

# **An Intelligent Multi-Floor Mobile Robot Transportation System in Life Science Laboratories**

Dissertation

for

obtaining of the academic title

Doktor-Ingenieur (Dr.-Ing.)

Faculty of Computer Science and Electrical Engineering

University of Rostock



Submitted by:

Ali Abduljalil Abdulla, born on 25, January 1984 in Baghdad, Iraq  
Rostock, Germany, 2016

**Reviewers:****1. Reviewer:**

Prof. Dr. -Ing. habil. Kerstin Thurow

Institute of Automation, University of Rostock, Germany

**2. Reviewer:**

Prof. Dr. -Ing. Hui Liu

Institute of Automation, University of Rostock, Germany

**3. Reviewer:**

Prof. Dr. William R. Hamel

Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, USA

**Data of Submission:** 1<sup>st</sup> , December, 2016

**Data of Defense** : 20<sup>th</sup>, December, 2017

## **DEDICATION**

To my family

## ACKNOWLEDGMENT

First, my great thanks go to Almighty **ALLAH** who helped me and gave me the ability to achieve this work.

I would like to express my deepest respect and sincere gratitude to my supervisors, Prof. Dr. **Kerstin Thurow** and Prof. Dr. **Norbert Stoll** for their guidance, help, encouragement and advice during my PhD study. Those efforts are highly appreciated.

Furthermore, I would like to extend my thanks to all mobile robot staff members. Especially, Prof. Dr. **Hui Liu** for his continuous encouragement and motivation. He was always accessible to answer all my questions.

My thanks are due to Dr. **Steffen Junginger**, Mr. **Peter Passow**, and Mr. **André Geißler** for their help and assistance during my stay in Germany. I also would like to express my gratefulness to Prof. Dr. **Mohit Kumar**, Mr. **Abdullah Askry**, and Dr. **Ahmed Al-karakchi**, for their assistance and help during my work.

Most importantly, I would like to extend my heartfelt thanks to my parents for their endless love, my mother who always prayed for me and my father who always available to support and guide me. My brothers and sisters for their massive encouragements. My wife, for her unwavering love and continued assistance, I cannot imagine the completion of this work without her assistance.



## ABSTRACT

In recent years, growing numbers of mobile robots have been employed for transportation tasks in indoor environments. In this dissertation, a multi-floor transportation system is presented for mobile robots in life sciences laboratories. There are several new and innovative aspects of this system. Firstly, a new method is presented for mobile robot mapping in multi-floor environments which employs two kinds of mapping. Relative mapping generates a global map with high accuracy and flexibility for expansion into any number of floors, while a path map consists of a set of information waypoints positions which are related to the global map. These waypoints are then utilized in the path planning stage to identify an obstacle-free path. The mapping methods utilize the SGM as a landmark reader to receive the information required.

A new method for mobile robot indoor localization is also proposed which evaluates the robot's position in the global map. The localization method also relies on a passive landmark reader to realize a low-cost solution capable of use in laboratories of any size. This method can be employed to localize any kind of robot inside a multi-floor environment.

A hybrid method for path planning is used to plan a path from the source point to the destination. This method has two working modes: path mapping and an artificial intelligence algorithm. The backbone path planning uses a flexible goal selection method which provides innovations in terms of the optimization of planning time and the number of paths required to reach the destination. The Floyd algorithm is employed to re-plan the path from the current position to the destination if backbone paths become unavailable due to complications such as collision avoidance, blocked paths, or failure in backbone tasks. This algorithm is optimized for indirect graph (bi-directional path) to reduce the required time by approximately -56%. The hybrid Backbone-Floyd method is developed to realize a robust, high-speed and flexible path planning approach. An ITMS facilitates the appropriate path planning method based on the current situation.

A new robust method for elevator operations for mobile robots is then presented which allows the robots to move between different floors in semi-outdoor environments. Two different working strategies are developed to guarantee secure glass elevator handling, using automated elevator and vision detection algorithms. Orders are sent to the elevator over Wi-Fi signals using the former, while the latter deals with entry, the detection of internal buttons and elevator door status, current floor estimation, and control of the kinematic arm for the button-pressing operation by the mobile robot. A pressure sensor attached to the microcontroller board and a smoothing filter with auto-calibration stages are proposed, which

together constitute a stable floor estimation approach for both the automated elevator and vision-based EHS.

A communication network socket has been established to control the human collision avoidance system that detects obstacles and determines a collision-free path for the robot using human-robot interaction. Meanwhile a kinematic arm module performs grasping, placing, and elevator button-pressing actions, and an internal management automated door controlling system controls the opening and closing of laboratory doors during the transportation task.

A charge management system is developed to direct the robot to the charge station smoothly for battery charging. To adapt this for a multi-robot system, a multiple charging station has been initialized with a unique ID for each robot.

Many experiments to assess and verify the performance of the multi-floor transportation system are reported. The experimental results show that the proposed methods and sub-systems developed for the mobile robot are effective and efficient in providing a transportation service in multiple-floor life sciences laboratories.

# CONTENTS

Contents .....	I
List of Figures .....	V
List of Tables .....	IX
List of Algorithms .....	XI
List of Abbreviations .....	XIII
Chapter 1    Introduction .....	1
1.1    Automation in Life Sciences Laboratories .....	1
1.2    Mobile Robots .....	2
1.3    Autonomous Navigation .....	3
1.4    Contribution .....	4
1.5    Outline of the Dissertation .....	6
Chapter 2    State of the Art .....	7
2.1    Introduction to Mobile Robots .....	7
2.2    Mobile Robots in Transportation System .....	8
2.2.1    Factory Transportation System .....	8
2.2.2    Medical Robot Transportation System .....	11
2.2.3    Shopping Robot .....	12
2.2.4    Mobile Robots in Laboratory Environments .....	13
2.2.5    Discussion .....	14
2.3    Navigation .....	15
2.3.1    Sensors .....	15
2.3.2    Mapping .....	21
2.3.3    Localization .....	24
2.3.4    Path Planning .....	32
2.3.5    Elevator Handling .....	36
2.3.6    Discussion .....	41

Chapter 3	General Concept .....	43
3.1	Multi-Floor Transportation System Control Strategy .....	45
3.2	Multi-Floor Transportation System Hierarchy.....	47
3.3	H2O Mobile Robot .....	50
3.3.1	General Introduction.....	50
3.3.2	StarGazer Localization Sensor .....	51
3.3.3	Kinect Sensor .....	53
3.3.4	The Intel RealSense F200 .....	54
3.3.5	The LPS25HP Pressure Sensor with STM32L053 Microcontroller .....	55
3.4	Communication System .....	56
3.4.1	Internal Structure of TCP/IP .....	56
Chapter 4	Navigation System .....	57
4.1	StarGazer Handling API .....	57
4.2	Multi-Floor Navigation System .....	58
4.2.1	Map Building.....	58
4.2.2	Indoor Localization.....	61
4.2.3	Path Map Creation .....	62
4.2.4	Path Planning.....	64
4.2.5	Internal Management Automated Door Controlling System.....	67
4.2.6	Internal Battery Charging Management System (IBCMS).....	69
4.2.7	Movement Core.....	70
4.2.8	Integration of the MFS with other related sub-systems.....	72
4.2.9	Experimental Results.....	77
4.3	Vision Landmarks Reader Error Management .....	82
4.4	Fine Position Correction Method.....	84
4.5	Hybrid Path Planning.....	85
4.5.1	Floyd Method.....	85
4.5.2	The Developed Internal Transportation Management System (ITMS) .....	86
4.5.3	Hybrid Method Experiment.....	88

4.6	Conclusion .....	91
Chapter 5	Elevator Handling.....	93
5.1	Introduction.....	93
5.2	Outside Elevator .....	94
5.2.1	Movement to Elevator Area.....	94
5.2.2	External Button Detection.....	95
5.2.3	Elevator Opening Detection .....	98
5.2.4	Experiments.....	99
5.2.5	Conclusion .....	102
5.3	Inside Elevator .....	103
5.3.1	Localization.....	103
5.3.2	Robot Movement Inside Elevator.....	103
5.3.3	Internal Button Detection .....	104
5.3.4	Internal Button Detection With F200 Hand Camera .....	106
5.3.5	Current Floor Estimation.....	106
5.3.6	Experiments.....	111
5.3.7	Conclusion .....	116
5.4	Automated Elevator Management System.....	117
5.5	Button Pressing Operation .....	119
5.5.1	The Complete Scenario .....	119
5.5.2	Validation Experiments.....	120
5.6	Elevator Handler Error Mangement System .....	121
Chapter 6	Summary and Conclusion.....	125
Chapter 7	Outlook.....	129
References:	.....	131
Appendices	.....	147
Appendix A.	Algorithms .....	147
Appendix B.	Network Protocol.....	151
Appendix C.	Developed GUI.....	155

List of Publications .....	157
Declaration.....	159
Curriculum Vitae (CV) .....	161
Theses.....	163
Zusammenfassung .....	167

# LIST OF FIGURES

Figure 1 : Different Types of Mobile Robot Application. ....	3
Figure 2: Virtual laboratories . ....	9
Figure 3: Work assistant mobile robot . ....	10
Figure 4: OzTug robot with vision sensor and predefined path . ....	10
Figure 5 Hospital robot service system . ....	12
Figure 6: Components of mobile robot transportation system . ....	13
Figure 7: Robot arm for gripping labware in the laboratory . ....	14
Figure 8 Simplified structure of incremental encoder . ....	17
Figure 9 Computer Vision . ....	19
Figure 10: HSL color space . ....	20
Figure 11 Voronoi Diagram . ....	22
Figure 12 Object Oriented Map . ....	22
Figure 13 Occupancy Grid Map . ....	23
Figure 14 Certainty Grid Map . ....	23
Figure 15 Quad tree Occupancy Grid. ....	24
Figure 16 Experimental environments for robot position estimation . ....	26
Figure 17 Mobile robot position based on Omni-camera and a probabilistic method . ....	27
Figure 18: Localization System based on RFID . ....	29
Figure 19: Sensor module and camera . ....	31
Figure 20: Modified A* path planning method when the robot is 2/3 larger than the map cell .....	33
Figure 21: DNC path planning concept . ....	35
Figure 22: Stairway detection by mobile robot in indoor and outdoor environments . ....	37
Figure 23: Mobile robot climbing stairs . ....	37
Figure 24: Elevator Door Detection with angle relations . ....	38
Figure 25 Results for image containing a single character and strong reflections . ....	39
Figure 26: PR2 Robot . ....	40
Figure 27: Complete structure of the multi-floor system. ....	45

Figure 28 Multi-floor system basic structure.....	46
Figure 29 Elevator handler strategy.....	46
Figure 30 Multi-floor transportation system hierarchy .....	47
Figure 31 H2O mobile robot .....	51
Figure 32 Artificial passive ceiling landmark for SGM.....	52
Figure 33 StarGazer module working modes .....	53
Figure 34 StarGazer localization sensor .....	53
Figure 35 Microsoft Kinect sensor .....	54
<i>Figure 36</i> The RealSense F200 camera and its Software development kit .....	55
Figure 37 Internal TCP/IP communication network .....	56
Figure 38 StarGazer handling API for changing the working mode.....	57
Figure 39 Relative map creation .....	60
Figure 40 Relative map in separate file .....	61
Figure 41 Localization method in multi-floor environment .....	62
Figure 42 Path waypoints parameters .....	62
Figure 43 Bi-directional Path Structure .....	64
Figure 44 path points and it is file name stored in a separated file .....	64
Figure 45 Waypoint distribution in the backbone method .....	66
Figure 46 Internal Transportation Management System.....	67
Figure 47 Internal Management Automated Door Controlling System (IMADCS).....	68
Figure 48 Automated door working mechanism inside movement core .....	68
Figure 49 Internal Battery Charging Management System flowchart .....	69
Figure 50 Multi-robot distributed around charge station.....	70
Figure 51 Movement core flow chart .....	72
Figure 52 Complete socket structure for the MFS.....	73
Figure 53 Data flow between the RRC, MFS, and H2O robot .....	74
Figure 54 Data flow between the MFS, elevator handler System, and RAKC.....	74
Figure 55 Data Flow for Automated Doors Control and CAS .....	75
Figure 56 The developed socket connection to all sub-system .....	76
Figure 57 Multi-floor main GUI.....	77



Figure 58 Real mobile robot transportation path on multiple floors .....	78
Figure 59 Execution of transportation task in multi-floor environment .....	78
Figure 60 A GUI of the IMADCS .....	79
Figure 61 Reflective ceiling landmark on the glassy wall.....	79
Figure 62 Ruler scale for robot position measurement.....	80
Figure 63 Robot movement position error .....	80
Figure 64 MFS<->RRC socket communication.....	81
Figure 65 Real mobile robot transportation path on single floor .....	82
Figure 66 Vision landmark reader error handling .....	83
Figure 67 Vision landmarks reader error movement scenario .....	83
Figure 68 Vision landmarks reader error management.....	83
Figure 69 Fine position correction based on the last movement direction.....	84
Figure 70 Floyd algorithm time cost .....	86
Figure 71. Hybrid Backbone-Floyd method.....	87
Figure 72 Shortest Path Procedure .....	87
Figure 73 Path generated by robot movement in multi-floor environment .....	88
Figure 74 Contrast experiment for fine method .....	90
Figure 75 Elevator handling sequential sub-process .....	93
Figure 76 Robot position outside elevator .....	95
Figure 77 Kinect calibration stage.....	96
Figure 78 HSL-based entry button detection .....	97
Figure 79 Entry button detection with RGB-D arm camera .....	98
Figure 80 Opening detection method.....	98
Figure 81 Glassy elevator environment and sun's direct effect on the Kinect camera.....	99
Figure 82 EEBD with/without direct sunlight and at different distances .....	100
Figure 83 Entry button detection based on new F200 camera.....	101
Figure 84 Ceiling light natural landmark (left) and artificial landmark (right) .....	103
Figure 85 Robot behavior inside elevator .....	104
Figure 86 Internal button detection method .....	105
Figure 87 Inside elevator BP detection and floor number extraction .....	106

Figure 88 Elevator BP detection and floor number extraction method .....	106
Figure 89 Elevator floor reader.....	107
Figure 90 Finite impulse response filter .....	108
Figure 91 Microcontroller flowchart .....	109
Figure 92 Range of height measurements for each floor .....	110
Figure 93 Embedded floor estimation inside MFS.....	110
Figure 94 Destination floor button recognition (DFBR): processed image without BP .....	111
Figure 95 Destination floor button recognition (DFBR): success rate without BP .....	112
Figure 96 Destination floor button recognition: processed image with BP.....	112
Figure 97 Destination floor button recognition: success rate with BP .....	113
Figure 98 Current floor reader (CFR) processed image .....	113
Figure 99 Pressure sensor data and the smoothing filter. ....	114
Figure 100 Daily pressure changing over the week.....	115
Figure 101 Automated elevator with ADAM socket and EMS .....	117
Figure 102 Transportation Experiment .....	119
Figure 103 Complete scenario .....	120
Figure 104 Complete operation scenario.....	121
Figure 105 arm pressing position error.....	121
Figure 106 Outside elevator error handling flow chart. ....	122
Figure 107 Inside elevator error handling flowchart .....	123

## LIST OF TABLES

Table 1 Static method and backbone method comparison (path required).....	66
Table 2 Accuracy, mean and standard deviation of important robot positions in cm.....	81
Table 3 Floyd time cost comparison .....	86
Table 4 Time consumption for mobile robot transportation in multi-floor environment. ....	88
Table 5 Comparison of methods developed for navigation to reach important points.....	90
Table 6 Repeatability and tolerance for elevator entry button detection (EEBD) .....	100
Table 7 Entry button detection in sunny weather and at different distance .....	101
Table 8 Entry button detection in cloudy weather at different distances.....	102
Table 9 Destination floor selection strategy .....	104
Table 10 False readings in DFBR without BP .....	112
Table 11 Height measurement in meters with different filter orders.....	115



## LIST OF ALGORITHMS

Algorithm 1: Localization Method in Multi-Floor Environment .....	147
Algorithm 2: Floyd searching process .....	147
Algorithm 3 Optimized Floyd for indirect graph algorithm.....	148
Algorithm 4: A Reactive Path Planning Strategy.....	148
Algorithm 5: Elevator internal button detection.....	149
Algorithm 6: Elevator floor detection .....	149



## LIST OF ABBREVIATIONS

AMR	autonomous mobile robot
AGV	automated guided vehicles
AUAV	Autonomous Unmanned Aerial Vehicles
AUV	Autonomous Underwater Vehicles
CELISCA	Center for Life Science Automation
PLM	Product line manager
PCAS	Process Control Adaptor System
LIMS	Laboratory Information managements system
HWMS	Hierarchical Workflow Management System
TACS	Transportation and Assistance Control System
RRC	Robot Remote Center
RBC	Robot Board Computer
SGM	StarGazer Module
RAKC	Robot Arm Kinematic Controller
CAS	Collision Avoidance System
ICP	Iterative Closest Point
ITMS	Internal Transportation Management System
IBCMS	Internal Battery Charging Management System
IMADCS	Internal Management Automated Door Controlling System
ITMS	Internal Transportation Management System
EHS	Elevator Handler System
RAKM	Robot Arm Kinematic Module
FMCWs	Frequency Modulated Continuous Waves
AFM	Any-time Motion Planning





# **Chapter 1      INTRODUCTION**

## **1.1 Automation in Life Sciences Laboratories**

The life sciences comprise various fields, including biochemistry, biotechnology, agriculture and medical technology. The automation of laboratories in these fields is being established in different areas, from applied research to production and drugs development. The automation of life science laboratories has many advantages, including saving staff time in performing repetitive tasks, data analysis, ensuring safety, reducing the need for staff training, reducing human errors, increasing productivity, and improving the throughput of processes and the accuracy of experiments [1]. The automation of life science laboratories is a relatively new field, and its development is causing huge changes in working processes [2]. This advancement was brought to realize the demands for higher capacity, throughput and efficiency. In accomplishing complicated tasks in a large system in automated laboratories, many sub-systems have to be integrated. In engineering, system integration is defined as the process of bringing together component of sub-systems into one system and ensuring that the sub-systems function together as a single system [3]. Advanced automated laboratories integrate many sub-systems such as high-throughput screening, analytical and measurement systems, product line managers (PLMs), and robotic systems. Two kinds of robotic systems using stationary robots [4][5] and mobile robots [6][7] are commonly used in laboratory automation processes [8][9] to increase productivity with higher precision and accuracy, and to decrease costs.

Mobile robots in automated laboratories are usually used for the transportation of samples and other material between sub-systems located in different places. Meanwhile automated laboratories typically lack space due to the large amount of equipment. Each sub-system in an automated laboratory has a limited work space, and thus, an optimal solution for a mobile robot navigation system to work conveniently and safely will require an effective method to avoid collisions with the building structure, equipment, other robots and humans.

At CELISCA, researchers are working to develop the automated life science laboratory of the future. For a facility such as a laboratory, a system consisting of several mobile robots has been established which allows samples, labwares and materials to be transported between individual automated sub-systems as support for laboratory staff or to replace them outside of working hours. In transportation systems for mobile robots, a major topic to be handled is the design of an appropriate navigation mechanism.

One of the key tasks for the mobile robot is the ability to detect its surrounding environment in order to interact with the environment that it is employed in. Thus different kinds of sensors have been utilized (see section 2.3.1). Many issues should be considered in realizing an efficient navigation system, including questions of the representation of the working environment (mapping), knowledge of the mobile robot's current position within the working environment (localization), and a strategy for motion that connects the source point to the destination point (path planning).

The present study concerns the use of mobile robots in a multi-floor transportation system, and this dissertation presents an innovative navigation system for mapping and localization, path planning, elevator handling, and other relevant systems including communication, automated door control, battery charging management, and an error management strategy. In addition, control techniques for other related sub-systems, including robot arm manipulation and collision avoidance working processes are designed.

## **1.2 Mobile Robots**

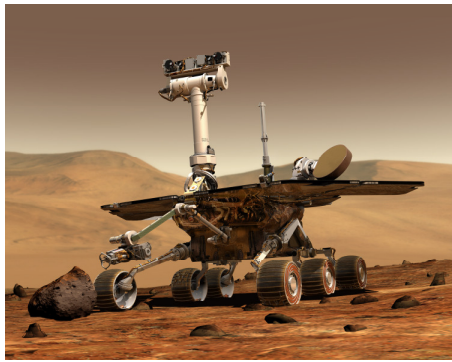
Over the last two decades, the world of robotics has witnessed steady development and invention. It can be described as the current top of the technical advancement. Robotics is a branch of mechanical engineering, electrical engineering and computer science that deals with the design, construction, operation, and application of robots as well as computer systems for their control, sensory feedback, and information processing. The field has developed quite quickly from industrial robots, which are typically attached to a fix surface to perform repetitive tasks, to more complicated machines enabling the replacement of humans in performing challenging tasks.

Mobile robots have attracted special attention since they are able to navigate in their environment and do not suffer from a limited working range in contrast to classical stationary industrial robots. Mobile robots can be autonomous mobile robots (AMRs) [10], which are stand-alone machines with their own controllers, and are used to perform tasks in an uncontrolled and unpredictable environment. Working without an onboard operator depends on a combination of software and sensor-based guidance systems to navigate through pre-defined paths in relatively controlled environments, such as in automated guided vehicles (AGVs) [11]. Mobile robots can play a widespread and critical role in many applications, including (but not limited to) medical and cleaning services, agriculture, operations in hazardous environments, construction and demolition, space and military applications, materials handling, support services for the elderly and handicapped, and entertainment. Figure 1 shows different types of mobile robot applications.

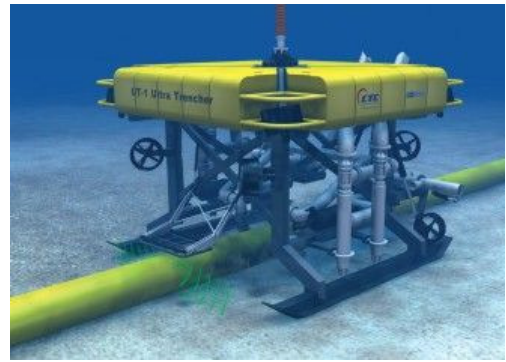
### 1.3 Autonomous Navigation

Autonomous robots are intelligent machines which have the ability to execute tasks without human intervention. They can receive information about their environment, work for a long time without human assistance, and realize safe navigation which avoids harming people, other assets, and themselves. Navigation is in this context a process which allows an autonomous mobile system to intelligently move and interact with its environment.

In contrast to certain mobile robots used for surveillance and cleaning services, which can utilize random movement in performing their tasks, the majority of mobile robot application requires a sophisticated navigation ability in order to move in the desired manner. Thus, an autonomous navigation system is a fundamental function for mobile robots.



*(a) Spirit: NASA's Mars Exploration Rover*



*(b) UT-1 Ultra Trencher: Underwater Ditch-Digging Robot to prepare the pipelines and cables installation*



*(c) RP-VITA: a medical robot which enables the doctors to remotely visit the patients using pc installed in the robot.*



*(d) KMR QUANTEC: mobile robot work in industrial application for heavy load transportation with autonomous navigation capability*

*Figure 1 : Different Types of Mobile Robot Application.*

The mobile robot has to solve four basic problems in order to achieve highly effective autonomous navigation, which are mapping, localization, path planning, and action.

Mapping uses the data gathered by sensors to represent the environment in a spatial model. Localization is the estimation of the robot's own position inside a map. Path planning is a strategy to create a path towards a destination position which should also allow obstacle avoidance. Once a path is created, action or path execution takes place which can adjust the motor action in response to changes in the surrounding environment.

For robots, sensors can be described as windows onto their environment. Sensors provide the robot with the necessary information about its internal state in addition to aspects of its surrounding environment. Many sensors with different functions and characteristics are available for autonomous robots. They can be classified based on resolution, accuracy, compatibility, and the type of data processed. Another general classification depends on the nature of the sensor's operation, which can be passive or active. Active sensors send and receive signals to sense their environment, while passive sensors merely collect data via observation without emitting any signals. A further classification is based on the sensor's working position, which can be external and internal. Internal sensors are responsible for monitoring the robot's state, including variables such as speed, temperature, or battery charge levels, while external sensors are responsible for measurements in the robot's environment and include range-finders or approximation sensors.

In summary, autonomous navigation is the ability of a mobile robot to collect sufficient information about the surrounding environment, to process and interact with it, and to move along a safe path in this environment, typically using a predefined path. The ability to sense the environment is an essential requirement for any autonomous system, and this role is performed by the sensors.

## **1.4 Contribution**

The main contribution of this dissertation is the design of a new multi-floor transportation system in laboratory environments, including the following components:

- a) A multi-floor mapping system is designed in which two kinds of maps are developed for autonomous mobile robots. Relative mapping provides a representation of the multi-floor global map, and path mapping is based on this to define the position of a set of rich waypoints which are then used as the basis for path planning and movement core.
- b) An indoor localization method is proposed which relies on a passive landmark reader to realize a low-cost robot position estimation solution to be used in laboratories of

any size. This method can be employed to localize any kind of robot inside a multi-floor environment.

- c) A hybrid method for path planning is developed which plans a path from the source point to the destination. This method has two working modes: path mapping and an artificial intelligence algorithm. Backbone path planning uses a multiple segmented shortest path planning method with flexible goal selection which provides innovations in terms of the optimization of the time required for path planning and the number of paths required to reach the destination. The Floyd algorithm is employed to re-plan the path from the current position to an intermediate destination if backbone paths become unavailable. It then returns control back to the backbone path planning to determine the subsequent path to the destination.
- d) An internal transportation management system is presented which controls the execution of the mobile robot transportation task after enabling the appropriate path planning method. In addition, it controls the working process of the relevant sub-systems, including the robot arm kinematic module, collision avoidance system, automated door controller, elevator handler, and automated elevator controller.
- e) An elevator handling system for a semi-outdoor environment is designed in which two different working strategies are developed to guarantee secure glass elevator handling using automated elevator and vision detection algorithms. Orders are sent to the elevator over Wi-Fi signals using the former, while the latter deals with the detection of entry and internal buttons and elevator door status, and controls the kinematic arm for the button-pressing operation. A pressure sensor is utilized as a stable floor estimation approach for both the automated elevator and vision-based elevator handler.
- f) The software architecture for the system is developed, and a huge program is divided into smaller parts based on the server-client structure to minimize complexity. Each part has a specific function, such as elevator handling, collision avoidance, and the robot arm's kinematic module. These functions work as servers while the multi floor system manages the transportation operations and utilizes the smaller part as plugin functions. Any number of mobile robots or devices can easily be added to this structure with new IPs.
- g) A charge management system is developed to direct the robot to the charging station smoothly for battery charging. To adapt this for a multi-robot system, a multiple

charging station has been initialized with a unique ID for each robot. Multiple charging stations have been established with unique ID's.

## **1.5 Outline of the Dissertation**

The organization of this dissertation is as follows:

**Chapter 2** gives a literature review covering mobile robots, robot applications in transportation systems, mobile robot localization, mapping methods, path planning, and elevator operation handling.

**Chapter 3** introduces the robot selected to perform transportation tasks in this research, along with the multi-floor system hierarchy with detailed description of each part, and the internal communication sockets.

**Chapter 4** focuses on the mobile robot navigation system in multi-floor environments, and starts by introducing the system developed for mapping, localization, path planning, error management, communication, automated door controller management, and charging management. Moreover, the experimental results to check all these developed systems are incorporated.

**Chapter 5** explains the problems of elevator handling and the solutions developed for elevator entry button detection under complex light conditions, internal button detection, localization inside the elevator, and door status detection. It also describes the automated elevator management strategy, error handling system for inside/outside elevator, height measurement system for floor estimation, finally the validation of all these developed systems with their results have been presented.

**Chapter 6** gives a detailed summary of the work described in this dissertation, along with conclusions concerning system performance.

**Chapter 7** presents recommendations for future research.

## **Chapter 2      STATE OF THE ART**

In the design of a robotic system, existing mechanisms, sensors and navigation strategy options should be taken into consideration. Knowledge of the advantages and shortcomings of existing systems will result in the selection of components and architectures best suited to any given task.

In this chapter, an introduction to mobile robots is presented, along with discussions of their application in transportation systems for multiple working environments, commonly used kinds of sensors, the employed mapping methods, various types of localization and planning methods and, finally, elevator recognition and handling.

### **2.1 Introduction to Mobile Robots**

At the end of 2014 there were more than 1.5 million robots around the world, virtually all of them being industrial robots. Arm manipulators represent a sector worth 10.7 billion dollars [12], where industrial robots are usually attached to a fixed surface to perform repetitive tasks with high speed and accuracy. Compared to the obvious success of industrial robots, mobile robots are still of minor importance. Mobile robotics is the branch of robotics dealing with movable robot systems that are able to locomote within an environment or terrain. In contrast to industrial robots they are attached to the mobile platform, so, they have the ability to move between different locations. Mobile robots can be divided into five main categories, which are: wheeled, tracked, legged, flying, and underwater robots. Wheeled robots are the most popular kind of mobile robots due to their low costs, simple control, and ease of design, construction and programming [13][14]. They use wheeled motors to drive themselves on a flat platform. The main disadvantages of wheeled robots are the risk of wheels slipping and the small area of contact with the ground, which lead to an inability to navigate well over rugged terrain or in areas with low friction. Three wheels are sufficient to provide efficient balance for the mobile robot, but additional wheels can decrease slippage while increasing costs.

Another kind of mobile robot is the tracked mobile robot. This kind uses tracks whose movement is similar to that of tanks [15]. Very effective in rugged terrain or loose environments, these robots have the ability to reduce slip due to large contact surface with the ground, and they can move over various different surfaces due to uniform weight distribution. However, they are less effective than wheeled vehicles due to their complex mechanics, and they may also damage the surface being moved over.

Recently, an increasing number of mobile robots use the principles of legged movement [16][17]. Legged robots are favored for utilization in very rough terrains since legs are more effective than wheels in such environments. Compared to wheeled mobile robots, legged robots separate the robot body from the terrain surface, and they can evade unwanted footholds and cope with relatively large obstacles. Their movement is in fact closer to human motion. On the other hand, legged mobile robots suffer from stability difficulties and high building costs as well as mechanical and programming complexity, and tend to have small batteries.

These three kinds of mobile robot are land-based. Two other kinds of mobile robots operate in the air and underwater. Air-based mobile robots are known as AUAVs (Autonomous Unmanned Aerial Vehicles)[18]. They were developed to execute military operations requiring repetitive, precise, and hazardous tasks to be performed over a long period of time. This kind of robot is efficient in surveillance operations due to the ability to fly in different weather conditions.

Underwater-based mobile robots are also known as Autonomous Underwater Vehicles (AUVs) [19]. These robots are designed to perform their tasks underwater, mostly at great depth. AUVs have a wide range of applications in different fields such as scientific research, military operations, and especially in marine geoscience. Their autonomous operation also makes them suitable for extreme environments, such as reconnoitering from the deepest hydrothermal vents to under polar ice-sheets. AUVs enable the imaging of the seafloor, and achieve high resolution mapping data.

Mobile robots are ubiquitous, and can be found in households and industry, earth and space exploration, and in military and surgical jobs. One of the fundamental types of tasks for the mobile robots, however, is transportation.

## **2.2 Mobile Robots in Transportation System**

### **2.2.1 Factory Transportation System**

Many different kinds of robotic system are utilized every day in manufacturing. Typically, the industrial robot has a jointed arm with an end effector, and is attached to a fixed surface to execute repetitive tasks in order to increase productivity. The fixed robot will always have a place in industrial fields but, on the other hand, the mobile robot promises the flexibility to serve in different places and to increase the range of robot applications, especially in transportation or delivery tasks.



Usually, mobile robots employed in factories are called autonomous guided vehicles (AGVs). The majority of industrial robot autonomous navigation systems are based on marks, wires, and sensors. Industrial robots improve work processes by decreasing costs, increasing productivity and accuracy, and saving staff resources.

**L. Ribas-Xirgo** presented a virtual laboratory for a manufactured plan simulation environment, which could be used for teaching and manufacturing development purposes [20]. Here, materials transportation and handling are accomplished by a KUKA youBot mobile robot, which implements the VRep physical simulator as shown in Figure 2. Despite the success of the experiment in teaching aspect, errors associated with sensors and actuators as well as floor irregularities were not considered in this virtual lab.

**Hong et al.** developed two types of assistance mobile robots for handicapped persons in real factory environments [21]. The first type utilized two-wheel differential mechanism for driving while the second type used an omnidirectional driving mechanism in order to minimize robot size and its turning radius. Experiments were conducted allowing a handicapped person to move boxes from one location to another. These experiments proved that disabled persons can work in a factory environment with the help of this kind of mobile robot, as shown in Figure 3. The omnidirectional mechanism provides the flexibility for the robot to move around in relatively small locations, but at the cost of lower efficiency.

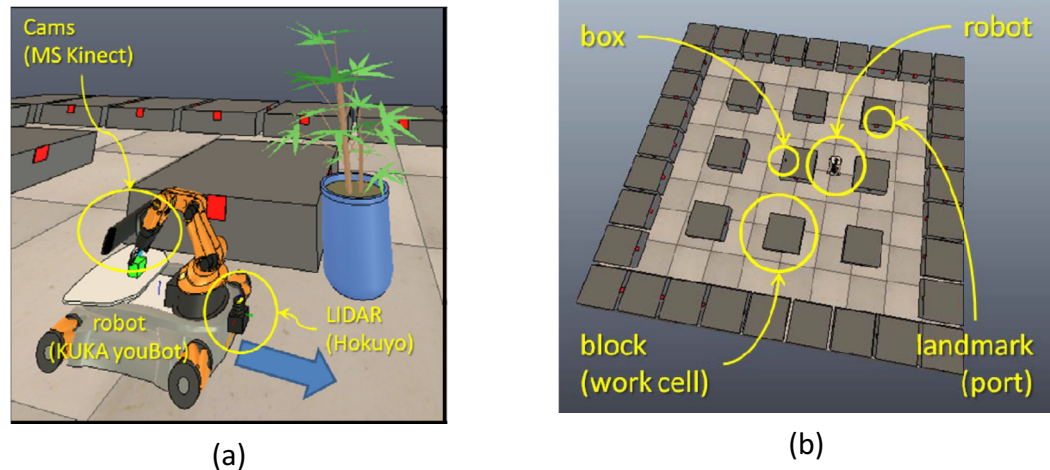


Figure 2: Virtual laboratories: (a) Sensor equipped with the model. (b) The robot in VRep simulator [20].



(a)



(b)



(c)



(d)

Figure 3: Work assistant mobile robot: (a) Moving. (b) Moving Around. (c) Working at table. (d) Moving a box [21].

**Horan et al.** developed a kind of mobile robot called OzTug for factory transportation [22]. This robot can transport very heavy loads up to 2,000kg. The development of OzTug used vision sensors with fuzzy logic to track a predefined line in the manufacturing environment, as shown in Figure 4. Simulation experiments were conducted in a 10\*10 m workspace to evaluate the performance of single and dual OzTug robots for load transportation. The results showed that the area error for a single robot attached to the load reached 16.84m<sup>2</sup>, compared to 4.34m<sup>2</sup> for load transportation based on a dual robot.

More about mobile robot transportation in factory environments can be found in published reviews [23]–[26].

