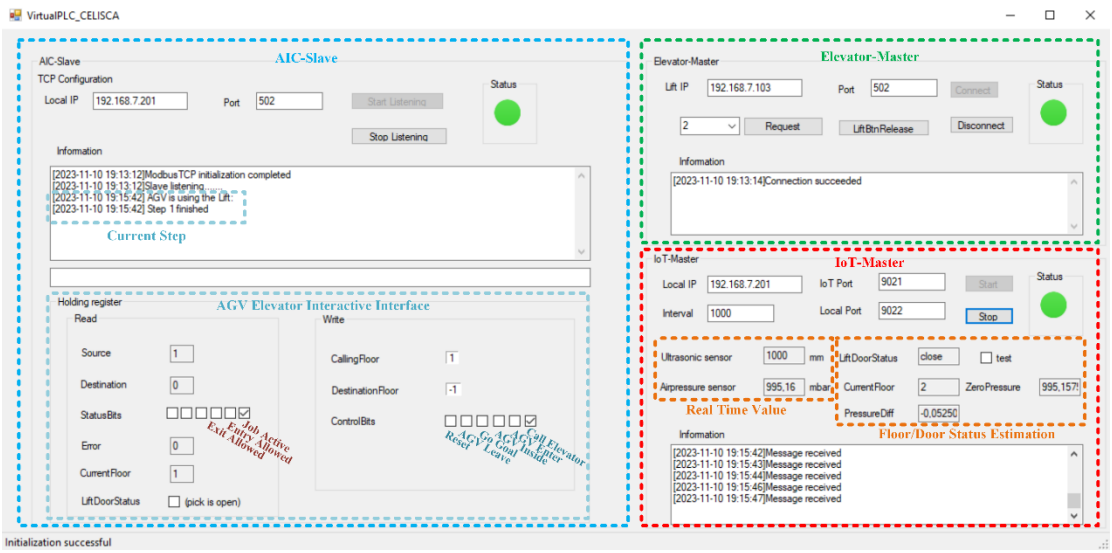


Figure 72 Virtual PLC GUI

Figure 73 shows the whole process logic of the virtual PLC assist mobile robot using the elevator.

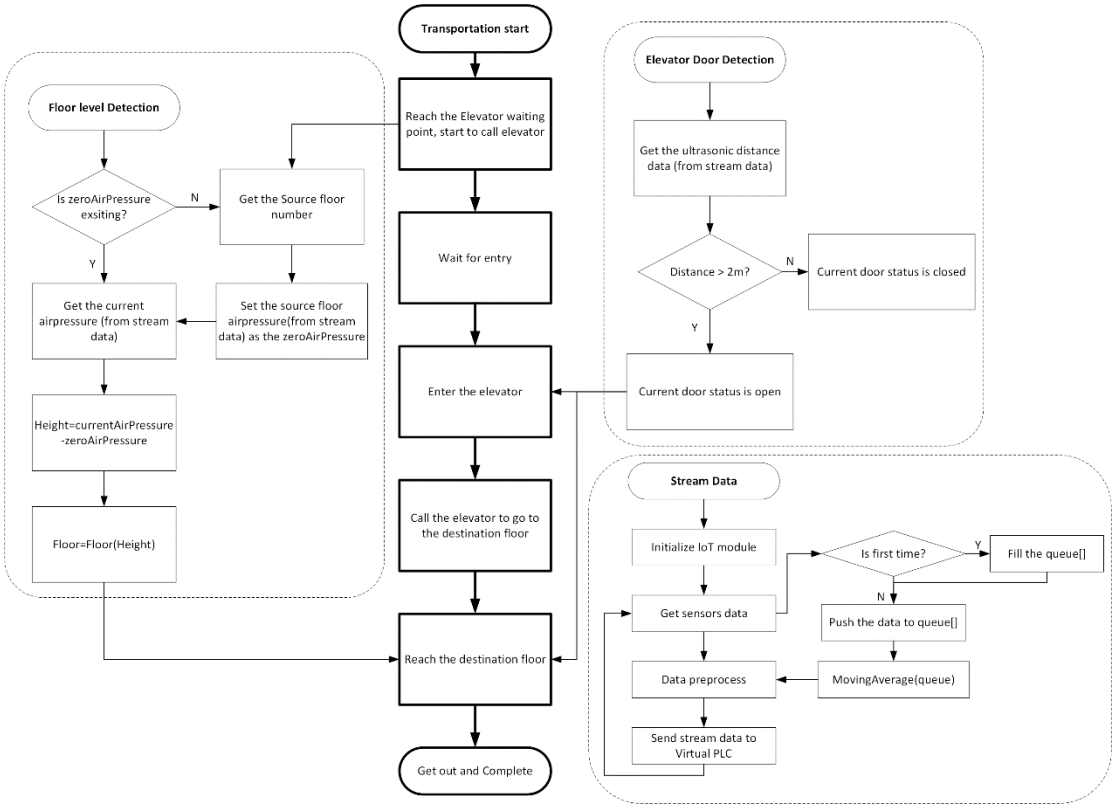


Figure 73 Virtual PLC Logic

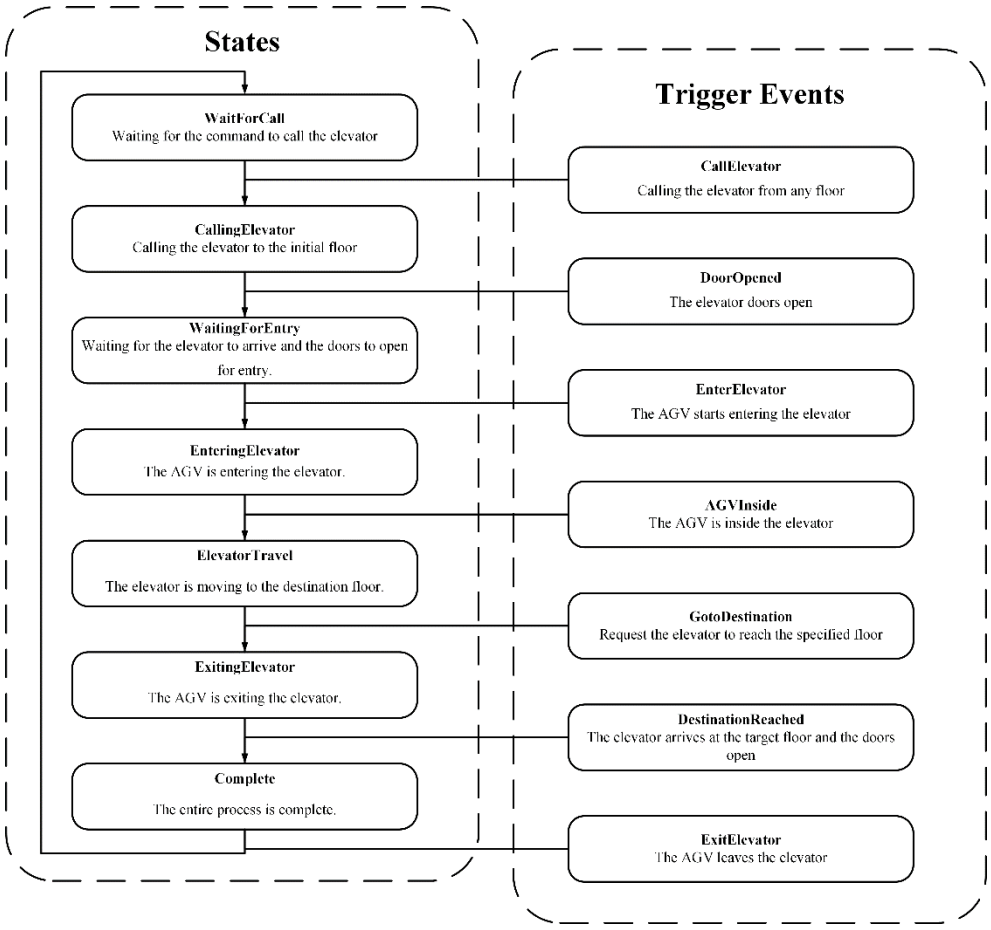


Figure 74 Elevator Interaction Processes Defined by FSM Principle

The design approach for implementing the interaction logic code is based on the FSM principle. FSM is a model used to design and implement computer programs, electronic systems, and logic networks, where the system's behavior is divided into a series of states. At any specific time, the system is in a particular state, and it transitions from one state to another based on inputs or events. Following the logic outlined in Section 5.3.2, the Virtual PLC employs an FSM to break down the interaction process into 7 states and corresponding 7 trigger events. The state transitions occur when specific trigger events are detected. For instance, to transition from the "WaitForCall" state to the next state, "CallingElevator," the "CallElevator" event must be triggered. This is illustrated in Figure 74. The code has been included in the appendix D.

5.3.4 Elevator Opening Detection

Before entering and after entering the elevator, as the robot waits for the doors to open, it is positioned to face the center of the elevator doors directly. This positioning allows the ultrasonic sensor to quickly capture the distance to the nearest obstacle. When the doors are closed, the distance recorded by the ultrasonic sensor from the robot to the elevator door is approximately 1,000 mm. Once the doors open, and assuming there are no other people or objects inside the elevator, this distance increases to about 4,400 mm. These distances fall within the accurate detection range of the ultrasonic sensor, minimizing the risk of false distance readings.

In the system, the door opening threshold distance of 2,000 mm. If the recorded distance exceeds this threshold, the elevator doors are determined to be open; otherwise, they are considered closed. Door status detection starts only when the elevator interaction begins, and finishes along with the floor estimation once the interaction ends.

The mobile robot uses LiDAR to detect obstacles in real-time. If the elevator doors begin to close while the robot is entering, it will stop and wait for the doors to open again. If the elevator is blocked by people or objects, the robot will wait for enough space to clear before proceeding. Additionally, if someone manually closes the elevator doors and go to the destination, the robot will wait for the next turn elevator. It does this by detecting sufficient space and making subsequent judgments.

5.3.5 Floor-Level Detection

The only elevator in the CELISCA is used by everyone, which means that it is inevitable for the mobile robot to encounter other people using the elevator and going to different floors. Accurate floor estimation is thus essential. Since the geometric structure of different floors is similar, it is challenging for laser sensors to distinguish between floors correctly. The elevator has a semi-outdoor structure that allows sunlight to stream in

through the glass, and the interior has highly reflective surfaces, making visual methods particularly challenging. Hence, a floor-level detection system is developed based on air pressure and BLE beacon for floor-level detection.

5.3.5.1 Air pressure-based Detection

Air pressure sampling is conducted across all floors of the CELISCA building. The IoT module is installed on the mobile robot, as shown in Figure 70, and the robot remains on each floor for three minutes. During this time, the air pressure sensor in the IoT module continuously records the air pressure values of the current floor. This dataset covers three minutes of air pressure data across five floors, but due to network fluctuations, the number of samples collected did not reach 180, only 171. Two clear characteristics can be observed from the Figure 75 air pressure data: a) the separation of air pressure between floors is distinct, with almost no overlap or confusion. b) The air pressure readings fluctuate consistently, with small upper and lower bounds. The air pressure readings fluctuate consistently, with small upper and lower bounds. The difference between the maximum and minimum air pressure readings on each floor stays around 0.3 mbar, except for an extreme case on Floor 3, where the difference reached 0.6 mbar. This indicates that determining floor levels based on air pressure sensors is theoretically feasible, but further optimization is still needed.

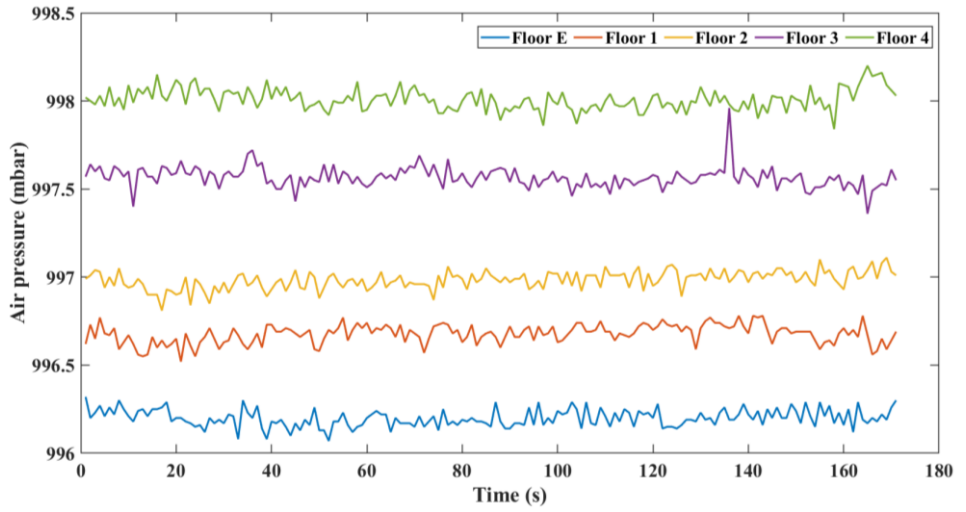


Figure 75 Raw Air Pressure Data

Additionally, continuous 3-day sampling was conducted. Figure 76 shows the air pressure records of five floors over three consecutive days, where the bar graphs represent the average air pressure values. The error bars correspond to the maximum and minimum air pressure values recorded.

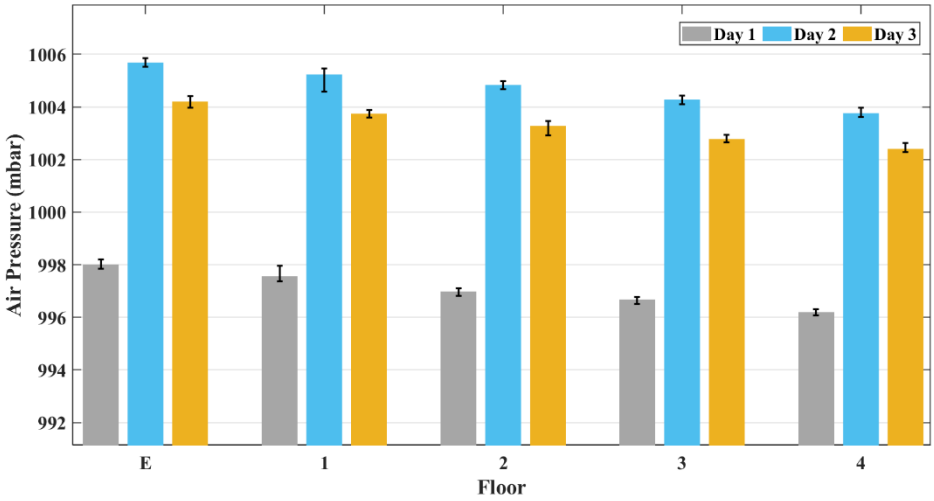


Figure 76 Three-Day Air Pressure Record

As seen in Figure 76, absolute air pressure changes every day, with readings on the same floor differing significantly from day to day. The air pressure value on Floor E from Day 1 is even lower than that of Floor 4 on the other two days. Therefore, absolute air pressure values are not suitable as a standard reference for floor levels. It is common knowledge that air pressure is affected by weather, temperature, and humidity, which can vary throughout the day. However, the study found that the pressure difference between floors within the same measurement is fixed and relatively stable, which led to consider air pressure differences as a method for floor level estimation.

The data displayed in Figure 75 is subject to fluctuations, which could be caused by several factors: a) Good ventilation conditions in the elevator shaft may result in fluctuating air currents. b) Minor mechanical errors and noise inherent to the barometer are unavoidable. c) The presence and movement of people in the laboratory may also affect air pressure readings. Although the overall noise generated is small, Figure 75 shows data from Floor 3 (purple data) encroaching into the air pressure range of Floor 4 in the latter half. To eliminate the impact of noise, a filter to sift through high-frequency noise is considered. Using the data from Figure 75, Figure 77 demonstrates the effect of applying a moving average, which significantly stabilizes the data. The gaps between floors become more consistent, and the smoothed data aids in subsequent floor estimation. The moving average method employed by the system can be expressed with the following Formula (4):

$$\bar{x}_t = \frac{1}{N}\sum_{k=0}^{N-1} x_{t-k} \tag{4}$$

Where, $\bar{x}_t$ represents the moving average of the air pressure at the time t, and x_t represents the air pressure reading at the time t. N represents the size of the moving average window. In this study, $N = 4$. This means that the moving average at any

given time t is the arithmetic mean of the air pressure readings at the current and previous three time points. Due to the volatility of the air pressure readings, a moving average that is too large would delay the determination of the arrival at the designated floor, while a window that is too small would not effectively filter out noise, hence the choice of 4. This method effectively smooths out short-term extreme fluctuations, providing a clearer perspective for floor analysis, as it does not overly delay the determination of the target floor while maintaining a sufficient level of noise filtering.

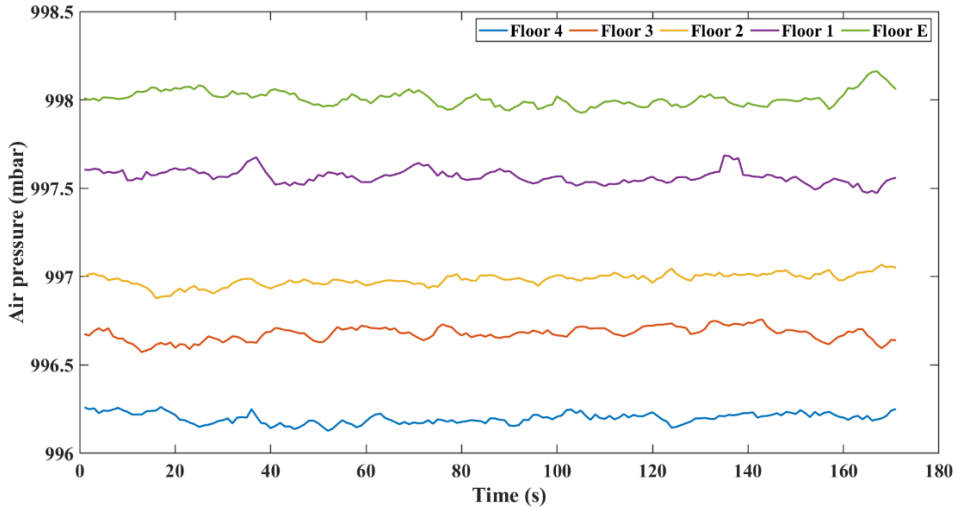


Figure 77 Air Pressure Data after Applying Moving Average Filter

To ascertain the floor level based on air pressure, a comparative analysis was conducted. It is articulated as follows:

Let $p_{f,t}$ as the air pressure reading on floor f at time t, and $p_{2,t}$ as the air pressure reading on floor 2 at the same time. The air pressure difference $d_{f,t}$ for each floor relative to floor 2 can then be calculated as follows:

$$d_{f,t} = p_{f,t} - p_{2,t} \tag{5}$$

To determine the average air pressure difference for each floor, calculate the mean $\overline{d}_f$ and standard deviation σd_f of the pressure differences as follows:

$$\overline{d}_f = \frac{1}{N}\sum_{t=1}^{N} d_{f,t} \tag{6}$$

$$\sigma d_f = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}(d_{f,t} - \overline{d}_f)^2} \tag{7}$$

Where, N represents the number of pressure difference measurements for floor f.

To create a practical and reliable range of pressure differences for each floor, define the pressure difference range R_f as:

$$R_f = [\bar{d}_f - k \cdot \sigma d_f, \bar{d}_f + k \cdot \sigma d_f] \tag{8}$$

Where, k is a coefficient that determines the width of the range based on the standard deviation. In this study $k = 2$.

To determine the air pressure difference ranges for each floor, an approach was taken to calculate the mean air pressure difference and its standard deviation in relation to the second floor. Based on these calculations, established a reasonable range using a coefficient k to determine the robot's floor location with a margin of error. By setting $k = 2$, allowed for a range that accommodates variations within half a standard deviation from the mean, providing a balance between sensitivity and robustness of the floor estimation process. Upon the foundation of these calculations, the ranges were fine-tuned based on a comparison with data collected during actual operation. The resulting ranges for each floor relative to the second floor are illustrated in Figure 78 below.

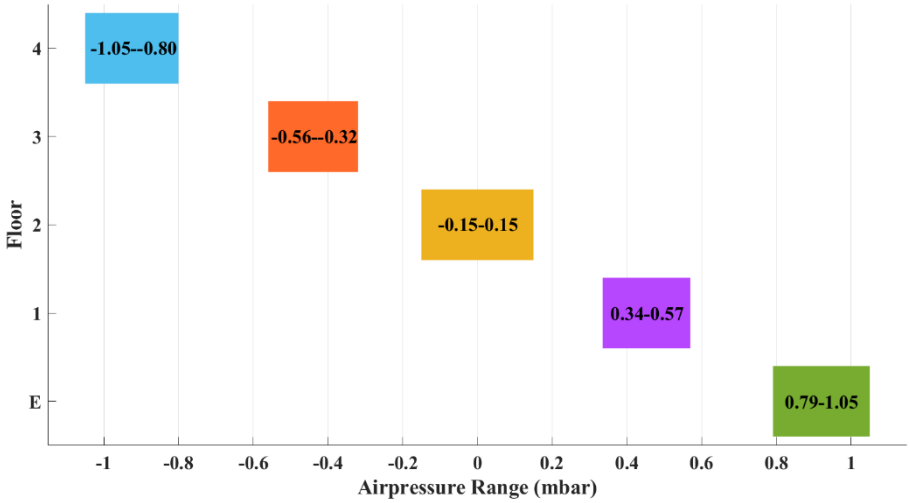


Figure 78 Air pressure difference ranges relative to the second floor

5.3.5.2 Air Pressure Estimation Accuracy Assessment

The first experiment conducted floor estimation for five floors over three consecutive days. The air pressure difference ranges were based on the second floor, as shown in Figure 78. The measurements were taken in sequence, starting with zeroing the air pressure at the second floor. The route was 2-4-3-E-1-2, staying for three minutes at each floor. The data were directly exported from the Virtual PLC. The accuracy rate was determined by comparing the estimated floors with the actual floors. The results are shown in Table 14.

Table 14 Floor Estimation Accuracy for Mobile Robot Over Three Days

Floor	Accuracy Rates (%)		
	Day 1	Day 2	Day 3
4	100.00	100.00	100.00
3	100.00	98.83	100.00
2	88.89	100.00	80.12
1	96.49	99.42	87.13
E	98.83	99.42	90.64

There were two types of errors: failure to recognize the current floor and misidentification of the floor. In this experiment, all errors were due to failure to recognize the current floor; there were no cases of misidentification, i.e., the air pressure difference was not within the range for any of the five floors, and there were no instances where the system assessed the robot to be on the third floor while it was actually on the second floor. The robot's logic to confirm floor arrival was to consider it had reached the target floor when the floor estimation matched the target and the door was open. This means that as long as the mobile robot reached the target floor, the floor detection was correct at any moment and the elevator door was open, it was considered to have arrived at the correct destination and began to leave the elevator. Typically, it takes only one minute for the mobile robot to travel from the start floor to the destination floor. Within one minute, under most weather conditions, there would not be significant changes in absolute air pressure. However, in this experiment, the time span reached 25 minutes. The longer the time, the greater the likelihood of changes in absolute air pressure. Therefore, the first floor reached had a higher success rate in floor detection because as long as the correct floor was detected once within the time frame, it was determined to have reached the correct floor. Therefore, the errors in this experiment did not affect the determination of reaching the target floor.

Additionally, a pattern emerged where the accuracy rate trended as 4>=3>E>1>2, consistent with the order of measurements. A significant factor is that air pressure fluctuates over time. Reliance on air pressure difference for floor determination is stable in the short term, but the accuracy of this method began to decline after just six minutes. The complete interaction process between the robot and the elevator is completed within 90 seconds. In the short term, the air pressure difference is stable and reliable. Moreover, with each elevator interaction, the air pressure is zeroed again, obtaining a new baseline air pressure based on the second floor. Expanding the range of air pressure difference can also increase accuracy.

5.3.5.3 BLE Beacon-based Floor Detection

BLE beacons are used to assist in floor detection alongside air pressure data. Although the positioning accuracy of BLE Beacon 4.0 is not very high, only meter-level, but it is sufficient for floor estimation. Figure 79 shows the installation of beacons in the elevator area. Each elevator door on each floor has a BLE beacon installed above it. The primary operating floors for the MOLAR robot are the 2nd and 3rd floors. The IoT module is installed on the mobile robot, and the Bluetooth module is used to receive the signal value from the beacons. It can receive the signal value of up to 6 beacons at the same time.

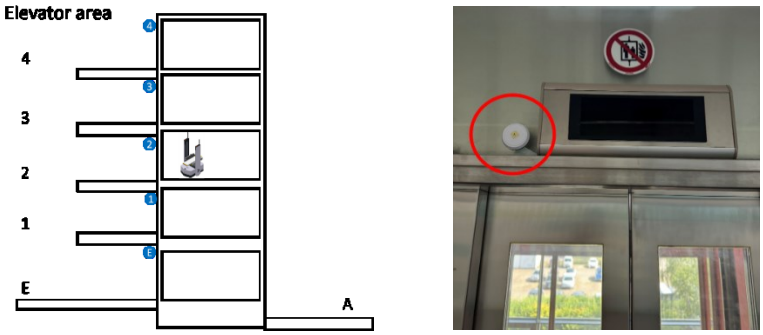


Figure 79 Layout of BLE beacons in the elevator

First, a distance experiment was conducted with the BLE beacons. In this experiment, one beacon was placed above the door with a transmitting power of 4 dB. The IoT module attached to the robot was sequentially placed at distances of 0.4, 0.6, 0.8, 1.6, 2, 3, 3.5, 4, 4.5, and 5 meters away from the beacon. The experiment was repeated three times, with the RSSI measured twice per second.

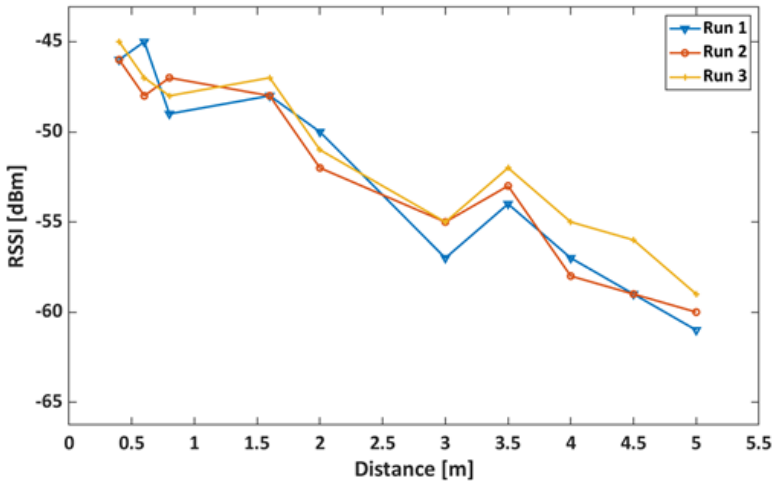


Figure 80 The relationship between RSSI and distance

The results in Figure 80 generally show that the RSSI decreases as the robot moves further away from the beacon. The standard deviations were less than 4% across the three runs. However, the actual signal strength at different positions can be influenced

by factors such as dense indoor environments, variations in the beacon's transmitting power, and obstacles in the surrounding environment. The resulting accuracy is sufficient for tracking samples and labware in a laboratory, even though it does not provide high accuracy within a centimeter range.

In the second experiment, 25 tests were conducted with the mobile robot operating in the elevator. The robot was manually controlled to move between floors: E -> 1 -> 2 -> 3 -> 4. The robot stayed on each floor for 30 seconds inside the elevator. The changes in signal strength from the beacons on each floor were recorded during the same time period. The results from one general case during test is shown in the Figure 81.

From the Figure 81, it is evident that upon reaching a particular floor, the signal strength of the beacon corresponding to that floor becomes the strongest. Using this proximity principle, it is possible to determine which floor the robot is closest to, aided by the information that the elevator doors have opened. The success rate of 25 tests is 100%. Success is defined as reaching the corresponding floor and receiving the maximum beacon signal value of the current floor.

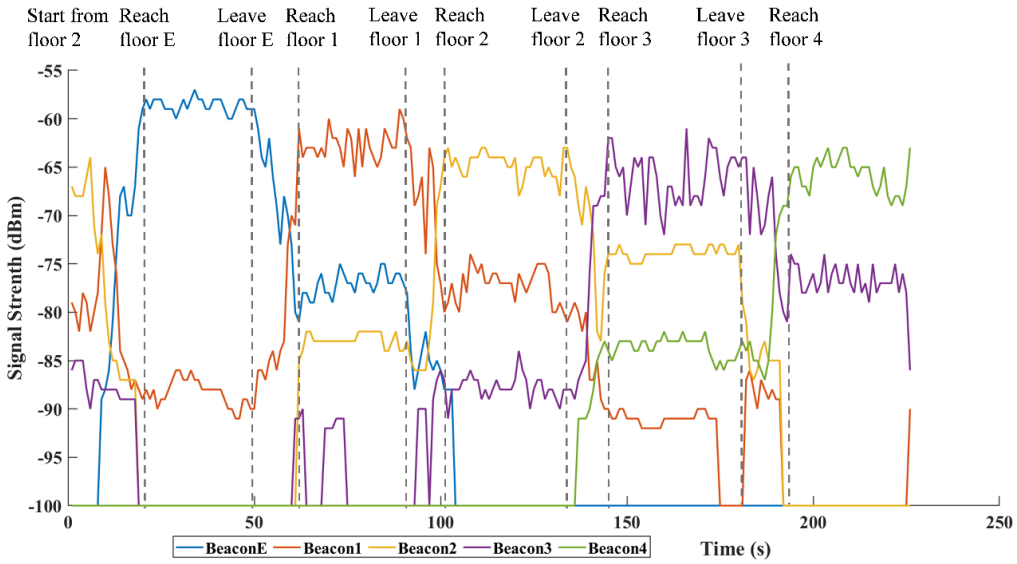


Figure 81 Signal Strength Changes of Beacons over Time

5.3.5.4 Hybrid Floor-Level Detection

In practical testing, using air pressure changes to obtain floor information is a stable and quick method. The only exception occurs under extreme weather conditions, where rapid air pressure changes can lead to inaccurate floor estimation. In such cases, the error typically means the mobile robot has reached the 3rd floor, but the floor reading is still transitioning between the 2nd and 3rd floors, making it difficult to determine that the

3rd floor has been reached. It does not mean that the air pressure sensor would incorrectly indicate the 2nd floor when the robot is on the 3rd floor. Therefore, the beacon is only needed as an auxiliary measure, with the primary floor detection still relying on the air pressure sensor. The basic logic is as follows:

```
if (floorBasedOnPressure != "10") {
    // If air pressure determined the floor
    CurrentFloor = floorBasedOnPressure;
} else if (floorBasedOnPressure == "10") {
    // If air pressure did not determine the floor
    CurrentFloor = floorBasedOnBeacon;
}
```

Where, "10" represents an error floor. When the floor cannot be correctly determined, "10" is displayed, which usually occurs when the elevator is between floors. Under normal conditions, the display would show "0", "1", "2", "3", or "4". In the software, there is an option to enable or disable the beacon assistant. "floorBasedOnPressure" is obtained by the air pressure sensor in the IoT module as described in Section 5.3.5.1, and the current floor value is obtained according to the air pressure difference calculation method. "floorBasedOnBeacon" is obtained by the Bluetooth communication module of the IoT module. The signal value of the beacon installed at each elevator entrance is obtained, and then the signal strength is sorted. The floor corresponding to the beacon with the strongest signal is the current floor. After the Virtual PLC obtains the two data, it deduces the current floor together according to the above logic. In the software, there is an option to enable or disable the beacon assistant.

In summary, the system uses air pressure to determine the floor at priority. When it fails, it switches to using the beacon. Currently, this hybrid floor-level detection system has been in normal operation in the laboratory for half a year. No errors have occurred during the floor detection process.

5.3.6 Multi-Robot Interaction with Elevator

Since the Virtual PLC and IoT module are not integrated into the mobile robot system, it is impossible to determine which robot is calling the elevator when multiple robots are operating simultaneously. The elevator usage signal sent by the AIC does not include information about the specific robot, making it unclear which IoT module's real-time data should be used to determine the elevator door status and current floor.

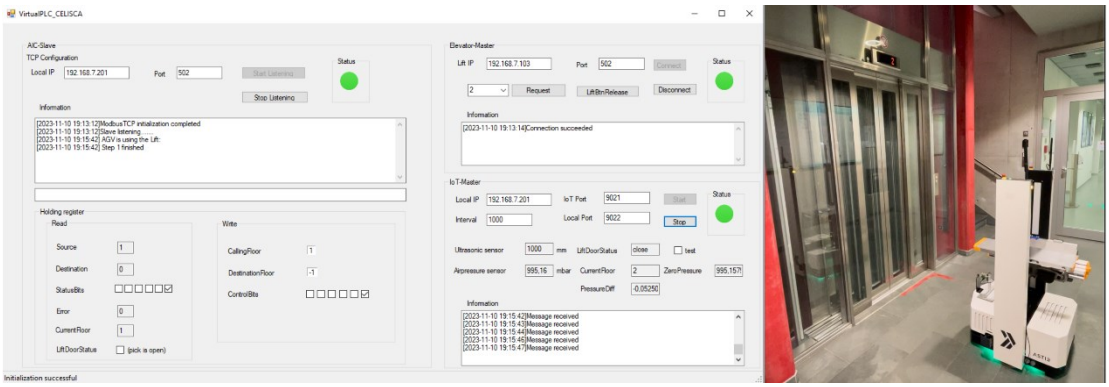
To solve this problem, two solutions were considered. The first solution involves using

the nearest beacon to identify the mobile robot closest to the elevator door beacon as the "robot using the elevator." In the FMS system, if multiple robots need to use the elevator, the system queues them based on the order of their requests. While this solution is easy to implement in code, it has practical issues. If other robots are also near to the elevator, the beacon signal fluctuations can result in a measurement accuracy of over 1 meter, which can lead to interference and uncertainty about which IoT module's data to use.

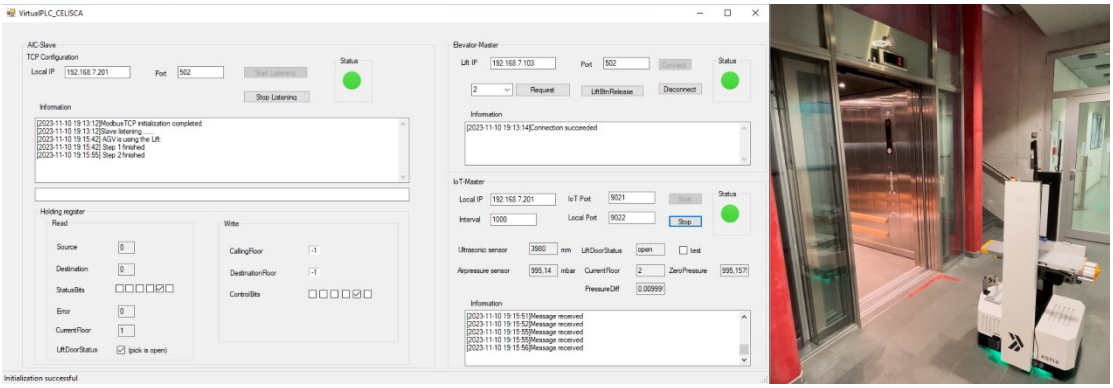
The final selected method involves the Virtual PLC sending a command to the AIC service via the HTTP protocol to obtain the status of the mobile robots upon receiving the door opening instruction from the AIC. This status command includes various parameters and states of the mobile robots, including their x, y coordinates, and orientation. By comparing the positions of all robots with the waiting points at the elevators on the 2nd and 3rd floors, the robot within the designated area is identified as the "robot using the elevator." The IoT module readings from this robot are then used to detect the elevator door status and floor level, thus eliminating interference from multiple robots being in the elevator area simultaneously.

5.3.7 Comprehensive Test

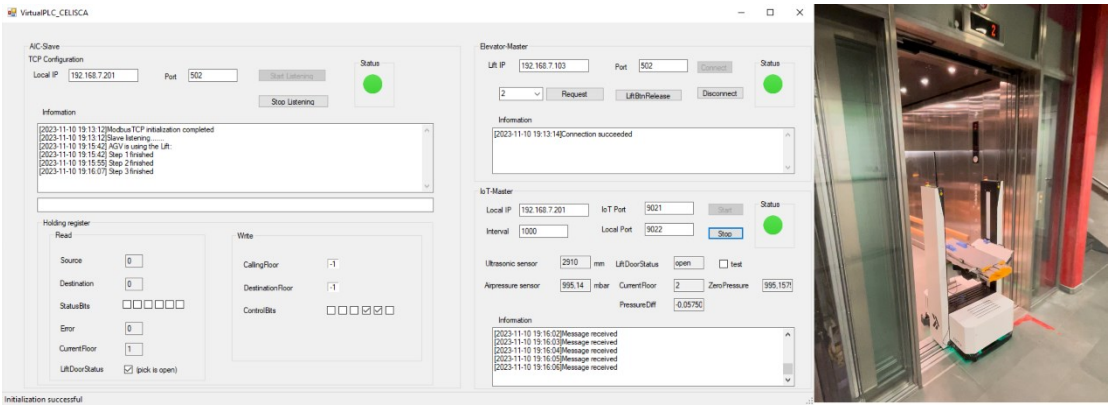
The experiment tested the performance of the entire interaction process. It involved directing the mobile robot from the second to the third floor, which required using the elevator. Figure 82 illustrates six key steps during the elevator interaction, correlating the Virtual PLC state with the actual interaction status between the mobile robot and the elevator. This experiment was repeated 100 times, traveling from the second to the third floor and back again. The success rate for completing the elevator interaction was 100%, demonstrating the system's stability.



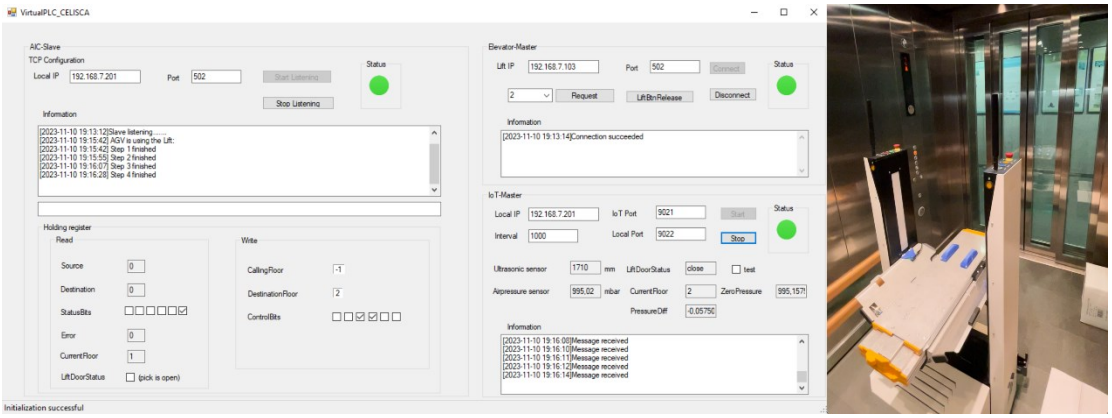
a) The mobile robot arrives at the elevator and initiates the call, activating the floor estimation system and door status detects.



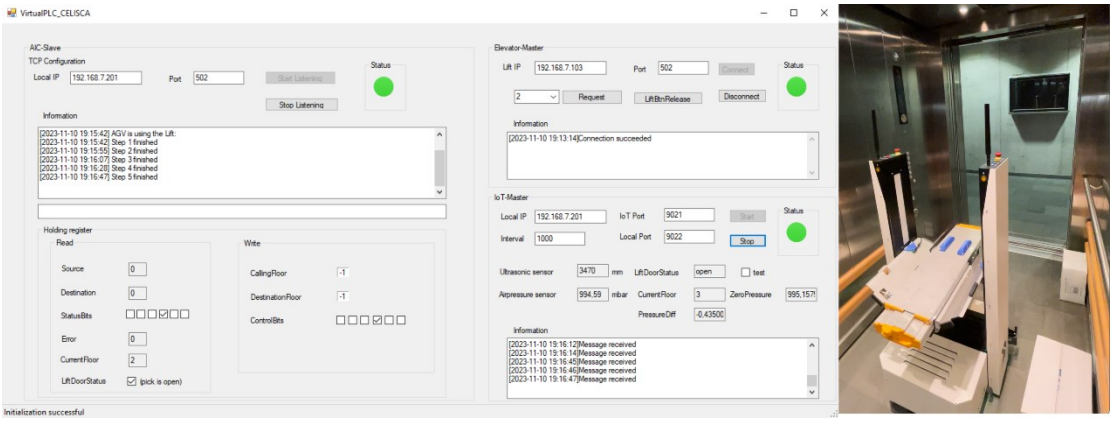
b) When the elevator reaches the source floor, the second floor, the IoT module detects the elevator doors opening, indicating the mobile robot to enter.



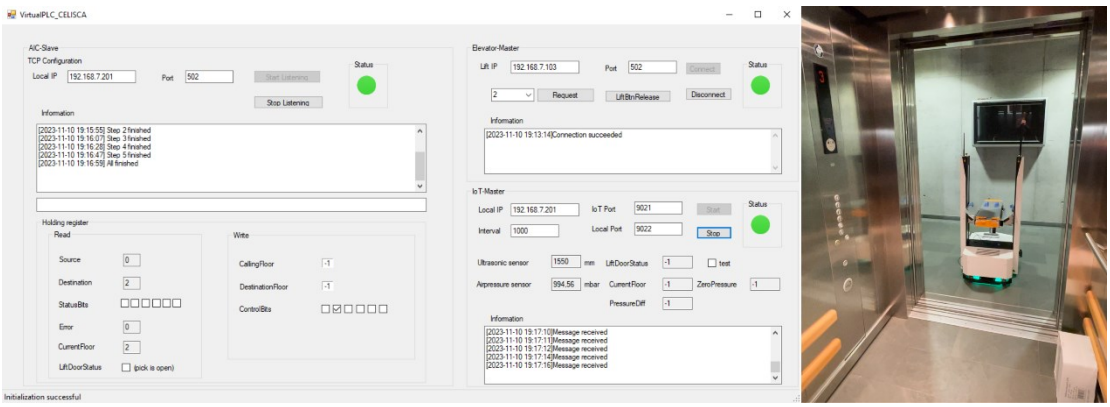
c) The mobile robot is entering the elevator.



d) Once inside, the mobile robot positions itself and calls the elevator to the target floor.



e) Upon arrival at the third floor, the IoT module detects the elevator doors opening, and the floor estimation confirms the third floor, indicating the mobile robot has reached its target floor.



f) The mobile robot exited the elevator, concluding the interaction, and the floor estimation and door status assessment systems are deactivated.