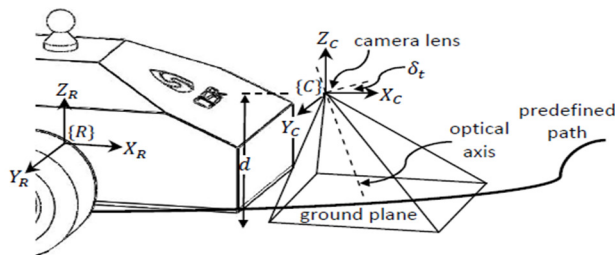


Figure 4: OzTug robot with vision sensor and predefined path [22].

### 2.2.2 Medical Robot Transportation System

Mobile robots used for transportation in medical fields perform tasks such as the delivery of drugs, laboratory supplies, food, and linen. The essential aim of these robots is to save the time of medical staff who can then focus more on patient care. Usually in hospitals huge quantities of articles need to be transported in large areas. Thus, accurate, safe, and low-cost delivery operations are required. Automating delivery work can enable hospitals to increase productivity, worker safety, and regulatory compliance, and may help to eliminate mistakes made by staff and missing drugs.

**Panasonic** developed a drug delivery robot called HOSPI [27]. HOSPI has been used for medical transportation since January 2011 in the Matsushita Memorial Hospital in Japan [28]. **Murai et al.** combined a sensor installed inside the HOSPI robot with a new localization method based on visible light communication [29]. In this method a special LED ceiling light was installed before each stairway, escalator, and in any places that the HOSPI should not enter. Experiments showed that this system is effective and convenient for hospital transportation. The main disadvantage of the method is that the lighted landmarks are difficult to maintain.

**Carreira et al.** presented a mobile robot concept for the transportation of meals called i-MERC [30]. This robot is supposed to keep meals warm and to guard against bacterial infection using an embedded heating system. It can transport meals between the kitchen, patient rooms, the dishwashing room, and the mobile robot station. However, the researchers failed to consider the challenges which would be faced in real world applications.

**Nguyen et al.** developed a transportation system for patients in the hospital environment [31]. They combined a mobile robot called Turtlebot with an intelligent bed, as shown in Figure 5. This bed can detect the robot and follow it without a nurse's help. Experiments showed that a control strategy should be developed to increase the stability of the overall system and other intelligent tasks must be considered (such as a collision avoidance strategy) in order to realize safe navigation in hospital environments.

**Cazangi et al.** developed a navigation strategy to control a service mobile robot in hospitals [32]. The robot used an RFID system to follow the doctor or nurses in order to assist them. However, the tracking navigation system was not evaluated in a real environment. Unfortunately, the success of the proposed system relies completely on the RFID precision in providing the robot navigation system with RFID tags movement information.

More information on the use of mobile robots in health applications can be found in [33]–[36].



Figure 5 Hospital robot service system: (a) Turtlebot mobile robot; (b) Intelligent Hospital Bed [31]

### 2.2.3 Shopping Robot

The development of mobile robots for shopping have been witnessed advances in recent years. Usually, shopping assistance mobile robots have an autonomous navigation system for use in malls or supermarkets. The main aim of the shopping robot is to render shopping more comfortable, especially for older and disabled people. The classical tasks performed by shopping assistance mobile robots are handling a shopping list, offering information about products, guiding the human to the desired product, and transporting the chosen products.

**Kohtsuka et al.** introduced an autonomous shopping cart to assist disabled and elderly customers [37]. This shopping cart carries goods and follows the customer. A laser range-finder is utilized in a collision avoidance system and to measure the distance to the customer. This proposed system has not yet been implemented in a real shopping environment. It uses a simple tracking method which means that the robot may easily lose track of the designated customer if another person passes between the latter and the shopping cart.

**Kulyukin et al.** developed a shopping mobile robot for visually impaired people [38], where RFID tags are distributed on the shelves for robot localization. This robot has the ability to navigate autonomously in the grocery store, guiding the customer to a good location, and transporting goods using a basket integrated in the robot. This robot system is expensive for large area store and it has not been evaluated in a real market environment.

**Matsuhira et al.** presented a mobile robot transportation system for shopping [39]. Figure 6 shows the system's components, which consist of an autonomous mobile robot guide containing the localization system and another mobile robot called the cart robot which has a shopping basket. Finally, a camera system is used to recognize between the human and robot. The system was validated in experiments and could work autonomously in a shopping environment. However, the system is very expensive since it requires two mobile robots assigned to each customer and a huge number of fixed ceiling cameras to cover a normal market size. Furthermore, the requirement of two robots for each customer would lead to a shopping environment crowded with more robots than humans.

More information about mobile robots for shopping can be found in [40]–[43].



*Figure 6: Components of mobile robot transportation system [39].*

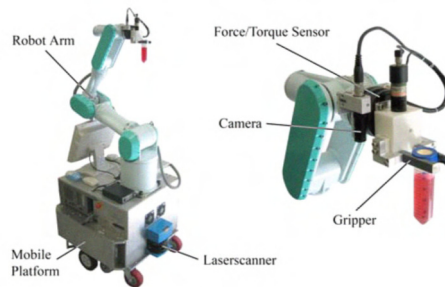
## 2.2.4 Mobile Robots in Laboratory Environments

The development of the automated laboratory requires a multidisciplinary strategy to investigate, improve, and benefit from available techniques to increase laboratory productivity and accuracy, decrease labor costs, eliminate human errors and pollution, execute experiments previously impossible in the laboratory environment, and save the time of scientific staff. Transportation systems in life sciences laboratories involve special requirements to integrate with existing automated systems to provide higher quality work processes and achieve the desired goal. The success of the transportation system depends particularly on other associated systems to transport the functionality to it in a suitable time.

**Wojtczyk et al.** presented a new approach for service mobile robots in laboratories. The robot has mobile parts integrated with a robot arm for the carrying of samples in life science laboratories [44]. The robot is equipped with two laser range-finders for localization and obstacle detection, a vision sensor for vial localization and detection, a gripper for vial grasping, and a force sensor to sense the pressure applied to the vial. This robot can identify a sample and grip it, as shown in Figure 7. The main disadvantages of this system are the high cost of the integrated laser range-finders, and the inability of the robot to handle dynamic obstacles in its path.

**Scherer et al.** presented a sample management system based mobile robot for automation laboratory [45]. In this system a differential driver with an odometer sensor, two laser range-finders, and a gyroscopic compass are used for navigation in the laboratory environment. A global map is built based on a set of laser reflector marks which are distributed in the robot's workspace. These landmarks have a global reference, and the A\* search algorithm is utilized for static path planning and a vision sensor is used for arm position compensation. Manual offline measurements were conducted to show the position accuracy. The accuracy in this work reached less than 1 cm. The main limitation of this system is its cost, which mainly related to very high cost of laser range-finders.

**Liu et al.** developed a new laboratory mobile robot transportation system (LMRTS) for life sciences automation [46]. In this system, multiple robots called H20 mobile robots (DrRobot Canadian Company) have been configured to realize a single-floor transportation system. The Stargazer module is utilized for mapping and localization and the hybrid Floyd-Dijkstra method is used for path planning. The main limitation of this system is that it works only in a single floor environment. For multiple robot the mapping, localization, and path planning configuration processes need to be repeated for each robot since the map completely implemented in the SGM. Thus, the mobile robot has to manually move around the building to build a new map separately for each robot, a new waypoints have to defined in the whole building based on the built map for path planning, finally assist points need to define open/close doors waypoints for automated door controller management system. These process have to be performed if any map needs to enlarge the working environment even with one landmark.



*Figure 7: Robot arm for gripping labware in the laboratory [44]*

More about mobile robotic applications in laboratories can be found in [49]–[52], [53].

### 2.2.5 Discussion

From previous research related to mobile robot applications in transportation tasks, the following conclusions can be drawn:

- a) In recent years, the development of transportation systems based on mobile robots has progressed rapidly to meet requirements such as high precision, routine task execution, transportation in hazardous areas, and low-cost manufacturing. On the other hand, the rapid development of robot technology is helping to reduce the production costs of highly efficient and reliable mobile robots.
- b) Land-based robots are presently the most common kind of robots. Each mobile robot locomotion mechanism has a specific environment to work in. Wheeled mobile robots are preferred compared to other kinds of land-based mobile robot in indoor environments due to their low cost, simple control, ease of design, construction and programming, and effectiveness on flat platforms.

- c) Most of the mobile robot transportation systems mentioned above are not suitable for laboratory environments for various reasons. For example, a laboratory transportation system based on a mobile robot requires higher precision of movement, lab equipment must be handled, integration with automation islands is necessary, and scheduling the robot's activity in accordance with the main laboratory's control schedule. Moreover, the mobile robot transportation systems developed so far cannot work in multi-floor environments, but only on a single floor of restricted size.

## **2.3 Navigation**

Navigation is the process of employing sensor data to create a model of the environment and then using it to guide the robot to the required goal. A navigation system has four main aspects: mapping, localization, path planning, and action. Mapping involves building an environmental model using information from the sensors, localization concerns the ability of the mobile robot to estimate its relative position within the map which has been built. Mapping and localization are later utilized to plan the robot's path to the destination. Once the path is realized, the robot's actions consist of its movements determined by decisions taken, with some movement constraints such as speed and the required time specified by a human operator. In this section, issues related to commonly used sensing systems, and mapping, localization, and path planning methods are clarified and their advantages, disadvantages and limitations listed in discussions of the relevant literature.

### **2.3.1 Sensors**

The robot must collect information about itself or its surrounding environment, and this requires sensors. Large numbers of different types of sensors for mobile robot systems are available with varying levels of accuracy, reliability, and functionality. The robotic navigation system relies completely on the sensory data received to enable the right decisions to be made. All sensors have a degree of error and uncertainty, which can be reduced via the integration and fusion of multiple sensors. Sensors can be classified as either passive or active. Passive sensors sense the environment and the robot's internal state by observation without emitting any signal, providing data on, for example, battery voltage, motor speed, and optical data. Active sensors, meanwhile, emit energy signals to detect the objects by their reflection.

#### ***2.3.1.1 Laser Range Finder***

The laser range finder is a favored sensor for obstacle range recognition and the mapping of surrounding terrain. It uses a laser beam to calculate the distance to an object. The most common principle for the laser range finder is the time of flight. This can be achieved by



sending a laser pulse, and the time required to detect the reflected pulse can be used to determine the distance to the object. Due to the high light speed (approximately  $3 \times 10^8$  m/s), this method is not suitable for high accuracy (sub-millimeter) measurement. The triangulation principle can be used to solve this problem. Triangulation uses a laser source beam and detector. The emission angle of the laser beam and the distance to the detector can be used to determine the distance to an object using the triangular relation. The laser range-finder has high precision and speed of reading even over long distances. On the other hand, it is one of the most expensive sensors available. The laser source can also consume a high level of power and may represent a hazard. In outdoor working environments the laser pulses can reflect from metallic surfaces which are closer to the laser source than the object. Also, the range finder may fail to detect an object if the distance to it is larger than 365 meter and the object is too near to the earth [52].

### ***2.3.1.2 Ultrasonic Sensors***

Ultrasound is a sound wave with higher frequency (40 kHz) than the upper limit of human hearing. The ultrasonic sensor is used to detect an object and to measure the distance to it. Ultrasonic transducers are used to convert DC signals into ultrasound pulses, and *vice versa* after detecting the object. The underwater range finder is a common use of the ultrasonic sensor. If there is an object to detect when the ultrasonic pulses are emitted, the pulses will be reflected back to the transmitter. The difference between the times of transmission and reception can be utilized to calculate the distance to the object (the time of flight principle). Ultrasonic systems provide efficient range detection over short distances and the main advantage of the ultrasonic sensor is its low cost. However, a wide ultrasonic beam may cause inaccurate object position within the beam. Differences in signal spread and speed and imprecise measurement of the arrival times of reflected signals are disadvantages of ultrasonic sensors, which may be associated with low accuracy and resolution and noisy signals [53].

### ***2.3.1.3 Radar Sensors***

Radar sensors use frequency modulated continuous waves (FMCWs) or microwaves for the reliable detection of any objects within their range. Therefore, radar sensors are appropriate for many applications in robotic systems, such as mapping, path planning, and obstacle avoidance.

The time of flight principle is also used in radar sensors to calculate the distance to the object. Radar sensors are efficient for long- and short-range detection and can work in extreme weather conditions such as fog and rain. Compared to other common types of active sensors such as laser range finders, radar waves do not reflect from metal surfaces, and radar sensors



have better accuracy resolution and error handling than ultrasonic sensors. However, the sensors experience problems in detecting multiple objects, since multiple reflected signals can produce blurred data. Another disadvantage of radar sensors is their high cost [54].

#### 2.3.1.4 Encoder

Any transducer which has the ability to create a coded reading of measurements can be called an encoder. Shaft encoders are digital transducers that are utilized to measure speed and angular displacement. In comparison with analog transducers, their digital counterparts show better resolution, reliability and accuracy, and cost less. Shaft encoders can be divided into absolute and incremental types based on their nature and the method used for the analysis of the output. Absolute encoders provide unambiguous positional values at any angle using rings only. Many rings are required to determine position and orientation. Incremental encoders, meanwhile, need to count pulses to maintain absolute position in the surroundings range. An external ring is utilized to calculate rotation, while two internal rings are used to find the rotational direction using binary pulses.

Encoders can use four techniques to generate signals, including optical, sliding contact, magnetic, and proximity sensor methods. The optical encoder is the most popular kind; it uses an opaque disk that has one or more rings, with transparent areas in each ring. Light emitting LEDs with the receiver on the other side of the disk are used to generate binary output, as clarified in Figure 8. The binary output signal is generated by interrupting the emitting light using the opaque areas. The binary pulses may be used to calculate disk speed and acceleration. Optical encoders have a long lifetime but are sensitive to jolts. Magnetic encoders work based on the Hall, or magneto-resistive, effect to generate the binary output signal. The Hall Effect is generated by passing a segmented metal disk among permanent magnetics. Magnetic encoders have an acceptable long lifetime and they are not oversensitive to shocks. Optical and magnetic encoders can both provide a relative resolution [55].

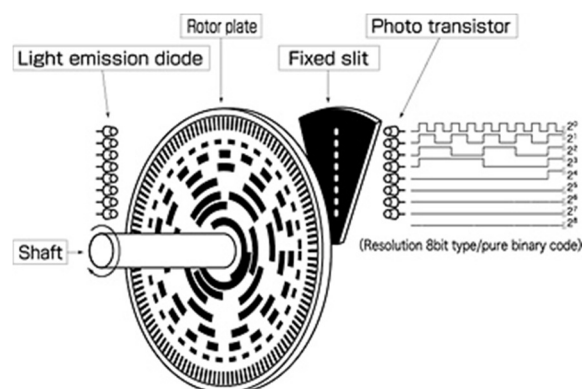


Figure 8 Simplified structure of incremental encoder [56]

### ***2.3.1.5 Potentiometers***

Potentiometers are the basis of passive sensors, performing as variable resistors. Potentiometers are voltage dividers used to measure electrical potential (voltage), and linear and rotary models are commonly used in robotic systems for the calculation of linear and angular displacements.

The cost of potentiometers ranges from a few cents to about 50 US dollars. The main advantage is that their resolution is virtually infinite, typically depending on manufacturer and cost. Resolution also depends on the number of rotations between the end stops in the rotary model. The main disadvantage of the potentiometer is its short lifetime. Potentiometers are utilized in a wide range of applications, in particular in electronic circuit input control, analog signal tuning, and positional measurement [55][57].

### ***2.3.1.6 Tachometers***

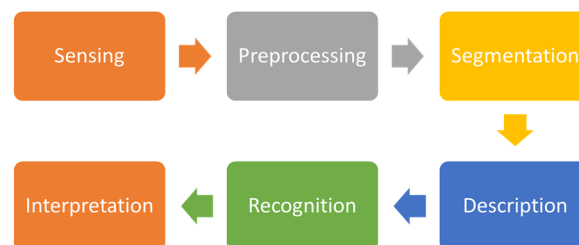
Tachometers are used to determine acceleration and velocity based on measurements of the rotational speed of a coupled shaft. In general, the tachometer works as a small DC generator in reverse mode, and it consists of multiple wound armature coils inside a permanent magnet. The voltage level produced is related to the shaft rotation speed, and passes from the rotary side out through the brushes. An attractive advantage of this sensor is its low cost, but it has a short lifetime due to brush erosion and also a lower response time due to the spikes generated at the commutator at high speed [58][59].

### ***2.3.1.7 Cameras***

Cameras or image sensors provide the highest data sets about the surrounding environment. They work by collecting the light reflected from objects and then directing it to highly light sensitive components. Vision systems may be employed for range detection, object identification, and tracking, and in mobile robots they can be used for aspects of navigation such as mapping, localization, and path planning. Digital cameras usually use the charge-coupled device (CCD) sensor due to its low noise, high resolution, lower consumption of power, and high light sensitivity. The high quality of images captured by CCD sensor cameras results from the special manufacturing process that enables the transmission of charge to the chip without any distortion. Some digital cameras, however, use the complementary metal-oxide semiconductor (CMOS) instead. CMOS sensors are usually more susceptible to noise, and offer lower quality, lower resolution and lower sensitivity, whereas their attractive features include low cost and long battery life. Cameras are usually one of the instruments used in large computer vision systems [60]. Current CMOS cameras already have improved image resolution and overall quality, and CMOS sensors will soon match CCD in these respects while retaining the advantages of long battery life and low cost. In general, the images usually

contain a huge amount of information, some of which is useless and so filtering processes are required to eliminate it [61].

Computer vision can be defined as the ability of a computer to understand the acquired images, including distinguishing objects and detecting range, using models created with the assistance of physics, geometry, and learning theory. In many computer vision applications, the system is pre-programmed to perform a specified task, but more recently learning methods are frequently used instead. Computer vision divides the understanding of the image up into several smaller processes, as shown in Figure 9. A camera is used to sense the environment and capture images. A preprocessing stage is used to eliminate noise and improve image quality. Image plane adjustment, color extraction, and a blurring process are used in this stage. Once preprocessing is completed, the image is segmented into separate regions or objects. Each region is analyzed to find features such as shape, size, and color. The extracted features are then employed to identify the objects. A special identifier is given for each object. The most difficult process is interpretation, which involves a consideration of how the object data are employed, what the relationships between objects are, and what effect they have on robot operation. The capabilities of computer vision to provide a huge amount of data using a single camera sensor at relatively low cost make it an appropriate choice for mobile robot operations [62].



*Figure 9 Computer Vision*

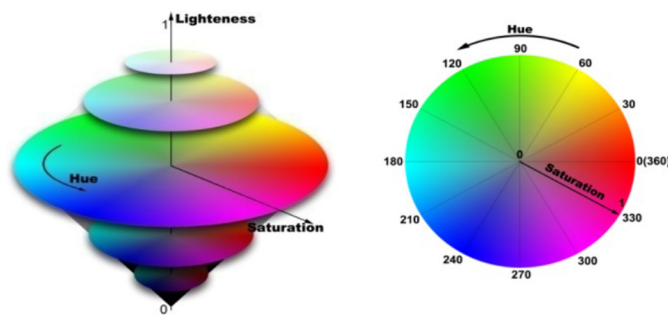
Color representation is one of the most important techniques used in graphical computer vision. In color theory, models mathematically describe how colors may be represented. A color space is one where the components of the color model are precisely defined, allowing viewers to know exactly what each color looks like. The RGB, CYMK, and HSL schemes are the most common approaches to color representation.

RGB is an additive color model where the Red, Green, Blue colors are combined to produce a large range of colors. Its name comes from the three channels R, G, B. These channels store the intensity of each color and the combination of intensities give different colors. The RGB scheme is used in sensing, representing, and displaying images in an electronic display device. This representation is simple and easy to understand. However, the camera gives different

combinations of the RGB channels when the quality of lighting changes, which renders this type of representation unstable in color detection applications.

CYMK is a subtractive color model dealing with how white light is reflected from a colored surface. It consists of the four channels cyan, yellow, magenta, and black. It is used mainly in printing devices due to the minimal cost of color representation compared to other methods.

HSL (hue-saturation-lightness) is one of the most common cylindrical-coordinated representations of points in an RGB color model, as shown in Figure 10. It was developed in the 1970s for computer graphics applications, and is used in color selection, image editing software, image analysis and computer vision [63]–[66]. From Figure 10 it can be seen that each color has a specific value represented by the hue value. The other two components are pixel lightness intensity, called lightness, and the amount of color saturation, termed saturation. From this representation, the hue value has color information which is stable in different lighting conditions.



*Figure 10: HSL color space [63]*

### **2.3.1.8 Discussion**

A sensor can be defined as a device that responds to inputs from the physical environment and then converts it into readable signals. These inputs may be heat, pressure, velocity, acceleration, light, or other measures, and the sensors can be classified according to many categories such as passive and active sensors. Active sensors sense the environment and the robot's internal state by emitting energy signals to sense its reflection while passive sensors merely observe.

Passive sensors consume less energy than active ones since they only need power to amplify their output signals, but they are usually associated with signal and noise problems. Many passive sensors are utilized in robotic systems, such as cameras, tachometers, and potentiometers, each having a specific function. Potentiometers are used to calculate the linear and angular displacements while tachometers determine the mobile robot's

acceleration and velocity, whereas cameras provide a huge amount of information about the robot's environment.

Each sensor has a certain area of coverage, for which it can detect the physical input that it is sensing. For example, laser range-finders and ultrasonic and radar sensors may all be used to determine the range to an object, but each type has advantages and disadvantages that make it more suitable in specific applications. Laser range-finders have high precision and speed performance but are very expensive. Meanwhile ultrasonic sensors are cheap but only work over short distances and have accuracy difficulties due to signal interference, and finally the radar sensor has better accuracy but is expensive. Range finder sensors are active sensors.

### **2.3.2 Mapping**

Maps are crucial components of navigation systems. The autonomous robot should have the ability to create a spatial model of the surrounding environment based on sensor data. Maps can be classified based on the type of sensor information into path, free space, object-oriented, and composite maps. These kinds of maps are different in terms of their data framework and resolution. Special maps are created to cope with specific path planning methods. In spite of this, maps are typically either local or global. Local maps describe a limited area surrounding the robot, while global maps are much wider and their limits may be, for example, buildings with multi-floors. Global maps are used to realize a general path to the destination, while a local map models the surrounding environment with more details. For the mapping, the resolution of the sensor used is an important aspect to be considered. For example, obstacles should lie within the range of their actual positions according to sensor accuracy. Typically, mapping and localization data is gathered by the same sensor, since robot localization is associated with the map built. Some maps depend on data from collective sensors to minimize the robot odometry, which is defined as a method to estimate changes in robot position using the motion data. This section summarizes the main types of mapping, describing the advantages, disadvantages, and limitations of each.

#### **2.3.2.1 Path Map**

Path maps consist of paths or movements. A road map is a collision-free set of paths between a starting position and an ending position. Therefore, these maps describe the connectivity of robot free space on the map [67].

This kind of map is either generated by human programming or learning methods. Maps created by human programs consist of room modules and landmark positions. In the learning method the robot is moved manually and the waypoints and motions are recorded. The path maps are stored internally and used to direct the robot to the destination. Usually path maps

are used in applications with only small variations in the robot's path; for example, in factory environments.

### 2.3.2.2 Free Space Map

Free space maps model the free space between obstacles instead of representing the obstacles themselves. Such maps are created by moving the robot around in unknown or unpredictable environments and collecting data about free areas, obstacles, and unknown areas which are then recorded in a graphic module. Many methods can be used to represent free space. Georg Voronoi introduced a method called the Voronoi diagram. This splits the environment into regions based on equal distance from the edge of the region to the points of neighboring regions (see Figure 11). Each region is classified as busy, empty, or unknown. The diagram is established by calculating the vertical bisector of each link line. The diagram border represents the free path for mobile navigation [68].

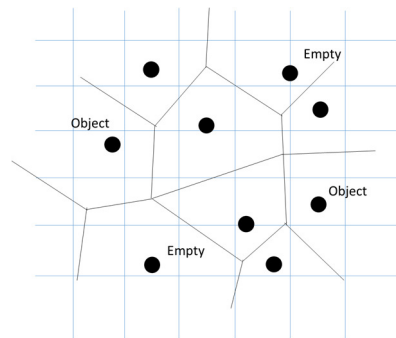


Figure 11 Voronoi Diagram

### 2.3.2.3 Object Oriented Map

This kind of map considers the positions of obstacles in the environment. When the area occupied by obstacles is calculated, the obstacle-free space (free area) is calculated by implication. Two methods are used to represent an obstacle in the map. Usually obstacles are registered as a group of  $x,y$  vertices related to the global reference. Otherwise, a connected list of  $x,y$  vertices and coordinates is used to represent the positions and orientations of obstacles, as shown in Figure 12.

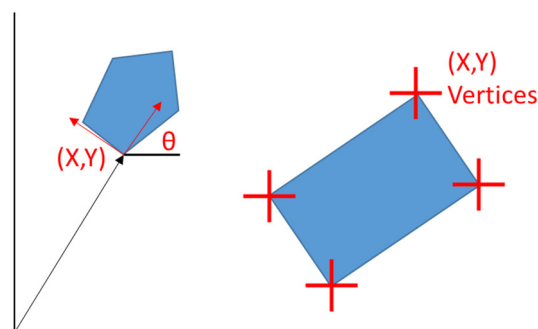


Figure 12 Object Oriented Map

This method is not suitable for unpredictable environments because it provides little data about the working environment whereas accurate sensor data is required to produce a map with higher accuracy.

#### 2.3.2.4 Composite Maps

In contrast to the free space map which ignores obstacles, and the objected-oriented map that considers obstacles, the composite map considers both free space and obstacle areas. The most common approach used to construct a composite map is grid mapping. Grid maps depend on the geometrical structure of the environment. Grid maps can be tessellated into boundary areas or points; each point or cell represents a limited area in the environment, and may contain information related to its state, such as equipped, dangerous, or reachable. Occupancy and certainty grids are the most popular kinds of grid maps. The occupancy grid collects sensor data for each cell, then applies a threshold to the incoming data to specify the cell's state in a binary representation as either occupied or free. The main disadvantage of this method comes from representing a partially occupied cell as fully occupied (see Figure 13). This problem has been tackled by using the same approach to occupancy but without data thresholding, instead using a range of numbers to represent each cell state (see Figure 14). However, such maps still suffer from limited resolution in world representation. To achieve better obstacle representation, quad tree maps may be used. In these, obstacle areas are divided into smaller sections as data is collected so as to increase accuracy (see Figure 15) [69][70][71].

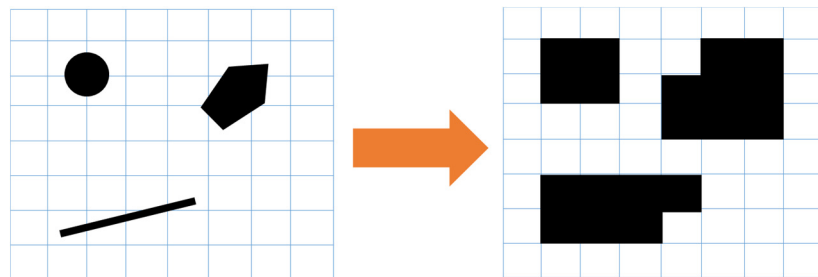


Figure 13 Occupancy Grid Map

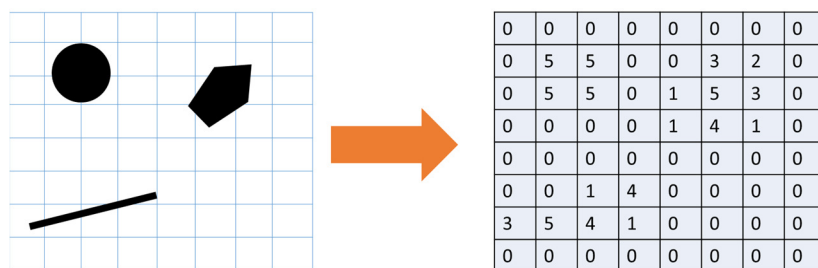


Figure 14 Certainty Grid Map

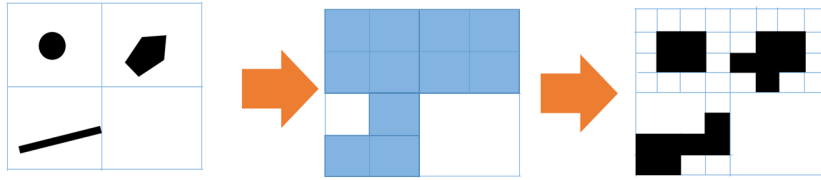


Figure 15 Quad tree Occupancy Grid.

### 2.3.2.5 Discussion

Mapping is a fundamental process for autonomous mobile robots. Different kinds of maps vary in terms of the aspects of the environment which need to be represented in building an appropriate model enabling the robot to perform its tasks.

Each type of map has advantages, disadvantages and limitations. The main advantages of the road map are easy implementation, robustness, and the ability to provide sufficient information for path planning stage and the avoidance of fixed obstacles, but it cannot be applied in unpredictable environments. Object-oriented maps provide little data about the working environment. This method has limited application since it requires accurate sensor data to realize a higher accuracy map. It is also not suitable for unpredictable environments. Grid maps have the advantage of combining measurements from multiple sensors, but it suffers from limited resolution of object profiles.

## 2.3.3 Localization

Robot localization indicates the ability of the robot to establish its own position and orientation in either absolute or relative frames of reference using sensor measurements [72][73]. As the robot moves, the estimation of its position changes. Thus, the robot has to continuously update its position through active calculation processes. It is hard to imagine an autonomous robot without a localization system, since the robot has to know its current position in order to move to the next point. Localization systems can be classified as those suitable for outdoor or indoor mobile robots.

### 2.3.3.1 Dead Reckoning

Dead reckoning is an estimation of position relative to the last known position, elapsed time and estimated robot speed [74]. Generally, this method uses an inertial measurement unit (IMU) or a control variable such as an encoder, and thus does not depend on external signals. Since each dead-reckoned position estimation relies on the previous one, the spatial error in the robot position generally grows with time due to an accumulation of sensor errors which is the main disadvantage of this technique. The error source can be the wheel slippage or the surface irregularities. This error can be reduced using precise sensors and accurate sensor calibration. Due to the cumulative error, the dead reckoning technique is generally unsuitable



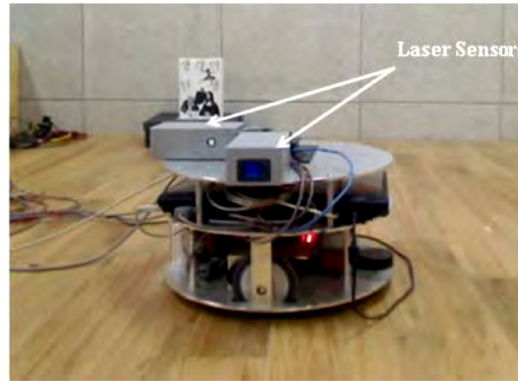
for large environments or when localization is required over a long time period. Even if the noise associated with the sensor used was completely eliminated, even any very small error in initial angle position will cause a reduction in localization accuracy. In spite of this, dead reckoning techniques are popular for the rapid estimation of robot position, especially in complex localization applications [75]–[77]. This technique is suitable for applications involving localization over short time periods. The most popular approach with this technique in mobile robots is odometry. Odometry is the use of signals from actuators on the robot's wheels [78] or legs [79] to evaluate changes in position over time. Usually, incremental rotary sensors (IRCs) are attached to the wheeled shaft axes. These sensors measure the rotation of the robot wheels and then transform this information into robot forward motion and steering speed. In addition to error accumulation problems, however, wheel slipping also affects the accuracy of this method. Odometry can also be used to determine the speed of a robot by integrating measurements from different exteroceptors such as optical mice, radar and sound wave sensors, cameras and gyroscopes [80]–[83].

Many researchers have used multi-sensor systems (sensor fusion) for robot indoor localization to improve resolution and increase reliability, and to realize a higher signal-to-noise ratio. In order to realize an efficient localization system based on sensor fusion, many difficulties need to be overcome. These are related to technical issues such as software and symmetric computational complexity, power consumption and the efficiency of the accumulation of data from multiple sensors. The disadvantages of using sensor fusion in localization systems include high cost and computational complexity as well as positional error which accumulates over time.

**Qingxin et al.** presented a strap-down inertial navigation system to determine the robot's path [84]. They adopted multiple sensors, including gyroscopes, accelerometers, encoders, and compasses, to determine the robot's position and azimuth. Results from simulation showed that the strap-down inertial navigation system could work with low-resolution sensors. The main disadvantage of the system is that it cannot continue at the same performance level while time goes on since the level of error steadily increases.

**Anjum et al.** presented a method for sensor-data-fusion based on an Unscented Kalman Filter (UKF) algorithm to increase the accuracy of localization of the mobile robot [85]. They obtained information on the robot's movement using many sensors, including accelerometers, gyroscopes, and optical encoders. Two different UKF models were utilized in this method to account for slip errors, and this method was validated in terms of its ability to translate the linear motion of the robot through a navigation area of two meters, where the robot moved one meter forward and then moved one meter backwards. The experimental

results showed that the proposed system can localize the robot with a positional error of 10 cm based on the sensor data and of 0.5 cm after applying the UKF algorithm. In the same experimental scenario, the positional error increased if wheel slip occurred to 23.8 cm with sensor data and to 9.6 cm after applying the UKF algorithm. Figure 16 shows the experimental environment.



*Figure 16 Experimental environments for robot position estimation [85]*

More about dead reckoning can be found in [86][87].

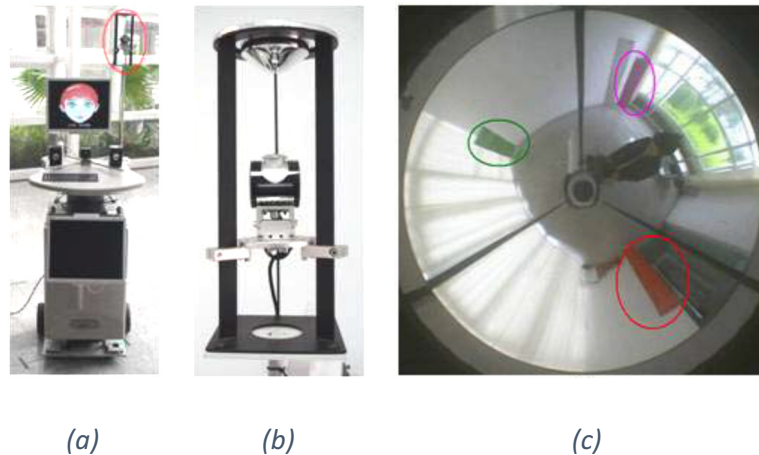
### ***2.3.3.2 Indoor Localization based on Maps***

To overcome the cumulative error associated with the use of dead reckoning techniques, the robot must compare the environment signal readings with a map of the known environment. Map-based localization approaches rely on the representation in the map, and localization methods can be classified based on the procedures used in processing sensor data. In localization-based map approaches, the sensor data is compared with the built map to estimate the robot's position. The robot searches in the global maps for a region which matches the local map built based on the sensor information.

**Li et al.** proposed a concept for an indoor localization system based on an electronic compass and visual data [88]. The electronic compass is installed on the robot's base to obtain the azimuth angle of the robot, and then data from a camera on the robot's head is used to localize the robot inside the map. Two separate memories are required in this method: the first is used to save the camera data, while the second stores the electronic compass data collected during the robot's movements. A virtual robotic system is used to simulate the proposed concept in an ideal environment. The simulated robot was shown to be able to navigate through an experimental virtual environment, but the design of the system would need to take many aspects into consideration to become applicable in real environments. For example, the memory size requirements would be huge, and the data processing modules complex especially with large scale maps. The sensitivity of the compass to noise might be another serious issue.