Figure 82 Complete Interaction Process between MOLAR robot and the Elevator

5.3.8 Error Handling Test

The elevator utilized by the mobile robot is a common resource in the building, heavily frequented by staff and used for transporting goods. This frequent usage poses several challenges, potentially disrupting the robot's smooth operation. To ensure reliable functionality, we've identified typical scenarios that may hinder the robot's interaction with the elevator and developed corresponding solutions:

- **Blocked entry**: The elevator may be fully occupied by people or goods, preventing the robot from entering. To manage this, a timed mechanism is implemented. If the robot cannot enter the elevator due to blockage, it triggers a re-call every 25 seconds, allowing the robot multiple attempts to enter the elevator.
- **Robot and people inside the elevator**: The elevator may stop at floors that are not the robot's destination due to use by other people. In such cases, the robot's control system, using real-time floor estimation, ensures that the robot remains in the elevator until the desired floor is reached. It does not exit prematurely; instead, it

commands the elevator to continue to the correct floor.

- **Air pressure variability in extreme weather**: Fast Changes in weather can lead to significant shifts in air pressure, affecting the accuracy of floor detection. To counteract this, Bluetooth Beacons have been installed at each elevator door to provide a backup floor localization system. When air pressure data is unreliable, the robot uses signals from the nearest beacon to determine its current floor.

5.3.8.1 1. First Test: Blocked Entry

This test simulates a scenario where an object blocks the mobile robot from entering the elevator. After people move away or the object is removed, the elevator doors have already closed, and the mobile robot fails to enter in time. The Virtual PLC then reopens the elevator doors, allowing the mobile robot to enter successfully.

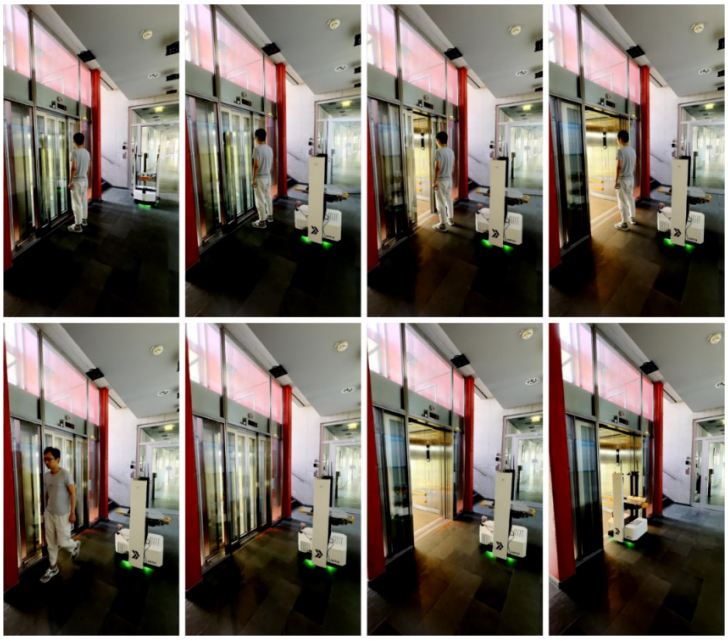


Figure 83 Elevator error handling test 1

The specific test scenario is shown in Figure 83. This scenario was tested 20 times, with the mobile robot successfully entering the elevator after each obstruction was removed. If an obstruction persists, the robot stays there, waiting until a staff member addresses the issue.

Simulation of an Object Blocking the Mobile Robot's Entry into the Elevator. The Mobile Robot moves to the front of the elevator. Someone blocks the Mobile Robot's entry into the elevator. After the person leaves, the elevator is closed. The Mobile Robot calls the elevator again, and the elevator door reopens, allowing the Mobile Robot to enter the elevator smoothly.

5.3.8.2 2. Second Test: Occupied Elevator

This experiment simulates a scenario where personnel stand at the front of the elevator, blocking the mobile robot from entering. The robot's LiDAR continuously scans the environment; if it detects an obstacle in its path, the robot will stop, preventing the interaction with the elevator from progressing to the next stage. The Virtual PLC will continue to call the elevator every 20 seconds until the elevator arrives again with sufficient space or a timeout of 10 minutes occurs, after which the mobile robot will enter. This scenario was tested 20 times, and each test was successful. However, it is crucial that the obstacle remains near the front of the elevator. If the obstacle is at the rear, the mobile robot will still enter.

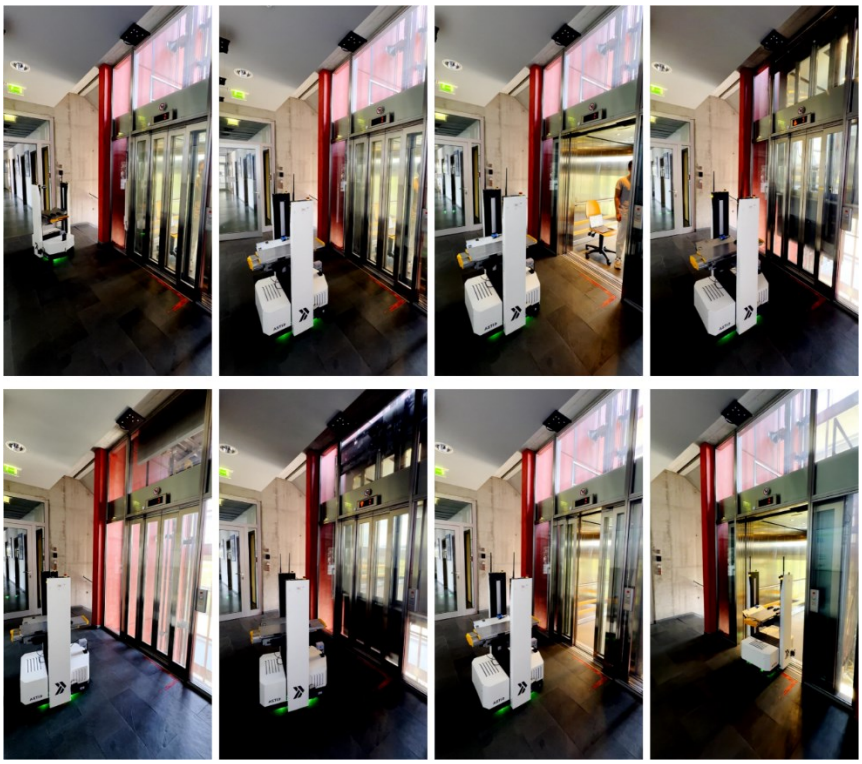
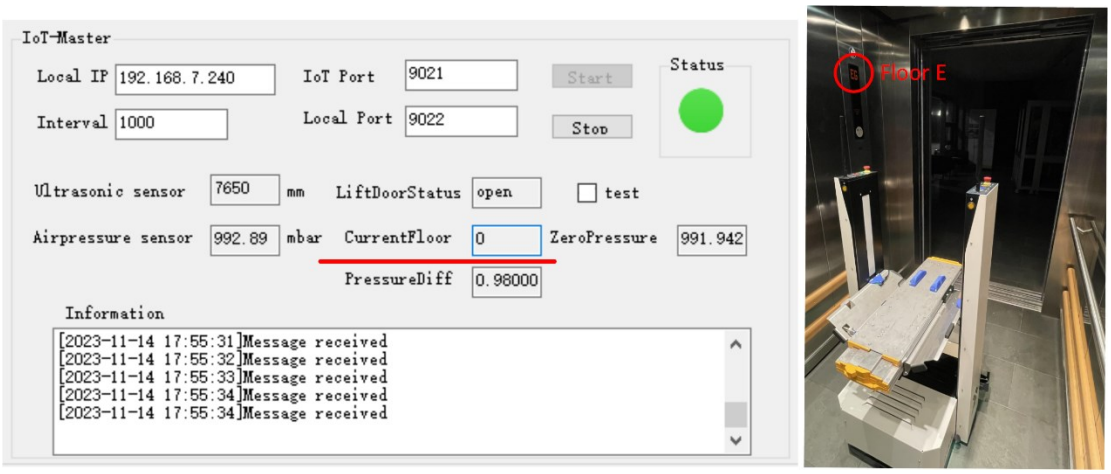


Figure 84 Elevator error handling test 2- Simulation of Elevator at Full Capacity

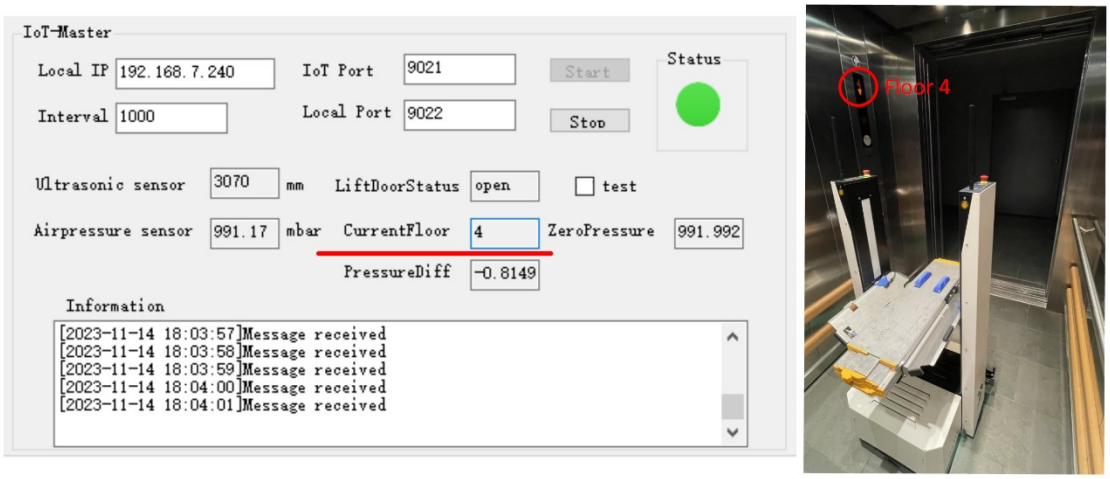
5.3.8.3 3. Third Test: Wrong Floor Detection

This test simulated a scenario where the elevator stops at the floors that are not the mobile robot's intended destination, due to the presence of other people using the elevator. The robot's transportation area was between the second and third floors of the laboratory building. Therefore, the system's response when the elevator arrived at floors E and 4 was tested. Upon the elevator reaching a floor apart from the intended one, the mobile robot detected the opening of the elevator doors. However, since the current floor did not match the target floor, the robot determined it had not yet arrived at the destination. Thus, it remained inside the elevator until it reached the correct floor, at

which point the robot successfully exited. This scenario was tested 20 times with a 100% success rate.



a)



b)

Figure 85 Elevator error handling test 3- Elevator Reaches the Wrong Floor, a) At the wrong Floor E, which corresponds to level 0 in the Virtual PLC system; b) At the wrong Floor 4, which corresponds to level 4 in the Virtual PLC system

5.3.9 Analysis

The series of tests described in this Section were conducted to evaluate and analyze the interaction between mobile robots, automatic doors, and elevators within a controlled laboratory environment. Each experiment was specifically designed to assess different aspects of the system's performance and reliability.

The Automatic Door Interaction Test successfully demonstrated that the mobile robot can effectively interact with the door PLC using the AIC system. Laboratory-wide WiFi coverage ensures that there are always at least two routers covering any given location, ensuring full signal coverage and reducing the likelihood of disconnections or delays.

The door opening and closing logic is designed to ensure that the robot can pass through without getting trapped by the door. Even if the mobile robot malfunctions, it can still close the door using the AIC. When the door is used by others, the robot verifies that the passage is clear before proceeding, and during this process, the door remains open. This ensures that the mobile robot can autonomously and smoothly enter different rooms and areas.

The Elevator Interaction Tests included multiple scenarios designed to evaluate the system under normal and error conditions. These tests were critical in validating the integrated system's functionality, including air pressure estimation accuracy, comprehensive interaction, and error handling:

The three-day precision assessment validated the reliability of the air pressure-based floor estimation system. Despite highlighting minor fluctuations in air pressure over 20 min that can affect accuracy, the tests demonstrated that air pressure remains very stable in the short term. The entire elevator ride process with the mobile robot taked less than two minutes. During this period, the method proves robust as air pressure is recalibrated with each elevator use.

The comprehensive interaction process involved the mobile robot using the elevator and demonstrated a 100% success rate, underscoring the robustness and stability of the interaction system. The Virtual PLC communicates simultaneously with three modules (the AIC, IoT module, and elevator PLC) performing logic control of the interaction process. The Virtual PLC software runs continuously on a PC without experiencing any crashes.

The error handling test provided insights into how the system manages common real-life scenarios. When an external obstacle blocks the mobile robot, the Virtual PLC repeatedly calls the elevator every 20 seconds. This low frequency ensures that the mobile robot does not impede others from using the elevator for transporting goods or disrupt its operations. Waiting for the next elevator when it is full and accidentally going to the wrong floor are typical scenarios. The error handling design allows the system to effectively manage various real-world situations where humans and robots share the same elevator.

Overall, the experiments conducted have demonstrated that integrating mobile robots with automatic doors and elevator systems via the Virtual PLC and IoT modules is not only feasible but also reliable. The system has shown its capability to handle standard operations and unexpected scenarios efficiently. These results confirm the potential for such automatic systems to be applied in multi-floor, multi-room environments.

6 Chapter 6 SAMI Integration for Workflow Control

The full automation of laboratory processes requires more than just the integration of equipment and environment. It requires a central system to manage all operations effectively. CELISCA has developed a multi-level management system that coordinates scheduling across all integrated devices, allowing for unified planning across different functionalities and diverse workflows within various workstations. Mobile robots are critical to this system, acting as the operational links that complete the automation loop. This Section details the integration of the mobile robot system into the SAMI control framework, including how the AIC system's functional interfaces were programmed into SAMI and the corresponding control logic was designed.

6.1 Introduction

The objective is to establish a workflow management system of integrated mobile robots within a highly decentralized laboratory environment. Traditional automation lines often lack the flexibility needed to meet the complex and dynamic demands of scientific laboratories. To address this gap, CELISCA has developed a versatile, multi-level workflow management system based on the SAMI platform. This advanced system connects all laboratory equipment, requiring only that these devices possess some form of communication interface. Importantly, there are no stringent specifications or open-source requirements for the equipment. In practice, laboratory personnel simply prepare materials and labware in designated storage areas and input the necessary devices and operational sequences into the system.

The system autonomously generates optimal workflows for each experiment, encompassing everything from the transportation of raw materials to the final analysis of samples. It also provides real-time monitoring of each step in the experiment and collects feedback from the devices currently in use. In the event of a malfunction, the system can issue timely error alerts. A hierarchical structure proves particularly beneficial in automated laboratories, especially those with complex experimental workflows. This significance is amplified when managing large volumes of interconnected heterogeneous automation systems, which require frequent checks on various physical conditions like temperature, concentration, or sample analysis.

The objective of the workflow management system is to integrate mobile robots and PCS-controlled workstations within a highly distributed laboratory environment. As shown in Figure 86, each workstation is equipped with a specific PCS that controls the operational sequence and conditions of the experiments performed at that workstation. Mobile robots play a crucial role in this ecosystem by bridging the physical gaps between these PCS-controlled workstations. They are responsible for the efficient

transportation of raw materials, samples, and labware from one station to another. Transportation operations across different rooms and even floors.

This hierarchical and interconnected approach enhances the laboratory's capability to conduct more complex and varied experiments efficiently. By automating routine tasks and integrating sophisticated monitoring and control functions, the system frees up researchers to focus more on analysis and less on the logistics of experiment setup and execution, pushing the boundaries of what can be achieved in scientific research environments.

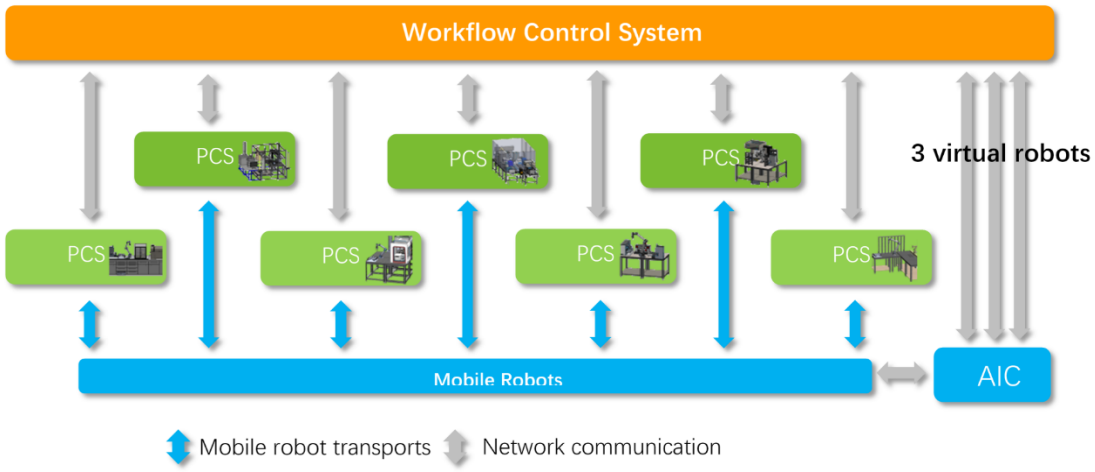


Figure 86 Workflow Control Structure

6.2 SAMI Management System Framework

Section 3.3.1 outlines the overall framework, presenting SAMI as the top-level control system for laboratories. The SAMI structure is effectively organized into a two-layer architecture, featuring a Global SAMI layer and several Sub SAMI layers. This hierarchical setup facilitates centralized management while allowing for specific device control, thereby supporting smooth operations within complex, automated research environments.

Global SAMI: This layer serves as the primary control center for mobile robots, overseeing task workflows, equipment statuses, and progress monitoring. It also provides visual representations of ongoing tasks. Two critical components within Global SAMI are "Keeper" and "Router."

Keeper: This is a comprehensive database that stores all pertinent information needed for the integrated system to function effectively. It encompasses everything from module-specific details to communication protocols and labware transportation information related to specific devices.

Router: This component is responsible for sending messages to designated Sub SAMI modules and the robot control software, known as AIC. It supports a wide range of communication protocols, including DDE, OLE, DCOM, ActiveX, and TCP/IP.

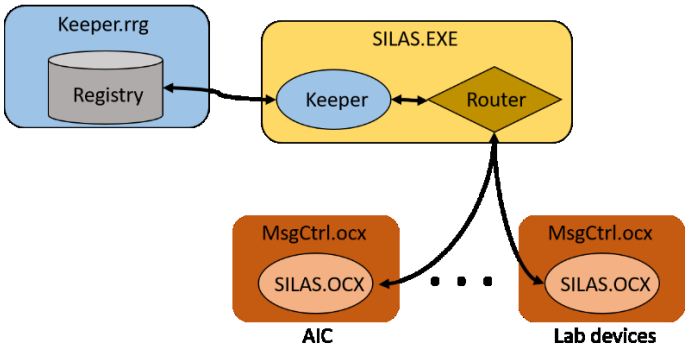


Figure 87 Architectural Overview of the SAMI System

Sub SAMI: Each independent workstation has its corresponding Sub SAMI. Sub SAMI modules integrate various types of device without specific API requirements and generate customized experimental solutions based on the lab devices in use. These solutions also include the estimated time required for completion, equipment preparation, and status checks.

Global SAMI utilizes the solutions generated by Sub SAMI to create an overarching workflow. While Sub SAMI can also generate workflows, it does so only for localized device-specific solutions. In contrast, Global SAMI has the capability to incorporate solutions from all Sub SAMI modules to construct a comprehensive workflow.

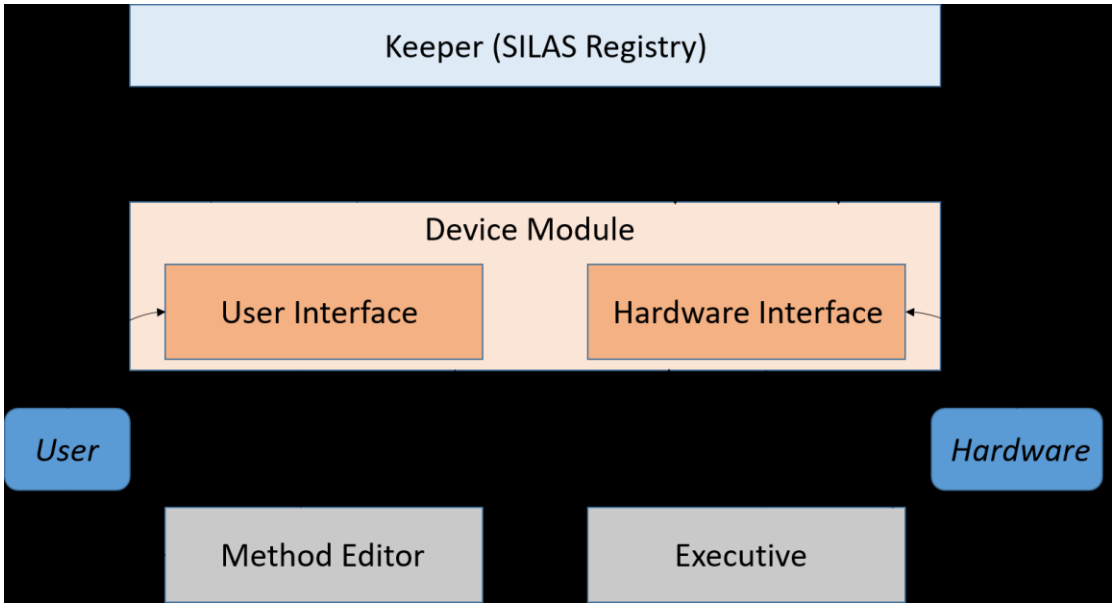


Figure 88 Interaction Framework of Modules in the SAMI System

Figure 88 illustrates the architecture of a modular automation system designed to facilitate seamless integration and communication between user interactions, software

components, and hardware devices. The system is structured to ensure efficient interaction and control across various layers, enhancing both flexibility and functionality.

Keeper (SILAS Registry), which functions as a central registry for all device modules within the system. It is responsible for managing critical functions such as fetching and setting data for modules, along with handling queries regarding module unloading and existence. This central registry ensures that all components within the system can be tracked and managed effectively, promoting system integrity and responsiveness.

The Device Module serves as the core interface between the user and the hardware, facilitating direct interactions and operational control. This module is bifurcated into two main interfaces:

User Interface: This interface allows users to directly interact with the system. It provides a configuration dialog for setting module parameters, action configurations for customizing device operations, and options to pause, resume, or abort operations. It also includes a status query feature that lets users obtain real-time status updates from the device.

Hardware Interface: This interface bridges the communication between the software and the physical hardware. It sends operational commands to the hardware and handles the responses, ensuring that commands are executed correctly and efficiently.

Users interact with the system through the User Interface of the Device Module, configuring settings, initiating device actions, and monitoring operational statuses. This direct interaction simplifies the process of managing complex workflows and device configurations. The hardware component comprises various lab devices that perform tasks based on commands received from the Device Module. These devices are integral to the system's operations, executing configured tasks as directed by the software.

Additionally, the system includes a Method Editor and an Executive, both of which are components of the previously introduced SAMI software. The Method Editor serves as a tool for creating and editing methods or workflows. It communicates directly with the Device Module, enabling users to send configurations, receive replies, and control the hardware effectively. The Executive functions as the execution engine of the system, running automated workflows created by the Method Editor and managing the tasks in real-time.

Overall, this system architecture not only promotes modularity and flexibility but also enables users to customize device operations through a centralized interface while maintaining precise control over the hardware behavior and system workflows. The centralized Keeper ensures that module data is consistent and well-managed across the

system, facilitating an efficient and responsive automation environment.

6.3 Robot Control API Access

In this section, the API's provided by the AIC system is discussed. Those APIs allow remote control and sending transport commands to mobile robots. The AIC's APIs are provided via HTTP protocol, supporting various operations including login, task assignment, task tracking, task cancellation, and retrieving robot status. These APIs enable the integration of mobile robots into the SAMI control system. All API endpoints are shown in Figure 89:

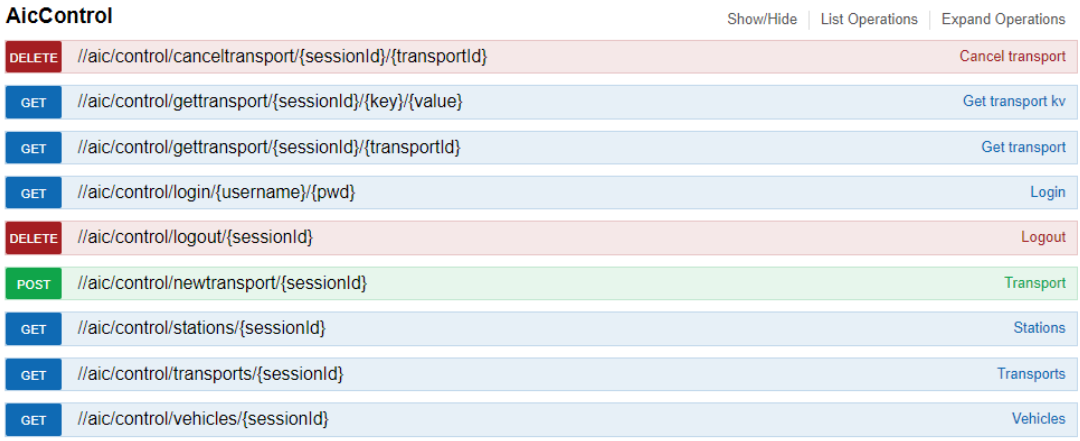


Figure 89 AIC HTTP APIs

Below are the commonly used API functions in the AIC system, that are crucial for efficient transportation scheduling of mobile robots:

- LoginAPI(): This method is used for administrator login. It provides the necessary session ID, which is required for all subsequent API calls.
- NewTransport(): This API assigns point-to-point transport tasks to mobile robots and generates a task ID for tracking the task execution status.
- TrackTransport(): Provides real-time updates on the tasks being executed by the mobile robots. It supports various task status updates, including queuing, scheduled, en route to pickup point, loading, loaded, en route to delivery point, unloading, completed, canceled, failed, and unknown status.
- CancelTransport(): Allows the cancellation of an ongoing transport task. This is particularly useful in emergencies or errors, allowing the immediate halt of the robot's movement.
- GetVehicleStatus(): Retrieves comprehensive status information of all mobile robots, including their current location, load status, availability, and battery level. This information helps the SAMI system confirm the availability of mobile robots during

the preparation phase to schedule upcoming transport tasks.

By implementing these functions into high-level control management and optimizing the related processes, the system ensures smooth workflow management and provides a high level of automation. Its real-time monitoring and control capabilities make it suitable for the needs of modern complex laboratories.

6.4 Upper Controller GUI

Based on the API's in Section 6.3, a MFC upper control program was developed using C++ to verify the practicality of these interfaces. Acting as the upper controller for the AIC software, it sends task commands to create transport orders and monitors the task status in real time. Tasks can be canceled at any time.

Figure 90 provides an overview of the communication class functionality between the global SAMI and AIC systems, detailing how these APIs implement the described features.

```
class ASTICommunication
{
public:
    ASTICommunication(void);
    std::string LoginAPI(const std::string& IPAddress, const std::string& textUser, const std::string& textPwd, std::string &Mes);
    std::string NewTransport(const std::string& IPAddress, const std::string& Source, const std::string& Target, const std::string& sessionID, std::string& Mes);
    std::string TrackTransport(const std::string& IPAddress, const std::string& sessionID, const std::string& taskID, std::string& Mes);
    std::string CancelTransport(const std::string& IPAddress, const std::string& sessionID, const std::string& taskID, std::string& Mes);
    std::string GetVehicleStatus(const std::string& IPAddress, const std::string& sessionID, std::string& Mes);
    std::vector<std::string> DivideMessage(const std::string& mes);
};
```

Figure 90 Communication Class based on API

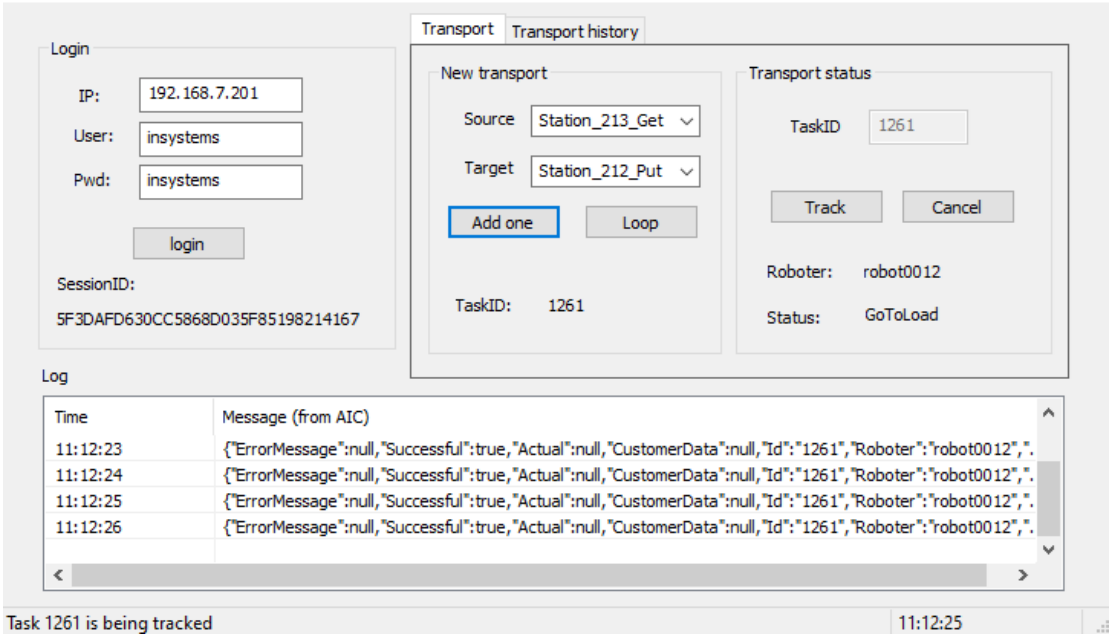


Figure 91 MFC Upper Control GUI

Figure 91 is the upper control software. From left to right, the interface includes five buttons, each representing a specific function:

- **Login**: Based on the input "IP", account and password, this function obtains

administrator access to the AIC and sends the subsequent commands using the SessionID. All other commands require the SessionID to be effective.

- **Add One**: Initiates the process of picking up labware from the source station and transporting it to the target station. The prerequisites are that a mobile robot is in auto mode in the AIC, a tray is placed at the source station by a human or robotic arm, and there is no tray at the target station. The mobile robot does not have a sensor to check for trays at the station, but plans include using an IoT module installed on the tray to track the real-time position based on the beacon.
- **Loop**: Allows the mobile robot to cycle between two stations for transportation. This function is used for repeated testing of stability, positioning accuracy, and laboratory demonstrations. It eliminates the need for staff to manually assign tasks, enabling the mobile robot to continuously perform transportation tasks in the laboratory. This function relies on two APIs: creating a new transport task and tracking the task status. Upon detecting task completion, the names of the Source station and Target station, along with the Get and Put suffixes, are swapped to reassign the transport task.
- **Track**: Queries the task status based on the TaskID. It displays both the current and historical task statuses, with 11 possible states ("Queued," "Scheduled," "GoToLoad," "Loading," "Loaded," "GoToUnload," "Unloading," "Finished," "Canceled," "Failed," "Unknown"). By default, it tracks the current task status.
- **Cancel**: Cancels the task based on the TaskID. If canceled before picking up the labware, the mobile robot will return to a nearby parking station. If canceled after picking up the labware, the mobile robot will stop in place, requiring manual mode for further human operation.
- **Log box**: Displays all JSON responses from the AIC system. The bottom left shows the current operation status, while the bottom right shows the current time. The second page records the history of each transport, including the start and end times and the outcome.

Figure 91 illustrates the process of sending a command and tracking the task. First, the login retrieves the current "SessionID" returned by the AIC. Then, a transportation command is sent, specifying to get the tray from station 213 and deliver it to station 212. After receiving the "taskID" from the AIC, real-time task status tracking begins based on this "taskID." Currently, robot 12 has taken the task and is on its way to station 213 to get the tray (GoToLoad).

6.5 SAMI Upper Controller

After validating the API with the MFC upper control program, it needs to be integrated into the SAMI software using C++. In the SAMI-related code, base functions for controlling devices have been defined through templates. To integrate the API communication class, developers need to embed it within these templates and override related virtual functions to adjust the logic for operating the mobile robot. Important virtual functions include:

Initialize(): Initializes the robot connection and settings, crucial for the robot's startup sequence.

Finalize(): Properly closes the robot connection and cleans up resources, ensuring a safe shutdown.

IsActive(): Checks if the robot is actively performing a task, critical for status monitoring and decision-making.

IsReadyToMove(): Determines if the robot is ready to start a motion, essential for ensuring readiness before commands are executed.

Status(): Provides detailed status information about the robot, which is vital for monitoring and troubleshooting.

Abort, Stop, Continue(): Control functions for handling emergency stops, regular stops, and resuming operations, crucial for operational safety and efficiency.

These functions provide a clear overview of how the robot interacts with its operational environment and handles different states and tasks. These functions demonstrate the comprehensive control capabilities built into the SAMI software for managing a mobile robot.

Once the communication class and functional logic are fully supplemented, these methods and logic will be called by the following SAMI GUI, which is universal. The two main features are "Prepare" and "Move". "Prepare" checks if communication is normal and if any robots are available for transport. Then "Move", similar to the MFC upper control software, sets the source and destination stations and then passes the task to AIC, starting task status monitoring. Tasks can be aborted during execution.

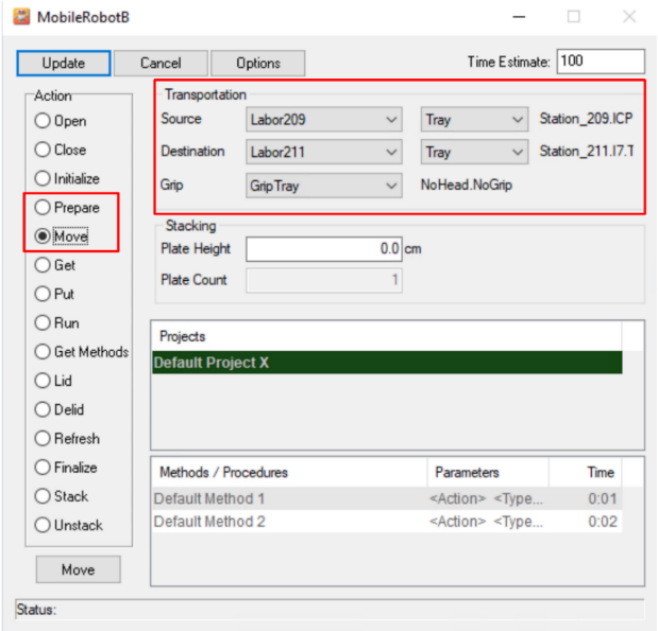


Figure 92 SAMI Robot Controller GUI

6.6 Applications

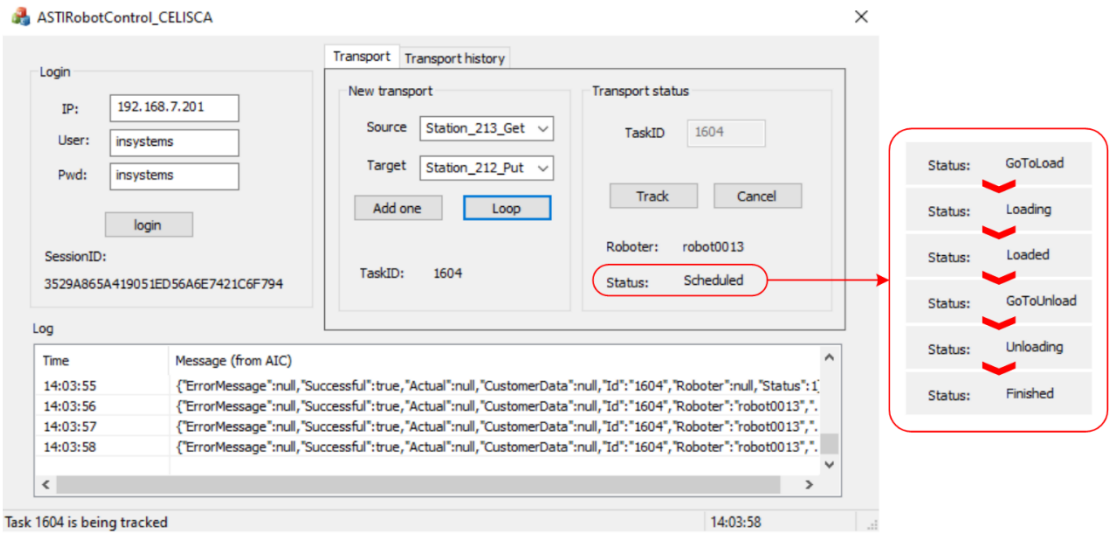
After integrating the mobile robot with SAMI, the mobile robot has been incorporated into CELISCA's control systems. The first application utilizes the MFC upper controller to create a Point-to-Point Transportation. The second application uses SAMI to generate a sequence of experimental steps, which includes actions on laboratory devices. The third application lets SAMI run two methods at the time, requiring the simultaneous operation of two robots. These applications demonstrate that mobile robots can perform transport tasks in the CELISCA laboratory and showcase their capability for collaborative operations among multiple robots.

6.6.1 Application I - Point-to-Point Transportation

This experiment demonstrates that a mobile robot can receive transportation commands from a remote upper control system to the AIC for point-to-point transport tasks. Using the MFC upper control software, the entire point-to-point transportation task is initiated, monitored, and completed.

The starting point is Station 213, and the destination is Station 212. It requires manual verification that a tray and labware are prepared at Station 213 and that there is no tray on the rack at Station 212. Then, the MFC upper control software sends a new transport command, including the source and target station information. Once a successful start response is received from the AIC, the MFC upper control software begins monitoring the current status of the corresponding task ID. Figure 93 a) shows the process of creating new transport commands and updating the status of the transporting mobile

robot. Figure 93 b) displays the AIC receiving the new transport order, creating a new task in the task queue, and assigning it to the appropriate mobile robot. At this time, only robot 13 is operational, so the two existing tasks are carried out by robot 13. The status of task ID 1604 is "Scheduled," corresponding to the status shown in Figure 93 a).



a)

Id	Created	Source	Target	Robot	State	Priority
1603	2024.07.02 13:59:59	Station_212_Get	Station_213_Put	robot0013	Unloaded	None
1604	2024.07.02 14:03:55	Station_213_Get	Station_212_Put	robot0013	Scheduled	None

b)

Figure 93 MFC upper control GUI sends the Transport order to AIC, a) MFC GUI; b) AIC

Figure 95 illustrates the actual execution process of robot's transportation task within CELISCA. The mobile robot starts from the waiting point near the charging station, first goes to Station 213 to get the tray, and then passes through the automatic doors between rooms 213 and 212 to reach Station 212 to put down the tray.

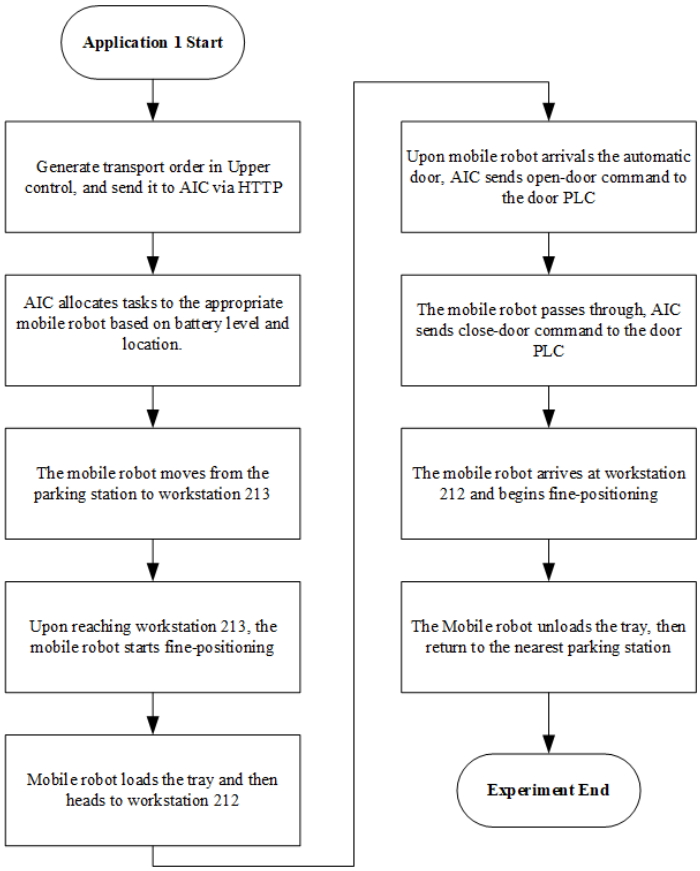


Figure 94 Flowchart of the mobile robot action of application 1

Figure 95 Transportation Processes from Station 213 to Station 212

The entire application 1 used robot 1, conducting a total of 50 tests with an average duration of 4 minutes and 20 seconds per test. Throughout the experiments, no errors occurred during the get/put of trays. In room 212, occasional network fluctuations caused the mobile robot to stop for about 20 seconds, waiting for the signal to recover before successfully completing the task. During the process of passing through automatic doors, the mobile robot exhibited shaking (minor directional adjustments when moving slowly through narrow areas). In the process of picking up and placing at workstation 213, the mobile robot experienced a failure in the first attempt at fine-positioning, followed by a successful second attempt.