A disadvantage of this approach is that it is sensitive to changes of the environment. Another approach focuses on detecting distinguishable patterns, called natural landmarks [89], in the sensor data. Landmarks are a special form of features which the robot can recognize using a sensor. Natural landmarks are usually defined in highly structured environments, such as corridors [90], hospitals [91], and factories [92]. The detection pattern is compared with stored pattern in the global map and based on these harmonization, the robot position is calculated based on classical geometry methods [93]. An example of the pattern matching method is shown in computer vision which uses well-distinguished features for robot localization.

**Duan et al.** presented a probabilistic method to estimate the mobile robot's position accurately [94]. They used omnidirectional vision and signal processing. The omnidirectional vision was used to realize the landmarks and bearing information. Then, a probabilistic mapping was constructed composed of the weight of coordinates. A Monte Carlo algorithm, which depends on particle filters, was then used for the robot to merge the probabilistic mapping with an odometer-based prediction model. Figure 17 shows the robot equipped with the Omni-camera.



*Figure 17 Mobile robot position based on Omni-camera and a probabilistic method: (a) Robot equipped with Omni - camera. (b) Omni-camera. (c) Experimental Work Space [94]*

**Lu et al.** presented a mobile robot localization method based on an external ceiling camera and an ultrasonic sensor embedded inside the robot [95]. This method used a fixed camera to gather data from the environment which was then translated into a small region based on the extracted edges. A\* algorithm was used to plan the robot's path to the destination, where during robot movement the ultrasonic sensor senses an obstacle and the current small region is marked as an obstacle on the map. Then the robot continues until it reaches the destination. The proposed system was validated in a small area (limited to ceiling camera view) using a

small robot called Parallax Scribbler 2 to prove its localization ability inside the segmented map. This method is not suitable for use by a large robot or in a large environment area due to the limited camera view. Also it is too expensive and would require the installation of many cameras to cover a large area.

**Hadda et al.** presented a method based on stereo vision to conduct mapping and localization [96]. This method used nodes and arcs to build a hybrid card for mapping. Node construction starts by extracting SIFTS (scale invariant feature transform) points and calculating their depth information. SIFTS is an algorithm used to extract feature vectors which are invariant to image translation, scaling, and rotation. The connection between nodes are called arcs, and arc length is calculated using an odometer. A method based on singular value decomposition was used for global localization in the map created, and experiments showed that the proposed method succeeded in mapping and localization with a positional estimation error of up to 26.5 cm. The main disadvantage of this system is that it is not fully automated since the local position of the mapping cards is given manually and the metric relationship between local maps is not supported by the mapping system together.

**Qian et al.** developed a 3D map building method [97]. This technique was based on a laser range finder and Kinect sensor. The laser range finder was used to build an accurate 2D map, and then the Kinect sensor was utilized to build the 3D map based on an image-matching method. Experimental results showed that the proposed technique gives robust performance in map construction. The limitations of this system are related to the use of the laser scanner, which is very expensive, and the Kinect sensor which exhibits unstable behavior in cases of reflective surfaces, and also the matching method takes a long time to build each scene.

**Sim et al.** utilized natural features with maximum edge density distribution as landmarks, with computer vision for mobile robot indoor localization [98]. A localization method was proposed which takes advantage of the strengths of image- and landmark-based methods. The images were coded to a group of visible features called landmarks, and experiments showed the feasibility and precision of the proposed method, although it failed in conditions of poor lighting.

**Lee et al.** developed a localization method based on a grid map, where the technique used sonar data to evaluate the robot's position [99]. The localization map was developed based on ultrasonic sensor data and an extended Kalman filter (EKF). Experiments were performed with many scenarios to validate the performance of the proposed technique. The results showed that the robot can complete its path to the destination even with wheel slippage or if the robot suddenly stopped, but data on the accuracy of the system and the reproducibility of the results was not obtained. The main limitation of the proposed localization approach,

which was based on landmark detection using a SURF function, is that it requires a long computation time.

Matching and position calculation operations with natural landmarks require a long time and high computational costs to realize accurate localization results, and the estimation of robot position may fail due to sensor noise. Thus artificial landmarks have been proposed to solve these problems. Artificial landmarks can be geometric shapes (circles, rectangles, lines, etc.) [100] or they may be expressed in another form such as a barcode [101]. Artificial landmarks are special, easily distinguishable objects which are placed in the robot's path for it to use in navigation [102].

Artificial landmarks are easier to extend and maintain compared to natural landmarks. There are two types of artificial landmarks: active and passive. An active landmark requires a power source while a passive landmark is made of reflective material and it's easy for expansion since it does not need any power and cables.

**Takahashi et al.** developed a new system for autonomous mobile robots to perform self-localization [103]. This system was developed based on multiple RFID readers and IC tags, as clarified in Figure 18. Experimental results showed that the proposed system was robust in robot localization. Unfortunately, this method is unsuitable for applications in large environments due to the high cost of installing IC tags.

Further examples of robot localization using RFID methods can be found in [104]–[107].

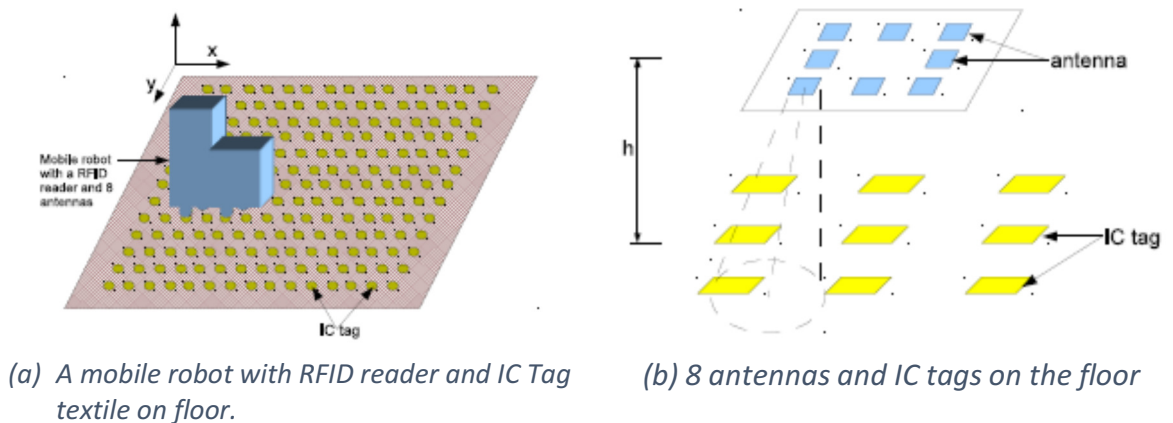


Figure 18: Localization System based on RFID [103].

**McCann et al.** proposed a new technique to improve mobile robot localization based on quick response (QR) code landmarks [108]. Each landmark has information about its 3D position relative to the robot map. The algorithm starts to search for landmarks inside the captured image from a Microsoft Kinect sensor camera, and then utilizes an improved Monte Carlo localization approach to estimate the robot's position based on depth information. Practical

experiments showed the improved performance of the probabilistic localization method in dynamic environments. The developed method gave a positional error of up to 9.9cm and requires an extra desktop station due to high computational complexity.

**Bae et al.** used coded infrared light for active landmarks in mobile robot indoor localization [109]. This method was carried out using IR LEDs with phototransistors. The LEDs were installed on the ceiling while the floor was divided into sectors with each sector having a characteristic ID. The coded infrared light gives information regarding the actual sector of the robot. This method requires huge numbers of IR sources to cover a large area, which make this solution very expensive and unsuitable for large- scale map applications.

**Nakazato et al.** presented a localization method based on invisible landmarks and an IR camera [110]. This method could be easily extended to cover a large area but its low accuracy and high cost would prevent its wide adoption. From experimental results, the values of positional and orientation accuracy were 10 cm and 5 degrees respectively.

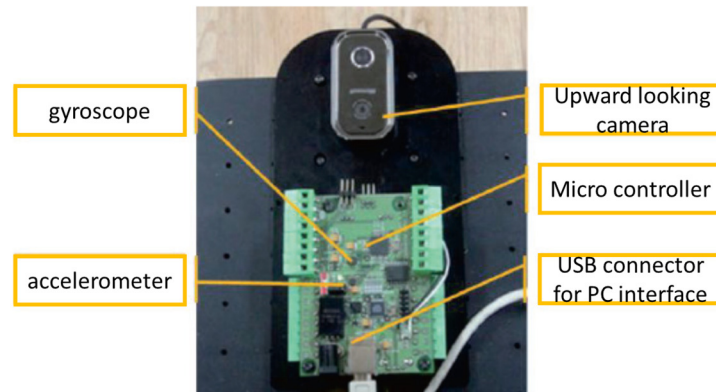
**Brassart et al.** presented an indoor localization method based on active beacons as landmarks [111], where robot position was calculated using a triangulation method in which reduced processing time was utilized, and a CCD camera linked to a coded infrared signal receiver was used for beacon detection. This would be a very expensive localization approach, making the proposed method inappropriate for application in large areas.

More cases of localization using landmarks can be found in [112]–[114].

**Bais et al.** proposed an algorithm for differential driver mobile robots for self-localization in a dynamic environment [115]. They used a combination of sensors which included a stereo camera, two digital encoders, a gyroscope sensor, two 10g accelerometers, and a magnetic compass. The algorithm finds two visual landmarks from the robot's environment, then starts to find the depth and orientation of the landmarks using the stereo vision system. When these landmarks are not detected, position estimation is initiated based on the other sensors and an extended Kalman filter. Simulation results showed that the proposed method could successfully localize the robot with a positional error of up to 6.2 cm as a maximum value for a relatively small area. Using the accelerometer to calculate the displacement distance from the starting point leads to cumulative error over time due to the double integration involved.

**Lee et al.** presented a new localization method for mobile robots based on data from both vision and motion sensors, as shown in Figure 19 [116]. They used a combination of data from a camera, mobile robot optical encoder, accelerometer, and gyroscope to estimate robot position, orientation, and velocity. This method is based on the measurement of differences between wheel encoder and accelerometer data to determine the robot's wheel slip. The

wheel slip and an extended Kalman filter are then utilized to specify the accurate position and estimate velocity. The proposed method was tested with different motion scenarios, and the results showed that the proposed system can localize the robot even if one of the four sensors used is inactive, but the authors did not assess the accuracy of the proposed system or the reproducibility of the results.



*Figure 19: Sensor module and camera[116]*

**Lee et al.** utilized a laser range finder and two dead reckoning sensors along with passive artificial landmarks to build a mapping cart [117]. The mapping cart was used to reduce human effort when building a large-scale map, and the method was to be applied in indoor T-city environments. The results showed that this method is convenient and efficient in building a large-scale map. For multiple robot this process needs to be repeated for each robot.

### **2.3.3.3 Discussion**

On the basis of a comparison of the various methods for indoor localization, the following conclusions can be drawn:

- 1) Most of the methods described have proven ability for indoor mobile robot localization in limited areas, but many difficulties would be encountered if they were employed in large areas.
- 2) Each indoor localization method has advantages, disadvantages, and limitations. For example, dead reckoning methods have the advantage of being simple and cheap and require a relatively short time for robot indoor localization. However, positional error will accumulate over time, and thus they are unsuitable. RFID reader and IC tag methods are robust but unsuitable for large environments due to the expensive installation of IC tags. Image vision methods give the robot accurate information about its environment, but fail to work properly when light levels are low and in certain complex situations. In addition, the required time is not satisfying. Methods using

multiple sensors may be efficient and stable, but the sensors could affect each other if employed in large areas.

- 3) In comparison with other existing indoor localization techniques, methods using artificial landmarks are not sensitive to lighting conditions, are relatively easy to install and maintain, and can cover large areas. Artificial landmarks have the advantage compared with natural landmarks of allowing a flexible and robust navigation system to be built. Passive landmarks are preferred over active landmarks due to their low cost, as they do not require a power supply, are easy to install and maintain (with no wires required) and have the ability to cover a large area.

### 2.3.4 Path Planning

Path planning is an important task that should be considered in designing an efficient navigation system [118][119]. Path planning can be defined as the control of motion from start point A to destination point B, avoiding collision and taking the shortest path to the destination. Many algorithms have been proposed and developed for this purpose.

Star (A\*) is a graphics algorithm that has been applied to find the optimal path to travel from a starting point to a destination. It is considered to be a best first search (BFS) algorithm. With this algorithm, cost has two functions: the real cost  $g(x)$  represents the cost to move from the starting node to node  $x$ ; and an approximate cost  $h(x)$  function calculates the cost from the current vertices to the goal vertices, which is a heuristic function:

$$F(x) = g(x) + h(x) \quad (1)$$

Many researchers have used this algorithm and modified it to be more applicable as follows.

**Yu et al.** presented a new path planning method based on a modified A\* algorithm [120]. The proposed method was used to find the safest and fastest path by adding a coefficient to the traditional A\* algorithm to avoid moving near to obstacles. The results showed that this method is more efficient than using the traditional A\* algorithm. It only works for single-node destinations, and cannot be applied in life sciences laboratory automation with multiple destinations.

**Goyal et al.** presented a new logical technique for path planning to solve the problem of the A\* algorithm bug when the mobile robot is larger than the map cell [121]. This problem causes the robot to collide with the corners of obstacles. The proposed method is based on enlarging the obstacle virtually by a defined factor before applying the A\* search algorithm. Simulation results showed that the robot avoids the obstacle safely but this method does not realize the shortest path to the destination, as clarified in Figure 20.



**Guo et al.** developed an improved D\* (dynamic A\*) algorithm for mobile robot path planning in unknown or dynamic environments [122]. They used the Mobot simulator environment and a WirobotX80 mobile robot to validate the algorithm. In spite of the success of this algorithm, it cannot escape from the restrictions of limited resolution in search directions ( $\pi/8$ ).

Dijkstra is a special case of the A\* algorithm where the heuristic function is equal to zero. This method is applied to plan the path from a single node to all other nodes. The computational cost of using this algorithm is considered to be a real cost ( $g(x)$ ) which represents the cost of motion from the starting node to node  $x$  [123]:

$$F(x) = g(x) \quad (2)$$

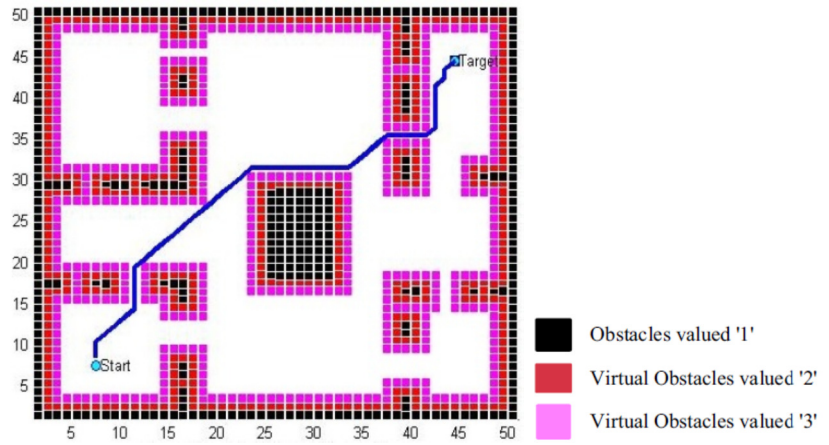


Figure 20: Modified A\* path planning method when the robot is 2/3 larger than the map cell [121]

**Park et al.** proposed a practical method for mobile robot path planning in an indoor environment, called the divide and search approach (DNC) [124]. The method gradually generates a hierarchical road-map that has a multi-level structure. The path is divided into small sections, and the shortest path algorithm (Dijkstra) is applied for each section separately until the destination point is reached, in order to minimize the amount of computation and time required for path planning. Experiments were conducted in an indoor environment to evaluate the performance of the proposed method. The results showed that it improves the time required and reduces computational complexity in comparison with unsegment maps. However, the proposed method plans the shortest path from a single node to destination nodes, which makes it unsuitable for multi-source applications.

The **Floyd algorithm** is a search inside graph algorithm used to generate the path between every pair of nodes, where the path chosen is enhanced until the optimal path is achieved. This method starts by initiating two matrices, one for path length and the other for path sequence, and iterating these matrices until the optimal path for all vertices is reached. The

time complexity for the Floyd algorithm is  $O(n^3)$ , where  $n$  is the number of vertices [123], [125].

Several studies have combined the latter two algorithms to minimize the computation time required.

**Liu et al.** presented a hybrid Floyd-genetic algorithm for a path planning in life science automation building [126]. Floyd algorithm was being used to plan the path in a small scale area while genetic algorithm was being applied for large scale area path planning. The proposed solution is unsuitable to be applied in multi-floor transportation environment, since: a) the genetic algorithm requires long time and for complex computational processes to find the accurately value of the fitness function and other various algorithm parameters [127]. b) The presented Floyd manner can't deal with unsorted waypoints to plan the large scale area. c) The hybrid method was proposed for single goal which make it unsuitable for the transportation tasks.

Other methods, including the ant colony system and fast marching method, have also been applied.

**Chia et al.** proposed robot path planning based on the ant colony system [128]. This is used to find the shortest path using the facility of reducing the procedure of ant colony searching food method. Simulation results showed that this method is efficient when there are obstacles in the path of the robot, but further development is needed for it to find the optimal obstacle space free path. The disadvantages of the proposed system include the time taken and level of computation needed (for example, 1000 iterations were required to find an accurate solution between 39 grid cells).

**Arismendi et al.** proposed an Any-time Motion Planning (AFM) approach for unorganized environments based on a fast marching method (FMM) [129]. A laser range-finder was used to provide real time information about the surroundings. This method applied two procedures. Firstly, a global path was planned which was then divided into sub-paths equal in distance, and secondly the robot moved using a global path from one node to another and, when there was an obstacle to movement, a new path was generated to reach the next node without affecting the global path. Simulations showed that this method minimizes the complexity of the process and reduces the required calculation time. Figure 21 clarifies this method.

More about path planning can be found in [130]–[134].

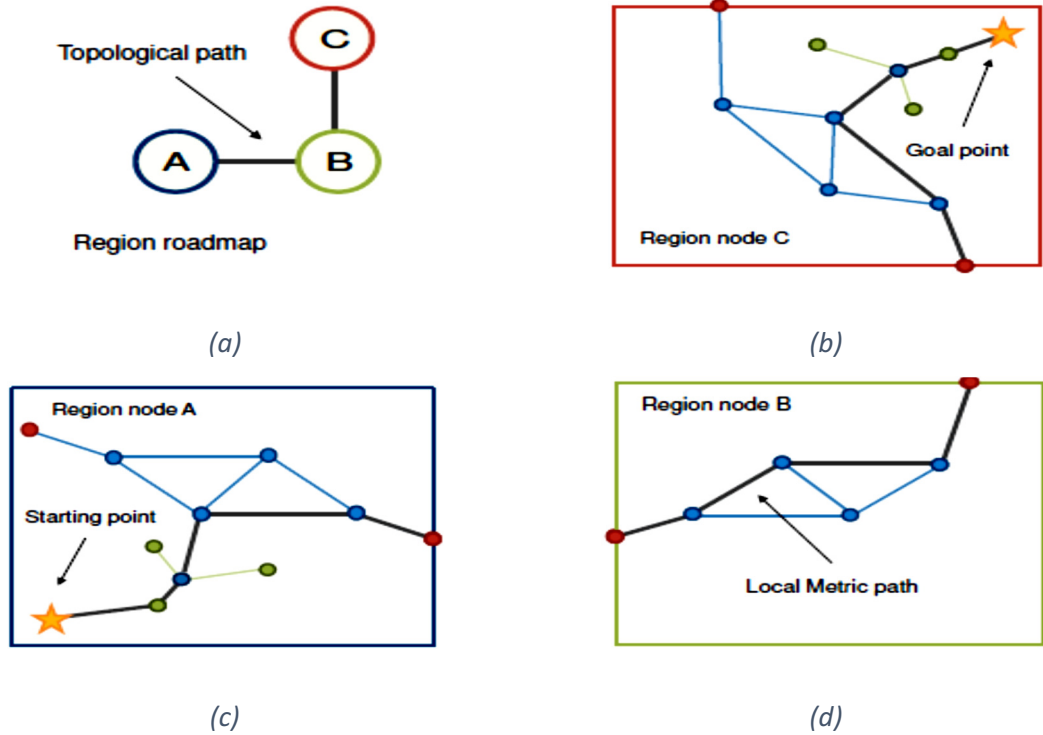


Figure 21: DNC path planning concept: (a) topological map from A to C; (b) nodes in destination region C; (c) nodes in intermediate region B; (d) nodes in start region A [129]

#### 2.3.4.1 Discussion

- a) Path planning has a critical role in mobile robot transportation, which can be defined as a technology used to search for a convenient path to reach the destination safely. The planning path usually needs to satisfy restraints such as collision avoidance, and the estimation of the shortest path.
- b) Two methods of path planning can be used in the laboratory environment. The first creates a number of fixed paths in applications with only small variations in the robot's paths. The second method depends on real-time data to plan a dynamic path. The latter approach usually uses intelligent algorithms for obstacles avoidance and to plan the shortest path. Many smart algorithms can be used to find the shortest path, including genetic algorithms, the ant colony, or the A\*, Floyd, and Dijkstra algorithms, and each of these algorithms has advantages, and disadvantages. For example, genetic algorithms provide robust performance but require a long time [127] and complex computational processes to find the most accurate value of the fitness function and various other algorithm parameters. The ant colony is efficient in planning the shortest obstacle-free paths and requires less time than a genetic algorithm [127], but still requires complex processing.

- c) The A\* is faster than other algorithms. This algorithm is used for path planning in known environments since it requires a heuristic function defined as the cost of movement from the current node to the target node. The A\* algorithm plans the path between a single starting point and single destination; thus, it cannot be applied for transportation tasks in life sciences laboratories which have multiple goals.
- d) The Dijkstra algorithm generates the shortest path from a single source to the destination node. This algorithm is a special case of the A\* algorithm when the heuristic function is equal to zero so the cost function represents only the real cost. For automated laboratories with multiple sources and destinations, the Dijkstra algorithm should be applied for each source node, which requires a long time compared with the Floyd algorithm.
- e) The Floyd algorithm is one of the most important graph searching algorithms. It is utilized to find the shortest path or distance between each pair of nodes, which makes it suitable for multiple source and destination nodes. The efficiency and simplicity of the Floyd algorithm mean that it has become very popular in engineering applications.

### 2.3.5 Elevator Handling

The mobile robot must either climb stairs or ride an elevator to move between different floors [135][136]. Legged robots and special designs of wheeled mobile robots can climb stairs. However, climbing stairs during transportation tasks adds further challenges to robot movement related to load capacity and balance preservation. An elevator handling system (EHS) includes many operations which must be accomplished in order to travel successfully on an elevator. The elevator entrance door and its open or closed status need to be recognized, the entry button must be detected, the destination floor button distinguished, and the current floor read.

**Delmerico et al.** proposed a stairway localizing method as an object on a map for a multi-floor system [137]. From depth information in images, geometric cues are leveraged to detect a stair, and then the relating position and parameter are extracted. The system can determine stair height, width, depth and pitch. Figure 22 shows the stair detection process in both outdoor and indoor environments, with mean error of 1.7 cm for height, 1.2cm for depth, and 17.3cm for width.

**He et al.** designed an autonomous mobile robot which has the ability to climb stairs while still providing a stable movement on a flat ground due to its special wheel design [138]. Experimental results proved its ability to climb 12 steps with a rise height of 160mm and width

of 245mm (see Figure 23). The limitations of the presented technique include the special wheeled design and the mobile robot's inability to transport liquids.

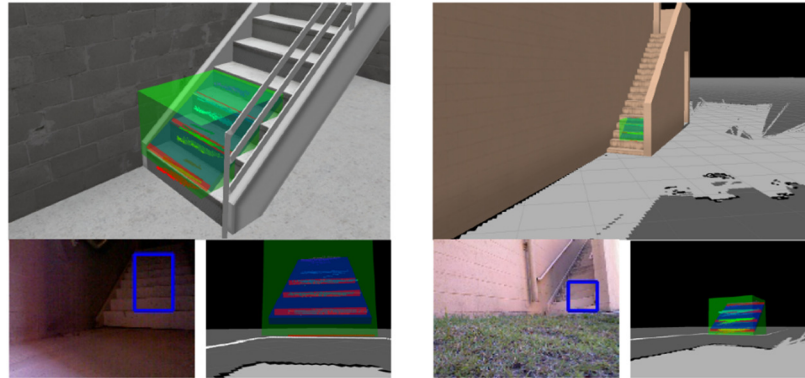


Figure 22: Stairway detection by mobile robot in indoor and outdoor environments [137]

**Beak et al.** used stereo vision in mobile robot navigation to propose an elevator entrance door detection algorithm [139]. This method includes three parts: image capture and filtering; the implementation of an algorithm to extract the elevator door; and the detection of the size of the elevator. In this algorithm, the elevator door detection method depends on both the camera angle and the distance to the door, which have to be more than 50 degrees and 8m away from the door respectively for the door to be detected, as shown in Figure 24. The latter two factors represent disadvantages of the method, and also the matching method requires a higher calculation time and it does not provide useful information about door position.

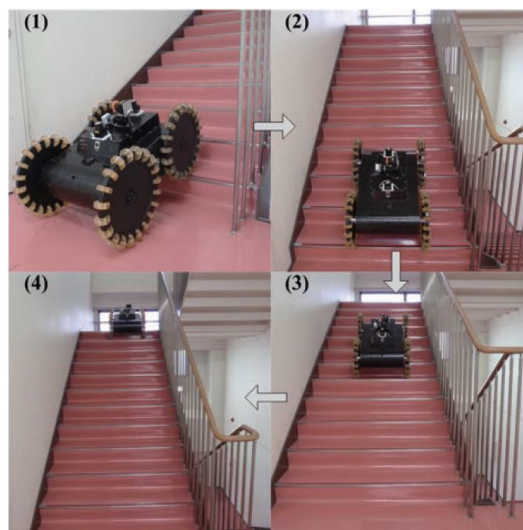


Figure 23: Mobile robot climbing stairs [138]

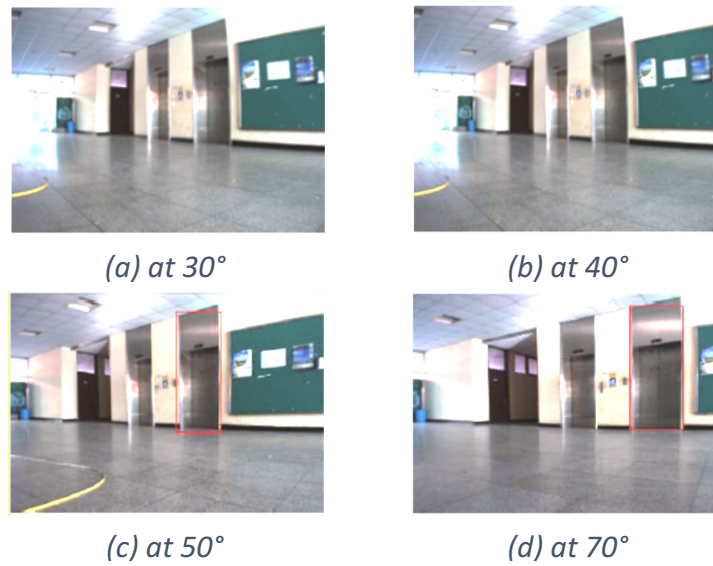


Figure 24: Elevator Door Detection with angle relations [139]

**Kang et al.** proposed a concept for elevator button detection using a Pioneer DX2 mobile robot [140]. A threshold method was used to transform a processed image into a binary one. For the detection of entry and floor number buttons, an artificial neural network was used to determine the best matching by discarding weak candidates. However, the performance of the proposed method was not validated. Its main disadvantages are that it can only recognize elevator buttons but does not deal with button pressing issues. Thus the robot could not work autonomously.

**Xinguo Yu et al.** applied computer vision for elevator button detection and recognition [141]. The detection approach was divided into button panel (BP) detection and button recognition. A method called multiple partial models was used to enhance BP detection, and combined structural inference, the Hough transform, and multi-symbol recognition were employed for button recognition. Experiments showed a detection success rate of 72.6% for external buttons and 85.4% for internal buttons. Here, it should be noted that it is impossible to detect a button when the detection of the BP has failed. Also, the proposed detection method does not provide real coordinates, which are significant in guiding the robot's arm in button pressing operations.

**Liu et al.** proposed an edge-based text region extraction algorithm [142]. The technique can detect text with respect to different font conditions, such as size, style, and color. Figure 25 shows the performance of the proposed algorithm from experiments performed on a fixed set of images. The results show that the precision rate of this method reached 91.8 % and the false positive error rate was 5 %. The disadvantages of the proposed method are that it was

not implemented with a real-time video stream to show its performance in a mobile robotic system. Also, it detects text without achieving character recognition, and finally it does not estimate the text position inside the environment.



*Figure 25 Results for image containing a single character and strong reflections: (a) original image. (b) extracted text [142]*

**Kurogi et al.** proposed a new method for elevator entry button detection [143]. This method uses a single camera with a laser scanner and 2D landmarks placed over the button. This method costs less than stereo vision but errors appear when the angle is greater than 0 degrees.

**Klingbeil et al.** applied a method based on a single camera and laser range finder to realize autonomously open doors [144]. The vision sensor is used to identify the knob or the locations of small buttons, while the laser range-finder is utilized to estimate the distance to the door. The mobile robot must be in front of the door at an angle of  $\pm 20$  degrees. The robot arm is then used to open the door or press elevator buttons. Experimental results showed that the proposed approach was successful in opening 31 of 34 doors. The three failures occurred when dealing with poor lighting conditions and glassy doors.

**D. Troniak et al.** proposed a multi-floor mobile robot transportation system that integrates stereo and monocular vision and a laser range-finder to sense the environment [145]. A map is built based on an SLAM algorithm, which is then used to find a path to the destination. A template-matching algorithm is used to find the elevator buttons. The accuracy of elevator button detection from the outside reached 85%, while it was 65% from the inside. This system requires a long time (4.6 s) to detect the elevator entrance button. Also the robot needs a long time to actually enter the elevator, which may lead to the elevator being missed, and thus human assistance is still required. Figure 26 shows the PR2 robot near the elevator.





*Figure 26: PR2 Robot [145]*

**Chen et al.** presented a mobile service robot which has the ability to plan its own path as well as the ability to call the elevator at each floor [146]. Wireless control was designed to manage the operation to call the elevator. An Xbee module is used to control the servo motor to pushing the button to call the elevator. An RFID is used to estimate the current floor when inside the elevator. Experimental results demonstrated that the proposed mobile robot can take the elevator and complete delivery tasks. This technique has many limitations, however, which would prevent its utilization in life sciences laboratories. Firstly, each floor needs an electronic circuit consisting of an Xbee module, servomotors, microcontroller and power supply positioned beside the elevator for the call button pressing operation. Also, an RFID antenna would need to be installed on each floor for floor estimation, and finally this technique cannot handle destination floor calling.

**Maneerat et al.** presented a simple floor estimation method [147]. This method is based on received signal strength (RSS) which is gained from the number of wireless sensor network sources installed in each floor. This method is very expensive since it requires at least four wireless network sources on each floor to realize an acceptable success rate.

**Chen et al.** presented a height measurement system based on the MS5534B pressure sensor [148]. This sensor was attached to an LPC1114 microcontroller with the necessary peripherals attached to build the hardware platform. An experiment was conducted in a hotel covering 37 floors to examine the stability of the proposed system. However, this system does not take into consideration the wide variation in pressure sensor readings for one location at different times.

More about elevator detection can be found in [149]–[152].



### 2.3.6 Discussion

- a) Unless the mobile robot has the ability to climb stairs, an EHS is an essential aspect of moving from one floor to another in a multi-floor transportation system. However, even if the mobile robot can climb stairs, there still challenges to be faced during delivery tasks related to load capacity and balance preservation. The EHS must include many operations such as the recognition of the elevator entrance door and its open or closed status, the detection of entry and destination floor buttons, and the estimation of the current floor.
- b) Many methods have been used to handle elevator operation. Each has achieved some degree of success, albeit with limitations. For example, a template matching technique has been utilized to detect elevator buttons which achieved a success rate up to 85% for the entry button and 63% for floor buttons inside the elevator. The disadvantage of this method is the excessive time required for button detection (4.5s and 4.3s respectively). An artificial neural network has also been proposed to achieve the best matching by discarding weak candidates for entry and internal buttons detection. However, this method has not been validated in a real elevator environment. A multiple symbol method has been adopted for both external and internal button panel detection, while a combination of image processing techniques was utilized to recognize external and internal elevator buttons. This method achieved satisfactory performance for button recognition but did not provide real coordinates, which are significant for guiding the robot's arm in button pressing operations.
- c) For current floor estimation, many techniques can be utilized, such as an RFID, the received signal strength (RSS) of a wireless network, and a height measurement system. Each approach can realize floor estimation but with some disadvantages and limitations. For example, the RFID technique suffers from high costs since RFID antennae should be installed on each floor, the floor estimation based on RSS has the same disadvantage of being very expensive since it would require at least 4 wireless network sources on each floor to achieve an acceptable success rate. Finally, the height measurement system depends on pressure sensors for floor estimation. This method is low-cost, requiring only one sensor for all floors, but would require frequent recalibration to overcome the problem of wide daily variations in atmospheric pressure.



## Chapter 3      GENERAL CONCEPT

In recent years, researchers at the Centre for Life Sciences Automation in Rostock, Germany, have developed a Hierarchical Workflow Management System (HWMS) to manage the entire process for establishing a fully automated laboratory [153]. The HWMS has a middle control layer divided into a Transportation and Assistance Control System (TACS), including the Robot Remote Centre (RRC) [154], and a Process Control Adapter System (PCAS) which is responsible for managing the laboratory's automated systems. The RRC is the layer between Robot Board Computer (RBC) and the TACS.

Hui, et al presented a single floor transportation system based on the H20 mobile robot [46]. In this system, the RRC and the RBC were developed. The RRC is a GUI developed to manage transportation tasks as follows. The transportation task is received, and is then forwarded with the planned path to the appropriate robot which has the highest battery charge value, and the transportation results are reported back to the TACS. Two hybrid methods are proposed for path planning from single source to single destination points. These methods can plan a path on a single floor but are unsuitable for use with transportation tasks with multiple sources and multiple destinations. The transportation task is initialized at the RBC level, and thus mapping and localization are implemented at this level. The H20 mobile robot is equipped with a StarGazer Module (SGM) working in mapping mode for mapping and localization. The SGM is a stand-alone sensor capable of easily building one connected map. For localization, passive landmarks are utilized with the SGM to realize a low-cost solution capable of use in laboratories of any size. The RBC of this system is adapted for a single-floor transportation system. Thus, an intelligent multi-floor navigation system is presented at the RBC level to handle a complex building structure with laboratories distributed on different floors. The single floor navigation strategy with mapping, localization, and path planning along with an internal door control management system cannot be utilized in a transportation system covering more than one floor, for the following reasons. The single-floor navigation system was applied using SGM in mapping mode, which stores the complete map inside the SGM and this module gives the position in the building map to the robot. Then, depending on the transportation task, the robot moves along the path generated. This method works for a single floor but the process takes a long time due to individual map building and path generation for each robot. Additionally, the SGM cannot deal with multi-floor environments in mapping mode, for two reasons. The SGM can build only one map at a time, and also it cannot build a map inside the elevator due to height limitations since it cannot read a position correctly when the distance from the landmark to the SGM is less than 1.1m. Also, the

execution of the planned path completely depends on the DrRobot library which cannot manage incoming error from the SGM during path execution. For internal door control in the single-floor transportation system, assist points in the planned path are marked with unique numbers for open and closed status. However, this technique is not suitable for a multiple-floor transportation system since bi-directional paths will be used to navigate from multiple sources to multiple destinations.