The likely cause of the shaking phenomenon in the mobile robot is that at low speeds, the control loop maintains the same frequency, resulting in more frequent left and right controls to follow the path over the same distance. The high precision of path planning and the DWA used in local path planning may make minor adjustments to maintain the path or avoid obstacles, which are more noticeable in narrow or complex environments.

At workstation 213, the only one of the six laboratory workstations where the mobile robot performs pick and place operations, the area to the right is obstructed by easily movable chairs, which may affect the initial positioning for fine-positioning, leading to a failure in the first fine-positioning attempt.

6.6.2 Application II – SAMI Method Experiment

The experiments conducted focus on enabling mobile robots to execute multiple transport tasks, integrating other lab equipment as part of the overall workflow. Figure 97 shows the SAMI EX Editor in action during the experiment preparation phase. This graphical method editing interface allows for customized planning of sequential or parallel workflows. Upon establishing the workflow, an optimization algorithm is run to derive the most efficient schedule. As illustrated in Figure 96, the process follows a time-based coordinate system and is segmented into five steps:

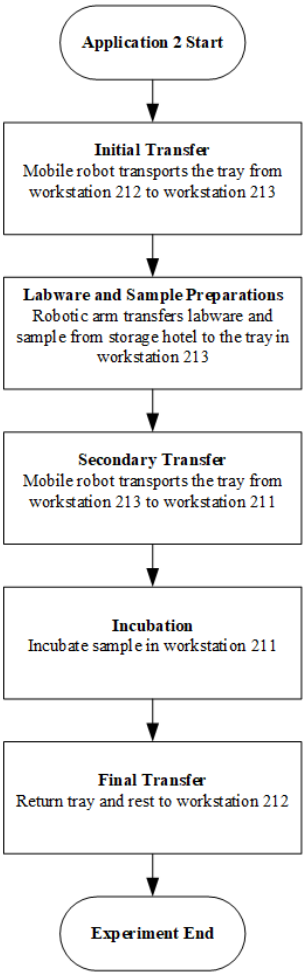


Figure 96 Workflow Flowchart of Application 2

The software allows for an adaptable and optimized coordination of multiple elements in real-time ranging from the mobile robots, various lab equipment, and even the detailed steps of scientific experiments. This leads to a maximized utilization of resources, streamlined operations, and an overall more efficient laboratory workflow.

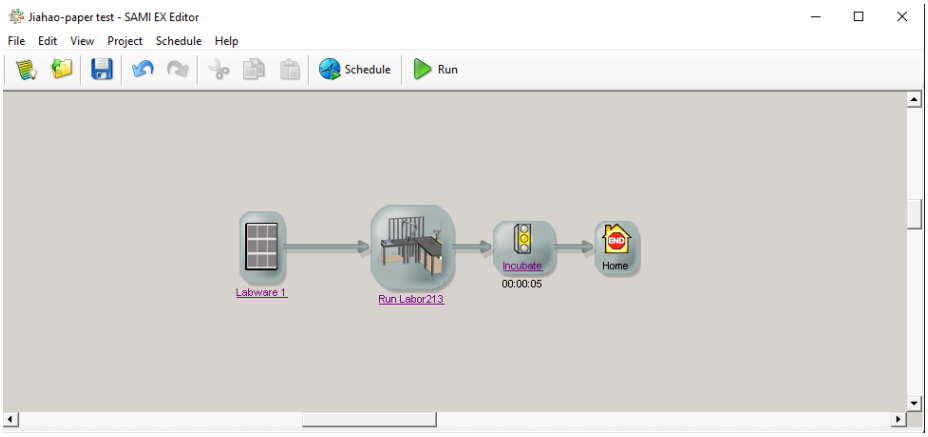


Figure 97 Workflow Setting in SAMI Editor (Application 2)

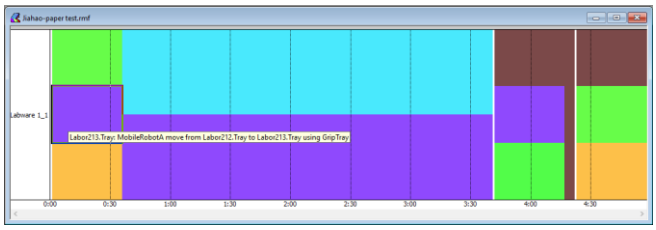


Figure 98 Generated workflow following successful scheduling (Application 2)

Once the optimal schedule has been determined, the workflow can be executed. Figure 99 provides a real-time view of the software interface during task execution, offering an overview of the interface:

Upper Box: Located at the top of the three boxes on the left side of the interface, this segment visually represents the current position of the tray within the workflow, using a grid-like graphical representation. The tray is shown in transit between the first and second target locations.

Middle Box: This area showcases circles representing three mobile robots. A robot encapsulated within a green circle indicates that it is actively engaged in a task. On the right side, the tray's location within Room 212 is displayed."

Lower Box: This section lists all the events that are scheduled to take place those that have already been executed, are currently in execution, or are yet to be executed. The current step being executed involves the mobile robot transporting the tray from workstation 212 to workstation 213.

The interface is designed to provide a comprehensive, real-time overview of task statuses, robot activity, and event sequencing. This level of real-time feedback enables teams to monitor progress and intervene if necessary, ensuring a more controlled and efficient workflow. A complete set of photographs capturing the mobile robot at various stages of the workflow is shown in Figure 100.

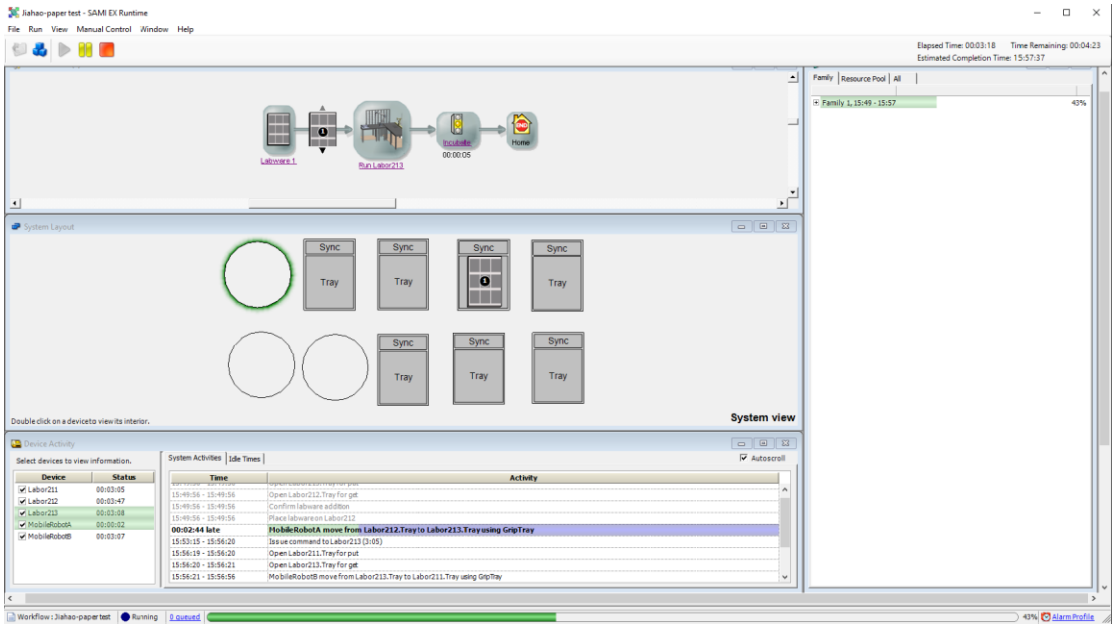


Figure 99 Real-Time Visualization of workflow progress (Application 2)

a) b) c)

d) e) f)

g) h) i)

j) k) l)

Figure 100 Image of the Experimental Workflow during Robot Transportation (Application 2), a) The robot initiates from the waiting point; b) The robot enters Room 212 through the automatic door; c) The robot picks up the tray from the Room 212 station; d) The robot transports the tray to the Room 213 station; e) After placing the tray, the robot returns to the waiting point; f) The robotic arm in the Room 213 station begins its operation, moving the required labware from the "Hotel" storage to the tray; g) Upon completion in the Room 213 station, SAMI assigns a new transportation task to the the robot; h) The robot exits Room 213 and enters the corridor; i) The robot transitions from the corridor to Room 211; j) The robot transports the tray to the Room 211 station; k) After incubation in Room 211, the robot transports the tray back to its starting point; l) The robot returns the incubated tray, now loaded with labware, to the Room 212 station.

This experiment is one of the simpler ones conducted in a laboratory setting. To enhance repeatability and save time, the incubation period for labware in Room 211 was shortened to 5 seconds. The entire experiment was repeated 20 times using robot 1, and it successfully completed the commands on all occasions while demonstrating stable performance. Each full cycle of the processes took an average of 20 minutes.

No errors occurred during the get/put processes. The mobile robot smoothly navigated through automatic doors. Additionally, the transition times between different pieces of equipment were brief. For instance, it took 20 seconds from when the mobile robot placed the tray in Room 213 to when the robotic arm began its operations. This time included the startup period for the robotic arm. Also, within 5 seconds after the robotic arm finished positioning the labware, the mobile robot received a new transport command and moved to the loading station.

This experiment demonstrated the feasibility of an automated laboratory system, where different devices and subsystems can be interconnected via a mobile robot, also proving the system's quick response capabilities.

6.6.3 Application III – Multi-Robot Running

Like Application II, this experiment is based on the SAMI system but initiates two simultaneous tasks, utilizing two mobile robots concurrently within the CELISCA laboratory. This experiment demonstrates the MOLAR robots' ability to operate and coordinate multiple robots simultaneously.

Initially, two methods are established in the SAMI EX Editor. The first method directs a mobile robot to repeatedly transport between Station 213 and Station 306, as shown in Figure 100. This involves navigating through multiple automatic doors and utilizing the elevator between the 2nd and 3rd floors. The second method involves repetitive transportation tasks between Station 212 and Station 209. The routes for both tasks intersect in Room 213, the south corridor on the 2nd floor, and the area in front of the 2nd-floor elevator, potentially leading to intersections. After inputting the tasks into the

SAMI EX Editor, scheduling is performed to generate an optimal solution, as shown in Figure 101, employing two robots for the transportation tasks.

The experiment then commences in the SAMI runtime GUI. As depicted in Figure 102, the first method's tray moves between Station 213 and Station 306, indicating ongoing transportation by the robot from getting the tray at Station 213 to placing it at Station 306. Similarly, the second method's tray moves between Station 212 and Station 209, showing the robot performing tasks from getting the tray at Station 212 to placing it at Station 209. These actions correspond to the two current tasks displayed at the bottom.

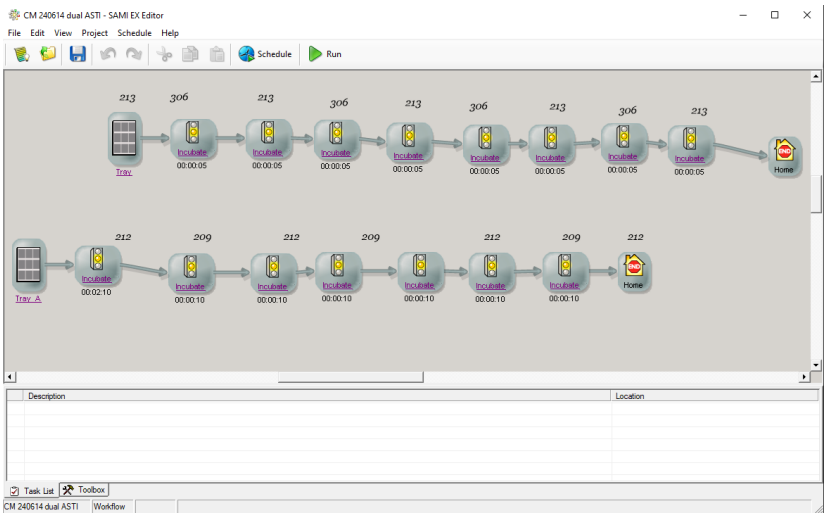


Figure 101 Workflow Setting in SAMI Editor (Application 3) see remark above

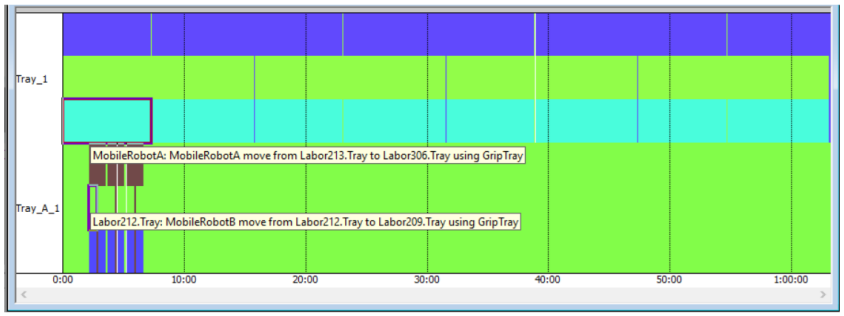


Figure 102 Generated workflow following successful scheduling (Application 3)

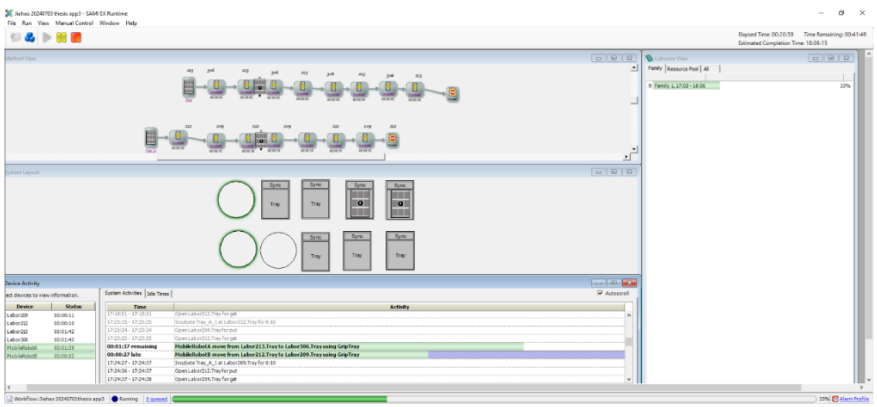


Figure 103 Real-Time Visualization of workflow progress (Application 3)

Figure 104 and Figure 105 track the detailed processes of the two mobile robots executing their tasks. Figure 103 follows robot 12's perspective, completing the first method's tasks between Station 213 and Station 306. Figure 105 follows robot 13's perspective, completing the second method's tasks between Station 212 and Station 209.

Figure 104 a) displays both robots starting simultaneously; robot 12 heads to Station 213 to get the tray. Since Station 213 is a single-robot area, robot 13 waits outside the area until robot 12 leaves, then it proceeds.

Figure 104 b) shows robot 12 traversing the 2nd south corridor to the elevator area, then taking the elevator to the 3rd floor, passing through the 3rd north corridor to reach Room 306, and putting the tray at Station 306.

Figure 105 a) depicts robot 13 getting the tray at Station 209, then passing through the 2nd north corridor, the elevator area, and the 2nd south corridor to reach Room 213.

Figure 105 b) shows robot 12 placing the tray at Station 213. Because it's a single-robot area, robot 13 moves to the waiting point until robot 12 departs. Robot 13 then passes Station 213 en route to Station 212 to place the tray.

a)

b)

Figure 104 Image of the Experimental Workflow during Robot 12 Transportation (Application 3)

a)

b)

Figure 105 Image of the Experimental Workflow during Robot 13 Transportation (Application 3)

This experiment was repeated 10 times, averaging 3 hours and 40 minutes each. It provided a more comprehensive assessment of system performance, not only involving the mobile robot's use for getting and putting trays and navigating through automatic doors but also including multi-floor transportation and multi-robot operations. It tested the daily transport interactions between robots and elevators and the movement planning of multiple robots in narrow areas.

Overall, the SAMI process was smooth during this experiment. It was possible to monitor the movement of two trays and the current step of the experiment in real-time on the SAMI EX Runtime software. However, there was an issue with robot deadlock. In the area in front of the elevator on the second floor, two mobile robots stood on opposite sides of an automatic door, each treating the other as an obstacle, which hindered normal movement. This was caused by the insufficient coverage of the single robot area, which was later adjusted to resolve the issue. No deadlock occurred during the interaction between the two robots in room 213.

During elevator use, when shared with other people, there could be failures due to the robot's inability to recognize the entire elevator if obstructed by goods inside, leading the robot to rotate within the elevator and misidentify its location. An extreme scenario occurred when, after reaching the destination floor and the robot correctly identified the floor and began to exit, the elevator doors happened to close completely (a rare occurrence as elevator doors are typically reopened several times), and someone else

called the elevator to another floor. Consequently, the robot exited on the wrong floor following its original exit routine. Hence, in situations where elevators are shared with others, there is a small chance that the robot-elevator interaction might fail. However, under normal circumstances with no human presence, the success rate of elevator interactions is 100 percent.

7 Chapter 7 Summary and Outlook

7.1 Summary

This dissertation explored the integration of mobile robots into intelligent, distributed life science laboratories to achieve comprehensive automation. The research highlighted significant advancements in mobile robot technology, particularly in navigation, scheduling, and integration within laboratory environments. A thorough review of state-of-the-art mobile robots emphasized their role in enhancing laboratory efficiency, reducing human error, and increasing productivity. Integration was detailed on three levels:

- **Robot System Level**

Development of a 2D environmental map with added semantic information for zone segmentation and functional areas, which aids the robots' movement and interaction in complex laboratory environments. Infrastructure for labware transportation stations and charging stations was established, along with a comparative performance analysis with older robot models like H20.

The MOLAR robot is not open-source for secondary development, which restricts further optimization of its navigation capabilities. The mapping was developed using LiDAR to establish a complete environmental map of CELISCA. The 2D mapping, based on the Gmapping algorithm, delineates only obstacles and free areas. Further segmentation defines rooms, corridors, elevator zones, and restricted areas linked by automatic doors and elevators. To optimize the robots' performance in complex lab areas, zones such as narrow areas, preferred lines, and single-robot areas are tested for their effectiveness and stability.

Lab station get/put setups include labware, trays for placing labware, and racks supporting the trays. Designed to transport up to nine pieces of labware simultaneously, these trays and racks incorporate a sloped connection surface to allow a tolerance of ±1 cm. Combined with fine-positioning at each station, an accuracy of 1 cm and 1° ensures a 100% success rate in the robots' handling processes.

In "auto" mode, robots automatically navigate to charging stations when battery levels fall below 40% and no tasks are assigned. Charging stations strategically placed on various floors facilitate proximity-based charging. Task distribution within AIC selects the nearest and most charged robot, optimizing task completion times.

A comparison of the MOLAR robot with the older H20 transport robot across various dimensions—such as navigation technology, positioning accuracy, and transport efficiency—shows significant improvements in MOLAR's performance. Its path

planning and obstacle avoidance capabilities feature smoother trajectories and more intelligent functionalities.

- **Infrastructure Level**

Utilizing the laboratory's internal WiFi network for communication with automatic doors and elevators, addressing the challenges associated with closed-system elevators through IoT-based solutions.

Control of automatic doors and elevators is crucial for a multi-floor laboratory. Lab automatic doors controlled via PLC manage opening, holding open, and closing actions. Elevator interactions are complex due to their closed-system nature, requiring knowledge of elevator door statuses and floor levels. IoT methods installed on robots use ultrasonic sensors for door status and air pressure for floor detection. Since air pressure is affected by weather, temperature, and humidity, relative pressure differences are used to ascertain floor levels. Before each elevator interaction, air pressure is zeroed, and then height differences are calculated based on pressure differences, with nearly 100% accuracy under normal conditions. In extreme weather, BLE beacon technology assists floor determination. Installed BLE beacon at every elevator door, the closest beacon's RSSI value indicates the current floor. A C#-developed Virtual PLC acts as middleware, interfacing with AIC, elevator PLC, and IoT modules, and designing interaction logic for robots and elevators, including error scenario handling. The integration of automatic doors and elevators allows the MOLAR robot to navigate freely throughout the laboratory.

- **Top-Level Control System Level:**

Incorporating mobile robots into the SAMI system positioned them as a crucial element of the automated experiment process, linking highly distributed lab devices and enhancing laboratory automation towards achieving a 24/7 unattended operation. The ultimate integration involves embedding the mobile robot system into CELISCA's top-level control system, SAMI, which connects all lab devices and schedules them to form optimized workflows. The MOLAR robot, serving as a bridge between different sites, transports materials and labware needed for experiments, achieving transportation between different experimental stages. SAMI uses APIs provided by AIC over HTTP to send transport commands, which include source and target stations, while monitoring the transport process. The integration of the MOLAR robot into SAMI began with the development of upper-level control software using C++ to validate the usability of AIC's APIs. The communication class was then written into the SAMI program, and control logic was designed, allowing Global SAMI to directly dispatch the MOLAR robot and connect highly distributed stations into a cohesive system.

Currently, the MOLAR robot has been performing transport tasks in CELISCA for an

extended period, undergoing a series of experiments to validate its performance in various aspects, including navigation systems (mapping, localization, and movement), obstacle avoidance, functional areas, and station handling of labware. Automatic door usage, elevator interactions, IoT module data transmission, automatic charging, communication sockets, and error handling are all part of this comprehensive evaluation. The experimental results demonstrate that the robot's navigation methods allow it to adapt to all indoor environments without the need for environmental modifications. The robot's station-based fine-positioning accuracy, combined with the sloped design of racks and trays, ensures the precision of the handling processes. Triangle devices installed at charging stations also ensure safety, automatic charging keeps the robots able to work over extended periods. The telescopic conveyor-based handling processes are stable, and continually optimized at each station during ongoing tests. Observations of the robot moving through CELISCA lead to continuous adjustments and optimizations of the map's semantic functional zones, ensuring smoother movement, especially in complex and narrow areas and zones where robots intersect. The automatic door control system functions correctly, maintaining doors open as robots pass through and ensuring they do not close prematurely. Even if someone opens a door ahead of a robot, it proceeds as normal. Elevator tests are conducted more frequently, combining routine multi-floor tests with dedicated elevator tests. Floor recognition and elevator door status detection are highly stable, and the handling of error scenarios aligns well with real lab conditions. SAMI's integration has been successful, allowing the robot to receive transport commands from SAMI and move immediately after the completion of actions at one site to begin operations at the next. Once transport is complete, the next site's equipment starts immediately. Through these integrations, the robot successfully unites the entire laboratory into a single automated entity, enhancing the level of automation.

7.2 Outlook

The MOLAR robot currently operates effectively in a laboratory setting, handling transportation tasks with ease. However, future efforts can further enhance its functionality:

1. Integration of IoT Modules on Trays: By incorporating IoT modules on trays, it becomes possible to track the position of each tray. Currently, stations lack devices to detect the presence of trays on racks, which poses safety risks if not checked by staff before initiating robot operations. The addition of IoT tracking modules can solve the safety concern.

2. In life science laboratory environments, which are more complex than other mobile

robot environments like factories and logistics warehouses, the precision instruments and narrow passages pose challenges to robot navigation. When robots enter these narrow spaces, the speed slow down to ensure safe passage. Future developments could focus on enhancing these safety zones or adjusting speed restrictions to increase the robots' movement speed in confined areas.

3. Customized Station Parameters for Each Robot: During the get/put processes, the parameters for stations (such as conveyor elevation, extension length, and the distance of the robot from the triangle device) are global parameters for all robots. Given the inevitable mechanical variations among robots, this can lead to inconsistencies in the accuracy of the get/put tasks. Ideally, station parameters should be individually calibrated for each robot to address these variations. Fortunately, the sloped design of the connection surfaces between trays and racks provides a tolerance that ensures successful get/put operations despite these differences.

4. The laboratory database features a Web GUI that displays real-time IoT information about the lab. It has already been successfully configured to update the real-time coordinates of the mobile robots to the database every 5 seconds and display this on the Web GUI, allowing for visual tracking of the robots' locations within the laboratory. Future updates could include more detailed robot-related information, such as tray details and transportation data (origin, destination, current status), to enrich the display on the Web GUI.

5. The elevator area is a shared space for people, robots, and goods. While the use of elevators by humans has been considered under various circumstances, special situations still need further optimization. For instance, if a mobile robot reaches its designated floor but finds the elevator exit blocked, or if it has recognized its arrival and initiated the exit process but the elevator doors close due to a call to another floor, the robot will not reposition itself and will continue with the exit process. Additionally, if the elevator environment is crowded upon the robot's entry, it may lead to the robot losing its localization. These corner cases require specific attention and optimization.

8 References

[1] G. Mies and P. Z. Zentay, "Machine Tools and Industrial Robots as Key Technologies to Enable Industry 4.0," *REPÜLÉSTUDOMÁNYI KÖZLEMÉNYEK (1997-TŐL),* vol. 32, no. 2, pp. 37-48, 2020.

[2] K. Thamrongaphichartkul, N. Worrasittichai, T. Prayongrak, and S. Vongbunyong, "A framework of IoT platform for autonomous mobile robot in hospital logistics applications," in *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2020: IEEE, pp. 1-6.

[3] Y. Bi, H. Hu, Y. Luo, B. Li, C. Wang, and M. Huang, "Research on Mobile Robot Platform to Assist Laboratory Management," in *Intelligent Robotics and Applications: 14th International Conference, ICIRA 2021, Yantai, China, October 22–25, 2021, Proceedings, Part III 14*, 2021: Springer, pp. 357-366.

[4] M. C. Moreno, Y. D. Angulo, and O. J. Suarez, "Teleoperated robot prototype for the manipulation and transport of chemical substances in laboratory: a low-cost adaptation," in *2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2022: IEEE, pp. 1-6.

[5] Joo, S.H., Manzoor, S., Rocha, Y.G., Bae, S.H., Lee, K.H., Kuc, T.Y. and Kim, M, "Autonomous navigation framework for intelligent robots based on a semantic environment modeling," *Applied Sciences,* vol. 10, no. 9, p. 3219, 2020.

[6] Y. Jia, B. Ramalingam, R. E. Mohan, Z. Yang, Z. Zeng, and P. Veerajagadheswar, "Deep-Learning-Based Context-Aware Multi-Level Information Fusion Systems for Indoor Mobile Robots Safe Navigation," *Sensors,* vol. 23, no. 4, p. 2337, 2023.

[7] P. Gao, R. Guo, H. Lu, and H. Zhang, "Correspondence identification for collaborative multi-robot perception under uncertainty," *Autonomous Robots,* vol. 46, no. 1, pp. 5-20, 2022.

[8] K. Thurow, "Strategies for automating analytical and bioanalytical laboratories," *Analytical and Bioanalytical Chemistry,* vol. 415, no. 21, pp. 5057-5066, 2023.

[9] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science,* vol. 364, no. 6446, p. eaat8414, 2019.

[10] J. Zhang, W. He, A. Zhao, L. Zhang, Q. Feng, and Y. Zhou, "Semantic map for service robot navigation based on ROS," in *Second Target Recognition and Artificial Intelligence Summit Forum*, 2020, vol. 11427: SPIE, pp. 226-233.

[11] S. Y. Alaba and J. E. Ball, "A survey on deep-learning-based lidar 3d object detection for autonomous driving," *Sensors,* vol. 22, no. 24, p. 9577, 2022.

[12] C. Ai, Z. Qi, L. Zheng, D. Geng, Z. Feng, and X. Sun, "Research on mapping method based on data fusion of lidar and depth camera," in *2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, 2021: IEEE, pp. 360-365.

[13] Balogh, M., Vidács, A., Fehér, G., Maliosz, M., Horváth, M.Á., Reider, N. and Rácz, S, "Cloud-controlled autonomous mobile robot platform," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021: IEEE, pp. 1-6.

[14] Kabzan, J., Valls, M.I., Reijgwart, V.J., Hendrikx, H.F., Ehmke, C., Prajapat, M., Bühler, A., Gosala, N., Gupta, M., Sivanesan, R. and Dhall, A, "AMZ driverless: The full autonomous racing system," *Journal of Field Robotics,* vol. 37, no. 7, pp. 1267-1294, 2020.

[15] R. Kim, V. Madabushi, E. Dong, and A. Mazumdar, "Increasing mobile robot efficiency and versatility through manipulation-driven adaptation," *Journal of Mechanisms and Robotics,* vol. 13, no. 5, p. 050906, 2021.

[16] M. Ciszewski, Ł. Mitka, P. Kohut, T. Buratowski, and M. Giergiel, "Robotic system for off-shore infrastructure monitoring," *Journal of Marine Engineering & Technology,* vol. 16, no. 4, pp. 310-318, 2017.

[17] L. Huiping and S. Hao, "Research status and development trend of bionic multi-legged mobile robot," in *2022 International Conference on Electronics and Devices, Computational Science (ICEDCS)*, 2022: IEEE, pp. 27-33.

[18] V. Saikhom and M. Kalita, "UAV for Remote Sensing Applications: An Analytical Review," in *International Conference on Emerging Global Trends in Engineering and Technology*, 2022: Springer, pp. 51-59.

[19] J. P. Panda, A. Mitra, and H. V. Warrior, "A review on the hydrodynamic characteristics of autonomous underwater vehicles," *Proceedings of the Institution of Mechanical Engineers, Part*

M: Journal of Engineering for the Maritime Environment, vol. 235, no. 1, pp. 15-29, 2021.
[20] G. A. Demetriou, "Mobile robotics in education and research," *Mobile robots-current trends,* vol. 27, p. 48, 2011.
[21] H.-o. Lim and A. Takanishi, "Biped walking robots created at Waseda University: WL and WABIAN family," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 365, no. 1850, pp. 49-64, 2007.
[22] Golombek, M.P., Cook, R.A., Economou, T., Folkner, W.M., Haldemann, A.F.C., Kallemeyn, P.H., Knudsen, J.M., Manning, R.M., Moore, H.J., Parker, T.J. and Rieder, R, "Overview of the Mars Pathfinder mission and assessment of landing site predictions," *Science,* vol. 278, no. 5344, pp. 1743-1748, 1997.
[23] M. Hirose and K. Ogawa, "Honda humanoid robots development," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 365, no. 1850, pp. 11-19, 2007.
[24] B. Tribelhorn and Z. Dodds, "Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007: IEEE, pp. 1393-1399.
[25] R. D'Andrea, "Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead," *IEEE Transactions on Automation Science and Engineering,* vol. 9, no. 4, pp. 638-639, 2012.
[26] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for BigDog," in *2010 IEEE international conference on robotics and automation*, 2010: Ieee, pp. 4736-4741.
[27] Tomzcak, K., Pelter, A., Gutierrez, C., Stretch, T., Hilf, D., Donadio, B., Tenhundfeld, N.L., de Visser, E.J. and Tossell, C.C, "Let Tesla park your Tesla: Driver trust in a semi-automated car," in *2019 Systems and Information Engineering Design Symposium (SIEDS)*, 2019: IEEE, pp. 1-6.
[28] V. S. R. Kosuru and A. K. Venkitaraman, "Advancements and challenges in achieving fully autonomous self-driving vehicles," *World Journal of Advanced Research and Reviews,* vol. 18, no. 1, pp. 161-167, 2023.
[29] I. Spectrum. " 6 Key Connectivity Requirements of Autonomous Driving." https://spectrum.ieee.org/6-key-connectivity-requirements-of-autonomous-driving (accessed 22-April, 2024).
[30] I. F. o. Robotics. "World Robotics 2023 Report: Asia ahead of Europe and the Americas." https://ifr.org/ifr-press-releases/news/world-robotics-2023-report-asia-ahead-of-europe-and-the-americas (accessed 14-April, 2024).
[31] "SOLUTIONS FOR HEALTHCARE." https://aethon.com/hospital-robots-healthcare/ (accessed November 18, 2024).
[32] "The First and Only FDA Authorized Microbial Reduction Robot for the Healthcare Environment." https://xenex.com/ (accessed November 18, 2024).
[33] "Intuitive da Vinci - World class robotic surgical systems." https://www.intuitive.com/en-us/products-and-services/da-vinci (accessed November 18, 2024).
[34] "People + robots working together make the best team." https://www.diligentrobots.com/ (accessed November 18, 2024).
[35] "Amazon Robotics." https://en.wikipedia.org/wiki/Amazon_Robotics (accessed November 18, 2024).
[36] "Shelf-to-Person Solutions." https://www.geekplus.com/robot/p-robot (accessed November 18, 2024).
[37] "Maximize goods-to-person (GTP). Accelerate throughput. Drive omnichannel." https://www.greyorange.com/solutions/by-technology/rack-to-person/ (accessed November 18, 2024).
[38] "Warehouse Robotics Automation." https://locusrobotics.com/ (accessed november 18, 2024).
[39] "Hotel Delivery Robots-Focus on guests and leave hotel room deliveries to Relay." https://relayrobotics.com/relay-delivery-robots-for-hotels/ (accessed november 19, 2024).
[40] "LG Electronics LG Introduces AI-Powered CLOi ServeBot for Hospitality and Healthcare." https://www.lgcorp.com/media/release/27812 (accessed November 19, 2024).
[41] "Henn na Hotel." https://group.hennnahotel.com/ (accessed November 19, 2024).
[42] "For better business just add Pepper." https://us.softbankrobotics.com/pepper (accessed November 19, 2024).
[43] "Who is Bossa Nova?" https://www.bossanova.com/ (accessed November 19, 2024).
[44] "LoweBot - A helping hand for customers and associates."

https://www.lowesinnovationlabs.com/projects/lowebot (accessed November 19, 2024).
[45] "KMR iiwa." https://www.kuka.com/de-de/produkte-leistungen/amr-autonome-mobile-roboter/mobile-roboter/kmr-iiwa (accessed November 19, 2024).
[46] "LD-series Vollständig autonome mobile Roboter." https://industrial.omron.de/de/products/ld-series (accessed November 19, 2024).
[47] "The Evolution of Your Intralogistics With the No. 1 Integrator for Automated Guided Vehicle Systems." https://www.mobile-robots.de/69/en/agv-world (accessed November 19, 2024).
[48] "Freight 500/1500." https://synertechusa.com/product/freight-500-1500/ (accessed November 19, 2024).
[49] "Build your autonomous workforce." https://ottomotors.com/amrs/ (accessed November 19, 2024).
[50] S. Kleine-Wechelmann, K. Bastiaanse, M. Freundel, and C. Becker-Asano, "Designing the mobile robot Kevin for a life science laboratory," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2022: IEEE, pp. 870-875.
[51] K. Thurow, L. Zhang, H. Liu, S. Junginger, N. Stoll, and J. Huang, "Multi-floor laboratory transportation technologies based on intelligent mobile robots," *Transportation Safety and Environment,* vol. 1, no. 1, pp. 37-53, 2019.
[52] M. Ghandour, H. Liu, N. Stoll, and K. Thurow, "Interactive collision avoidance system for indoor mobile robots based on human-robot interaction," in *2016 9th International Conference on Human System Interactions (HSI)*, 2016: IEEE, pp. 209-215.
[53] I. Kramer, R. Memmesheimer, and D. Paulus, "Customer Interaction of a Future Convenience Store with a Mobile Manipulation Service Robot," in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 2021: IEEE, pp. 1-7.
[54] N. K. Ilampooranan, G. V. K. Damodaran, P. Rajagopal, S. Narasimhan, A. Maitra, and S. Sriram, "Design and Development of a Shopping Assistance Robot," in *Proceedings of the 2023 6th International Conference on Advances in Robotics*, 2023, pp. 1-7.
[55] R. Lin, H. Huang, and M. Li, "An automated guided logistics robot for pallet transportation," *Assembly Automation,* vol. 41, no. 1, pp. 45-54, 2021.
[56] A. A. Abdulla, "An intelligent multi-floor mobile robot transportation system in life science laboratories," Dissertation, Rostock, Universität Rostock, 2017, 2016.
[57] J.-H. He, Y.-L. Chen, H.-H. Chiang, and H.-C. Li, "Dynamic obstacle avoidance based on a dynamic space-time grid map for mobile robots," in *2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021: IEEE, pp. 1-3.
[58] A. N. Kumaar and S. Kochuvila, "Reinforcement learning based path planning using a topological map for mobile service robot," in *2023 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 2023: IEEE, pp. 1-6.
[59] Z. Liu, L. Gao, F. Liu, D. Liu, and W. Han, "Fusion of weighted Voronoi diagram and $\mathrm {A}^{\ast} $ algorithm for mobile robot path planning," in *2022 2nd International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT)*, 2022: IEEE, pp. 403-406.
[60] H. Gao, X. Zhang, J. Wen, J. Yuan, and Y. Fang, "Autonomous indoor exploration via polygon map construction and graph-based SLAM using directional endpoint features," *IEEE Transactions on Automation Science and Engineering,* vol. 16, no. 4, pp. 1531-1542, 2018.
[61] Z. Xiang and J. Su, "Towards customizable robotic disinfection with structure-aware semantic mapping," *IEEE Access,* vol. 9, pp. 35477-35486, 2021.
[62] M. Luperto, M. Antonazzi, F. Amigoni, and N. A. Borghese, "Robot exploration of indoor environments using incomplete and inaccurate prior knowledge," *Robotics and Autonomous Systems,* vol. 133, p. 103622, 2020.
[63] M. Mielle, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "SLAM auto-complete: Completing a robot map using an emergency map," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2017: IEEE, pp. 35-40.
[64] Gomez, C., Fehr, M., Millane, A., Hernandez, A.C., Nieto, J., Barber, R. and Siegwart, R, "Hybrid topological and 3d dense mapping through autonomous exploration for large indoor environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020: IEEE, pp. 9673-9679.
[65] K. Borodacz, C. Szczepański, and S. Popowski, "Review and selection of commercially available IMU for a short time inertial navigation," *Aircraft Engineering and Aerospace Technology,* vol. 94, no. 1, pp. 45-59, 2022.
[66] H. Wang *et al.*, "Mobile Robot Indoor Positioning System Based on K‐ELM," *Journal of Sensors,* vol. 2019, no. 1, p. 7547648, 2019.
[67] Q. Liang and M. Liu, "A tightly coupled VLC-inertial localization system by EKF," *IEEE*

Robotics and Automation Letters, vol. 5, no. 2, pp. 3129-3136, 2020.

[68] Z. Cui, Y. Wang, and X. Fu, "Research on indoor positioning system based on VLC," in *2020 Prognostics and health management conference (PHM-Besançon)*, 2020: IEEE, pp. 360-365.

[69] W. Guan, S. Chen, S. Wen, Z. Tan, H. Song, and W. Hou, "High-accuracy robot indoor localization scheme based on robot operating system using visible light positioning," *IEEE Photonics Journal,* vol. 12, no. 2, pp. 1-16, 2020.

[70] P. Louro, M. Vieira, and M. A. Vieira, "Bidirectional visible light communication," *Optical Engineering,* vol. 59, no. 12, pp. 127109-127109, 2020.

[71] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A backbone-floyd hybrid path planning method for mobile robot transportation in multi-floor life science laboratories," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016: IEEE, pp. 406-411.

[72] E. Bernardes, S. Viollet, and T. Raharijaona, "A Three-Photo-Detector Optical Sensor Accurately Localizes a Mobile Robot Indoors by Using Two Infrared Light-Emitting Diodes," *IEEE Access,* Article vol. 8, pp. 87490-87503, 2020, Art no. 9089000, doi: 10.1109/ACCESS.2020.2992996.

[73] T. Grami and A. S. Tlili, "Indoor mobile robot localization based on a particle filter approach," in *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2019: IEEE, pp. 47-52.

[74] Pérez-Navarro, A., Torres-Sospedra, J., Montoliu, R., Conesa, J., Berkvens, R., Caso, G., Costa, C., Dorigatti, N., Hernández, N., Knauth, S. and Lohan, E.S, "Challenges of fingerprinting in indoor positioning and navigation," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*: Elsevier, 2019, pp. 1-20.

[75] W. Lv, Y. Kang, and J. Qin, "Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 6, pp. 1241-1253, 2017.

[76] I. Nagai, J. Sakai, and K. Watanabe, "Indoor self-localization using multiple magnetic sensors," *Journal of Robotics and Mechatronics,* Article vol. 31, no. 2, pp. 203-211, 2019, doi: 10.20965/jrm.2019.p0203.

[77] T. Thewan, C. Seksan, S. Pramot, A. Ismail, and K. Terashima, "Comparing WiFi RSS filtering for wireless robot location system," *Procedia Manufacturing,* vol. 30, pp. 143-150, 2019.

[78] J. L. C. Villacrés, Z. Zhao, T. Braun, and Z. Li, "A particle filter-based reinforcement learning approach for reliable wireless indoor positioning," *IEEE journal on selected areas in communications,* vol. 37, no. 11, pp. 2457-2473, 2019.

[79] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. Hacker, "An ibeacon based proximity and indoor localization system," *arXiv preprint arXiv:1703.07876,* 2017.

[80] A. Prorok, L. Gonon, and A. Martinoli, "Online model estimation of ultra-wideband TDOA measurements for mobile robot localization," in *2012 IEEE International Conference on Robotics and Automation*, 2012: Ieee, pp. 807-814.

[81] T. Yang, A. Cabani, and H. Chafouk, "A survey of recent indoor localization scenarios and methodologies," *Sensors,* vol. 21, no. 23, p. 8086, 2021.

[82] W. Cui, Q. Liu, L. Zhang, H. Wang, X. Lu, and J. Li, "A robust mobile robot indoor positioning system based on Wi-Fi," *International Journal of Advanced Robotic Systems,* vol. 17, no. 1, p. 1729881419896660, 2020.