In a mobile robot multi-floor transportation system, the RBC is developed to realize the functions of a multiple floor navigation system with a Robot Arm Kinematic Module (RAKM), Elevator Handler System (EHS), and Collision Avoidance System (CAS). In this dissertation a new multi-floor system is developed which includes mapping, indoor localization, path planning, an Internal Management Automated Door Controlling System (IMADCS), communication system, Internal Battery Charging Management System (IBCMS), and an Elevator Handler System (EHS) as well as controlling the working process of the relevant systems including collision avoidance and the RAKM, as clarified in Figure 27. The CAS sends a permission request to the MFS after an obstacle detection. In some complex locations e.g. elevator or narrow area the CAS cannot properly work due to gesture detection distance limitation of the Kinect sensor thus, the MFS ignores the permission request in these locations. If the MFS responses to the CAS permission request, the MFS sends the robot position with the current waypoint and postpones the transportation task, then executes a series of the CAS requested movements till avoid the human. For grasping/placing/pressing operation, the MFS guides the robot to the destination position with high precision and sends a grasp, place, or press order to the RAKM with a specific information related to the target specifications, and wait the process success message.

The MFS integrates with the EHS to interact with the elevator environment and to realize the necessary information regarding to: button positions, current floor number, and elevator door status. The IMADCS developed to provide the MFS the ability to control the automated door in the robot path during transportation tasks while the Robot motion center provides the way to control the robot hardware by getting the robot sensors readings and execute the required movement. The client-server connection architecture module (asynchronous socket) is enabled to control the interaction of the MFS with these sub-system over Ethernet. A TCP/IP command protocol based server client structure is used to guarantee the reliability and the expandability. So any kind of devices can be added into the communication network conveniently with a new IP.

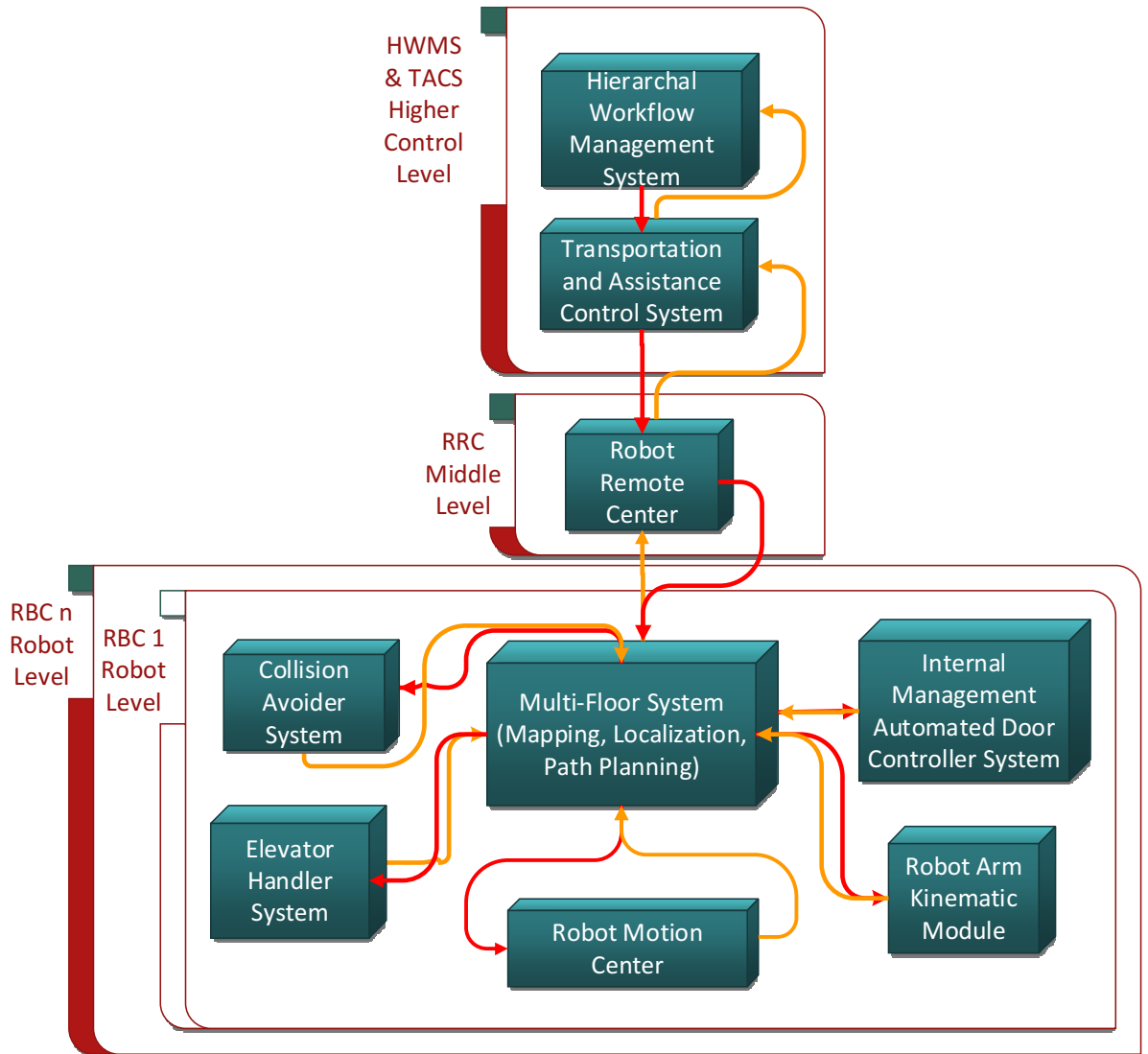


Figure 27: Complete structure of the multi-floor system

### 3.1 Multi-Floor Transportation System Control Strategy

The basic structure of a multi-floor transportation system, which is based on the H20 robot's capabilities and specified transportation requirements, is shown in Figure 28. From the figure below, the mobile robot is responsible for transporting labwares from one island in the laboratory (floor A) into other laboratory islands on different floors. The robot uses ceiling landmarks to gain position information and the elevator as the connection between the floors to reach the destination in the transportation task. The elevator handler strategy is clarified in Figure 29. Communication with the higher levels of the laboratory management system is used to send and receive information.

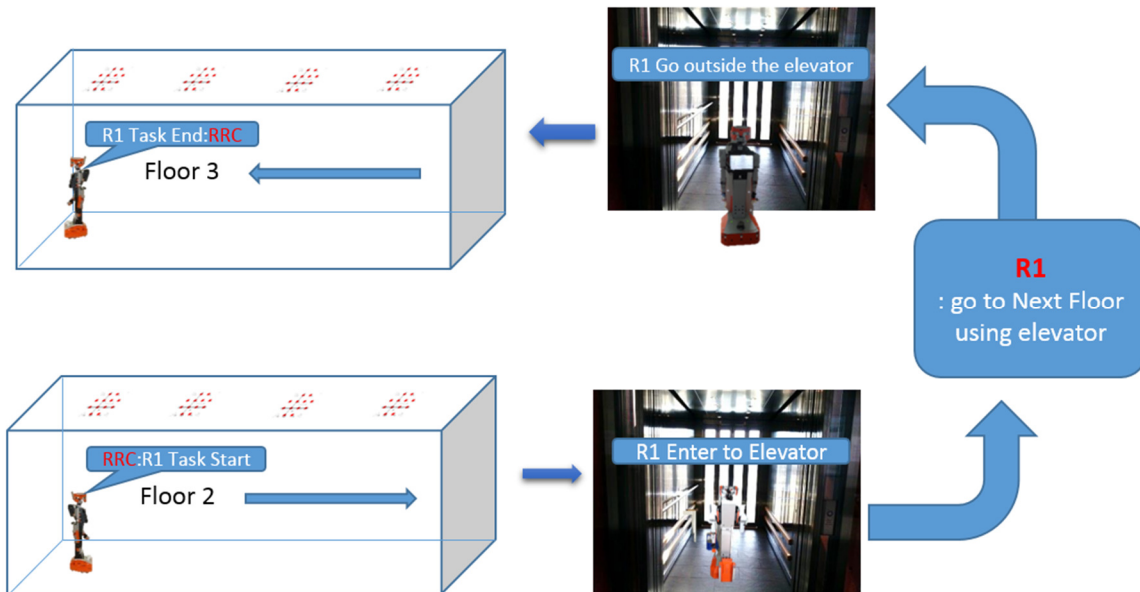


Figure 28 Multi-floor system basic structure

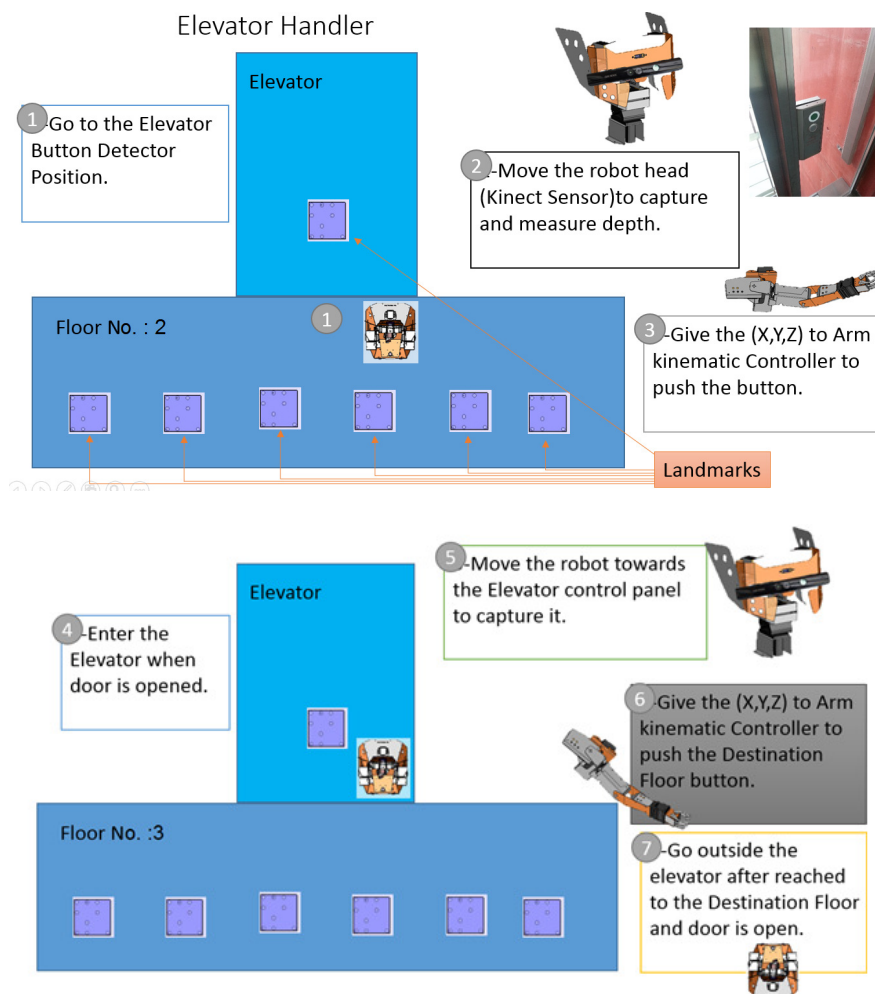


Figure 29 Elevator handler strategy

### 3.2 Multi-Floor Transportation System Hierarchy

Figure 30 presents the hierarchy of the multi-floor transportation system. From this figure, it can be seen that the multi-floor transportation system based on mobile robots has two main aspects: multi-floor navigation and elevator handling.

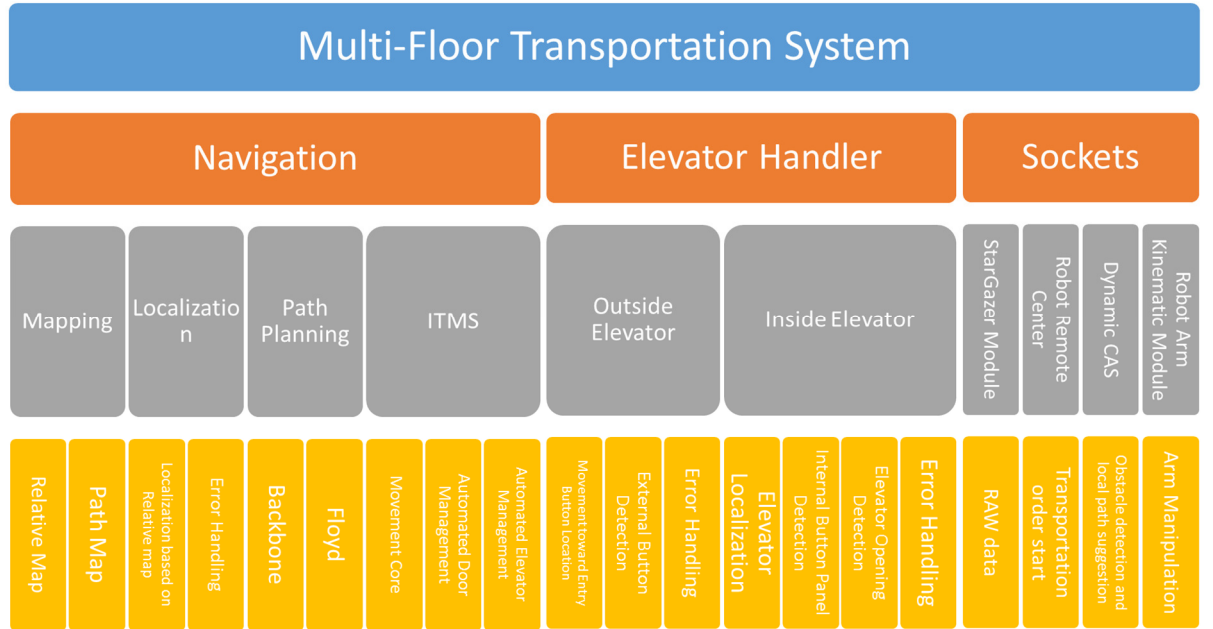


Figure 30 Multi-floor transportation system hierarchy

Navigation consist mainly of mapping, localization and path planning. Mapping and localization are based on ceiling landmarks and a SGM is employed in the multi-floor navigation system. In comparison to other exiting mapping and localization techniques, the use of artificial landmarks is not affected by differences in lighting conditions. They are also easy to install and maintain, and can cover large areas. The SGM can recognize 4096 landmarks, each landmark can localize 1.6-6.5 m in diameter based on the ceiling height, therefore it can cover  $4096 \times \text{localization range}$ . Passive landmarks are selected rather than active landmarks since the former do not consume power and easier in expansion since no wires are needed. The SGM is a low-cost localization sensor for large indoor environments which is accurate, robust, reliable, and less affected by ambient light [155]. The multi-floor environment adds more challenges for map-building since the map must represent positions as X,Y co-ordinates and floor number. In the mapping method, the SGM in 'alone' working mode is used as a HEX reader since it does not have the ability to build more than one floor map inside it. The landmark ID is utilized to define the current floor, and the information extracted with the IDs is used to build the relative map. Two kinds of mapping are employed to produce a relative map (metric map) which is used as a global map in the multi-floor environment with a unique reference point. Meanwhile the path map is used to realize an

obstacle-free set of paths between a starting position and the destination position. The path map relies on the relative map to specify a waypoint position inside it.

A localization method based on the relative map is used to find the mobile robot's position inside the multi-floor environment. This is necessary to allow the robot to decide its next position and plan a path to the specified goals during transportation. An error handler management system is utilized to deal with incorrect landmark readings, since the SGM may accumulate many errors while the robot is moving in a complicated environment which includes a lot of reflected and direct sunlight. The system detects a wrong ID reading if the SGM reads a non-stored landmark ID, or if the calculated distance to reach the next position is larger than the normal distance.

A path planning approach is developed to plan the robot's path to the destination. A new static path method with dynamic goal selection is designed to realize obstacle-free paths which direct the robot to the required goal. This method optimizes the planning speed as well as the number of paths used to reach the destination. However, the developed method cannot deal with unexpected dynamic obstacles. Thus, another path planning method is developed using a Floyd searching algorithm, due to its efficiency and simplicity, to dynamically plan the path from any point to an intermediate destination. The Floyd method is implemented when the dynamic obstacle avoidance integrated with the multi-floor system or if static paths become unavailable for any other reasons. A smart management system is created to select between these two methods so as to achieve high speed and flexible path generation.

Other systems required are automated door management to control the automatic doors on more than one floor. Multiple sockets connection to connect to distributed doors in multi-floor will implemented with the reconnection ability. This system will receive the door number and translated it into a specific door server and pin number. This system will serve the navigation system by adding the ability to open or close a specific door during robot transportation.

A charge management system is also developed to control the robot charging station safely and smoothly. The system can deal with multiple robots, and multiple charging stations are installed in different locations with a specific ID for each robot, where each robot has the ability to recognize its charge status. This system will continuously monitor battery parameters of voltage and temperature and robot position on entering and leaving the charge station.



The MFS has a number of system APIs to connect the RAKM, the EHS, the collision avoidance controller, and the automated door controller in order to utilize the functions from these sub-unit to carry out transportation orders. In general, the MFS controls the execution of the following mobile robot operations during the multi-floor transportation process. The EHS allows the robot to detect the elevator environment by recognizing the elevator buttons and reading the real-time floor numbers. The collision avoidance system detects obstacles in the robot's path and achieves collision avoidance using human-robot interactions, and the RAKM completes grasping, placing and button-pressing manipulations. The IMADCS controls the opening and closing of all of the laboratory doors during the robot's movement, and the robot motion control system provides the robot with different kinds of motion.

To enable transportation between laboratories on different floors the robot could either use the elevator or climb the stairs. Since the robot employed for the transportation task is a wheeled mobile robot, it must use the elevator due to its inability to climb stairs. Using stairs has many challenges even for legged mobile robots fulfilling transportation tasks due to load capacity and balance observation. The elevator handler has many essential issues to consider, such as entry and destination button recognition, and recognizing the current floor. At CELISCA, a glassy elevator is used to connect different floors. This adds more challenges in elevator handling operations due to the effect of direct sunlight on the sensor employed. For entry button detection, a preprocessing stage has been added to combat the effect of sunlight. An image segmentation and blob analysis strategy is utilized to extract the button's position. This method has higher speed compare to the SURF feature extraction detection method. The basis of the method utilized for entry button detection is to remove the background and keep the required color range, and thus color filters should be used. HSL color filtering is chosen rather than RGB filters due to its stability in different light conditions. Since the button detection method is applied in a real application, the real coordinates of the detected buttons should be extracted for the pressing operation. Thus, an RGB-D sensor is utilized for mapping between the image coordinate and the depth sensor to extract the real-world coordinates. Internal button detection uses the same concept as for entry button detection, where the label for each button should be distinguished to select the destination floor button. Optical character recognition (OCR) is used to solve this problem rather than the template matching technique due to its speed and stability under different light conditions. After detection of the destination floor button, the real-world coordinates should also be extracted for the pressing operation. The current floor reading technique uses OCR with different constraints (such as a larger font) and without depth translation. In addition to the main issues associated with elevator handling, there are other aspects that the system must deal with to successfully interact with the elevator. The door status must be detected as either open or closed. The

detection method used may be simple and involve a range-finder sensor. An ultrasonic range finder sensor is preferred to use rather than other common laser or radar types since it works efficiently over short distances and costs less. For localization inside the elevator, either a ceiling light or an installed landmark with a SGM reader sensor can be used, and passive landmarks have been chosen due to their robustness. The MFS has APIs to both the EHS and the arm kinematic module in order to control the pressing operation. The EHS sends the extracted real-world position to the arm, whose API then translates this position into the required joint movement based on an inverse kinematic solution [156]. Finally, the combined movements of all the joints direct the arm end effector to the button for pressing. The elevator should be handled in a safe manner to complete multi-floor transportation tasks. Thus, another method is utilized where the elevator is equipped with an interface to control the call operation over Wi-Fi. In this strategy the MFS calls the elevator over a Wi-Fi signal for the robot's current position and then selects the destination floor based on the logical organization of the transportation tasks. For example, the destination floor for the placing operation must be chosen if the robot has completed the grasping stage. Both strategies are combined as a hybrid method for safe elevator handling, and the management system selects the best elevator handling method via the following procedure. First the robot uses the button recognition method and arm to press the correct button and tries to get the elevator three times. After that if it fails, the elevator is called over a Wi-Fi signal. The current floor reading from the elevator can be either based on computer vision, as clarified earlier, or on the height measurement system. The LPS25HB pressure sensor and the STM32L053 microcontroller are used as a height measurement system hardware platform and are configured and programmed to sense the environment and detect the current floor number. The height measurements give a robust floor estimation approach, which operates by applying a smoothing filter with auto-calibration stages to cancel the effect of wide variations in pressure sensor readings, and the second method requires new hardware but is more stable in cases where the floor number mark is blocked by a person. In cases of leaving the elevator on the wrong floor, the elevator error handler has the ability to return the robot to the elevator and repeat the elevator operation processes until reaching the destination selected in the transportation order.

### **3.3 H20 Mobile Robot**

#### **3.3.1 General Introduction**

One type of mobile robot produced by the Canadian Dr. Robot Company, called the H20 [157], has been used to show the efficiency of the multi-floor system. This robot is shown in Figure 31. It is equipped with many sensor/actuators and includes components such as an on-board

computer, PMS5005 controller board, localization module sensor, dual robotic arm, ultrasonic sensor, infra-red (IR) proximity detection sensors, IR motion sensor, motorized robotic head, arm camera and main camera system, and a battery.

The localization and additive (RGB-D), LPS25HP pressure sensors are explained here too due to their importance.

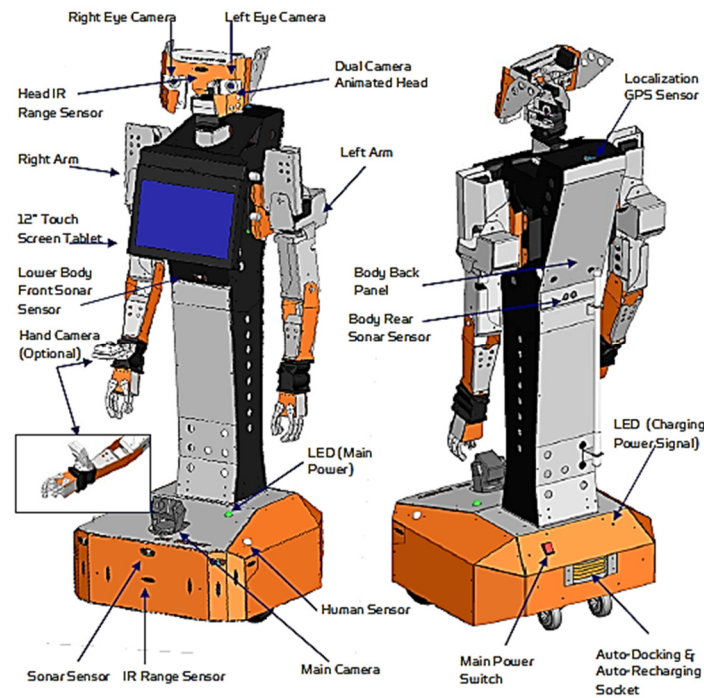


Figure 31 H20 mobile robot [140].

### 3.3.2 StarGazer Localization Sensor

The StarGazer Module (SGM) is an active sensor for mobile robot indoor localization. It has high resolution, accuracy, and speed [155]. The SGM consists of an IR projector, an IR camera, an internal flash memory and an on-board processor. Indoor localization based on passive landmarks can work with the SGM in dark environments since the infrared light is less affected by ambient light. Each landmark is assigned with a unique ID, and the SGM senses the reflected IR rays from the passive landmark installed at the ceiling. The artificial passive landmark has either 3\*3 or 4\*4 reflective circles made of retro-reflective coated film which strongly reflects IR light. The SGM sends IR light and then the landmark reflects this ray, as shown in Figure 32a. The 4 corners are used for height measurements and landmark orientation, while the other points (12 in a 4\*4 configuration) extract the landmark ID, and each reflective circle is multiplied by the corresponding weight to generate the ID (see Figure 32b). The maximum possible number of landmarks for the 4\*4 configuration is  $2^{12}$ , which is equal to 4,095 landmark IDs, and normally the distance between adjacent landmarks is 2m

(which varies according to ceiling height). Thus localization based on the SGM can cover a large area. The sensor sends data over the COM port in ASCII code, and takes twenty readings per second. It has a repetitive precision of 2 cm and an angle resolution of 1 degree. The raw data format is:

$$\begin{matrix} F \\ \sim^{\wedge} \quad | \quad li|\pm a|\pm x|\pm y|z \quad \backslash \\ Z \end{matrix}$$

where i represents the ID number, a is the landmark orientation, x is position in the X axis, y is Y axis position, and z is landmark height.

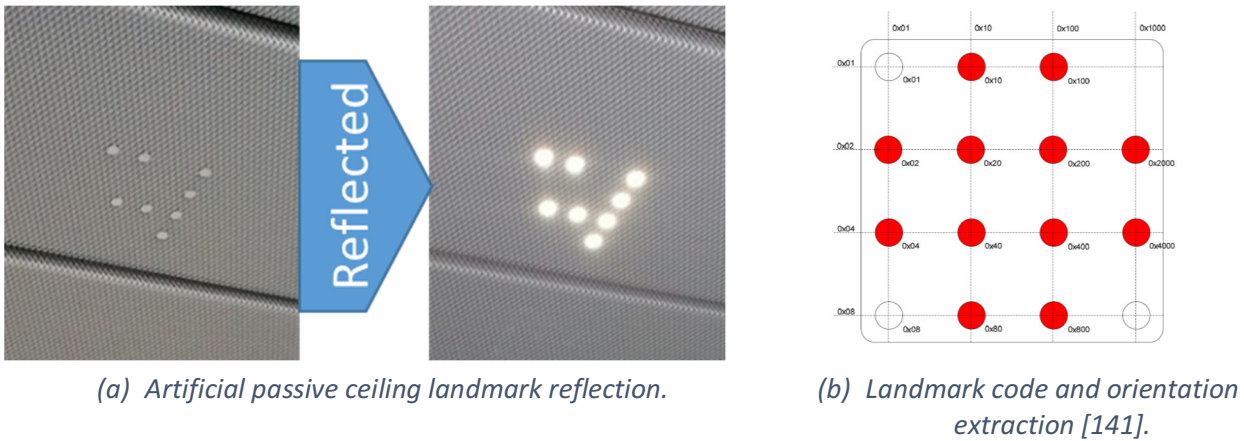


Figure 32 Artificial passive ceiling landmark for SGM

The sensor works in two modes. The mapping mode has one reference landmark whereas coordinates and angle measurements are given according to the landmark's reference position. In mapping mode, the SGM requires the configuration of map size, reference landmark, and type of landmark. Map building is then easily achieved by moving the SGM around the building to collect information on the relationships between landmarks. Information acquired from the ceiling landmarks gives the robot the ability to localize itself on the map according to the landmark's reference position. This working mode cannot build more than one map since it uses only an x and y position.

In 'alone' mode, every landmark represents a reference point and the coordinate and angle measurements are given according to the current landmark's reference position. Figure 33 demonstrates the difference between the SGM working modes, and Figure 34 shows an artificial passive landmark installed in the life sciences laboratories.

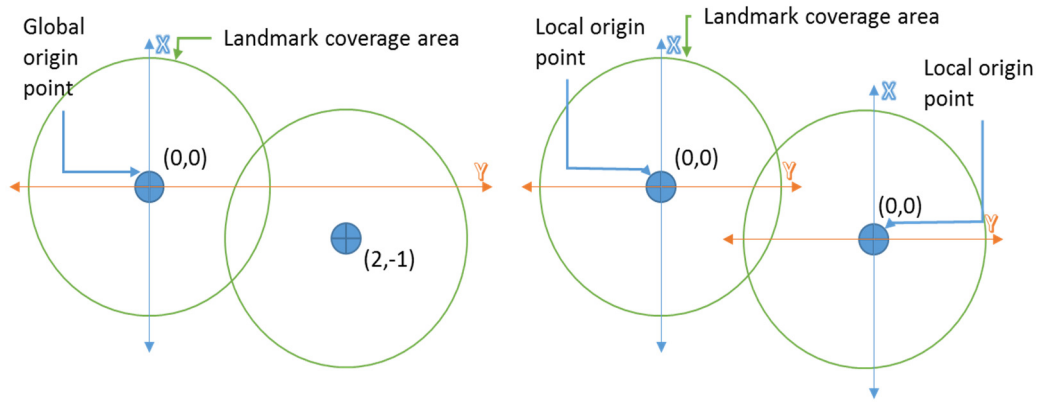


Figure 33 StarGazer module working modes

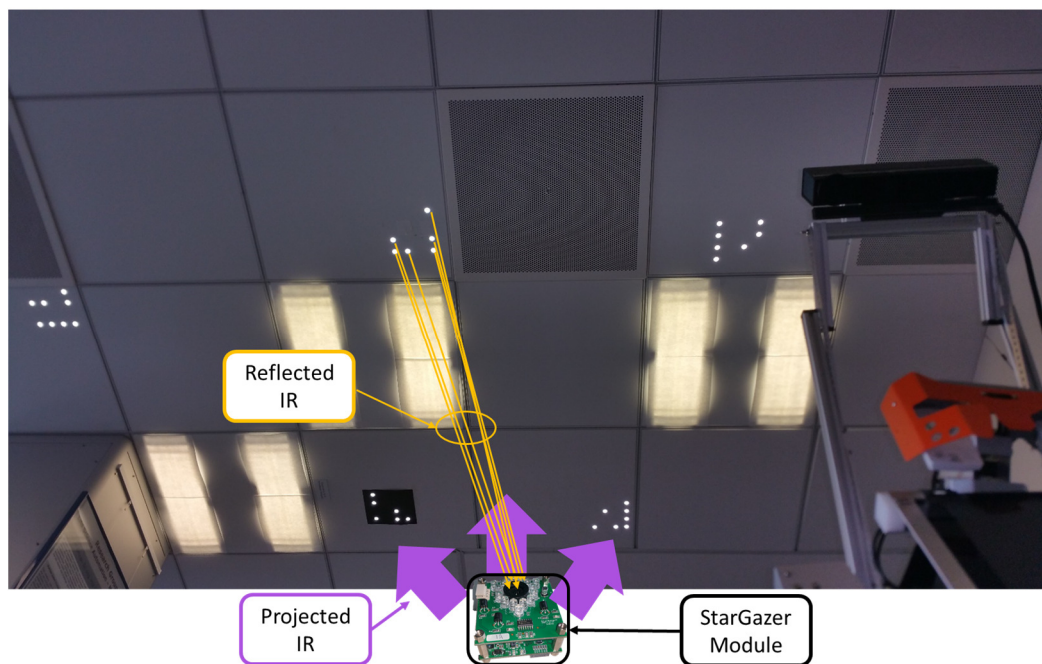


Figure 34 StarGazer localization sensor

### 3.3.3 Kinect Sensor

The Kinect sensor is a motion sensing input device developed by Microsoft for gaming and Windows PC. It provides the ability to control and interact with a game or specific application using gestures and spoken sounds as commands. This sensor consists mainly of an RGB camera, an IR camera, an IR laser projector array, a multi-array microphone, and a processor as shown in Figure 35. The RGB camera is used to recognize the face, to show the video while the IR camera and IR projector work together as a depth sensor, and a structure of light technique is used to find the depth for each pixel. The RGB sensor has a 640\*480 resolution at 30 frames per second, while the depth sensor has a resolution of 320\*240 and can measure depth between 45 to 4500 cm and the camera angles for horizontal and vertical are 57 and 43 degrees respectively. The software provides 3D motion for the human body and face and

voice recognition [158]–[160]. This sensor is usually used due to its beneficial characteristics of low cost, high accuracy and reliability and its ability to extract the real position in Cartesian coordinates of the detected object by mapping the object's position in the captured image with depth sensor information. The position of the Kinect sensor is the reference of these measurements.

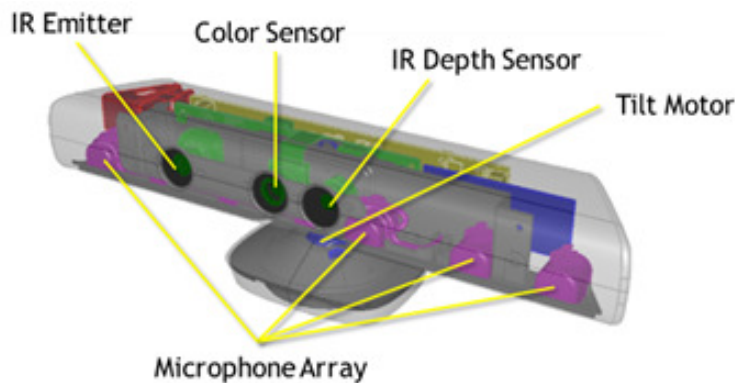
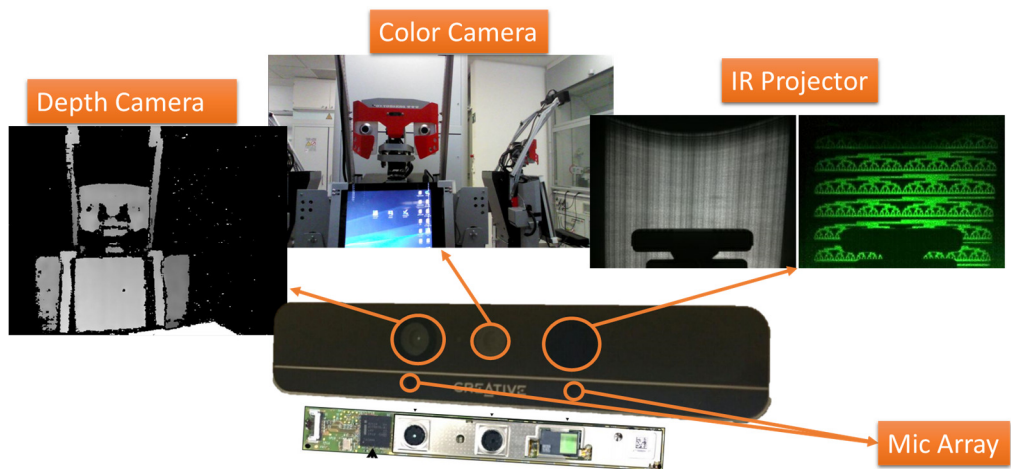


Figure 35 Microsoft Kinect sensor [142]

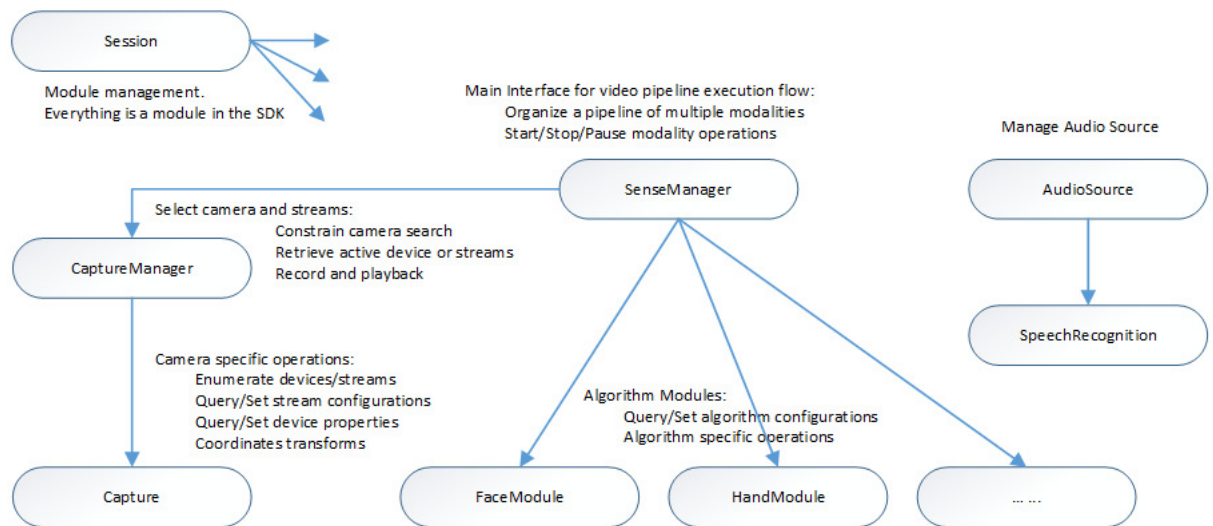
### 3.3.4 The Intel RealSense F200

The Intel F200 camera was developed by the Intel company to work over short distances based on coded light technology to extract the depth position, and it has full HD camera resolution (1,080 pixels at 30 frames per second), a working range from 20-120 cm for depth sensing, gesture tracking at 20-60 cm, and face recognition at 25-100 cm. Figure 36.a shows the components of RealSense F200 camera. It works under the Windows 8, 10 operating system. From the software development kit (SDK) shown in Figure 36.b, this sensor is capable of face, gesture, and speech recognition, and acquisition of raw data (RGB image, depth data, IR Image). This hand camera works at short range, and therefore requires higher robot positional accuracy due to the limitations of the camera view over a short distance.