[83] A. Alexandr, D. Anton, M. Mikhail, and K. Ilya, "Comparative analysis of indoor positioning methods based on the wireless sensor network of bluetooth low energy beacons," in *2020 International Conference Engineering and Telecommunication (En&T)*, 2020: IEEE, pp. 1-5.

[84] D. Mankotia, S. Agrawal, and S. Singh, "Error minimization in bluetooth based indoor localization of a mobile robot using Cuckoo search algorithm," in *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, 2014: IEEE, pp. 283-288.

[85] R. C. Luo and T. J. Hsiao, "Dynamic wireless indoor localization incorporating with an autonomous mobile robot based on an adaptive signal model fingerprinting approach," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 3, pp. 1940-1951, 2018.

[86] Z. Wang, M. Liu, and Y. Zhang, "Mobile localization in complex indoor environment based on ZigBee wireless network," in *Journal of Physics: Conference Series*, 2019, vol. 1314, no. 1: IOP Publishing, p. 012214.

[87] F. Martinelli, "Simultaneous localization and mapping using the phase of passive UHF-RFID signals," *Journal of Intelligent & Robotic Systems,* vol. 94, pp. 711-725, 2019.

[88] J. Wang and Y. Takahashi, "Particle Smoother - Based Landmark Mapping for the SLAM

Method of an Indoor Mobile Robot with a Non-Gaussian Detection Model," *Journal of Sensors,* vol. 2019, no. 1, p. 3717298, 2019.

[89] Magnago, V., Palopoli, L., Fontanelli, D., Macii, D., Motroni, A., Nepa, P., Buffi, A. and Tellini, B, "Robot localisation based on phase measures of backscattered UHF-RFID signals," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2019: IEEE, pp. 1-6.

[90] R. Yadav and L. Malviya, "UWB antenna and MIMO antennas with bandwidth, band-notched, and isolation properties for high-speed data rate wireless communication: A review," *International Journal of RF and Microwave Computer-Aided Engineering,* vol. 30, no. 2, p. e22033, 2020.

[91] D. Shi, H. Mi, E. G. Collins, and J. Wu, "An indoor low-cost and high-accuracy localization approach for AGVs," *IEEE Access,* vol. 8, pp. 50085-50090, 2020.

[92] X. Zhu, J. Yi, J. Cheng, and L. He, "Adapted error map based mobile robot UWB indoor positioning," *IEEE Transactions on Instrumentation and Measurement,* vol. 69, no. 9, pp. 6336-6350, 2020.

[93] X. Li and Y. Wang, "Research on the UWB/IMU fusion positioning of mobile vehicle based on motion constraints," *Acta Geodaetica et Geophysica,* vol. 55, pp. 237-255, 2020.

[94] X. Xu, X. Liu, B. Zhao, and B. Yang, "An extensible positioning system for locating mobile robots in unfamiliar environments," *Sensors,* vol. 19, no. 18, p. 4025, 2019.

[95] Y. Liu, R. Sun, J. Liu, Y. Fan, L. Li, and Q. Zhang, "Research on the positioning method of autonomous mobile robot in structure space based on UWB," in *2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, 2019: IEEE, pp. 278-282.

[96] T.-L. Habich, M. Stuede, M. Labbé, and S. Spindeldreier, "Have I been here before? Learning to close the loop with lidar data in graph-based SLAM," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2021: IEEE, pp. 504-510.

[97] Z. Sun, B. Wu, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "Frontier detection and reachability analysis for efficient 2D graph-SLAM based active exploration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020: IEEE, pp. 2051-2058.

[98] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. and Leonard, J.J, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics,* vol. 32, no. 6, pp. 1309-1332, 2016.

[99] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010: IEEE, pp. 22-29.

[100] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE international conference on robotics and automation (ICRA)*, 2016: IEEE, pp. 1271-1278.

[101] R. Munoz-Salinas, M. J. Marin-Jimenez, and R. Medina-Carnicer, "SPM-SLAM: Simultaneous localization and mapping with squared planar markers," *Pattern Recognition,* vol. 86, pp. 156-171, 2019.

[102] M. Avgeris, D. Spatharakis, N. Athanasopoulos, D. Dechouniotis, and S. Papavassiliou, "Single vision-based self-localization for autonomous robotic agents," in *2019 7th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2019: IEEE, pp. 123-129.

[103] K.-T. Song and Y. C. Chang, "Design and implementation of a pose estimation system based on visual fiducial features and multiple cameras," in *2018 International Automatic Control Conference (CACS)*, 2018: IEEE, pp. 1-6.

[104] W. Lv, Y. Kang, and J. Qin, "FVO: floor vision aided odometry," *Science China Information Sciences,* vol. 62, no. 1, p. 12202, 2019.

[105] C. Duan, S. Junginger, J. Huang, K. Jin, and K. Thurow, "Deep learning for visual SLAM in transportation robotics: A review," *Transportation Safety and Environment,* vol. 1, no. 3, pp. 177-184, 2019.

[106] S. D. Murthy, S. Srivenkata Krishnan, G. Sundarrajan, S. Kiran Kassyap, R. Bhagwanth, and V. Balasubramanian, "A robust approach for improving the accuracy of IMU based indoor mobile robot localization," in *ICINCO 2016 - 13th International Conference on Informatics in Control, Automation and Robotics, Doctoral Consortium*, 2016, vol. 2016-January, pp. 436-445, doi: 10.5220/0005986804360445. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85013080815&doi=10.5220%2f0005986804360445&partnerID=40&md5=f931240766b1fcc

61e91cca22e044e9e
[107] X. Li, Z. Yan, L. Huang, S. Chen, and M. Liu, "High-Accuracy and Real-Time Indoor Positioning System Based on Visible Light Communication and Mobile Robot," *International Journal of Optics,* Article vol. 2020, 2020, Art no. 3124970, doi: 10.1155/2020/3124970.
[108] Y. Zhang, L. Hsiung-Cheng, J. Zhao, M. Zewen, Z. Ye, and H. Sun, "A multi-DoF ultrasonic receiving device for indoor positioning of AGV system," in *Proceedings - 2018 International Symposium on Computer, Consumer and Control, IS3C 2018*, 2019, pp. 97-100, doi: 10.1109/IS3C.2018.00032. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063193279&doi=10.1109%2fIS3C.2018.00032&partnerID=40&md5=5796e7975baca09031a4ac414034195b
[109] D. Shen, Y. Xu, and Y. Huang, "Research on 2D-SLAM of indoor mobile robot based on laser radar," in *ACM International Conference Proceeding Series*, 2019, doi: 10.1145/3351917.3351966. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85073248594&doi=10.1145%2f3351917.3351966&partnerID=40&md5=02bd24bae1f8e429b19201922c75c1a6
[110] Fu, Q., Yu, H., Lai, L., Wang, J., Peng, X., Sun, W. and Sun, M, "A robust RGB-D SLAM system with points and lines for low texture indoor environments," *IEEE Sensors Journal,* Article vol. 19, no. 21, pp. 9908-9920, 2019, Art no. 8756267, doi: 10.1109/JSEN.2019.2927405.
[111] W. Cui, Q. Liu, L. Zhang, H. Wang, X. Lu, and J. Li, "A robust mobile robot indoor positioning system based on Wi-Fi," *International Journal of Advanced Robotic Systems,* Article vol. 17, no. 1, 2020, doi: 10.1177/1729881419896660.
[112] D. Mankotia, S. Agrawal, and S. Singh, "Error minimization in Bluetooth based indoor localization of a mobile robot using Cuckoo Search algorithm," in *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems, MedCom 2014*, 2014, pp. 283-288, doi: 10.1109/MedCom.2014.7006019. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84988299772&doi=10.1109%2fMedCom.2014.7006019&partnerID=40&md5=3a6f89d45aefcab8cc4cc91b4edd11b5
[113] R. C. Luo and T. J. Hsiao, "Dynamic wireless indoor localization incorporating with an autonomous mobile robot based on an adaptive signal model fingerprinting approach," *IEEE Transactions on Industrial Electronics,* Article vol. 66, no. 3, pp. 1940-1951, 2019, Art no. 8360741, doi: 10.1109/TIE.2018.2833021.
[114] F. Bernardini, A. Motroni, P. Nepa, A. Buffi, P. Tripicchio, and M. Unetti, "Particle swarm optimization in multi-antenna SAR-based localization for UHF-RFID tags," in *2019 IEEE International Conference on RFID Technology and Applications, RFID-TA 2019*, 2019, pp. 291-296, doi: 10.1109/RFID-TA.2019.8891990. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85075716041&doi=10.1109%2fRFID-TA.2019.8891990&partnerID=40&md5=855e55953f1a4f196db65d7e81e59d0a
[115] Y. Liu, R. Sun, J. Liu, Y. Fan, L. Li, and Q. Zhang, "Research on the positioning method of autonomous mobile robot in structure space based on UWB," in *2019 International Conference on High Performance Big Data and Intelligent Systems, HPBD and IS 2019*, 2019, pp. 278-282, doi: 10.1109/HPBDIS.2019.8735462. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068381330&doi=10.1109%2fHPBDIS.2019.8735462&partnerID=40&md5=8a9062966f807c98f4d12b0842811244
[116] Song, Q., Li, S., Yang, J., Bai, Q., Hu, J., Zhang, X. and Zhang, A, "Intelligent Optimization Algorithm - Based Path Planning for a Mobile Robot," *Computational intelligence and neuroscience,* vol. 2021, no. 1, p. 8025730, 2021.
[117] A. Zhang, S. Ma, B. Li, M. Wang, X. Guo, and Y. Wang, "Adaptive controller design for underwater snake robot with unmatched uncertainties," *Science China Information Sciences,* vol. 59, pp. 1-15, 2016.
[118] N. J. A. Jesuthas and S. Somaskandan, "Path-finding and Planning in a 3D Environment An Analysis Using Bidirectional Versions of Dijkstra's, Weighted A*, and Greedy Best First Search Algorithms," in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, 2022: IEEE, pp. 1-8.
[119] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu, and T. Liu, "The EBS-A* algorithm: An improved A* algorithm for path planning," *PloS one,* vol. 17, no. 2, p. e0263841, 2022.
[120] D. W. Sari, S. Dwijayanti, and B. Y. Suprapto, "Path Planning for an Autonomous Vehicle based

on the Ant Colony Algorithm: A Review," in *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*, 2023: IEEE, pp. 57-62.
[121] X. Cheng, J. Li, C. Zheng, J. Zhang, and M. Zhao, "An improved PSO-GWO algorithm with chaos and adaptive inertial weight for robot path planning," *Frontiers in neurorobotics,* vol. 15, p. 770361, 2021.
[122] Y.-H. Cheng and C.-Y. Kang, "Application of Genetic Algorithm to Path Planning Problem of Automatic Navigation Parking Spaces in Parking Lots," in *2023 Sixth International Symposium on Computer, Consumer and Control (IS3C)*, 2023: IEEE, pp. 118-121.
[123] M. Thangaraj and R. S. Sangam, "Intelligent UAV path planning framework using artificial neural network and artificial potential field," *Indonesian J. Electr. Eng. Comput. Sci.,* vol. 29, no. 2, p. 1192, 2023.
[124] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research,* vol. 5, no. 1, pp. 90-98, 1986.
[125] J. Luo, Z.-X. Wang, and K.-L. Pan, "Reliable path planning algorithm based on improved artificial potential field method," *IEEE Access,* vol. 10, pp. 108276-108284, 2022.
[126] Z. Zhu, J. Xie, and Z. Wang, "Global dynamic path planning based on fusion of a* algorithm and dynamic window approach," in *2019 Chinese Automation Congress (CAC)*, 2019: IEEE, pp. 5572-5576.
[127] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine,* vol. 4, no. 1, pp. 23-33, 1997.
[128] Q. Zhang, L. Li, L. Zheng, and B. Li, "An improved path planning algorithm based on RRT," in *2022 11th International Conference of Information and Communication Technology (ICTech))*, 2022: IEEE, pp. 149-152.
[129] J. Li, L. Li, J. Qiang, H. Wang, and Y. Cao, "Fast Path Planning Based on Bi-Directional RRT* for Mobile Robot in Complex Maze Environments," in *2023 42nd Chinese Control Conference (CCC)*, 2023: IEEE, pp. 4768-4772.
[130] B. Li and B. Chen, "An adaptive rapidly-exploring random tree," *IEEE/CAA Journal of Automatica Sinica,* vol. 9, no. 2, pp. 283-294, 2021.
[131] X. Gu, S. Neubert, N. Stoll, and K. Thurow, "Intelligent scheduling method for life science automation systems," in *2016 IEEE International conference on multisensor fusion and integration for intelligent systems (MFI)*, 2016: IEEE, pp. 156-161.
[132] A. N. Ragozin and A. D. Pletenkova, "Building a Forecast Using a Linear Prediction Filter for the Purpose of Detecting Anomalies in the Signals of Automated Process Control Systems," in *2023 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2023: IEEE, pp. 560-566.
[133] S. Guan, W. Hu, H. Zhou, Z. Lei, X. Feng, and G. P. Liu, "Design and implementation of virtual experiment for complex control system: A case study of thermal control process," *IET Generation, Transmission & Distribution,* vol. 15, no. 23, pp. 3270-3283, 2021.
[134] S. Neubert, B. Göde, X. Gu, N. Stoll, and K. Thurow, "Potential of laboratory execution systems (LESs) to simplify the application of business process management systems (BPMSs) in laboratory automation," *SLAS TECHNOLOGY: Translating Life Sciences Innovation,* vol. 22, no. 2, pp. 206-216, 2017.
[135] S. Guan, W. Hu, H. Zhou, and G.-P. Liu, "A method of remote experiment for complex energy system: an example for process control experiment of thermal power plants," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, 2020: IEEE, pp. 1501-1506.
[136] A. Patel and J. Patel, "LabMIMO: An Open-source Virtual Laboratory for Process Control," *Journal of Engineering Education Transformations,* vol. 36, no. 1, 2022.
[137] I. Papagianopoulos, G. De Mey, A. Kos, B. Wiecek, and V. Chatziathasiou, "Obstacle detection in infrared navigation for blind people and mobile robots," *Sensors,* vol. 23, no. 16, p. 7198, 2023.
[138] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "An automated elevator management and multi-floor estimation for indoor mobile robot transportation based on a pressure sensor," in *2016 17th International Conference on Mechatronics-Mechatronika (ME)*, 2016: IEEE, pp. 1-7.
[139] J. Palacín, R. Bitriá, E. Rubies, and E. Clotet, "A Procedure for Taking a Remotely Controlled Elevator with an Autonomous Mobile Robot Based on 2D LIDAR," *Sensors,* vol. 23, no. 13, p. 6089, 2023.
[140] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics,* vol. 14, no. 3, pp. 409-425, 2021.
[141] R. Sun, X. Wang, C. Feng, H. Xiao, Z. Xu, and J. Cao, "Design and Implementation of Elevator

Running with the Door Open Failure Monitoring System," in *International Conference in Communications, Signal Processing, and Systems*, 2022: Springer, pp. 44-51.
[142] J. Essien, "Integration of Ultrasonic Range Finder Technology with IoT for Smart Automated Door Control Systems."
[143] J. G. Ramôa, V. Lopes, L. A. Alexandre, and S. Mogo, "Real-time 2d–3d door detection and state classification on a low-power device," *SN Applied Sciences,* vol. 3, no. 5, p. 590, 2021.
[144] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A robust method for elevator operation in semi-outdoor environment for mobile robot transportation system in life science laboratories," in *2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES)*, 2016: IEEE, pp. 45-50.
[145] H. Wu, H. Liu, T. Roddelkopf, and K. Thurow, "BLE Beacon-based floor detection for mobile robots in a multi-floor automation Laboratory," *Transportation Safety and Environment,* vol. 6, no. 2, p. tdad024, 2024.
[146] B. Srisura, P. Tillapart, S. Charoenvikrom, and S. Channarukul, "Altitude Calibration toward Floor Change Detection," in *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, 2021: IEEE, pp. 1-6.
[147] J. Zhao, Y. Chen, and Y. Lou, "A human-aware robotic system for mobile robot navigating in multi-floor building with elevator," in *2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, 2019: IEEE, pp. 178-183.
[148] S. Manzoor, E.-J. Kim, G.-G. In, and T.-Y. Kuc, "Performance evaluation of YOLOv3 and YOLOv4 detectors on elevator button dataset for mobile robot," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 2021: IEEE, pp. 890-893.
[149] J. Liu, Y. Fang, D. Zhu, N. Ma, J. Pan, and M. Q.-H. Meng, "A large-scale dataset for benchmarking elevator button segmentation and character recognition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021: IEEE, pp. 14018-14024.
[150] E.-H. Kim, S.-H. Bae, and T.-Y. Kuc, "Mobile service robot multi-floor navigation using visual detection and recognition of elevator features (ICCAS 2020)," in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, 2020: IEEE, pp. 982-985.
[151] S. Müller, B. Stephan, T. Müller, and H.-M. Gross, "Robust Perception Skills for Autonomous Elevator Operation by Mobile Robots," in *2023 European Conference on Mobile Robots (ECMR)*, 2023: IEEE, pp. 1-7.
[152] Z. Dong, D. Zhu, and M. Q.-H. Meng, "An autonomous elevator button recognition system based on convolutional neural networks," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2017: IEEE, pp. 2533-2539.
[153] Neubert, S., Gu, X., Göde, B., Roddelkopf, T., Fleischer, H., Stoll, N. and Thurow, K, "Workflow management system for the integration of mobile robots in future labs of life sciences," *Chemie Ingenieur Technik,* vol. 91, no. 3, pp. 294-304, 2019.
[154] R. Sarif, M. F. R. Al-Okby, T. Roddelkopf, and K. Thurow, "MOBILE GAS SENSING FOR LABORATORY INFRASTRUCTURE," *IIUM Engineering Journal,* vol. 25, no. 1, pp. 178-207, 2024.
[155] S. Neubert, T. Roddelkopf, M. F. R. Al-Okby, S. Junginger, and K. Thurow, "Flexible IoT gas sensor node for automated life science environments using stationary and mobile robots," *Sensors,* vol. 21, no. 21, p. 7347, 2021.
[156] Thurow, K., Gu, X., Göde, B., Roddelkopf, T., Fleischer, H., Stoll, N. and Neubert, S, "Integrating mobile robots into automated laboratory processes: a suitable workflow management system," *SLAS TECHNOLOGY: Translating Life Sciences Innovation,* vol. 26, no. 2, pp. 232-235, 2021.
[157] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005: IEEE, pp. 2432-2437.
[158] J. Huang, K. Thurow, S. Junginger, H. Fleischer, H. Liu, and V. Q. Do, "IoT based Labware Tracking during Mobile Robot Transportation," in *2023 12th International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2023: IEEE, pp. 405-410.

9 Appendix

9.1 Appendix A: Robots Specifications

Table A1 ASTI robots Specifications

Parameter	Values
Dimensions (L x W x H)	737 mm x 630 mm x 1,515 mm
AMR weight	85 kg (with Telescopic conveyor about 150 kg)
Load capacity	Max. 50 kg (Telescopic conveyor max. 35 kg)
Lift (min./ max. transfer height)	Mobile from 440 mm to 1,100 mm
Lifting speed	0.035 m/s
Speed belt skids	up to max. 0.2 m/s
Max. driving speed	1.2 m/s
Reverse speed	0.3 m/s
Accuracy of navigation	+ 50 mm, 3"
Accuracy of fine positioning	+ 10 mm, 1°
Passable edges	<= 5 mm
Passable gaps (e.g. at lifts)	<= 3 cm
Max. floor slope	3°
Turning circle	800 mm (rotates on the spot)
Battery	8 cells LiFeYPO4 with balancing boards and temperature control, 24VDC,40Ah
Steering and drive system	2 driven wheels (differential drive) and 4 freely rotating wheels.
Certifications	CE (personal safety according to ISO 3691 -4)
Navigation technology	natural navigation based on environmental structures and free movement in the area
Communication	WLAN for order management and state transmission

9.2 Appendix B: Navigation Configurations

Table B1 Key Parameters in Gmapping Configuration

Name	Type	Value	Description
scan_topic	STRING	"scan"	The topic name for incoming laser scanning data. Gmapping uses this data to update and maintain the map.
minimum_score	REAL	0.0	This parameter sets the minimum score required to process each scan. It helps to determine whether a scan can effectively align and contribute to map building.
publish_transform	BOOLEAN	false	Decides whether to publish the robot's pose transformations.
transform_publish_period	REAL	0.0	Sets the period for publishing the pose, which is crucial for real-time systems to ensure that pose updates are synchronized with other parts of the system.
tf_delay	REAL	0.05	This parameter sets a delay for the transformation timestamps, used to address timing synchronization issues.
linearUpdate	REAL	1.0	It controls how far the robot moves linearly before updating the map. Smaller values lead to more frequent updates, which can improve map accuracy but also increase computational load.
angularUpdate	REAL	0.5	It controls how much the robot rotates before updating the map, similar to linearUpdate, and has a significant impact on map accuracy and performance.
resampleThreshold	REAL	0.5	When the variation in particle weights exceeds this threshold, a resampling process is triggered. This parameter is crucial for the efficiency and quality of the particle filter.
particles	INTEGER	30	The number of particles used to represent potential positions in the particle filter. More particles typically mean higher accuracy and computational costs.
xmin, xmax, ymin, ymax	REAL	-100,100, -100,100	These parameters define the physical dimensions of the map, determining the map's size and coverage area.