(a) F200 camera components





(b) F200 software development kit [161]

Figure 36 The RealSense F200 camera and its software development kit

### 3.3.5 The LPS25HP Pressure Sensor with STM32L053 Microcontroller

The LPS25HP is an integrated chip working as a digital barometer, consisting mainly of an absolute pressure sensor as a sensor bias, Wheatstone bridge piezo resistances representing the sensing elements, and a multiplexer to select either temperature sensor data or pressure data as output signals, an analog to digital converter (ADC), temperature compensation, and an IC bus to communicate with other devices through the I2C or SPI interfaces which are also integrated. The main features of the LPS25HP pressure sensor are an absolute pressure range of 260-1260 hPa, high resolution reaching 0.01 hPa with 24-bit data output, low current consumption reaching 4 $\mu$ A with supply voltage of 1.7 to 3.6 V, and temperature compensation is embedded inside.

When pressure is applied to the sensing element, an imbalance occurs in the resistor element of the Wheatstone bridge. Its output signal in addition to the temperature sensor data are converted by the low noise ADC and then a temperature compensation stage is applied. Finally, the digital results are sent over the I2C/ SPI bus with the data-ready indicator. The SPI/I2C bus with the data-ready pin makes this sensor ready to connect directly to the microcontroller system.

The STM32L053 microcontroller has been chosen in combination with the LPS25HP pressure sensor through the SPI due to its low cost and power consumption, high-performance RISC architecture, and high speed with 32 MHz oscillation frequency. It has a lot of peripherals, including 51 GPIO, 16 14-bit ADC, 1 channel 14-bit DAC, 64KB program memory, 8KB RAM, 2kb data memory, USB 2.0 port, USART, 4 SPI, 2 I2C, and 9 timers, which provide wide facilities to be employed in various applications.

### 3.4 Communication System

A network socket can be defined as a communication end-point between two programs running over a network [162]–[164]. The transportation system utilizes a server-client architecture where the server generates one socket for each client. There are internal and external implemented sockets. The internal sockets are utilized to communicate with the robot's hardware, receiving sensor information and controlling the robot movements, while the external sockets are established in the multi-floor transportation system for data exchange with the RRC, RAKC, CAS, EHS, RMC, and the automated elevator and door controllers (see section 4.2.8).

#### 3.4.1 Internal Structure of TCP/IP

The wireless IEEE 802.11g network is utilized for laboratory transportation. This has a suitable width, and fast data channel areas [154] to provide an appropriate wireless communication environment for all of the developed hardware components and systems. The client/server architecture is adopted for data exchange between the multi-floor system (MFS) and the mobile robot, where the MFS represents the client socket and the onboard computer inside the mobile robot is the server socket. Figure 37 shows the complete network architecture for the mobile robot. The SGM is connected to the internal switch through module 2, while the motion motors are connected through module 1. The MFS reads position information from the SGM and sends movement orders through the robot on-board computer. A TCP/IP-based protocol is established due to its reliability.

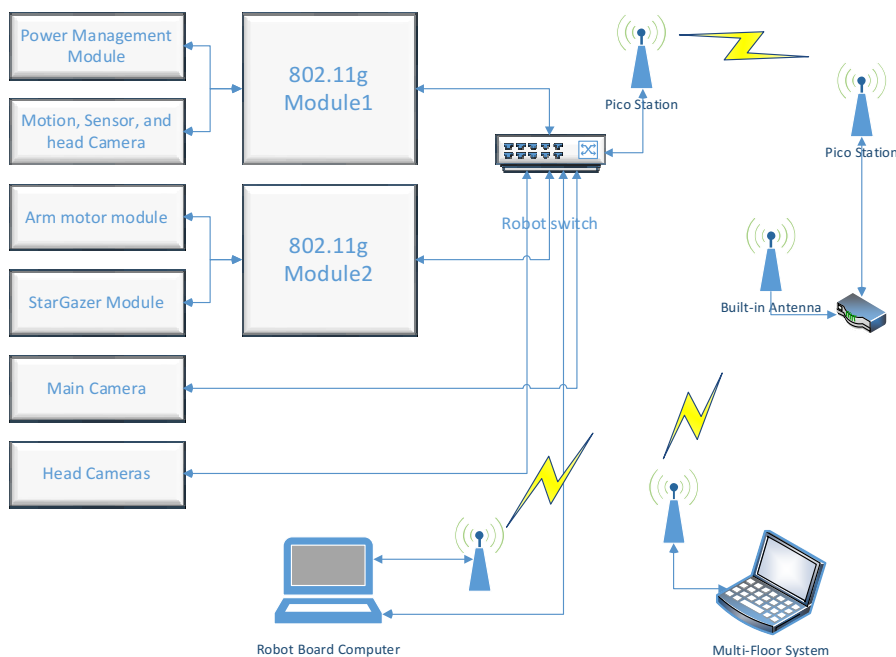


Figure 37 Internal TCP/IP communication network



## Chapter 4 NAVIGATION SYSTEM

In multi-floor transportation systems, certain features are required such as high speed transportation, coverage of a large laboratories area, low cost, and low sensitivity to the effects of changes in light conditions. The SGM in 'alone' mode is used to satisfy these requirements. In mobile robot navigation systems, many basic issues have to be handled, including mapping, localization, path planning, the movement core unit, and the EHS, IMADCS, IBCMS, navigation error handling system, and communication systems. In this section, a new and robust navigation system is presented to assist with transportation tasks in multi-floor laboratory environments.

### 4.1 StarGazer Handling API

It is essential for the API for the multi-floor navigation system to choose the right working mode, as explained in section 4.3. The H20 mobile robot has an API to use the map working mode for map building and localization without giving the possibility to change the working mode. Thus the API is developed to communicate over a Wi-Fi connection and to send the control term to set the mode to the "Alone Mode". This is a fundamental step in a multiple floor navigation system, which can also change the threshold algorithm to manual or auto, and the threshold value. Figure 38 shows the GUI created to set the working mode.

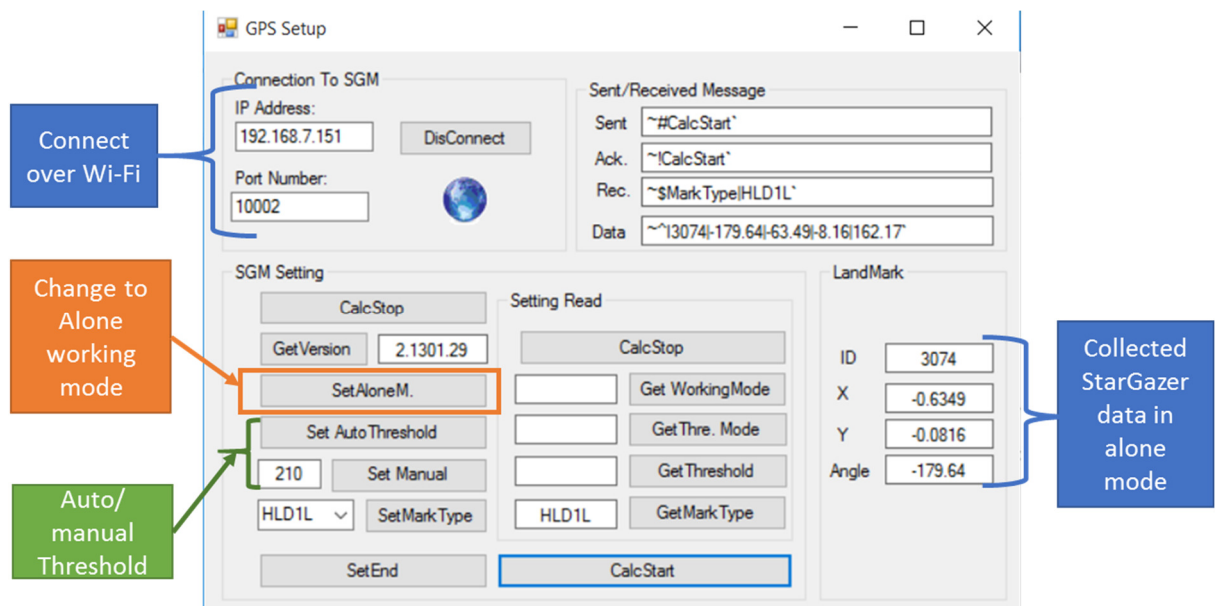


Figure 38 StarGazer handling API for changing the working mode

## 4.2 Multi-Floor Navigation System

To serve a multi-floor transportation system, a robust navigation system including mapping, indoor localization, path planning, and other relevant systems is required. This section describes the mapping, indoor localization, the path planning processes, the IMADCS, IBCMS, vision landmark reader error management, and fine position correction method, along with the results of experiments assessing the performance of these components.

### 4.2.1 Map Building

The first version of the navigation system used the SGM in mapping mode, which stores the complete map inside the SGM. This module gives the position in the building map to the robot so that, according to the transportation task, the robot moves along the path generated [46]. This method works for single floors and the process requires a long time due to individual map building and path generation for each robot. The required time varies according to the number of robots, map size and the number of paths required.

The SGM cannot store multiple floor maps in mapping mode due to the discontinuity between floors. In the proposed multi-floor navigation system, a new mapping method based on SGM in the 'alone' working mode is developed to develop relationships between all landmarks according to a single reference in the multiple floor environment called the relative map. This map is then used with a navigation program to reach the required destination based on the planning path. The structure of the relative map can be formulated as follows:

$$RelativeMap = \begin{bmatrix} \begin{bmatrix} IDF_{1,1} & IDF_{1,2} & \cdots & IDF_{1,n-1} & IDF_{1,n} \\ XF_{1,1} & XF_{1,2} & \cdots & XF_{1,n-1} & XF_{1,n} \\ YF_{1,1} & YF_{1,2} & \cdots & YF_{1,n-1} & YF_{1,n} \\ \theta F_{1,1} & \theta F_{1,2} & \cdots & \theta F_{1,n-1} & \theta F_{1,n} \end{bmatrix} \\ \begin{bmatrix} IDF_{2,1} & IDF_{2,2} & \cdots & IDF_{2,n-1} & IDF_{2,n} \\ XF_{2,1} & XF_{2,2} & \cdots & XF_{2,n-1} & XF_{2,n} \\ YF_{2,1} & YF_{2,2} & \cdots & YF_{2,n-1} & YF_{2,n} \\ \theta F_{2,1} & \theta F_{2,2} & \cdots & \theta F_{2,n-1} & \theta F_{2,n} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} IDF_{m-1,1} & IDF_{m-1,2} & \cdots & IDF_{m-1,n-1} & IDF_{m-1,n} \\ XF_{m-1,1} & XF_{m-1,2} & \cdots & XF_{m-1,n-1} & XF_{m-1,n} \\ YF_{m-1,1} & YF_{m-1,2} & \cdots & YF_{m-1,n-1} & YF_{m-1,n} \\ \theta F_{m-1,1} & \theta F_{m-1,2} & \cdots & \theta F_{m-1,n-1} & \theta F_{m-1,n} \end{bmatrix} \\ \begin{bmatrix} IDF_{m,1} & IDF_{m,2} & \cdots & IDF_{m,n-1} & IDF_{m,n} \\ XF_{m,1} & XF_{m,2} & \cdots & XF_{m,n-1} & XF_{m,n} \\ YF_{m,1} & YF_{m,2} & \cdots & YF_{m,n-1} & YF_{m,n} \\ \theta F_{m,1} & \theta F_{m,2} & \cdots & \theta F_{m,n-1} & \theta F_{m,n} \end{bmatrix} \end{bmatrix} \quad (3)$$

From equation 3 can be seen, the map consists of multiple layers, each representing one floor. Each layer contains the map elements which are a landmark ID number (IDF), an X position (XF) according to the starting point, a Y position (YF) according to the same reference point, and finally the angle ( $\theta F$ ) which is also related to the reference point of the current floor. Figure 39 shows the relative map creation sequence. This map is created manually using the following special procedures:

(a) Collecting data is accomplished using a representation of the difference between two landmarks, including the distance and angle relations. This step was conducted using a moving cart which was positioned between every pair of landmarks and then the relations ID, XID, YID, and AngleID were recorded.

(b) Entering the data into an Excel spreadsheet required a mathematical equation to find the relationship between landmarks. Consider:

$$X_{n+1} = X_n + \Delta X_n \quad (4)$$

$$Y_{n+1} = Y_n + \Delta Y_n \quad (5)$$

where  $\Delta X$ ,  $\Delta Y$  is the measured distance between the current two landmarks based on SGM reading. Next, consider:

$$\theta_{n+1} = \theta_r - \theta_{LM} \quad (6)$$

where  $\theta_r$  is the reference point angle and  $\theta_{LM}$  is the installed landmark orientation (0, 90, 180, and -90). An Excel spreadsheet was prepared to easily determine the relationships after entering all of the data from the first stage.

(c) For coding into the MFS, the relative positions for each landmark (ID, X\_relative, Y\_relative, and Angle\_relative) were taken from the Excel spreadsheet and added to the floor layer. A unique reference point for each floor was defined to unify all the readings against the same reference:

$$RelativeReference = \begin{bmatrix} X_{Re}F_1 & X_{Re}F_2 & \cdots & X_{Re}F_{n-1} & X_{Re}F_n \\ Y_{Re}F_1 & Y_{Re}F_2 & \cdots & Y_{Re}F_{n-1} & Y_{Re}F_n \end{bmatrix} \quad (7)$$

where  $X_{Re}F_1$ ,  $Y_{Re}F_1$  is a constant value for each floor to extract a unique reference in the whole map.

These procedures are performed manually and require some time to be done accurately. The time taken to build a relative map depends on the operator's experience with the system, the map size, and the required number of paths. For example, the time required for the proposed method is 10 minutes to build a map in SGM mapping mode plus 10 minutes for the creation

of obstacle-free paths for one robot [46]. This time will be multiplied by the number of robots in calculating the overall time required for multiple mobile robots. The method presented in this dissertation takes only 20 minutes for map creation plus 10 minutes for obstacle-free path creation. Once realized, it can be applied to any number of robots, and thus it minimizes the map/path generation time compared to the earlier applied method.



Figure 39 Relative map creation: (a) landmark data acquisition; (b) entered and process acquisition data; (c) relative map class

Initially the map was embedded inside the code as a class, which makes it difficult to update or add any new waypoints without changing the code design. Thus, a different method to store the relative map in a separate file is utilized, as shown in Figure 40. A special function was developed to read the file and translate it into a relative map class. The separate map file method adds flexibility, and makes it simple to make changes such as modifying or enlarging the map for any number of floors, and to increase the map size.



Finally, the  $x, y$  SGM readings are translated into real coordinates in a global multi-floor map:

$$x_G = x_{SGO} + x_R + x_{FR} \quad (11)$$

$$y_G = y_{SGO} + y_R + y_{FR} \quad (12)$$

where  $x_R, y_R$  are relative values from the relative map and  $x_{FR}, y_{FR}$  are the multi-floor reference for the current floor, which is used as a multi-floor unique reference.

The indoor localization procedures are illustrated in Figure 41. The map matching method is coded in a way that easily allows an expansion to  $m$  number of floors and  $n$  number of landmarks.

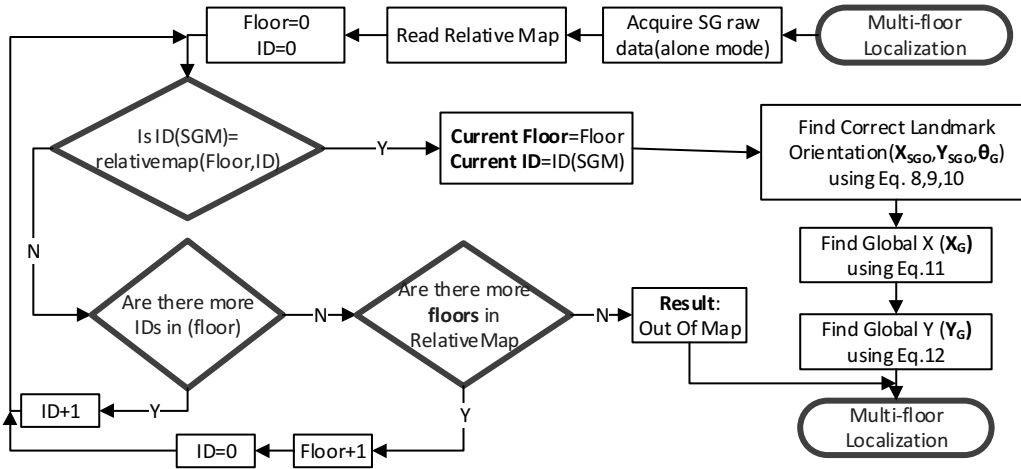


Figure 41 Localization method in multi-floor environment

### 4.2.3 Path Map Creation

After building the relative map via the method described above and using the map matching method to find the robot's position in the multiple-floor global map, the optimal paths have to be created. A path generally consists of a list of waypoints, each of which includes information used by the movement core unit and the Internal Transportation Management System (ITMS) for static obstacle-free dynamic path planning and IMADCS. For the movement core unit, the waypoints are utilized (see Figure 42) to specify the next position with the defined movement speed and direction, position and angle tolerance, and correction parameters.

	ID	X	Y	Angle	StopTir	Movem	Toleran	TurnSp	Collisor	Revers	Target/	DoorCo	FinalPo	XBC	YBC	Position	Connec	SeqNo
▶	2932	20.27	7.76	90	0	60	20	60	☑	☐	1	0	☑	0	0	L021...	1	1
	2932	20.27	7.02	90	0	60	20	60	☑	☑	1	0	☐	0	0		4	2

Figure 42 Path waypoints parameters

Each path can be used to connect two stations in addition to the middle point between them, and it can be used as a bidirectional path for movement, as shown in Figure 43 which includes: position 1 <-> position 2 [0,1,2,3,4,5,6], position 1 <-> middle point [0,1], middle point <-> position 2 [1,2,3,4,5,6]. Thus, the proposed path creation method is easier, more flexible and decreases the time required for path creation compared with the method described in a previous study [154]. The position name parameters are embedded in the waypoints and the coding method extracts these sub-paths from one path. The Door Controller field makes the process of path building including the doors easier than before [46], and the movement core uses these numbers with the help of a class in a look-up table to identify the door server, door port, and door pin.

The laboratory staff building the path can more easily add the door by choosing its door number without updating the code or specifying the door's server IP port and pin parameters. XBC, YBC are used to calibrate the robot's movement in the other direction by adding these values to the next point position in the reverse direction. However, in a narrow path area, it is necessary for the robot motor to have a small portion of orientation error when starting to move, and these values are used to solve shifting problems that may arise in the bi-directional path. The ITMS uses the position name field to extract the waypoint number, which is essential for: generating a movement path, and to detect path error. Other fields include the following:

- (a) The (ID,X,Y,Angle) is used to store the actual waypoint position of the robot after translating the current landmarks vision reading into a multiple-floor global map based on the relative map.
- (b) The stop time is used to add a specific delay between successive waypoints, and this delay is useful when waiting the RAKC.
- (c) The robot's speed of movement is controlled by movement speed field.
- (d) The movement acceptance tolerance is selected according to the tolerance parameters.
- (e) The rotational speed is selected for each point to achieve high precision in rotation.
- (f) A collision avoidance checkbox is used to enable or disable the simple collision avoidance method, which merely checks if the path is free from obstacles before moving forward based on the ultrasonic distance sensor.



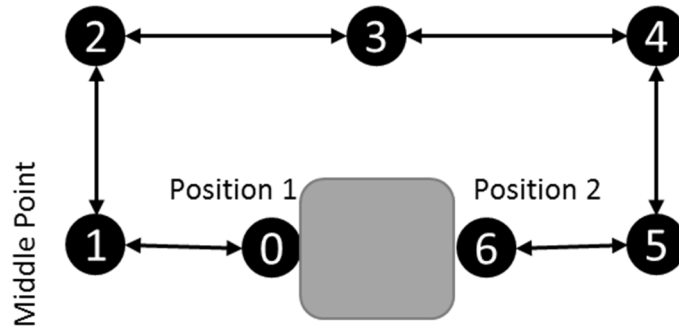


Figure 43 Bi-directional Path Structure

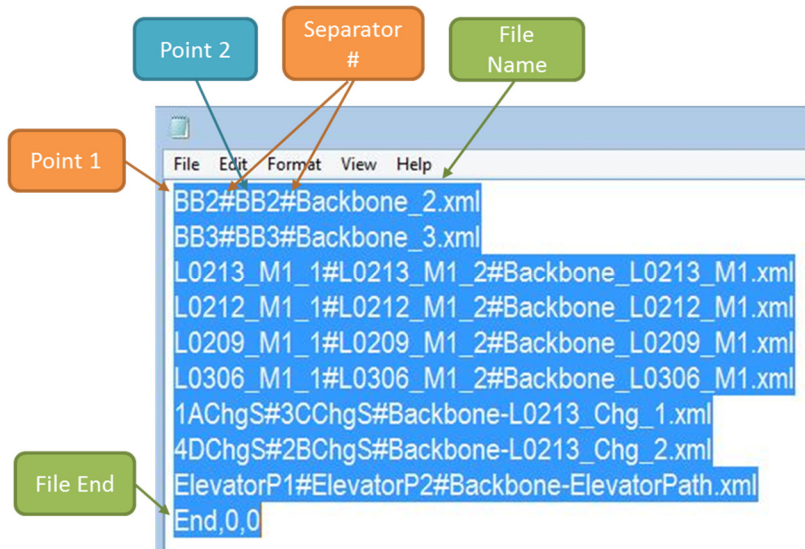


Figure 44 path points and it is file name stored in a separated file

Every path which is generated is stored in a separate XML file in addition to a unique text file which stores the path file names with their related points (see Figure 44). Each unique file is used later to specify the path for the current requested goals. This method is used to easily access, add, and modify any paths to cover a large-scale environment.

#### 4.2.4 Path Planning

Path planning can be defined as a method of controlling motions from a starting point to the goal point avoiding collisions and taking the shortest path to the destination. Many algorithms have been proposed and developed for this purpose [121], [124], [125], [165]. Compared to standard ones, the path planning issue poses more challenges for mobile robot transportation in life science laboratories. For example, since the proposed path planning method will be applied in real robot transportation systems, real-time planning results are desired. However, there are many combinations of the starting positions and destinations, and so the desired path planning method should be able to deal with dynamic path combinations quickly and flexibly. As a first version of path planning problem handling, a new path planning method



(named the backbone method) is presented here, which consists of an essential path for each floor and a group of flexible sub-paths for all laboratory stations. The corresponding ITMS software has also been developed by adopting the proposed backbone method. Since the complete multi-floor transportation system has to be integrated with other systems such as for collision avoidance and to add more intelligent capabilities to the navigation system, a hybrid backbone-Floyd path planning method with the developed ITMS is also presented in this section.

#### ***4.2.4.1 The Backbone Method***

This method is based on building a main path for each floor. This path has multiple key points leading to sub-paths that complete the transportation path, as shown in Figure 45. This method decreases the number of paths required to cover a large map with many stations (with multiple starting and destination points). In addition, this method makes path creation easier than with a normal static path by decreasing the total number of paths required. The number of paths required in the static path method is  $N*(N-1)$ , where  $N$  is the number of laboratory stations. The total number of path required in the backbone method is  $((\text{No. of Floor}) * \text{backbone path} + N/2 + \text{one elevator path (multi-floor case)})$ . For example, a single floor in the working environment consists of six transportation station points with two charge station points, giving results of 56 paths in the static method and 5 paths using the backbone path planning method. Table 1 shows the reduced number of paths with the backbone method especially in a multi-floor environment. When the robot receives the transportation task from the RRC, the ITMS divides the transportation task into small tasks and begins executing these tasks as follows:

- (a) Search inside the created paths to find the way from the charge station to the point near the backbone path.
- (b) Load the current floor (backbone) essential path and, according to the source station position, either move directly to the grasping station key point or move through multiple paths (current path toward elevator, elevator path, and grasping floor essential path to the grasping station key point) if the grasping station is on another floor.
- (c) Use the current grasping sub-path to reach the grasping station and execute the required operation.
- (d) Repeat (b) and (c) taking into consideration that the movement is towards the placing station.
- (e) Load the current floor backbone path and either move until reaching the charge key point on the path or move through (current path toward elevator, elevator path, and charging

floor essential path to the charging station key point) if the charging station is on a different floor.

(f) Use the charging path to reach the charge station and wait for the new transportation task.

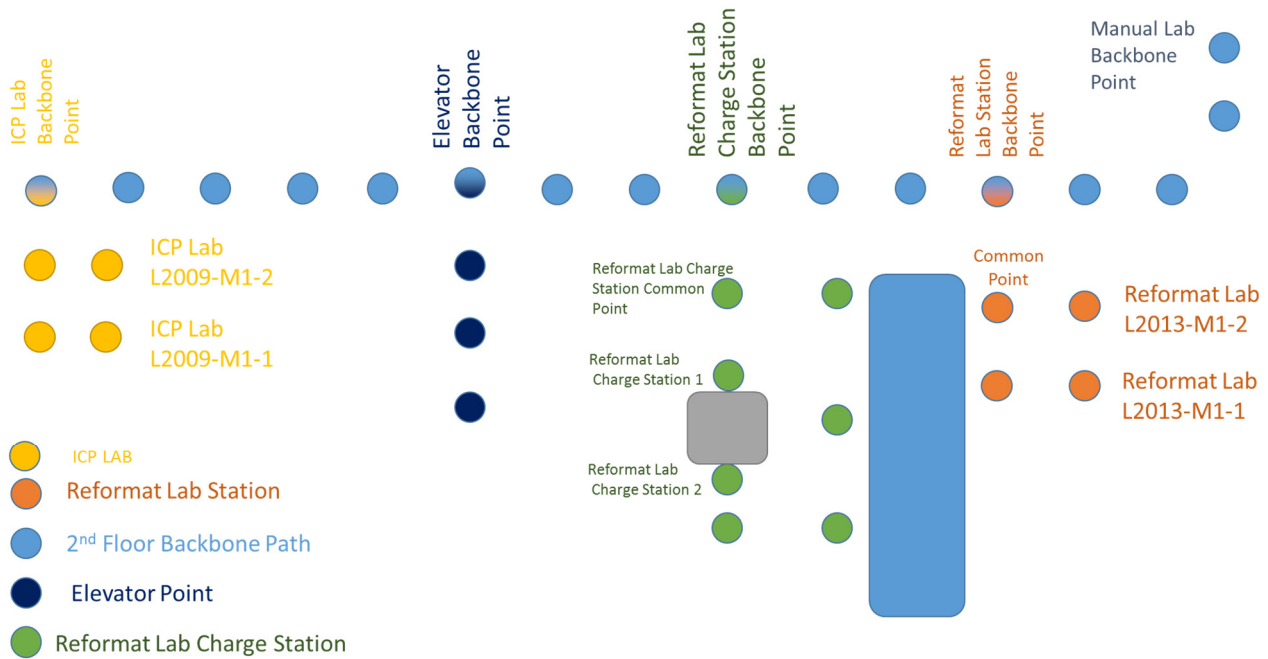


Figure 45 Waypoint distribution in the backbone method. Each color represents a path while mixed colors represent a waypoint connecting the essential path to a sub-path on the 2nd floor in Celisca

Table 1 Static method and backbone method comparison (path required).

No. of Transportation Station Point	Normal Paths Method	Backbone path method
8 (2nd) floor	56	5
12(1st+2nd) floor	132	9
16(1st+2nd+3rd) floor	240	12
20(1st+2nd+3rd+4th) floor	380	15

#### 4.2.4.2 Internal Transportation Management System (ITMS)

The ITMS is a transportation management system inside the MFS. It has been coded based on a multiple thread technique to increase efficiency and enhance the ability to access the MFS at any time. The management system shown in Figure 46 divides the incoming transportation task into small tasks whose execution is continuously monitored. It also informs the RRC level about the current transportation state (robot busy, transportation start, RAKM problem, wrong path/points parameters, and transportation finished). The ITMS chooses the right path and points, sends them to the movement core, and waits until the end of this task. Then

another path is loaded until the entire transportation task has been completed. When there is a problem with any task, the ITMC stops other tasks immediately and informs the RRC. The mapping, localization, Backbone path planning, and ITMS were published in [166].

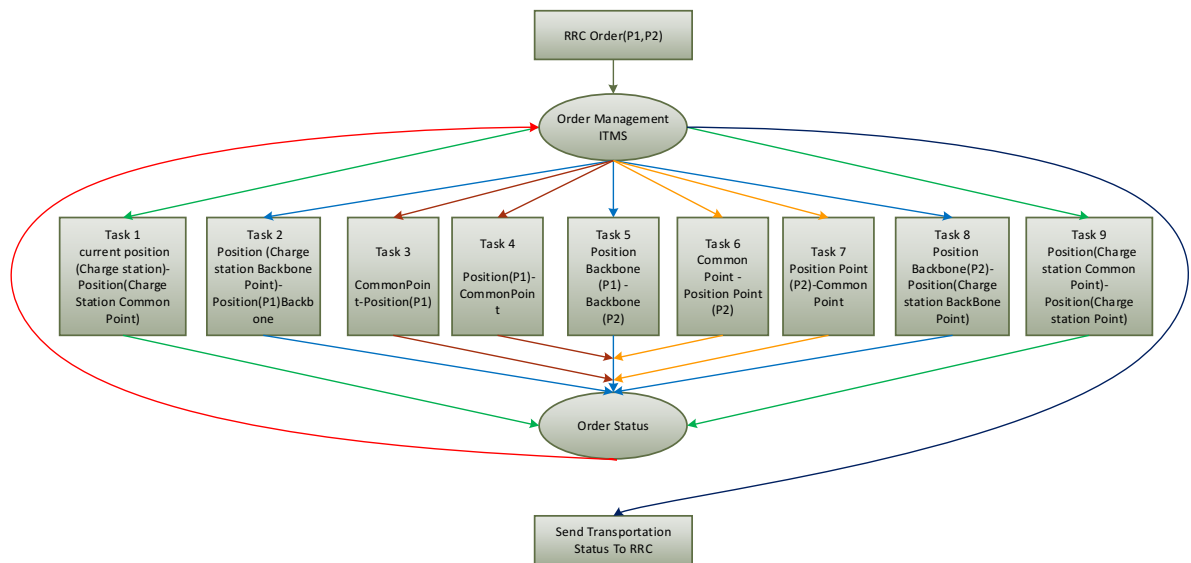


Figure 46 Internal Transportation Management System

#### 4.2.5 Internal Management Automated Door Controlling System

The IMADCS is a component used to control the doors during robot transportation and receive information on their status. The life science laboratories are equipped with automated doors and ADAM sockets. These are digital acquisition and control systems operating over a computer network which have the ability to collect data, remotely control the required devices for accomplishing integration between the automated and enterprise systems through network technology. They are distributed around the building to minimize the cables required for automated door connections. To handle the distributed ADAM modules, five sockets are built to manage thirteen automated doors in the multiple-floor laboratories (see Figure 47). A standalone method is created to manage these door controllers without affecting the main system using the following sequences. First the server status is checked and reconnected when required, then the control order to open or close the door is sent, and the door status is read. The movement core (see section 4.2.7) sends an order to the IMADCS to control a specific door according to the developed bi-direction path, and when the current waypoint door control is at 0 and the next waypoint has a specific number (as shown in Figure 48), the movement core sends an open command to the IMADCS with the specified number from the waypoints. Meanwhile, in the case of the current waypoint having a specific door number and the next waypoint having 0 in the door field, the movement core will send the door closed to the IMADCS with the specified door number. The IMADCS executes the incoming order by translating the door number into a specific server, port, and

pin number based on a look-up table embedded inside the unit. Finally, the IMADCS selects the door and performs the requested control order. A validation experiment regarding the integrated automated door management is shown in section 4.2.9.

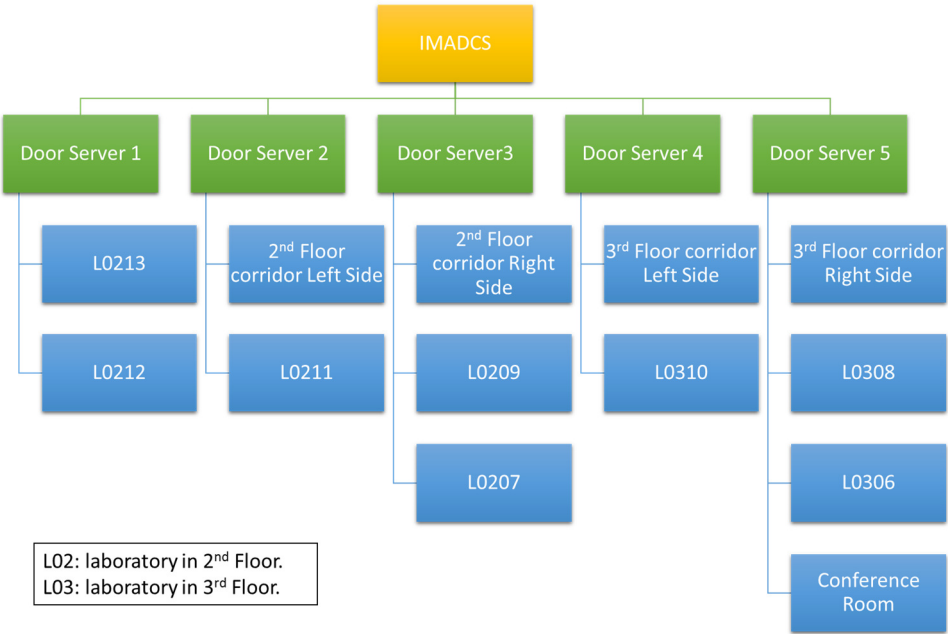


Figure 47 Internal Management Automated Door Controlling System (IMADCS)

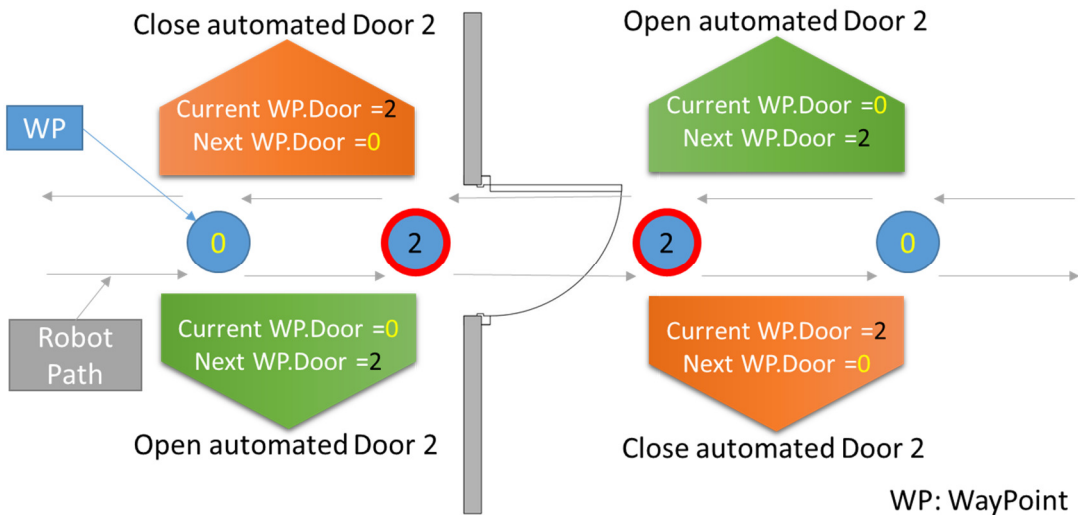


Figure 48 Automated door working mechanism inside movement core

#### 4.2.6 Internal Battery Charging Management System (IBCMS)

The H2O mobile robot is equipped with a motion and power control API, which includes a battery management system utilized by the embedded movement core. However, with a newly-developed multi-floor movement core, a new battery management system is required since newly designed system is applicable for single and multi-floors so it requires advance battery management system. Therefore, a new battery charging management system is embedded in the MFS to monitor the robot's battery and keep it in a fully charged state. This system collects information about the voltage and temperature of the battery and the robot's position. The information collected is used to make decisions from the IBCMS to charge or release the robot from the charging station, as illustrated in Figure 49. If the robot is idle, near to a charging position, and the voltage is less or more than the minimum or maximum threshold values respectively, then the robot takes the appropriate action. If not, the monitoring of the battery's current state continues. These actions include controlling the robot to move to the charge / wait point based on the movement core. The battery's status is checked every second in order to meet the requirements of the RRC level, while the decision about charging or releasing is made by the IBCMS every minute at this level. For charging, the robot repeatedly moves backwards in small steps until it makes contact with the charging station. The method of small reparative movements was chosen to keep the charging station safe and to give enough time for the robot hardware to update the charge status.

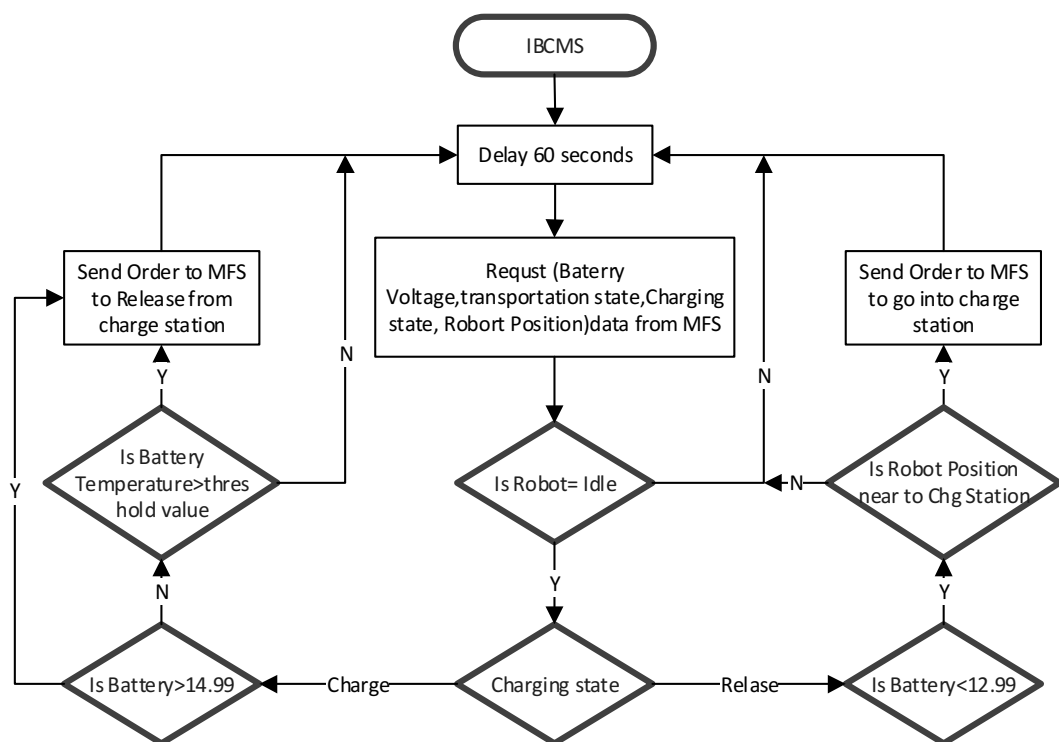


Figure 49 Internal Battery Charging Management System flowchart

In a multi-robot environment, the robot cannot know whether or not any of the charging stations are already occupied by other robots without a higher level system to manage robot charging operations, and so it is necessary to assign the charging station with a specific ID for each robot. The mobile robot utilizes these stations to keep its battery in a fully charged state ready for transportation tasks. Figure 50 shows the mobile robots distributed among the charge stations with the path map. The experiment described in section 4.2.9 shows the robot's ability to reach the charging station position after the execution of a transportation task.

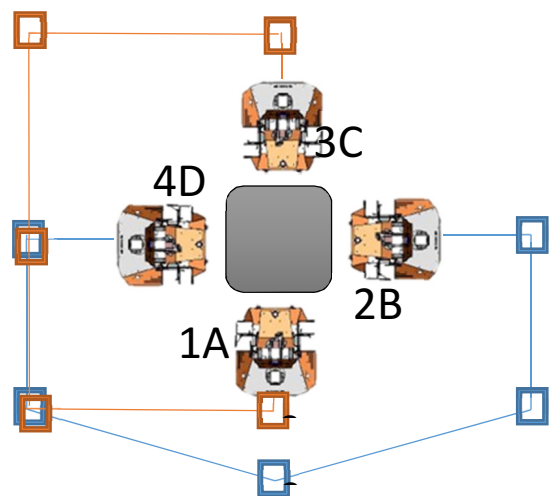


Figure 50 Multi-robot distributed around charge station

### 4.2.7 Movement Core

The movement core is a control unit which controls the robot's motion as it moves through a set of points defined by the path planning stage. It moves from one point to another safely and robustly by utilizing the other functions, including the IMADCS, EHS, RMC, vision landmark error handler, and the fine positioning method. Figure 51a shows an overview of the movement core method. After receiving the movement order from the ITMS with a specified point number for the starting and destination points, the movement core calculates the distance from the robot's current position toward the next waypoint and then controls the robot movement towards the calculated position. In cases where the distance is greater than the acceptable tolerance range of the last operation, the distance calculation and the robot movement is repeated until a point with acceptable tolerance is reached. Then iterate these procedures are repeated until the required destination is reached and the ITMS is informed that the current task is completed. Figure 51b illustrates the decisions made about movement from one point to another, starting by receiving the raw data from the SGM sensor and then

performing the localization processes required to find the robot's position inside a multi-floor global map. Based on the results of the localization procedure, the movement core calculates the angle towards the next waypoint, and then uses the results to rotate the robot repeatedly until it has reached the appropriate angle towards the next waypoint. After rotating the robot accurately, the movement core calculates the distance to the next point by updating the SGM reading, extracting the distance, and controlling the robot's movement until it reaches the position with an acceptable level of error. The door controller must take the decision to open the door earlier, before the robot reaches it, to give enough time for the automatic door to open, and then the decision to close the door is taken after the robot has passed through the door. The door number is specified in path waypoints. The movement core also controls the EHS to hand over the robot's current floor and to call the elevator at a specified time, and utilizing the ITMS transportation status (P1PositionDone, P2PositionDone, ChgPositionDone) to specify the destination floor when the robot is inside the elevator. Equations 13-19 are used to determine the distance and angle to the next points. Eq. 13 is used to find the distance  $l$  between the robot's position and the next waypoint. The robot's position is  $(x_2, y_2)$ , while point  $(x_1, y_1)$  is the current target point. The angles  $\theta_r$ ,  $\theta_c$  are computed using eqs. 14 and 19. According to the robot's current co-ordinates, equations 15-18 are used to find  $\theta_{ct}$  for each region.

$$l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (13)$$

$$\theta_c = \sin^{-1}\left(\frac{y}{l}\right) \quad (14)$$

$$\theta_{ct} = -(180 + \theta_c) \quad (15)$$

$$\theta_{ct} = (\theta_c - 180) \quad (16)$$

$$\theta_{ct} = \theta_c \quad (17)$$

$$\theta_{ct} = -\theta_c \quad (18)$$

$$\theta_r = \theta_{ct} + \theta_{sg} \quad (19)$$

where  $\theta_{sg}$  is the SGM angle that represents the alignment angle between robot and passive landmark,  $\theta_r$  is the required angle to rotate the robot towards the next waypoint,  $\theta_c$  is the angle between the robot and the next waypoint related to the Cartesian coordinates, and  $\theta_{ct}$  is the intermediate calculations angle.

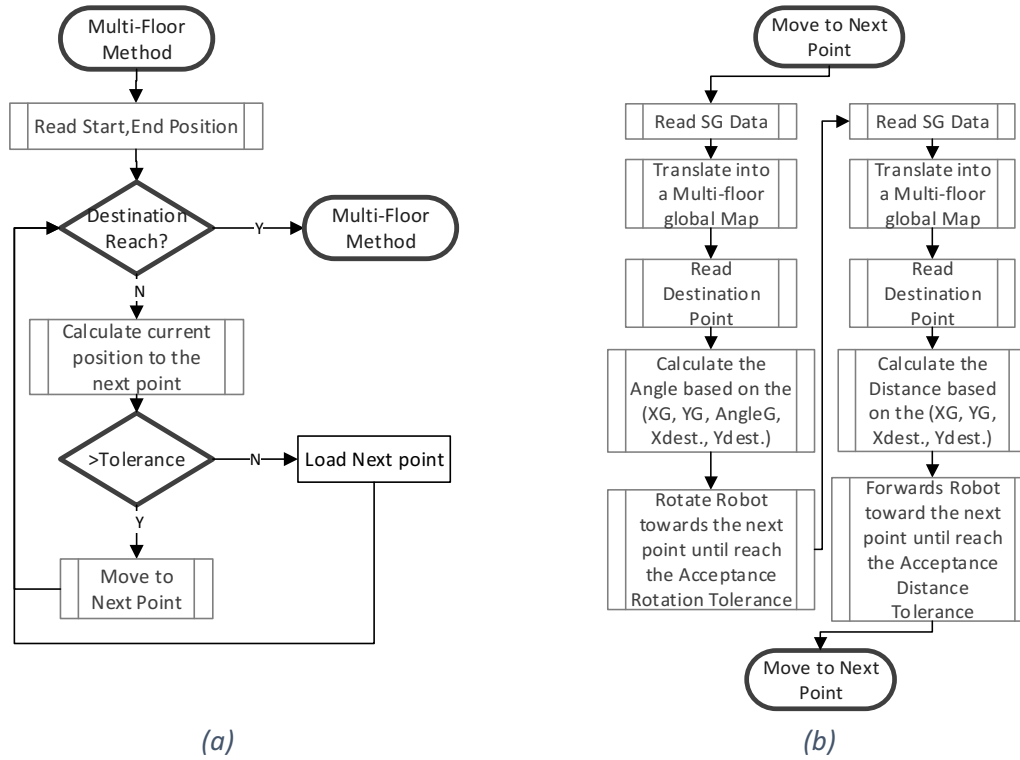


Figure 51 Movement core flow chart: (a) overview of the movement core; (b) movement decision from one point to another

## 4.2.8 Integration of the MFS with other related sub-systems

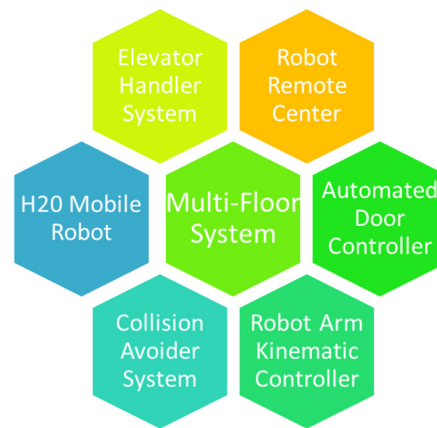
The use of sockets is convenient in allowing the system to update various types of robots in addition to add more system easily. The MFS communicates with numerous API using sockets (see Figure 52), to perform the required transportation tasks in an efficient manner. The client-server connection architecture module (asynchronous socket) is enabled to control the interaction of each sub-system with the others over Ethernet. TCP/IP-based protocol is established due to its reliability. All of the sockets used are explained separately below, including the purpose and data flow diagram.

### 4.2.8.1 Robot Remote Center and H20 Motion and Power Control

A server socket is created in the MFS to manage the connection with the RRC. This connection has a decisive role in receiving orders concerning transportation tasks, and sending back information about the robot's status, and status of transportation tasks. Figure 53 clarified the data flow between these sockets. The MFS receives transportation order, gives the real status of the robot including voltage and status for all power sources that provides the required robot power in addition to send the robot position in multi-floor map as Cartesian coordinates and orientation. The MFS also sends data about the success of transportation tasks or details of any problems which occurred with the robot during the transportation operation. The MFS is utilized to work with a distributed laboratories in multiple floor, which



may cause the connection to the RRC to be lost due to the weakness of the Wi-Fi signal in some locations. Therefore, a heartbeat monitor technique is utilized to predict these disconnections in advance and to do the required processing to handle it. This technique is only used for the connection with the RRC which has a wireless connection whereas other sub-systems are implemented in the same robot's onboard computer. Another connection is made between the H20 mobile robot and the MFS. Through this connection, the MFS commands the robot to move in a straight line or rotate for a specific time, asks for data from vision landmark readings and power source status as well as ultrasonic distance sensor readings. The H20 mobile robot processes these request and sends an acknowledgement that it has received the order by sending either the request back or the information requested. The command format between the MFS, RRC, and RMC is listed as tables in Appendix B.



*Figure 52 Complete socket structure for the MFS*

#### ***4.2.8.2 Elevator Handler System and Robot Arm Kinematic Module***

The EHS is a stand-alone program created to handle the elevator operations for multi-floor navigation. This system has the ability to detect the elevator environment through the following process. The exact position of the elevator entry button is first extracted under different light conditions, then the destination key is selected, and the current floor number is read. All this information is sent back over the socket to the MFS. The MFS receives the position information and sends it as an order to the RAKC (details about the developed arm kinematic module are found in [167]) to press the button using the detected X, Y, Z position information. Additionally, the RAKC socket is used for transportation tasks to send a grasping and placing orders with the current distance measured between the station and the robot. The RAKC inform the MFS when the requested operation (grasp, place, or press button) has been performed. The MFS system waits for a predefined time and then, if the RAKC fails to complete the task, it informs the RRC and stops the current transportation task to save the time. Figure 54 show these data flows. The command format between the MFS, EHS, and RAKM is listed as tables in Appendix B.

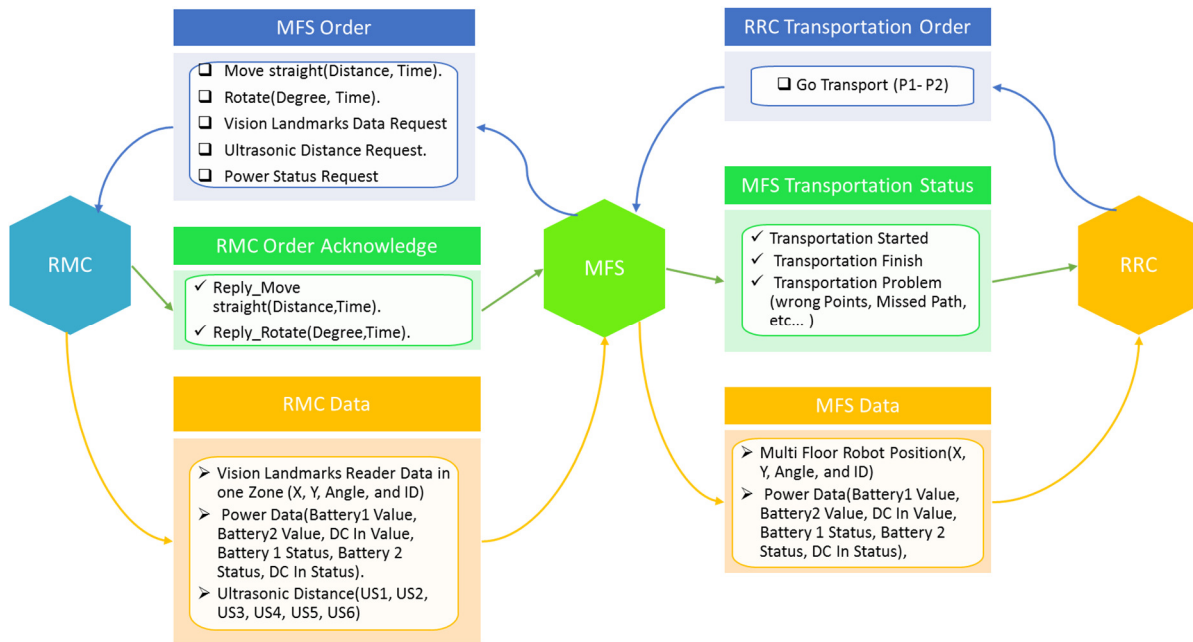


Figure 53 Data flow between the RRC, MFS, and H20 robot

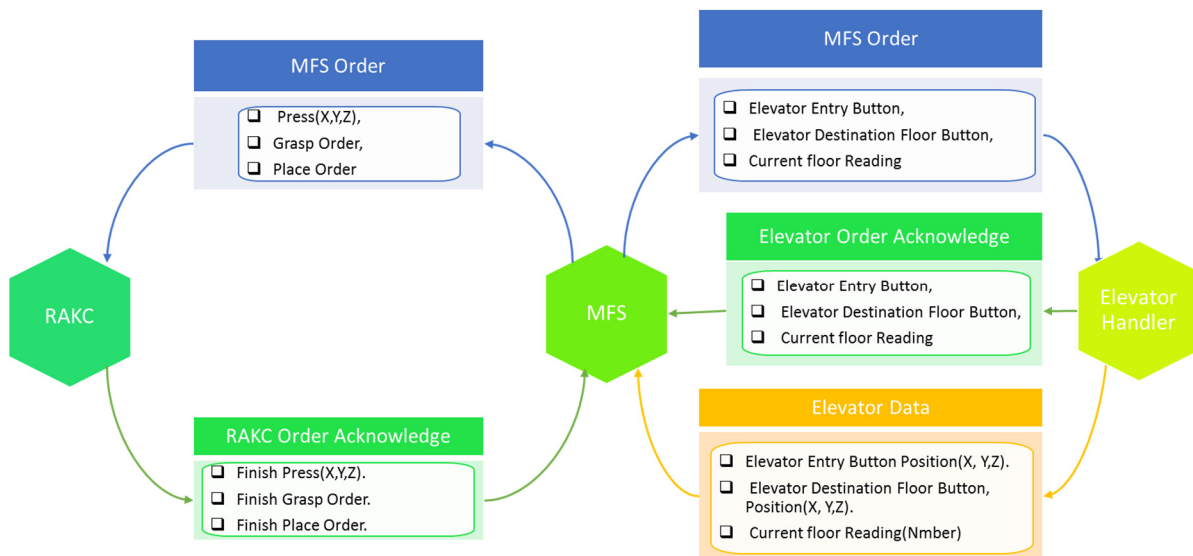


Figure 54 Data flow between the MFS, elevator handler System, and RAKC

#### 4.2.8.3 Automated Door Control and Collision Avoidance System (CAS)

As clarified in section 4.2.5 five sockets were established to manage thirteen doors in the robot's path during multiple floor transportation tasks. The automated door controller provides the ability to open and close doors, and to provide information on door status which is utilized later to easily manage the automatic doors. The approach applied to manage door control has the following sequence. The server connection is monitored and it can be reconnected if required. Then an order to open or close the door is sent. Finally, the door status is read. Another socket to the CAS (details about the developed CAS are found in [168])

is created to utilize this system in avoiding obstacles in the robot's path. The MFS sends an order over the socket to enable the CAS when the execution of a transportation task starts, and then the CAS monitors the robot's path. If any obstacle is found, the CAS sends an "obstacle detected" message to the MFS. The MFS either responds or ignores the incoming message according to the robot's position (the CAS failed to handle the robot in some complex positions for e.g. elevator or narrow area). When the MFS responds to the CAS request, it pauses the transportation operation and provides the CAS with the robot's current position and the next waypoint position. The CAS finds a simple route to avoid the obstacle and sends a movement request to the MFS for execution, including movement forwards or backwards or rotation to the right or left. After the obstacle has been avoided, the CAS sends an "obstacle-free" message. At the end of the collision avoidance maneuver, the robot may have moved outside of the planned area, and thus a hybrid path planning method (see section 4.5) is utilized to solve such a problem. Figure 55 shows the data flow between the MFS, CAS, and automated door controller. The XML format is used in message coding, and this message is converted into a single string with the UTF8 Encoding format. The message length is embedded in the message header before being sent, and is used to make sure that both socket sides receive the complete message. The command format between the MFS and CAS is listed as tables in Appendix B.

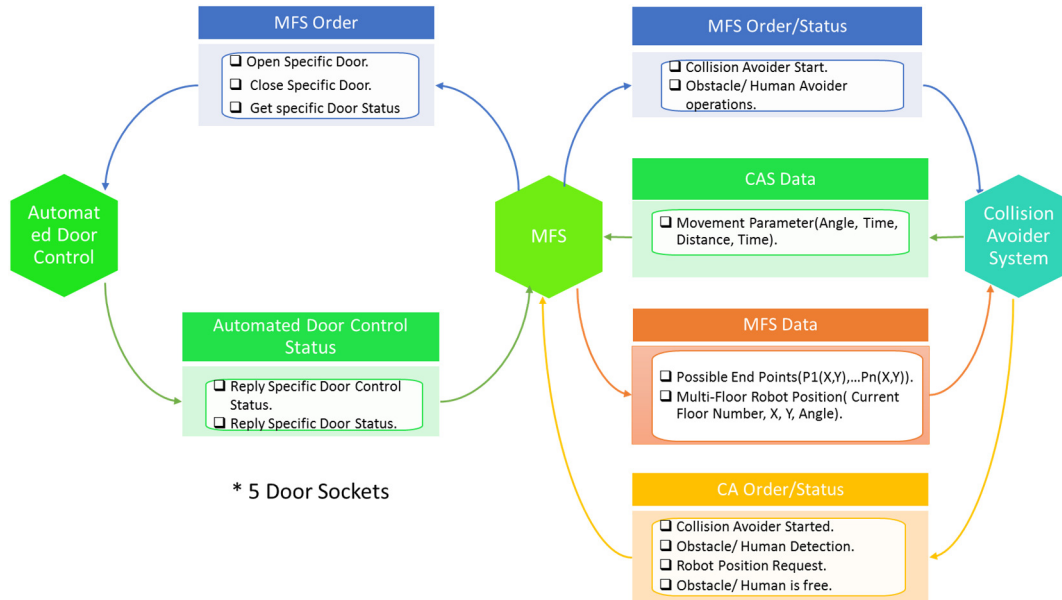


Figure 55 Data Flow for Automated Doors Control and CAS

As a summary, the client-server connection architecture module (asynchronous socket) is established to control the interaction of each sub-system with the MFS over Ethernet. TCP/IP-based protocol is utilized due to its reliability, expandability, and flexibility. Figure 56 demonstrates the graphical user interface of the developed sockets inside the MFS. From this

figure it can be seen, the server socket which connects the MFS with the RRC level, and the client sockets which are: A) motion and power control socket: to get the robot sensor readings (Battery voltage and temperature, ultrasonic distance) and execute the movement orders (move straight/rotation). B) Five automated door controllers' sockets: to send the door open/close orders. C) Elevator Handler System socket: to request for buttons position/floor number. D) Collision avoidance system socket: to realize the human robot interaction function. E) Robot arm kinematic module socket: to send the grasping/placing/pushing orders and wait for the operations acknowledge in a defined time. F) Automated elevator socket connection: to call the elevator to a specific floor. And G) socket connection to StarGazer module to get the raw data directly. The MFS manages all these sockets to interact with different parts of a mobile robot for labwares transportation tasks in automated laboratories. The other side of the communication module is established in the related sub-system where a client socket is adopted in RRC level [154] and server sockets are created in: elevator handler system, robot arm kinematic module [169], and collision avoidance system [168].

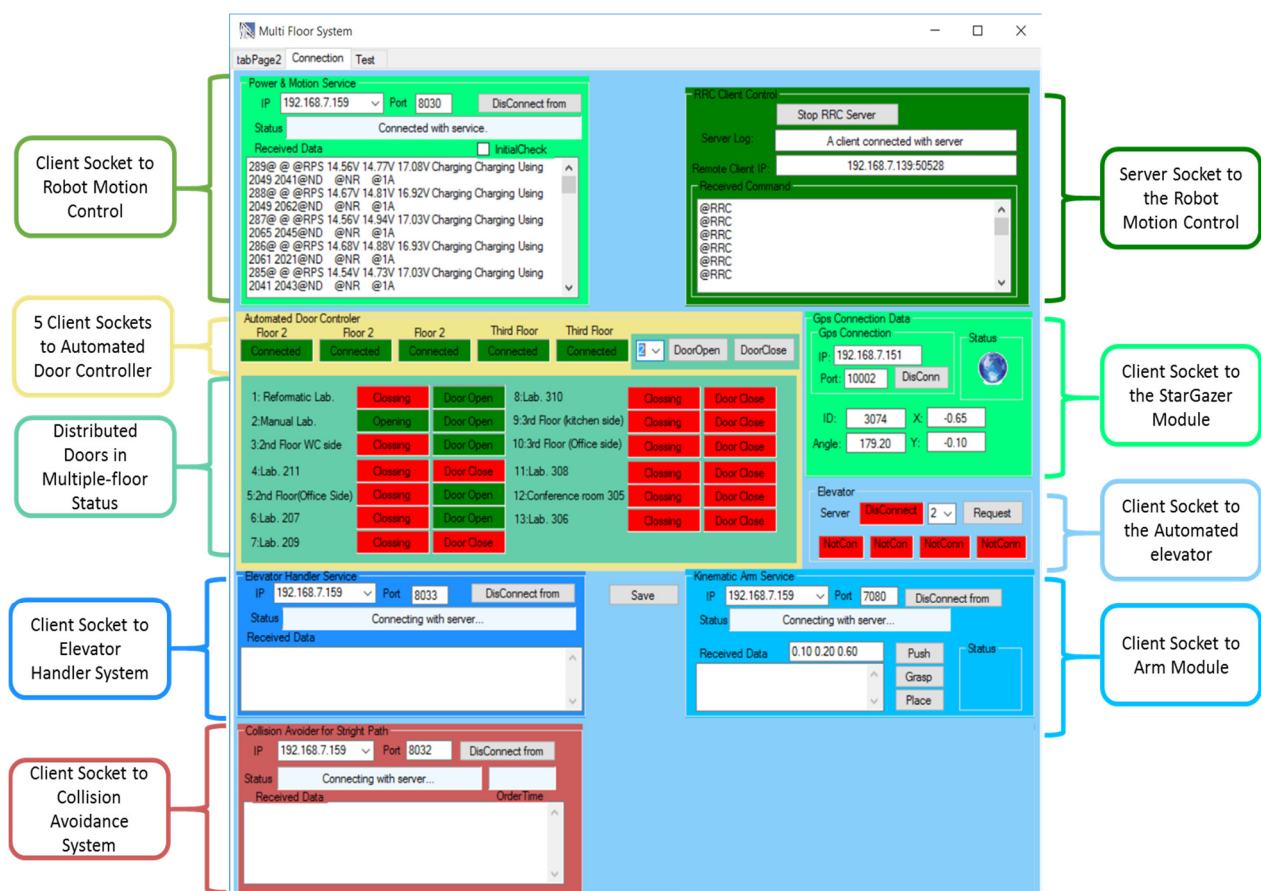


Figure 56 The developed socket connection to all sub-system

### 4.2.9 Experimental Results

The experiments described in this section were implemented with an H20 mobile robot in the laboratories at CELISCA (Rostock, Germany). The robot's linear velocity is 0.166 m/s and the angular velocity is 0.29 rad/sec. The landmarks were installed over the important points to realize high precision position estimation, with an SGM accuracy of  $\pm 1$ mm from the landmark's center area.

The first set of experiments were conducted in order to evaluate the performance of the backbone path planning method, the relative map, localization based on the relative map, the communication system, the transportation management system, and finally the movement core unit. A transportation task between laboratories distributed among different floors was executed twenty-five times in the CELISCA building. Figure 57 shows the multi-floor GUI (see Appendix C for more detailed). Figure 58 illustrates the path created by the robot during its movement between the 2nd and 3rd floors, including the elevator, in these real life sciences laboratories. Figure 59 shows the H20 mobile robot while executing the desired transportation task. During this experiment, the IMADCS (see Figure 60) was also examined, since four automatic doors distributed among different floors had to be handled. Only one connection has been lost to the automated door from 160 orders send to it. The ultrasonic sensor was used to detect the door's open or closed status. In four out of the twenty-five experiments, the robot required human assistance to complete its task due to incorrect SGM readings in a complex area when a reflection of a landmark appeared on the elevator's glass walls, as shown in Figure 61. This experiment was performed only twenty-five times due to the occurrence of this error which had to be dealt with, and the SGM reading error handler system is explained in section 4.3 below. The average transportation time to navigate through an estimated distance of 75 meters was 15:33 (mm:ss) in the range (13:22<->17:43).

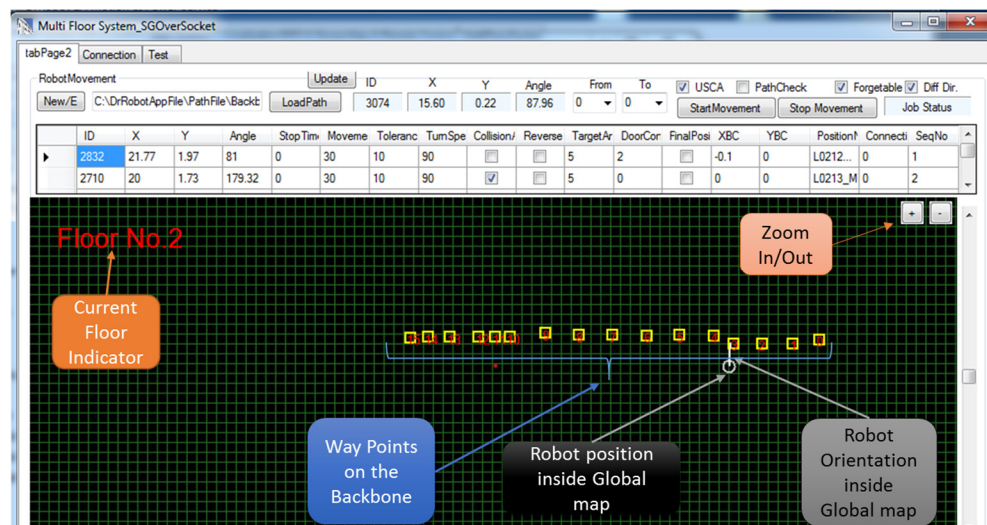


Figure 57 Multi-floor main GUI



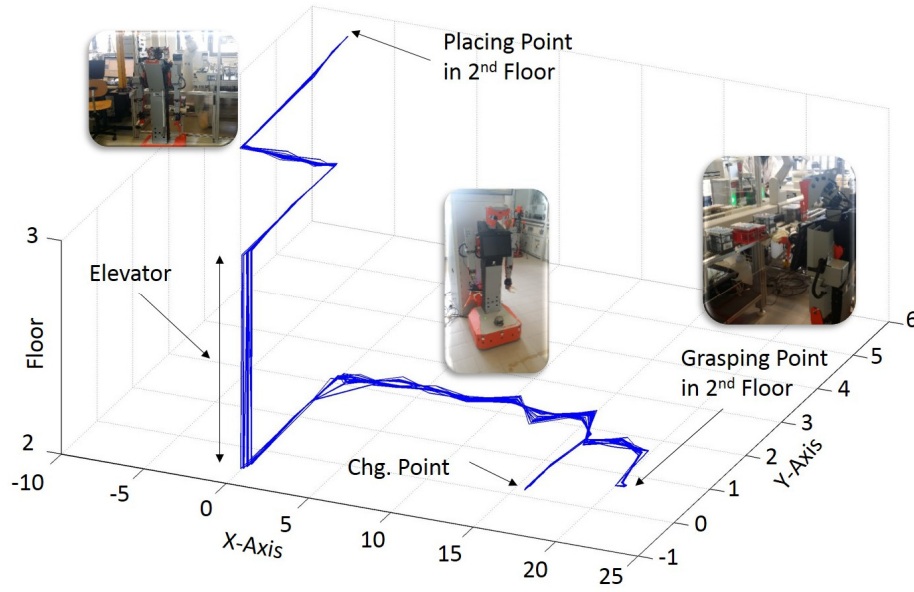


Figure 58 Real mobile robot transportation path on multiple floors (m)

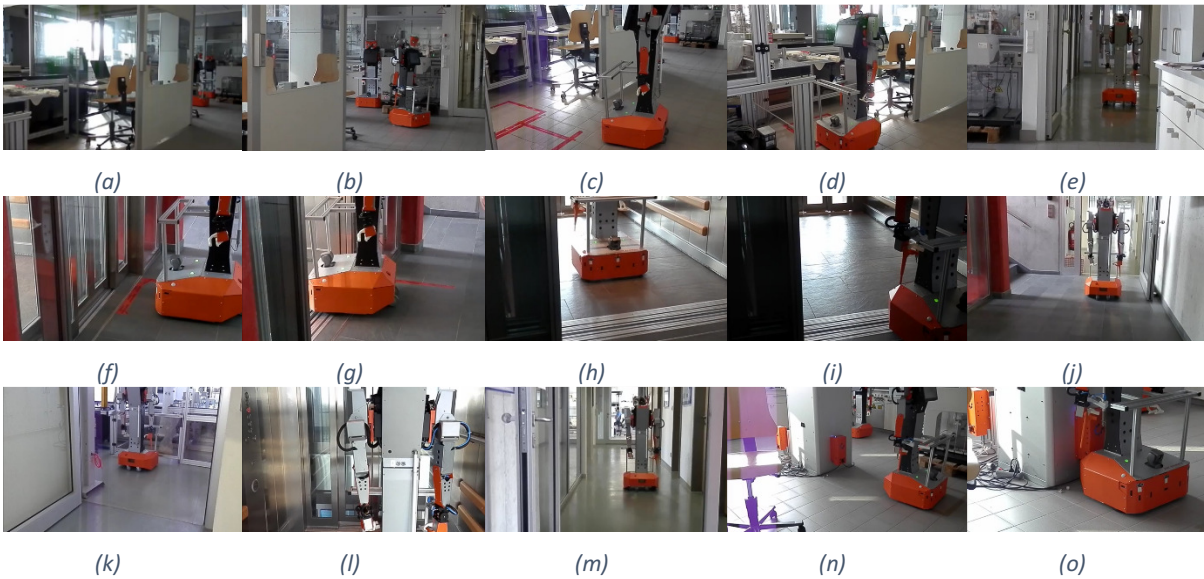


Figure 59 Execution of transportation task in multi-floor environment:

(a) robot at charge station; (b) leaving the charge station towards the current floor essential path until the grasping station is reached (c); (d) robot executes the grasping order, returns to the essential path to reach the elevator (e), and waits until the elevator door is open (f); robot enters the elevator and leaves it once the destination floor is reached (g) - (i); moving along the destination essential path towards the placing station (j), (k). After finishing the transportation task, the mobile robot returns to the charging station via the elevator, with the charging station essential path shown (l), (m); the charging procedure includes rotation and movement backwards until contact is made with the charging station (n), (o).