Table B2. Key Parameter Settings for Particle Filtering

Parameter	Set Value	Description
min_particles, max_particles	300, 2000	Sets the range of particles used in the filter to balance accuracy and computational demand.
update_min_d, update_min_a	0.1, 0.05	Determines the movement or rotation required before updating the particle cloud to optimize updates and efficiency.
laser_min_range, laser_max_range	0.1, 20.0	Specifies the operational range for LIDAR; readings outside this range are ignored to improve efficiency.
resample_interval	2	Dictates the frequency of resampling to manage particle degradation based on system dynamics.
kld_err, kld_z	0.05, 0.99	Adjusts the number of particles dynamically based on environmental uncertainty to enhance efficiency and accuracy.
odom_alpha1 to odom_alpha5	0.02, 0.005, 0.02, 0.001, 0.01	Noise parameters for the odometry model, critical for accurate motion modeling and localization.
laser_model_type	likelihood_field	Defines the LIDAR model, impacting data interpretation and suitability for different environments.
transform_tolerance	0.5	Sets time sync tolerance to handle synchronization of data from various sources and prevent errors.
gui_publish_rate, save_pose_rate	10.0, 15	Controls the update frequency of the interface and pose saving, affecting performance and responsiveness.
beam_skip_distance, beam_skip_threshold, beam_skip_error_threshold	0.5, 0.3, 0.9	Manages laser data preprocessing to streamline calculations and update particle weights faster.

Table B3. Move_base key parameters

Parameter	Value	Description
update_frequency	1.0 Hz	Sets the frequency at which the global costmap is updated. Higher frequencies offer more current information but increase computational demand.
transform_tolerance	0.3 seconds	The allowable delay for transforming data between different frames, accommodating message passing and processing delays.
global_frame	"map"	Specifies the reference frame for the global costmap, typically set to "map" to align with the global map.
robot_base_frame	"base_link"	Defines the frame attached to the mobile base of the robot, used for transformations between the robot and the global frame.
resolution	0.05 meters	The resolution of the global costmap, measured in meters per pixel. Finer resolutions provide more detailed maps but increase computational load.

rolling_window	false	Determines if the global costmap updates as a rolling window centered on the robot, commonly used in local costmaps for dynamic settings.
width	10.0 meters	Defines the width of the global costmap. Larger dimensions provide more environmental context but increase memory usage and processing time.
height	10.0 meters	Defines the height of the global costmap. Larger dimensions provide more environmental context but increase memory usage and processing time.
track_unknown_space	false	Indicates whether areas without information are tracked as unknown space, beneficial for exploration tasks.
obstacle_range	12.0 meters	Sets the maximum distance from which an obstacle will be included in the costmap, influencing the robot's perception of distant objects.
raytrace_range	15.0 meters	Defines the distance for clearing obstacles in the costmap using raytracing from laser scans, helping maintain an accurate representation of clear space.

9.3 Appendix C: Get/Put Configuration

Get/Put general configuration of telescopic conveyor is used as follows:

<iofunction type="setvariable" name="hubhoehetransfer" value="945" />

<iofunction type="setvariable" name="DashDistance" value="60" />

<iofunction type="setvariable" name="LiftTransferHeightDelta" value="55" />

<iofunction type="setvariable" name="TelescopeExtensionLength" value="940" />

<iofunction type="setvariable" name="DashVelocity" value="50" />

Each station adjusts its configuration based on its actual size:

Station 211:

```
<instructionset name="Loading_Station_211_Get" basedon="Loading_MC">
  <replacedata>
    <data old="940" new="955" />
    <data old="945" new="940" />
  </replacedata>
```

Station 212:

```
<instructionset name="Loading_Station_212_Get" basedon="Loading_MC">
  <replacedata>
    <data old="945" new="965" />
  </replacedata>
```

Station 209:

```
<instructionset name="Loading_Station_209_Get" basedon="Loading_MC">
        <replacedata>
          <data old="945" new="995" />
          <data old="940" new="855" />
        </replacedata>
```

Station 313:

```
<instructionset name="Loading_Station_313_Get" basedon="Loading_MC">
        <replacedata>
          <data old="945" new="955" />
          <data old="940" new="955" />
        </replacedata>
```

Station 306:

```
<instructionset name="Loading_Station_306_Get" basedon="Loading_MC">
        <replacedata>
          <data old="945" new="860" />
          <data old="940" new="930" />
          <data old="-300" new="-150" />
        </replacedata>
```

9.4 Appendix D: Virtual PLC

9.4.1 D.1 Communication protocol in Virtual PLC

- AIC <-> Virtual PLC

Exchange Robot-Elevator interaction information

Parties	AIC	Virutual PLC
Protocol	Modbus TCP	
	Master	Slave
Data Type	Holding Register	

	Write	Read
Content	“CallingFloor” “DestinationFloor” “WriteBits” (Table 13)	“SourceFloor” “DestinationFloor” “ReadBits” (Table 13) “ErrorCode” “CurrentFloor”

- Virtual PLC -> Elevator PLC

Call the elevator to the specified floor

Parties	Virutual PLC	Elevator PLC
Protocol	ADAM TCP	
	Master	Slave
Data Type	SingleCoil	
	Write	Read
Content	“CallToFloorE” “CallToFloorA” “CallToFloor2” “CallToFloor3”	No reply

- Virtual PLC <-> IoT module

Collect IoT modules sensors data

Parties	Virutual PLC	IoT Modules
Protocol	UDP Boardcast	HTTP
Data Type	String	JSON
Content	"celiscaIOTInfra"+”TimeStamp”+ Message+TimeInterval	Airpressure Data UltraSonic Data BLE Beacon RSSI

- Virtual PLC <-> AIC

Get robots current status

Parties	Virutual PLC	AIC
Protocol	HTTP	
	Client	Server
Data Type	String	JSON
Content	URL+ VehicleStatus+ SessionId	Robot Status: "Mode", "Isloaded", "Name", "Orientation", "Position" "Soc", "TransportId"

- Virtual PLC -> DataBase

Update robots position to laboratory database

Parties	Virutual PLC	DataBase
Protocol	ODBC(Open Database Connectivity)	
Data Type	OdbcType.Int, OdbcType.Real	
Content	"FloorNo", "XPosition", "YPosition", "Id"	

9.4.2 D.2 Interaction Core Code

This code is the six steps of robot-elevator interaction.

Step 1: Activate job upon receiving a call to the elevator

```
// Checks if the elevator call is initiated and no other process is active
if (writeBit6.Checked == true && readBit1.Checked == false && entryElevatorDoor
== false && AGVisUsingLift == false)
    if (CallingFloorValue.Text == "1" || CallingFloorValue.Text == "2")
    {
        if (readBit1.InvokeRequired == true)
            Invoke(new MethodInvoker(delegate ()
            {
                AGVisUsingLift = true; // Set the robot as using the lift
                readBit1.Checked = true; // Mark the read bit for start
                SourceValue.Text = CallingFloorValue.Text; //Set the source floor
                sourceFloor = CallingFloorValue.Text; //Store source floor value
                entryElevatorDoor = true; //Phase of entering the elevator door
                elevatorRequest(Convert.ToInt32(CallingFloorValue.Text));
// Call the elevator to the source floor
                ElevatorEntryTimer();//Start the timer for entering the elevator
            }));
```

```
        }
    }
```

Step 2: Arrive at the called floor, deactivate job active and enable entrance allowance

```
// Checks if elevator door is open and jobactive is enabled
if (doorStatus == true && writeBit6.Checked == true && readBit1.Checked == true
&& readBit2.Checked == false && entryElevatorDoor == true && AGVisUsingLift ==
true)
{
        Invoke(new MethodInvoker(delegate ()
        {
            stableEntryFloorNum = true;//Fix floor number reading during
entrance
            CurrentFloorValue.Text = sourceFloor; //Display the current floor
            readBit2.Checked = true; // Enable the entrance
            readBit1.Checked = false; // Reset the job active bit
            SourceValue.Text = "0"; // Clear source floor display
            ElevatorEntryTimer();// Start entry timer
        }));
}
```

Step 3: Close the entrance allowance once the robot enters the elevator

```
// Checks if the entrance is allowed and the elevator is ready for operation
if (readBit2.Checked == true && writeBit4.Checked == true && entryElevatorDoor
== true && AGVisUsingLift == true)
{
        Invoke(new MethodInvoker(delegate ()
        {
            readBit2.Checked = false; // Disable the entrance allowance
            entryElevatorDoor = false;//Entering the elevator door is end
        }));
}
```

Step 4: Restart job active after the robot is inside and the destination is set

```
// Checks if destination bits are set and job active is not already set
if (writeBit3.Checked == true && writeBit4.Checked == true && readBit1.Checked
== false && LiftGoDes == false && AGVisUsingLift == true)
    {
            Invoke(new MethodInvoker(delegate ()
            {
                DestinationValue.Text = DestinationFloorValue.Text;
//Set the destination floor display
                readBit1.Checked = true;//Set job active for moving to the
```

```
destination
                destinationFloor = DestinationFloorValue.Text;
// Store the destination floor
                LiftGoDes = true; // Mark as ready to go to the destination
                elevatorRequest(Convert.ToInt32(DestinationFloorValue.Text));
// Call the elevator to the destination
                ElevatorGoDestinationTimer();
// Start the timer to handle elevator going to the destination
                stableEntryFloorNum = false;
// Unset the stable entry floor number
            }));
    }
```

Step 5: Exit allowed after reaching the destination floor

```
// Check the current floor and door status, then enable exit permissions.
if (doorStatus == true && CurrentFloorValue.Text == destinationFloor &&
writeBit3.Checked == true && writeBit4.Checked == true && readBit1.Checked ==
true && entryElevatorDoor == false && exitElevatorDoor == false && LiftGoDes ==
true && AGVisUsingLift == true)
{
        Invoke(new MethodInvoker(delegate ()
        {
            stableExitFloorNum = true; //Fix floor number reading during entrance
            CurrentFloorValue.Text = destinationFloor;//Display current floor
            readBit1.Checked = false; // Deactivate the job active status
            readBit3.Checked = true; // Enable the exit allowed status
            exitElevatorDoor = true; // Set the exit door status to true
            DestinationValue.Text = "0";//Reset destination floor on the display
            if (tmrElevatorDestination.Enabled == true)
// Stop the goto destination timer
            {
                TmrGoBox.Invoke(new Action(() =>
                {
                    TmrGoBox.Checked = false;
                }));
                tmrElevatorDestination.Stop();
                tmrElevatorDestination.Dispose();
            }
            ElevatorExitTimer();// Start the exit timer
        }));
}
```

Step 6: Reset and finalize the elevator usage after the robot exits

```
//Check if the robot leaves the elevator
```

```
if (readBit3.Checked == true && writeBit4.Checked == false && writeBit2.Checked
== false && exitElevatorDoor == true && AGVisUsingLift == true)
{
        Invoke(new MethodInvoker(delegate ()
        {
            //Reset all status
            AGVisUsingLift = false;
            readBit3.Checked = false;
            exitElevatorDoor = false;
            LiftGoDes = false;
            zeroPressureTag = false;
            baseFloortag = false;
            baseFloor = "-1";
            zeroAirPressure = 0;
            LiftDoorStatusBox.Invoke(new Action(() =>
            {
                LiftDoorStatusBox.Text = $"-1";
                CurrentFloorBox.Text = $"-1";
                CurrentAirPressureDiffBox.Text = $"-1";
                ZeroPressureBox.Text = $"-1";
            }));

            LiftDoorStatusBox.Invoke(new Action(() =>
            {
                LiftDoorStatusBox.Text = $"-1";
                DoorStutasBit.Checked = false;
            }));

            usingLiftRobotNum = 0;
            isFindRightRobot = false;

        }));
}
```

Declaration of Education Honesty

I declare that this dissertation, titled "Mobile Robots in Laboratory Automation – A Contribution to Total Automation" was independently authored by me and has not been submitted to any other university. All borrowed content has been clearly cited. I have employed AI tools to refine the language in this dissertation. This dissertation has not been used to obtain any other academic qualifications.

Jiahao Huang

Rostock, October 2024

Abstract

In life science laboratories, the demand for automation is increasingly high. Laboratory automation is more complex than factory automation due to the varied laboratory environments and the need to adapt to different experimental requirements. This dissertation explores the integration of mobile robot systems into complex, distributed life science laboratories to achieve total laboratory automation. The study focuses on enhancing navigation, scheduling, and integration capabilities within these environments. A significant portion of the study involves adapting mobile robot systems to meet the unique demands of life science laboratory settings, emphasizing the deployment of a versatile mobile robot, the MOLAR, designed to streamline operations across various laboratory processes.

At the robot system level, the dissertation introduces the navigation methods of mobile robots. A detailed 2D environmental map was developed using LiDAR, incorporating semantic information to aid in precise movement and interaction in a complex laboratory environment. It also includes the design of transportation tools and charging strategies, and experiments were conducted to test the practical effectiveness of the robot navigation.

At the infrastructure level, the research advanced the integration of mobile robots with the laboratory's existing WiFi network, enabling efficient communication with automated doors and elevators. This section addresses the challenges of using laboratory automatic doors and elevators by developing a dedicated GUI in C# for real-time monitoring and control of the interactions between mobile robots and the elevator system, using IoT technology for precise detection of elevator doors and floors.

At the workflow control level, the mobile robot's control system, AIC, is integrated into the laboratory's high-level Workflow Management system. The mobile robot acts as a bridge connecting distributed workstations within the laboratory, receiving transport commands from SAMI. This enables a fully automated, 24/7 operational laboratory.

This dissertation provides comprehensive insights into the methodologies used for integrating mobile robots into life science laboratories. It offers substantial contributions to the field of robotics and automation, demonstrating the potential of mobile robots to enhance laboratory accuracy, efficiency, and productivity.

Zusammenfassung

In Lebenswissenschaftlichen Laboren steigt der Bedarf an Automatisierung stetig. Laborautomatisierung ist aufgrund der vielfältigen Laborumgebungen und der Notwendigkeit, sich an verschiedene experimentelle Anforderungen anzupassen, komplexer als die Automatisierung von Fabriken. Diese Dissertation erforscht die Integration mobiler Robotersysteme in komplexe, verteilte Labore der Lebenswissenschaften, um eine vollständige Laborautomatisierung zu erreichen. Der Schwerpunkt liegt auf der Verbesserung von Navigation, Planung und Integration innerhalb dieser Umgebungen. Ein wesentlicher Teil der Studie befasst sich mit der Anpassung mobiler Robotersysteme an die speziellen Anforderungen von Laboren der Lebenswissenschaften, wobei der Einsatz eines vielseitigen mobilen Roboters, des MOLAR, hervorgehoben wird, der darauf ausgelegt ist, Operationen in verschiedenen Laborprozessen zu optimieren.

Auf der Ebene des Robotersystems führt die Dissertation die Navigationsmethoden mobiler Roboter ein. Eine detaillierte 2D-Umweltkarte wurde unter Verwendung von LiDAR entwickelt, die semantische Informationen beinhaltet, um präzise Bewegungen und Interaktionen in einer komplexen Laborumgebung zu unterstützen. Sie umfasst auch das Design von Transportwerkzeugen und Lade strategien, und es wurden Experimente durchgeführt, um die praktische Wirksamkeit der Roboternavigation zu testen.

Auf der Infrastrukturebene hat die Forschung die Integration mobiler Roboter mit dem bestehenden WiFi-Netzwerk des Labors vorangetrieben, um eine effiziente Kommunikation mit automatischen Türen und Aufzügen zu ermöglichen. Dieser Abschnitt behandelt die Herausforderungen bei der Nutzung von automatischen Labortüren und Aufzügen durch die Entwicklung einer speziellen GUI in C# für die Echtzeitüberwachung und -steuerung der Interaktionen zwischen mobilen Robotern und dem Aufzugssystem unter Verwendung von IoT-Technologie zur präzisen Erkennung von Aufzugstüren und -etagen.

Auf der Ebene der Arbeitsablaufsteuerung ist das Steuerungssystem des mobilen Roboters, AIC, in das hochrangige Workflow-Management-System des Labors integriert. Der mobile Roboter fungiert als Brücke, die verteilte Arbeitsstationen

innerhalb des Labors verbindet und Transportbefehle von SAMI erhält. Dies ermöglicht ein vollautomatisches, rund um die Uhr betriebenes Labor.

Diese Dissertation bietet umfassende Einblicke in die Methoden zur Integration mobiler Roboter in Labore der Lebenswissenschaften und leistet bedeutende Beiträge zum Bereich der Robotik und Automatisierung. Sie demonstriert das Potenzial mobiler Roboter, die Genauigkeit, Effizienz und Produktivität von Laboren zu erhöhen.

List of Publications

- **J. Huang**, S. Junginger, H. Liu, and K. Thurow, "Correcting of the unexpected localization measurement for indoor automatic mobile robot transportation based on a neural network," *Transportation Safety and Environment,* vol. 6, no. 2, p. tdad019, 2024.

  https://doi.org/10.1093/tse/tdad019

- **J. Huang**, S. Junginger, H. Liu, and K. Thurow, "Indoor positioning systems of mobile robots: A review," *Robotics,* vol. 12, no. 2, p. 47, 2023.

  https://doi.org/10.3390/robotics12020047

- **J. Huang** and H. Liu, "A hybrid decomposition-boosting model for short-term multi-step solar radiation forecasting with NARX neural network," *Journal of Central South University,* vol. 28, no. 2, pp. 507-526, 2021.

  https://doi.org/10.1007/s11771-021-4618-9

- **J. Huang**, S. Junginger, T. Roddelkopf, H. Liu, and K. Thurow, "IoT-Based Solution for Mobile Robots Utilizing Closed-System Elevators in Life Science Laboratories," in *2024 7th Iberian Robotics Conference (ROBOT)*, 2024: IEEE, pp. 1-6.

  https://doi.org/10.1109/ROBOT61475.2024.10796928

- **J. Huang**, H. Liu, S. Junginger, K. Thurow, "Mobile Robots in Automated Laboratory Workflows", *SLAS Technology*, p. 100240, 2024.

  https://doi.org/10.1016/j.slast.2024.100240

- **J. Huang**, K. Thurow, S. Junginger, H. Fleischer, H. Liu, and V. Q. Do, "IoT based Labware Tracking during Mobile Robot Transportation," in *2023 12th International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2023: IEEE, pp. 405-410.

  https://doi.org/10.1109/ICCAIS59597.2023.10382284

- M. F. R. Al-Okby, S. Junginger, T. Roddelkopf, **J. Huang**, and K. Thurow, "Ambient Monitoring Portable Sensor Node for Robot-Based Applications," *Sensors,* vol. 24, no. 4, p. 1295, 2024.

  https://doi.org/10.3390/s24041295

- C. Duan, S. Junginger, **J. Huang**, K. Jin, and K. Thurow, "Deep learning for visual SLAM in transportation robotics: A review," *Transportation Safety and Environment,* vol. 1, no. 3, pp. 177-184, 2019.

https://doi.org/10.1093/tse/tdz019

- K. Thurow, L. Zhang, H. Liu, S. Junginger, N. Stoll, and **J. Huang**, "Multi-floor laboratory transportation technologies based on intelligent mobile robots," *Transportation Safety and Environment,* vol. 1, no. 1, pp. 37-53, 2019.

  https://doi.org/10.1093/tse/tdy002

- K. Thurow, S. Junginger, **J. Huang**, "Mobile Robots in the Lab. When Robots take the Elevator". Wiley Analytical Science Magazine, 2024, 6

Curriculum Vitae

Full name	Jiahao Huang
Place of birth	Hunan, China
Date of birth	26.09.1994
Address	Friedrich-Barnewitz Str.12, 18119 Rostock
Nationality	Chinese
E-Mail	csuhjh@gmail.com

Education

2021-2025	Ph.D. student at the University of Rostock, Center for Life Science Automation Research group “Mobile Robtics” under the supervisory of Prof. Dr.-Ing. habil. Kerstin Thurow Research topic ”Integrating Mobile Robots for Total Laboratory Automation”
2017-2020	M.Sc. Traffic Equipment and Information Engineering at the Central South University of China Research topic ” Dynamic Path Planning and Control of Robot in Intelligent Environment”
2012-2016	B.Sc. Traffic Equipment and Information Engineering at the Central South University of China

Skills

Programming	Proficient in C++, C#, and MATLAB; familiar with Linux operating system, ROS; completed 300+ coding problems on Leetcode
Autonomous Driving	Knowledgeable about the cascade process of autonomous driving, planning and control algorithms, vehicle kinematics/dynamics models
Machine Learning	Experienced in designing neural networks for time series prediction, including NARX and LSTM networks