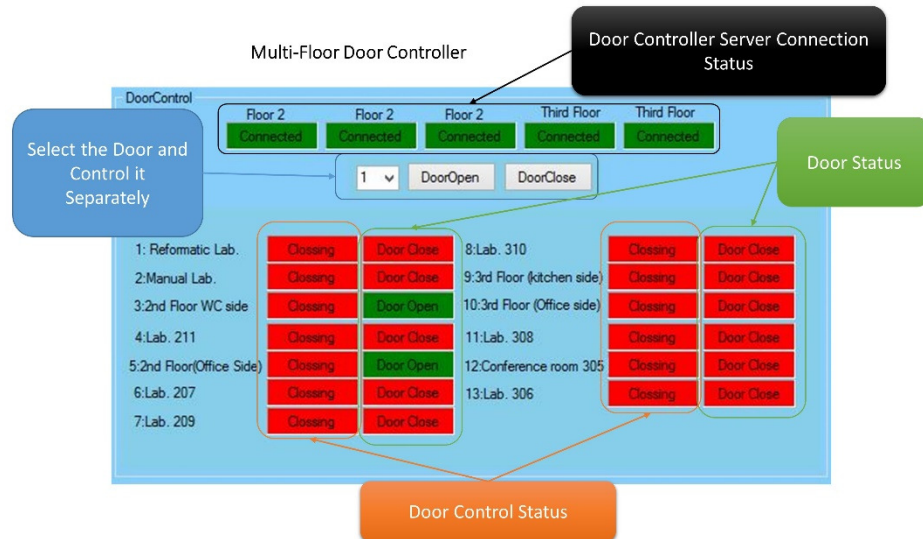


Figure 60 A GUI of the IMADCS

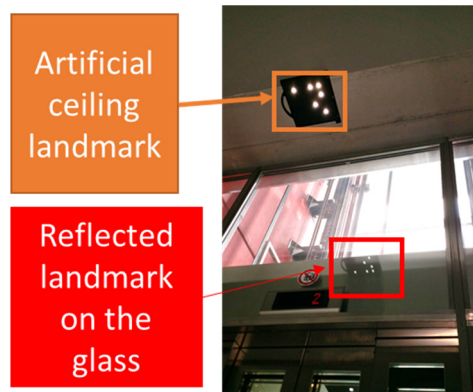


Figure 61 Reflective ceiling landmark on the glassy wall

The second experiment was conducted to determine the repeatability of the important points (grasping, placing, and charging) in the transportation path. Repeatability is defined as the ability of the mobile robot to reach the same position over a period of time, while accuracy is the difference between the point reached and the optimal point. To acquire the robot's position, a paper scale was fixed in the laboratory floor for both x and y axes as shown in Figure 62 for the grasping, placing, and charging points. The middle of this scale indicates the expected robot position and the initial reference get from multi-floor localization system. During the experiment, the robot's position on the scale is recorded for both x and y axes in the important points, and then the actual position is added to the measurement collected from the scale for accuracy analysis. The scale on the floor measurement technique is more accurate than SGM, which has a tolerance equal to 2mm in the center area of the landmark and 2cm in the edge of the landmark zone. However, the measurement based on the fixed floor scale also depends on the viewing angle of the observer. This experiment has been performed in the multi-floor environment fifty times, and each time the robot moved from

the charging station towards the grasping position, then moved until it reached the placing position, and returned back to the charging station. This experiment had a success rate of 100% in reaching the important points and then returning to the robot's initial position at the charging station. Figure 63 clarifies the repeatability of the mobile robot's movement to the important stations in the transportation path. In the figure it can be seen that the maximum repeatability range occurs in the grasping position. The complicated and narrow area involved caused a lower mapping accuracy in this position than in the other positions tested. The accuracy of achieving the elevator entry button position was also investigated as shown in Figure 63d. The accuracy, mean and standard deviation for each of these points are listed in Table 2.

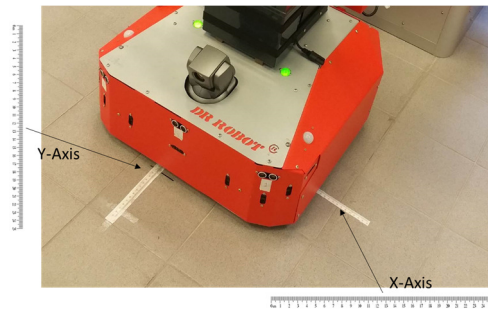


Figure 62 Ruler scale for robot position measurement

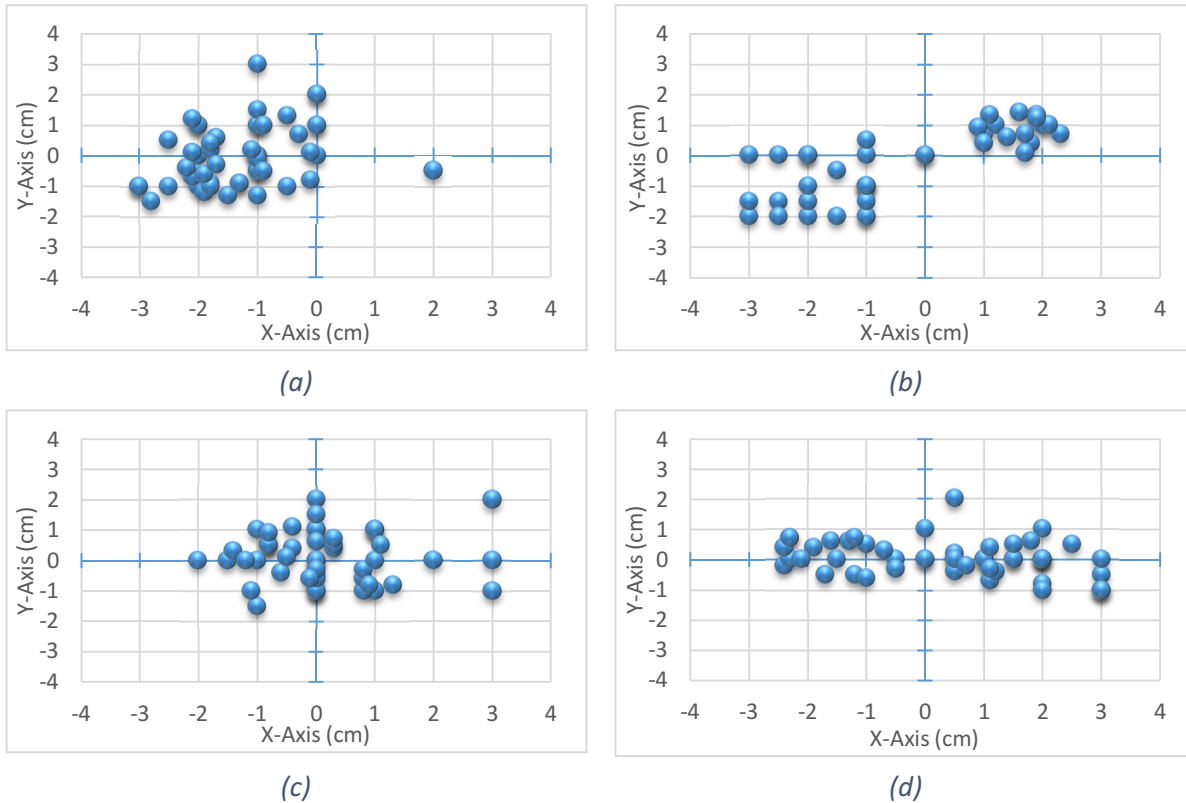


Figure 63 Robot movement position error: (a) grasping; (b) placing; (c) charging station; (d) entry button

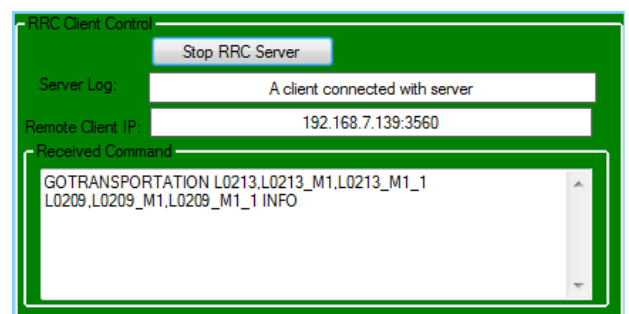


Table 2 Accuracy, mean and standard deviation of important robot positions.

	Axis	Target	Accuracy	Mean	S.D	Repeatability	Unit
Grasping point	X	2,060	1.28	2,058.71	0.95	±2.5	cm
	Y	56	-0.39	56.39	0.92	±2.25	cm
Placing point	X	-655	-1.58	-653.41	0.67	±1.5	cm
	Y	540	-0.90	540.90	0.65	±1.25	cm
Charging point	X	1559	-1.45	1560.49	0.95	±2	cm
	Y	33	0.06	32.94	0.58	±1.5	cm
Elevator entry point	X	-14	-0.32	-13.68	1.05	±2.5	cm
	Y	95	-0.162	95.16	0.82	±1.75	cm

The final experiment tested transportation tasks on a single floor to validate the integration of the MFS with the multi-robot management level (RRC). Once the RRC is working, the connected robots are listed as shown in Figure 64a. The RRC sends the transportation order to one of the available robots that has the highest charge status to execute the requested task. The MFS of the chosen robot receives this order as raw data (see Figure 64b), then the ITMS parses it to receive the grasping/placing position as X, Y and floor number. The execution of this task starts from the charging station, navigates towards point 1 (grasping), and is then directed to the placing position (point 2) and finally returns to the charging station. This experiment was repeated fifty times with a success rate of 100% in receiving the order from the higher level, controlling the automatic doors in the robot's path, and navigating to the grasping, placing, and charging positions. The path generated for the mobile robot's movements is shown in Figure 65. The transportation task takes around 9 minutes on average.

Robot ID	Robot IP	Connection Status	Transportation Status
1A	192.168.7.159	Connected	NoTask
2B	192.168.7.168	DisConnected	NoTask
3C	192.168.7.178	DisConnected	NoTask
4D	192.168.7.188	Connected	NoTask
5E	192.168.7.198	DisConnected	NoTask



(a) RRC GUI, connected robots;

(b) MFS GUI, transportation order received from RRC.

Figure 64 MFS<->RRC socket communication

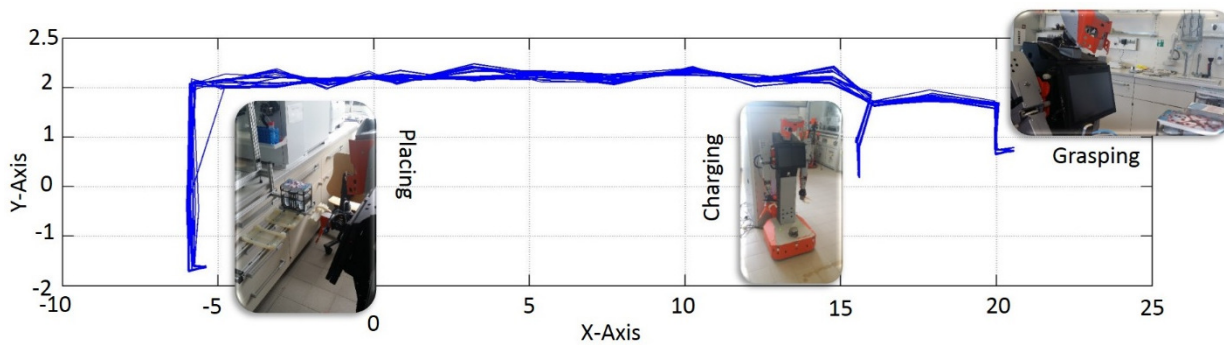


Figure 65 Real mobile robot transportation path on single floor (m)

### 4.3 Vision Landmarks Reader Error Management

The SGM reading collects errors which occur while the robot is moving in a complex environment which, for example, includes reflection and direct sunlight. The error handling for the SGM was developed to overcome these problems of incorrect reading. Figure 66 shows the system, which mainly consists of the error handling core which is responsible for analyzing input error and choosing the actions required for error handling. Two actions can be taken in cases of error detection. Firstly, the SGM could repeat the ID readings ten times to eliminate the wrong ID. Secondly, the robot could move backwards or forwards till realizing the right ID.

The system detects a wrong ID reading if the SGM reads a non-stored landmark ID or if the calculated distance to reach the next position is larger than the normal distance, such as when the reading is 10m while the specified distance between waypoints is 3m on average.

The error handling core continually monitors these two expected input errors. If the system detects any error, the first scenario is to keep reading until the correct ID is received. Usually, the SGM error handling works well by suspending the robot's movement until the right reading is received, but this may take a long time or might even fail, especially in the elevator environment where a lot of light reflection occurs. These delays may significantly affect the whole time taken to execute transportation tasks. This problem was resolved by specifying the number of repeat attempts as ten successive wrong readings. The second scenario described above starts immediately when the first has failed. The error handling system controls the robot in moving in the backward or forward direction until the correct ID is received five times as maximum range, in each time repeat the first action until get the right ID.

Figure 67 shows the implemented method for correction by movement, while the error handling core scenarios are shown in Figure 68.

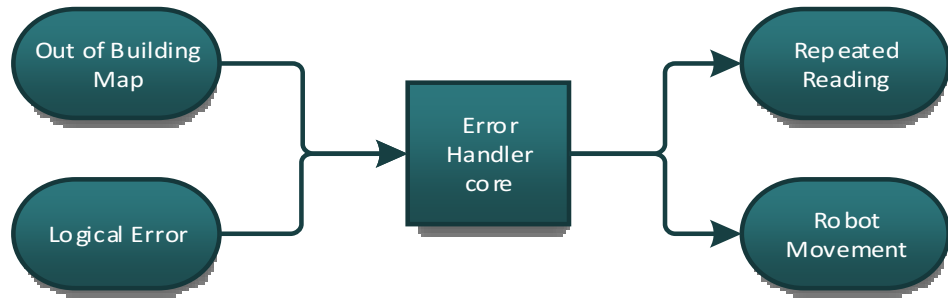


Figure 66 Vision landmark reader error handling

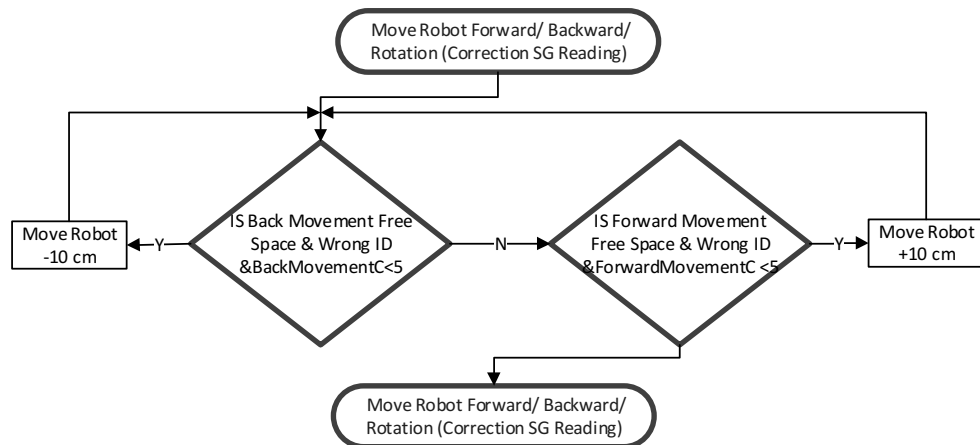


Figure 67 Vision landmarks reader error movement scenario

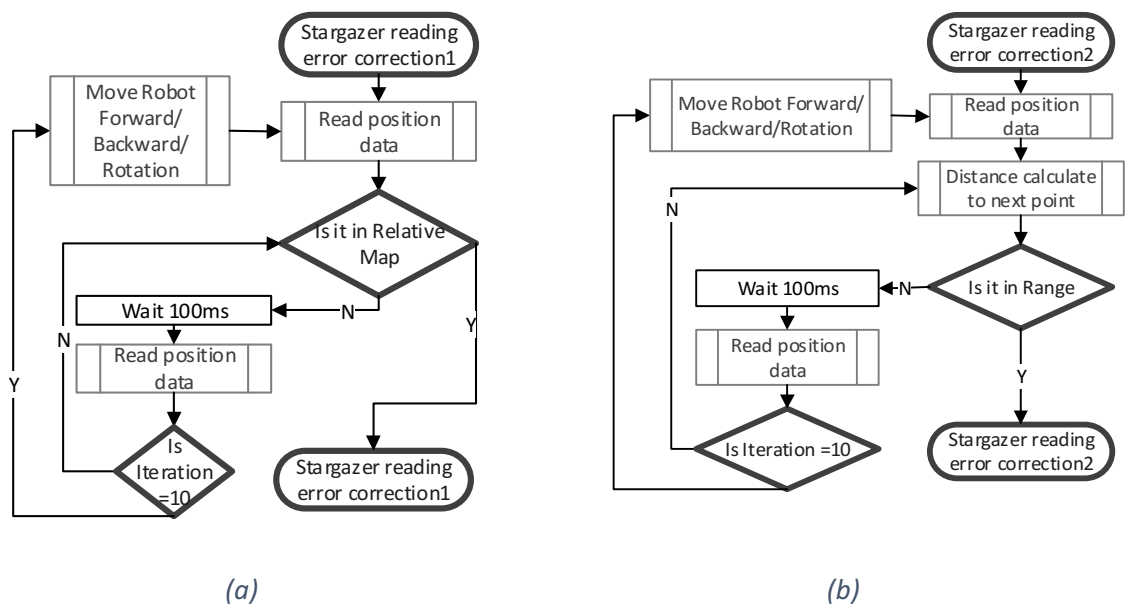


Figure 68 Vision landmarks reader error management: (a) read it ID out of relative map; (b) read it ID on relative map

## 4.4 Fine Position Correction Method

In laboratory automation, a high transportation speed over large areas is important to minimize the required time for the whole laboratories operation. But for an H20 mobile robot to move at such speeds adds challenges related to decreased movement accuracy. This would directly affect the arm manipulation performance as the H20 robot arm has a limited workspace, especially in multiple labware arrangement where in 2 cm robot error position the arm can handle 4 pieces of labware from the same position [169]. However, the H20 robot arm fails to grasp or place one of these labwares if the error in the robot's position is increased by 1 cm. The robot linear velocity increased to 0.2m/s and the fine function takes the responsibility of error correction, the angular velocity adapted with the required rotation angle degree to get balance between speed and accuracy. For example, if the rotational angle required is more than 30 degrees then the highest rotation speed of 0.34 rad/s is used, and for a lower angle the angular speed will decrease to achieve the highest accuracy. From the results of the experiment shown in section 4.2.9, it can be seen that the maximum repeatability range is in the grasping position, reaching  $\pm 2.5$  cm in x axis and  $\pm 2.25$  in y axis. The complicated and narrow area for the grasping position with high robot speed causes higher positional error in this position compared to the other positions tested.

The fine positioning method was developed to overcome this problem and to achieve a higher positional accuracy. This method uses two techniques. Firstly, the robot's speed is controlled in order to minimize movement error caused by wheel slip and to give more time for the motor encoder to be read and updated. Secondly, a position correction is added in the X or Y direction (depending on the latest movement) as shown in Figure 69. The H20 robot has a differential driver, and so it is not easy to correct its position in the right/left position. Thus the fine positioning method records the direction of the last movement and utilizes this direction to correct the robot's position until reaching to the motor accuracy limitation which is 1cm. The validation of fine position correction is demonstrated in section 4.5.3.

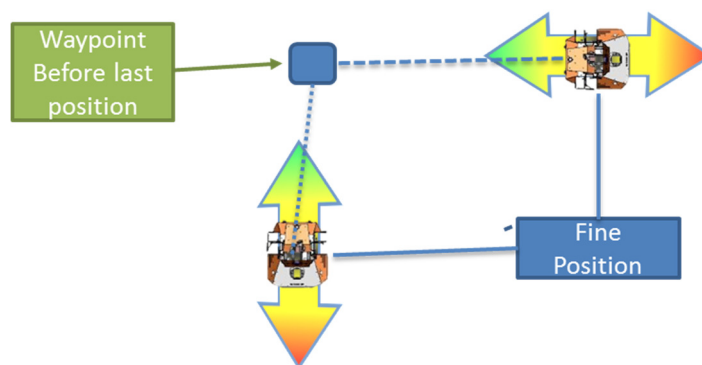


Figure 69 Fine position correction based on the last movement direction

## 4.5 Hybrid Path Planning

The backbone method is an easily implemented and rapid path planning method, but it cannot deal with unexpected changes in the environment with dynamic obstacles or other complex cases such as failure in one of its tasks, CAS movement, or blocked paths. Thus, to retain its advantages and eliminate the limitations, a hybrid backbone-Floyd path planning method was developed. The criterion used to switch from the backbone method to hybrid path planning is the unavailability of the backbone method because of failure in one of its tasks, after CAS movements, a blocked path, or failed arm manipulation. The Floyd algorithm is employed to re-plan the global path based on the robot's current positions and intermediate destination positions for grasping, placing, and charging when the global path based on the backbone method is unavailable. After completion, control is returned to the backbone method to complete the transportation task towards the final destination.

### 4.5.1 Floyd Method

The efficiency and simplicity of the Floyd algorithm has made it very popular in engineering applications. Its working processes can be explained as follows. Firstly, two square matrices are initialized where the distance matrix (DM) lists the distance between each two nodes and the sequence matrix (SM) stores the intermediate nodes between source and destination nodes. Then the sequence matrix is updated by considering the intermediate nodes, which are any nodes existing in the path between the first (source) node and the last (destination) node. The updating process is reiterated to find the intermediate nodes between each pair of nodes. The detailed computational process of the Floyd algorithm is given in Appendix A.2. Finally, the DM will contain the distances between all pairs of nodes, which will be represented in the SM.

Since the Floyd algorithm works with an indirect graph (a path map developed as bi-directional path), the time taken is important in laboratory automation. The Floyd algorithm can be optimized to minimize the calculation time by reducing the number of comparison tasks by half. The element of the distance matrix above the diagonal in the indirect graph has the same value as that under it, and this feature is used to optimize the calculation by applying the Floyd only to the element over the diagonal. The result is then duplicated for the elements under the diagonal for both the distance and sequence matrices. The optimized Floyd algorithm for indirect graph clarifies in Appendix A.3.

An experiment was conducted to validate the performance of the optimized Floyd method compared to the classical version in the developed MFS. This experiment was repeated ten times for each kind, with the same multi-floor path map, and the time taken was recorded. The time cost of each method is shown in Figure 70 while Table 3 gives the cost average time.

The variations in time taken by the algorithm is because the Windows operating system cannot run the application in real time due to multi-task. In general, it can be noted that the optimized Floyd algorithm reduces the required time by approximately 56 %.

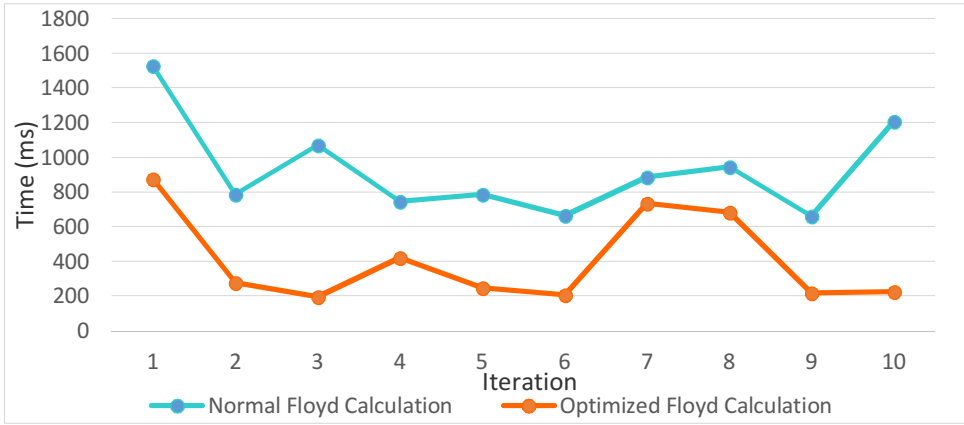


Figure 70 Floyd algorithm time cost

Table 3 Floyd time cost comparison

Floyd	It1	It2	It3	It4	It5	It6	It7	It8	It9	It10	Average
Normal (ms)	1,53	788	1,07	749	787	667	888	947	664	1,21	930.3
Optimized (ms)	877	280	196	424	251	209	736	685	218	227	<b>410.3</b>

#### 4.5.2 The Developed Internal Transportation Management System (ITMS)

The ITMS is equipped with a smart management technique to select the appropriate backbone or Floyd path planning method. As demonstrated in *Figure 71*, the ITMS implements the backbone method by dividing the incoming transporting task into a group of sub-tasks and continuously monitors the execution of these tasks. If the ITMS detects a failure in the execution of any task related to unexpected obstacles or any other reason, it enables the Floyd path planning method to build a new path related to the current destination point (grasp, place or charge) and controls the robot in reaching that position. After completion, the ITMS returns control to the backbone method. The reactive path planning strategy clarified as algorithm in Appendix A.4. The ITMS also monitors the execution of the functions embedded in the RBC (RAKM, EHS, IMADCS, and CAS) to predict possible failure cases during the transportation process, which gives the ability to higher levels to initiate the appropriate procedures earlier. If a failure is detected, the ITMS stops the current transportation task, informs the RRC, and controls the robot until the charging station is reached. The task 0 is added to cause the robot to leave the charging station smoothly and the last task (10) is used to align the robot's position accurately with respect to the charging plug. Task 10 works in cases where the robot reaches the charging station but is not aligned with the charging plug,

where the motor encoder is used to move forward by 50cm and then to use the multi-floor movement core with the charging station as the goal.

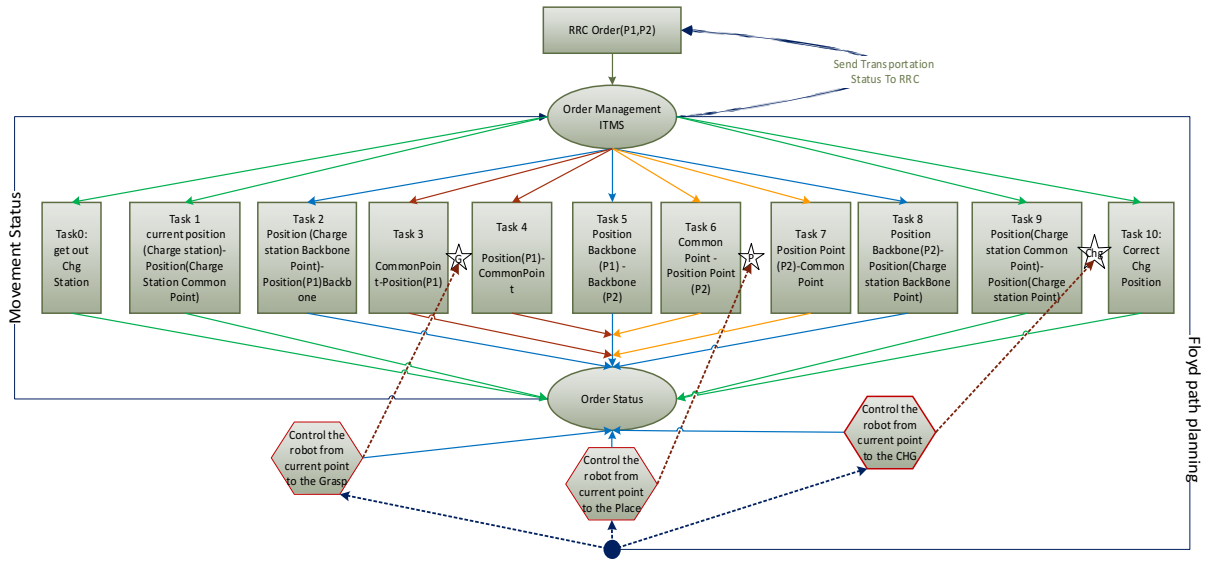


Figure 71. Hybrid Backbone-Floyd method

Since the CAS generates its own path to avoid obstacles, the robot will be in an undefined point at the end of the CAS task. In this case, the ITMS starts with special procedures to find the shortest path from the current point to the destination (see Figure 72) as follows. The current robot floor and x,y position is received from the MFS based on SGM data, and the three waypoints nearest to the current robot position are found. Then the robot point distance (DRP) is calculated, which is the distance between the robot's current position and each waypoint. The Floyd algorithm is used to find the length between each one of these waypoints and the destination point (LPD), and the total distance from the current robot position to the distance ( $DRP + LPD$ ) is calculated for each waypoint. The shortest distance is chosen, and this technique can also be utilized if for any reason the robot is in an unknown position. The developed Hybrid path planning method was published in [170].

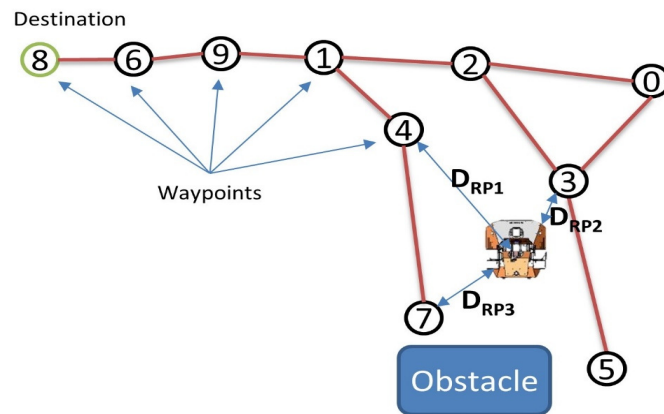


Figure 72 Shortest Path Procedure

### 4.5.3 Hybrid Method Experiment

Two experiments were conducted to validate the efficiency of the system and the new planning strategy. The details of each experiment are given below.

The first experiment was performed to validate the efficiency of the hybrid backbone-Floyd path planning method and the SGM error handler strategy, and to assess improvements in the speed of executing transportation tasks on the second and third floors at Celisca. The robot speed was 0.2 m/s while the angular speed was 0.34 rad/s. An H20 mobile robot was employed for the multi-floor transportation task. This test was repeated 40 times, and each time the robot navigated an estimated distance of 75 meters through an area containing 51 waypoints. The backbone path planning method was applied to plan the multi-floor transportation task four times, then deliberate errors were made four times in each task to force the ITMS to select the Floyd algorithm to find the shortest path to the next important grasping, placing or charging station. During this experiment, SGM error handling was run 10 times to correct the SGM reading. The hybrid path planning method succeeded in planning the path efficiently in all scenarios and the SGM error handling approach showed high performance in receiving the correct readings in the complex building structure with reflected light. Figure 73 shows the repeated path in multi-floors environment.

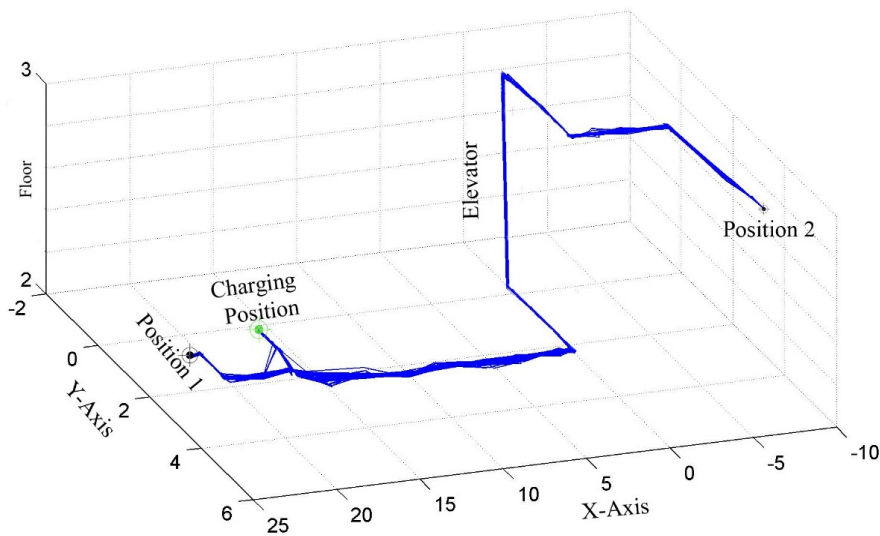


Figure 73 Path generated by robot movement in multi-floor environment (m)

**Error! Not a valid bookmark self-reference.** presents a comparison between the original speeds (0.166m/s, 0.29 rad/s) and those with the new error compensation technique (0.2 m/s, 0.34 rad/s) showing the minimum, maximum, and average times for all of these transportation operations. It is clear that the time reduction achieved was 246 seconds on average. The robot works in laboratories transportation; thus it must work with appropriate speed for labwares balance preservation during transportation process. Additionally, the

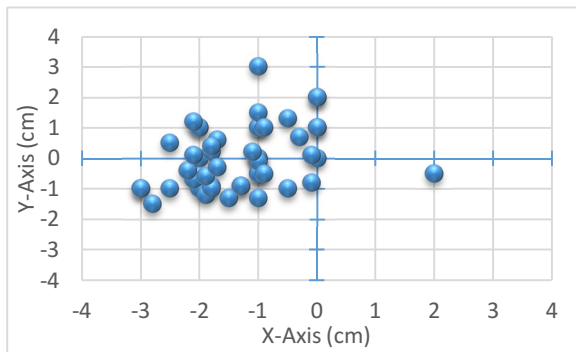


accuracy of robot movement decreases with increasing speed, since the robot hardware needs a certain time to receive the optical encoder measurements.

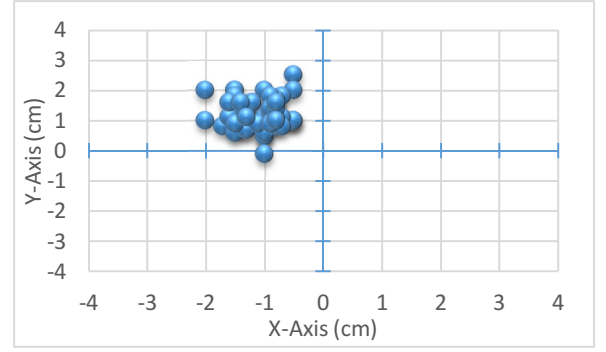
*Table 4 Time consumption for mobile robot transportation in multi-floor environment.*

	Min(MM:SS)	Max(MM:SS)	Average(MM:SS)
<b>Time required (Section 4.2.9)</b>	13:22	17:43	15:33
<b>Time required</b>	10:46	12:44	11:26

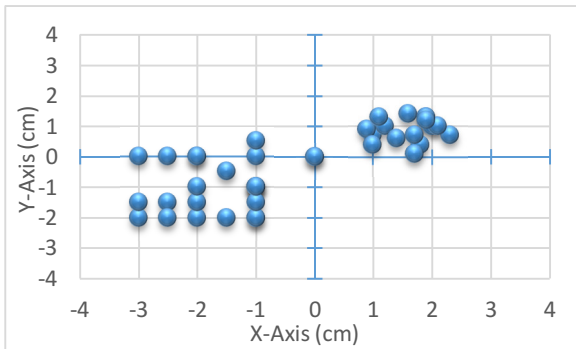
The second experiment was conducted to determine the repeatability of the different grasping, placing and charging points after employing the fine position correction method during the transportation task. This test has been performed on a single floor fifty times, and each time the robot moves from the charging station towards the grasping station, then moves until reaching the placing station, and finally returns to the charging station. In this experiment, a 100% success rate was achieved. Figure 74 clarifies the repeatability of the mobile robot's movement to the important stations in the transportation path. To clear the effectiveness of the fine position method in term of the repeatability and accuracy, the results from section 4.2.9 are also included here for comparison. In Figure 74a it can be noted that the precision and repeatability results show great improvements in the grasping position. In this position, the repeatability range has decreased from 5 cm to 1.5 cm in x axis.



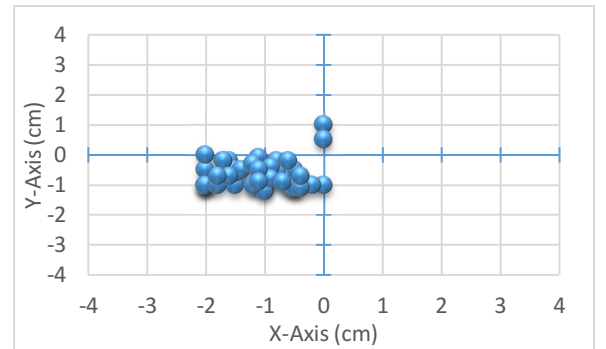
*(a) grasping position without Fine function*



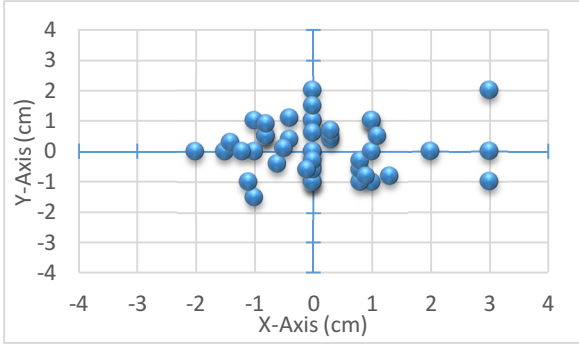
*(b) grasping position with Fine function*



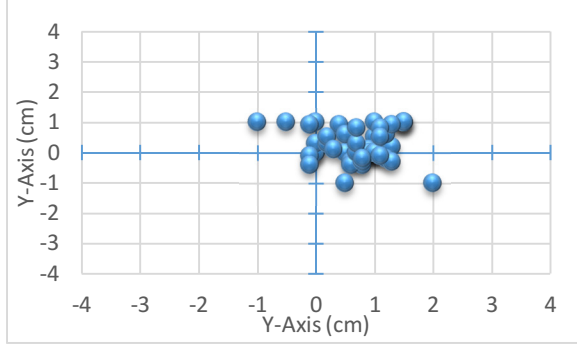
*(c) placing position without Fine function*



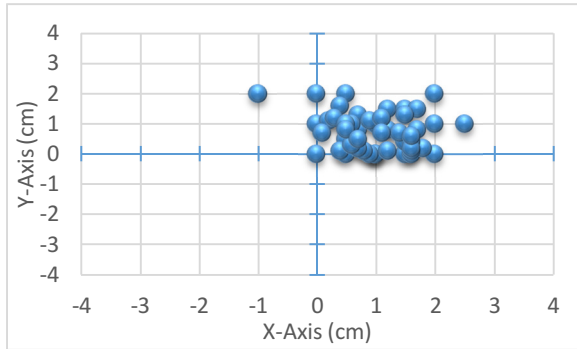
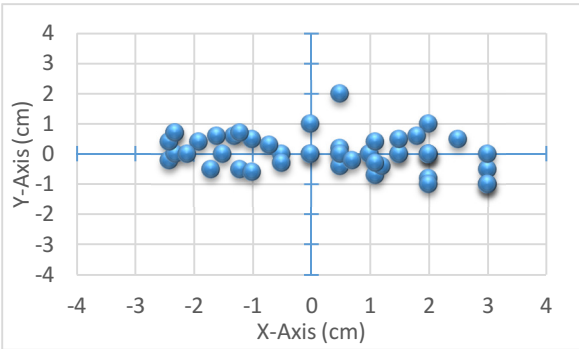
*(d) placing position with Fine function*



(e) charging position without Fine function



(f) charging position with Fine function



(g) elevator entry position without Fine function (h) elevator entry position with Fine function works

Figure 74 Contrast experiment for fine method

Table 5 Comparison of methods developed for navigation to reach important points.

			without				With Fine Method				Unit
	Axis	Optimal	Accuracy	Mean	S.D	Repeatability	Accuracy	Mean	S.D	Repeatability	
Grasping point	X	2,060	1.28	2,058.7	0.95	±2.5	1.096	2,058.	0.38	±0.75	cm
	Y	56	-0.39	56.39	0.92	±2.25	-1.21	57.21	0.51	±1.3	cm
Placing point	X	-655	-1.58	-653.42	0.67	±1.5	-1.088	-653.9	0.58	±1	cm
	Y	540	-0.94	540.90	0.65	±1.25	-0.65	540.6	0.43	±1.1	cm
Charging point	X	1,559	-1.45	1,560.45	0.95	±2	-1.01	1,560	0.68	±1.75	cm
	Y	33	0.06	32.94	0.58	±1.5	-0.77	33.77	0.64	±1	cm
Elevator entry	X	-14	-0.32	-13.68	1.05	±2.5	-0.68	-13.3	0.59	±1.5	cm
	Y	95	-0.16	95.16	0.82	±1.75	-0.25	95.25	0.53	±1	cm

## 4.6 Conclusion

In this chapter, the following conclusions are drawn from the innovative work conducted:

- New mapping, localization, and path planning approaches are developed with other related systems, including an IMADCS, IBCMS, and an error management system for a multi-floor navigation system for multiple H2O mobile robots.
  - a) Two methods are developed for mapping. Firstly, relative mapping is responsible for generating a global map in a multiple-floor environment with high accuracy and flexibility for expansion into any number of floors. This map gives a three-dimensional representation with a unique reference for all floors. Mapping methods utilize the SGM as the landmark reader to gather the information required. The path map consists of a set of rich information waypoints, the position of which is related to the relative map. Each waypoint has information about the incoming door number, the robot's maximum speed, acceptance tolerance, and movement direction, enabling the dynamic collision avoidance and fine positioning functions. These waypoints are then utilized in the path planning stage to find obstacle-free transportation paths.
  - b) An indoor localization stage for multiple robots is used to extract the positions of robots on multiple floors as x, y co-ordinates and floor numbers in the global map for any number of robots. The incoming data from the landmark reader is translated into positions based on the relative map to find a robot's current floor from the landmark ID and the related x, y position.
- A multi-segmented short path method named backbone path planning is presented to optimize the number of static transportation paths required with a dynamic selection mechanism. This is a fast and easy implementation path planning method, but it cannot handle changes in the environment such as dynamic obstacles. Thus, a hybrid Backbone-Floyd path planning method is presented to retain the advantages of the backbone method and to overcome its limitations. The criterion used to change from backbone to Floyd path planning is the unavailability of the backbone method because of a failure in one of its tasks, after CAS random movements, or the occurrence of a blocked path. Also Floyd algorithm is used after failing in robot arm grasping and placing operation for directing the robot movements towards the charging station which save the life science automation laboratories time by informing the scheduler level (HWMS) earlier to take an advance procedure and make robot ready for next transportation task. The Floyd algorithm is utilized to plan the path from the current position to the intermediate grasping, placing, or charging destination according to the status of the ITMS status. For example, if the

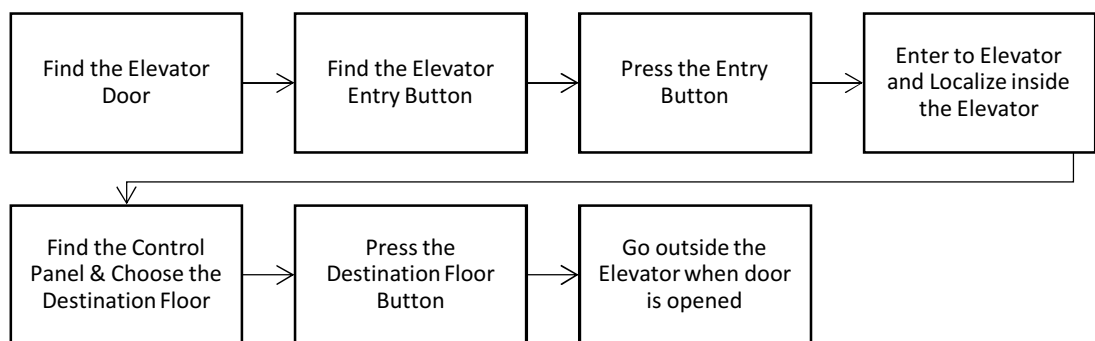
backbone is not available before the robot reaches the grasping station, the ITMS enables the Floyd algorithm to plan the path from the current position to the grasping station, and then path planning control is returned to the backbone method. In addition to selecting the appropriate path planning method and specifying the intermediate destination for the Floyd algorithm, the ITMS is also used to implement the backbone tasks when it is chosen and to manage the multiple source and destination planned path execution operation. Finally, a new movement core is developed to execute the planned paths.

- An indoor management automated door controller system (IMADCS) is developed to manage the control of thirteen automated laboratory doors distributed over multiple floors. ADAM modules are installed around the building to receive the IMADCS commands and to then forward them to the door controller through an Ethernet network.
- An accurate position error management system is developed to achieve highly accurate robot positioning suitable for the limited H2O robot arm workspace to adapt with multiple laboratory transportation tasks.
- A communication network is established for data exchange between the MFS and the RRC, RAKC, CAS, EHS, robot motion controller (RMC), and the automated door control system. In general, the MFS waits for orders from the RRC to start a transportation task. The arm controller waits for the order to grasp, place, or push with the relevant information from the MFS to perform the required manipulation. When the MFS receives an "ObstacleDetected" message from the CAS, it either cancels its request based on the current robot's position or pauses the transportation operation and executes a series of movement requests (distance in time, angle in time) until the path becomes free, and then the MFS utilizes the dynamic path planning method to regenerate a path to the destination. All these systems work inside the robot computer to achieve an intelligent robot movement system.
- A robot charging management system is developed to serve multiple robots in a multi-floor environment to monitor the robot batteries and maintain them in a fully charged state.

## Chapter 5 ELEVATOR HANDLING

### 5.1 Introduction

The elevator is the key for mobile robots to move between levels in multi-floor environments. At CELISCA, the elevator utilized is a glassy elevator and its shaft mainly consists of a metal frame with glass panels while the passenger cabin has two sides with metal walls while the other two have metal frames with glass panels. According to the literature survey in section 2.3.5, many issues must be handled for the robot to successfully interact with the elevator, as shown in Figure 75. The essential processes for the mobile robot to use the elevator are as follows. The elevator door must first be located, starting by finding the way to the elevator entrance based on a mapping program. Then the elevator entry button should be identified, using digital image processing to find the button landmark in pixel coordinates and extracting the depth information with the help of the Kinect sensor. This information is subsequently fed to the next stage of pressing the entry button, where the button coordinates data is sent to the arm's kinematic module which performs the required calculation to press the button and informs the MFS if the pressing operation is not possible. After this the robot must enter the elevator, moving a predefined distance into it after the door has opened, and then localizing itself based on recognizing a landmark installed inside the elevator. Next the control panel must be found and the destination floor chosen. Here the robot moves towards the control panel according to a ceiling landmark, finds the appropriate X,Y,Z coordinates and then feeds this information to the next stage. The required calculation is then performed to press the destination floor button, using the inverse kinematic module to determine the possibility of pressing it. Having pressed the button, the elevator door eventually opens and, after checking that the destination floor has been reached, the destination map is loaded and the robot finally leaves the elevator.



*Figure 75 Elevator handling sequential sub-process*

The transparent and reflective surfaces in a glassy elevator environment add challenging problems to the elevator operations due to the direct effect of sunlight on the vision sensor. In this chapter, the elevator handling system is divided into two parts. The outside elevator system handles the movement to the elevator, entry button detection, elevator entrance door status recognition; while the inside elevator system deals with robot localization inside the elevator, movement inside it, internal buttons detection, and current floor reading.

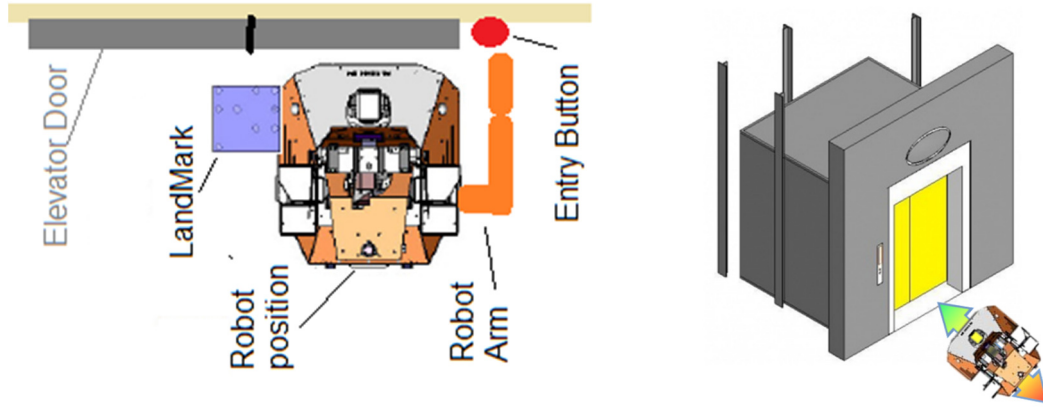
## **5.2 Outside Elevator**

This section covers what the robot requirements before entering the elevator. This includes the finding of the elevator entrance door, the recognition of the entry button, the determination of the real position, the transfer of this information to the RAKM, the detection of the elevator door's status, and finally going inside the elevator. All these aspects are explained in detail in the following sections.

### **5.2.1 Movement to Elevator Area**

The first step in elevator handling is the determination of the elevator zone. Then the required analysis is performed and processes are activated to reach the target destination. This stage is performed based on predefined waypoints on the map of the elevator area, which depend on the reflective artificial landmarks installed near the elevator in order for the robot to read the exact position and direction. Based on this information, the robot moves until it reaches the best position, as shown in Figure 76a. This movement is the basis for the next two stages which are entry button detection and elevator entrance door status recognition. The mobile robot is equipped with a dual arm utilized for the button pressing operation. For this purpose, a new end effector design is added to the arm and an inverse kinematic solution is applied for arm manipulation. These arms have a limited workspace, and thus it is necessary to control the robot until it reaches a predefined position with high accuracy so that the arm is able to work in its work space. The robot must locate itself in a specific position that allows the robot to press the entry button and to enter the elevator when it opens. The importance of the predefined position is related to minimizing the possibility of collision with the elevator door, especially after a labwares holder is added to the robot body which increases its length by 10 cm and its width by 8 cm. The arm workspace is increased or decreased according to the robot's position near the elevator, and the robot should be in front of the elevator entrance door within 29-31 cm according to the predefined position (2 cm workspace). This distance will allow the robot to press the entry button and enter the elevator when the door is opened. A hybrid method was used to achieve higher positional accuracy, which utilizes the fine correction function based on SGM (see Figure 76b) and a motor encoder correction based on

an ultrasonic sensor as follows. First the movement core is utilized until the predefined position is achieved. As reported in Table 2, the movement repeatability reached 5 cm for the x axis and 3.5 cm for the y axis. Then a fine method is used to achieve the predefined position with a repeatability of 3 cm for x and 2cm for y as shown in Table 5 based on the last movement direction. Finally, the ultrasonic sensor with motor encoder is employed to ensure that the robot reaches the exact position within a range of 1-2 cm by finding the range to the door.



*(a) Entry position movement idea*

*(b) Higher Accuracy Robot Positioning*

*Figure 76 Robot position outside elevator*

### 5.2.2 External Button Detection

As mentioned above, the glass elevator at CELISCA has a passenger cabin with two sides made of glass panels and metal frames, and the transparent surfaces add a big challenge for entry button detection, especially in sunny weather. To combat these difficulties a pre-processing calibration stage was applied using image statistics (minimum pixel values) to update the Kinect RGB camera hardware parameters in successive approximations which lead to the calibration of the camera hardware parameters (exposure time, sharpness, and saturation) to compensate for the influence of direct sunlight. Figure 77 demonstrates the calibration steps. The entry button detection technique uses computer vision and a depth sensor. It uses a Microsoft Kinect sensor to acquire the RGB image and depth information. Since the entry button and its panel are made from the same reflective material, the detection of the button is difficult to realize. Thus, a landmark has been installed around the entry button with a specific shape and color so as to enable an easy and applicable recognition process for the mobile robot. The elevator entry button detection process can be explained as follows. Firstly, the Kinect sensor is initialized with the required frame rate, image resolution, and depth resolution. Then an image is captured from the Kinect RGB sensor, and an RGB channel filter is used to remove the selected band of colors. Subsequently, patterns of connected pixels

(objects) are searched for inside the image, with candidates selected based on a specific width and height. After checking if the chosen candidates have a square shape with the specified acceptance distortion and color, a depth snapshot is collected from the Kinect sensor; then mapping is performed between the position in pixels and depth to extract the real X, Y, and Z coordinates. Finally, calibration is performed between the Kinect positions to the arm reference for the arm-pressing operation.

This method achieved a success rate of 88.5% in stable lighting conditions, as clarified in section 5.2.4, but the working environment has different lighting and sunlight conditions which may easily affect the detection process using the RGB color system. Thus, the success rate may be reduced significantly. This problem has been solved using HSL color representation which is more stable against changes in lighting conditions as explained later in the experimental section. The detection algorithm based on HSL color representation uses data on color, shape, and size to find the elevator entry button, and starts by searching for the object in the captured image. The process then applies the HSL filter to remove the background and retain the required color range for the button based on HSL color representation, extracts the edge of the detected object, finds the shape of the button based on these edges, and applies various filters related to shape such as size and shape distortion in order to increase the success rate. Finally mapping between the RGB and depth pixels is performed to derive the real and accurate Cartesian coordinates. The flow chart of button detection is shown in Figure 78.

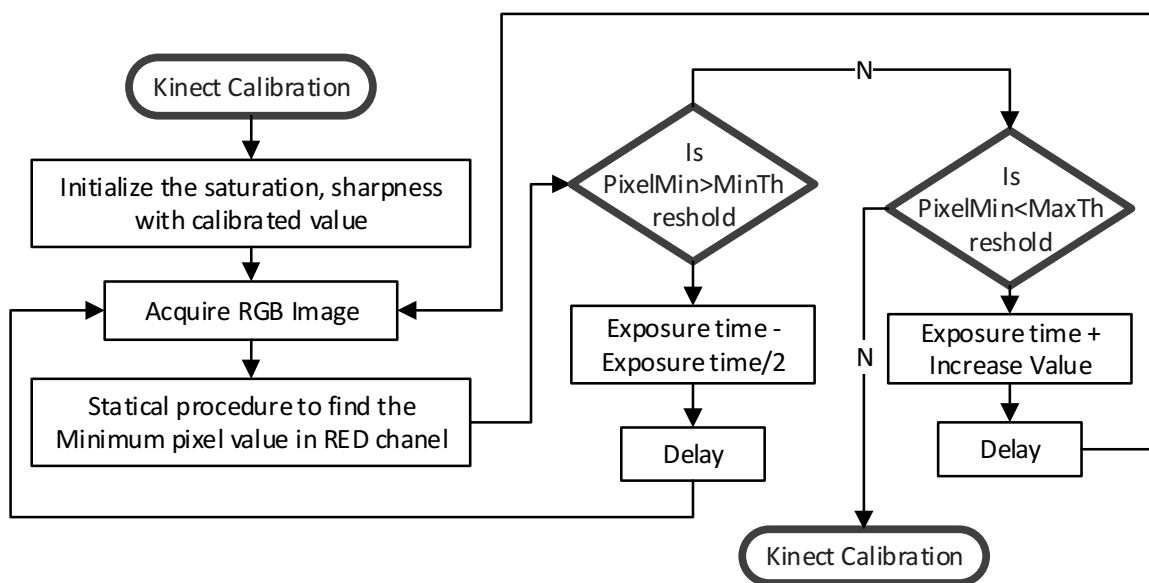
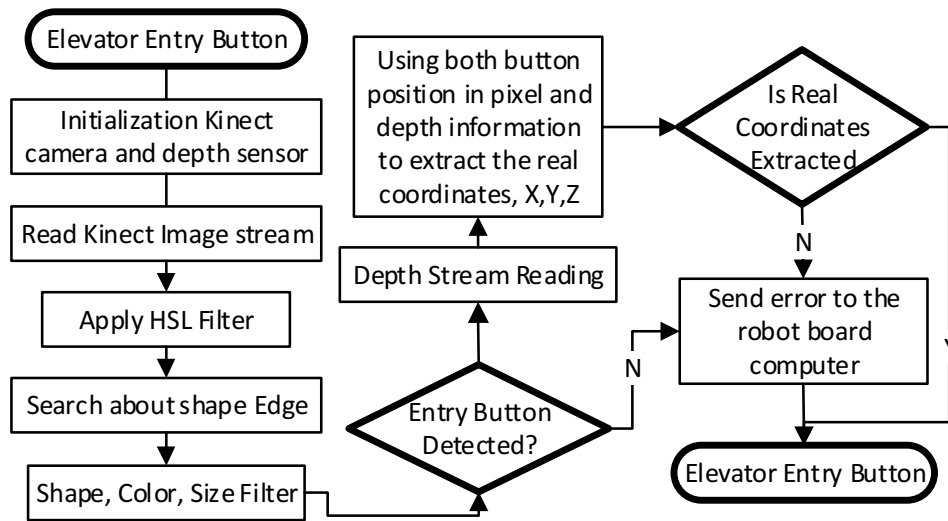


Figure 77 Kinect calibration stage





*Figure 78 HSL-based entry button detection*

From the experimental results described in section 5.2.4, it is clear that the HSL method with the enhancement function worked with a high success rate of 99% in different light conditions, and the depth data start to work when the distance exceeds 50 cm. Detection approaches based on the Kinect sensor cannot extract the real coordinates when the distance is less than 50 cm, however, and the H2O arm cannot reach the extracted button position with a distance in excess of 50 cm due to the weak arm joints and limited workspace. A solution to this problem would be that, once the position of the entry button is detected using a distance of more than 50 cm, the robot can move forward and press the button. Here, the blind robot movement after entry button detection is not accurate, and the arm requires highly accurate positional data to press the button which makes such a solution useless. Thus a new RGB-D sensor was utilized instead of the Kinect sensor to overcome this problem.

The RealSense F200 camera (Intel) works over short distances based on coded light technology to extract the depth position. Its working range is from 20-120 cm. It was used with the HSL filter and shape, size, and limited distortion to extract the button landmark. A new GUI was developed (see Figure 79) based on the same idea as the HSL method to extract real coordinates for the pressing operation.

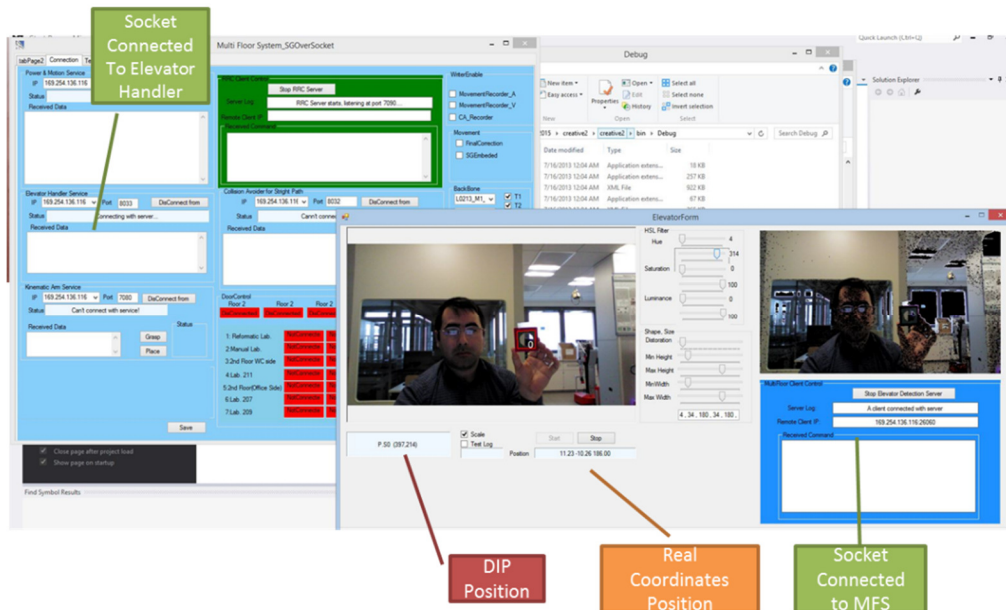


Figure 79 Entry button detection with RGB-D arm camera

### 5.2.3 Elevator Opening Detection

Once the entry button has been detected, its real position has been extracted, and the button has been pressed by the robot's Kinematic arm module, the robot has to enter the elevator. The mobile robot has to monitor the status of the entrance door and check if there is free space in the elevator when the door has opened. As described in chapter 3, the H20 mobile robot is equipped with many sensor modules, including IR distance approximation and ultrasonic detection. Since the elevator door consists of a metal frame with glass panels, the IR sensor module will not work, and so the ultrasonic sensor has been chosen as a range finder sensor to detect the door's status. A new method was established for elevator door status recognition based on the ultrasonic sensor as clarified in Figure 80. This receives the current data from the ultrasonic sensor and performs the required calculation to determine whether the elevator door is open or closed, and in addition detects whether or not a space in the elevator is free. This information comes from the main API to the multi-floor system.

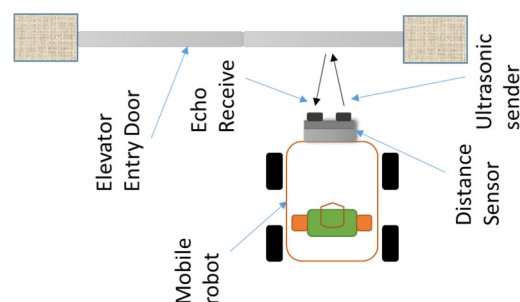


Figure 80 Opening detection method

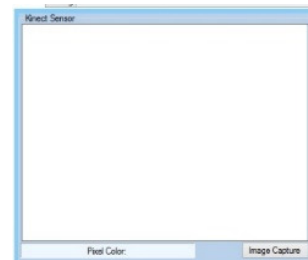
### 5.2.4 Experiments

In this section, the calibration of the Kinect sensor is validated. Entry button detection based on the Kinect sensor using the RGB and HSL method is examined as well as the F200 camera with the HSL method.

a) In the experiment for its calibration, the Kinect camera was first placed in front of the elevator area (see Figure 81a). The Kinect calibration method was executed, with camera views captured before and after the calibration. From Figure 81b it can be seen that the influence of the direct sunlight causes the captured image to be a full white image, which makes any image processing useless. Figure 81c shows the image after applying the Kinect calibration stage, and it is clear from this image that the calibration stage which controls the Kinect RGB sensor has achieved image enhancement making it suitable for image processing even under direct sunlight.



(a) Glassy elevator environment



(b) Kinect captured image.



(c) Kinect captured after enhancement

Figure 81 Glassy elevator environment and sun's direct effect on the Kinect camera and calibration

b) A second experiment verified the entry button detection based on the Kinect sensor with RGB and HSL methods. For entry button detection, a RGB-D Kinect sensor was positioned at different distances (45 cm to 60 cm) from the elevator entry and the results were recorded for each distance. The measurements were taken in different light conditions of cloudy and sunny weather. These procedures were executed for the RGB filter method and then with the HSL filter method and finally for the extra image enhancement function. Figure 82 shows the

data collected in a chart which shows the performance of the methods developed at different distances and with two weather conditions. The RGB method worked well under stable light but completely failed in direct sunlight. The enhancement function failed with the RGB method due to the effect on the RGB channel values. The HSL filter method showed a success rate of more than 99% under normal weather conditions, but it also failed in direct sunshine. The enhancement function with the HSL algorithm exhibited high performance and changes the system's capability from total failure to a success rate of more than 99% under direct sunshine. The Kinect depth sensor was also analyzed and, as clearly shown in Figure 82, it starts to work at a minimum distance of 50 cm. Table 6 shows the repeatability of entry button detection using the HSL method under sunlight conditions.

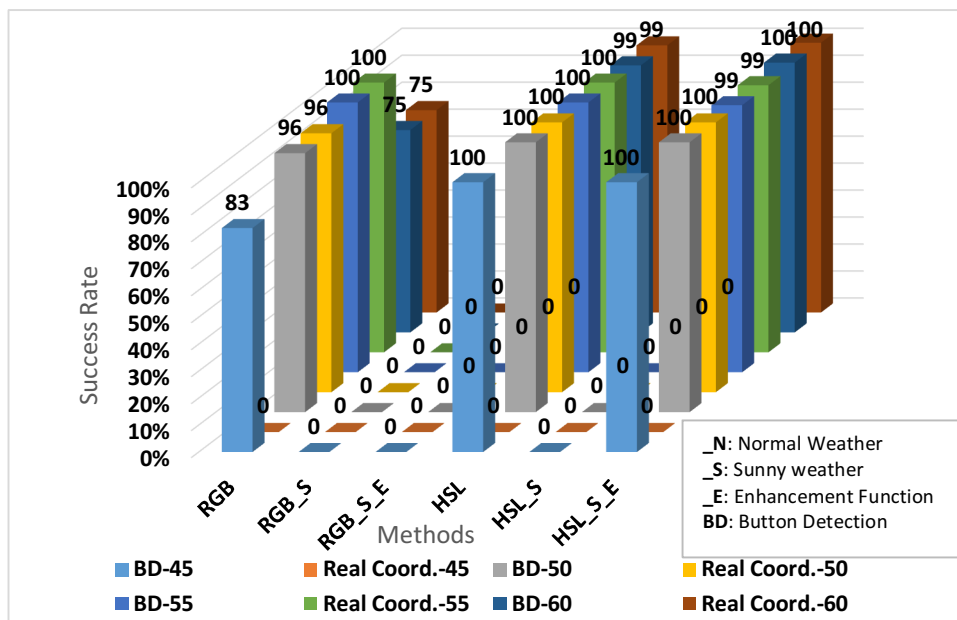


Figure 82 EEBD with/without direct sunlight and at different distances

Table 6 Repeatability and tolerance for elevator entry button detection (EEBD)

	Mean	STDEV	Repeatability	Unit
<b>X.50</b>	0.54	0.03	0.09	cm
<b>Y.50</b>	-11.60	0.01	0.05	cm
<b>Z.50</b>	49.84	0.06	0.20	cm
<b>X.55</b>	-1.45	0.01	0.10	cm
<b>Y.55</b>	-11.09	0.01	0.10	cm
<b>Z.55</b>	55.10	0	0	cm
<b>X.60</b>	0.05	0.07	0.21	cm
<b>Y.60</b>	-10.75	0.02	0.11	cm
<b>Z.60</b>	59.65	0.14	0.60	cm

A further experiment was conducted to examine the performance of the F200 vision sensor with the HSL filter method for entry button detection. The experiment was repeated 100 times for each distance (13, 20, 30, and 40 cm) and under different lighting conditions. The experiment results are shown in the chart in Figure 83, and it is clear that the entry button detection based on the F200 camera gave stable depth detection even at 13 cm under different lighting conditions, and entry button recognition was 100% successful in normal weather. Sunny weather reduced the success rate of the detection operation to 96% when the distance was 40 cm. The standard divisions, means, and tolerance of entry button detection in sunny and cloudy weather are shown in Table 7 and Table 8, where tolerance can be defined as the possible limits of variation in the value of positional error. The tolerance values rely on F200 hardware. In these tables, the extracted entry button position measurements are reference to origin point of the utilized camera. Thus, The x and y values can be positive/negative due to entry button position from the camera.

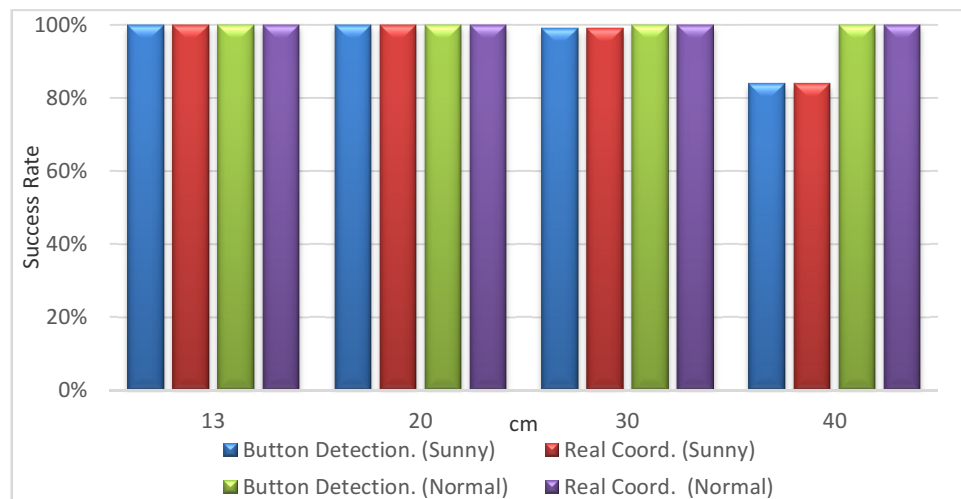


Figure 83 Entry button detection based on new F200 camera

Table 7 Entry button detection in sunny weather and at different distance

	Mean	STDEV	Tolerance	Unit
<b>X.13</b>	-3.955	0.004	$\pm 0.014$	cm
<b>Y.13</b>	0.869	0.014	$\pm 0.014$	cm
<b>Z.13</b>	13.100	0.00	0	cm
<b>X.20</b>	-3.665	0.008	$\pm 0.0215$	cm
<b>Y.20</b>	1.290	0.011	$\pm 0.021$	cm
<b>Z.20</b>	20.000	0.00	0	cm
<b>X.30</b>	-4.434	0.025	$\pm 0.0315$	cm
<b>Y.30</b>	1.775	0.057	$\pm 0.066$	cm
<b>Z.30</b>	29.978	0.042	$\pm 0.05$	cm
<b>X.40</b>	-3.337	0.037	$\pm 0.046$	cm
<b>Y.40</b>	2.156	0.003	$\pm 0.003$	cm
<b>Z.40</b>	39.937	0.049	$\pm 0.05$	cm

*Table 8 Entry button detection in cloudy weather at different distances*

	<b>Mean</b>	<b>STDEV</b>	<b>Tolerance</b>	<b>Unit</b>
<b>X.13</b>	0.870	0.019	$\pm 0.0305$	cm
<b>Y.13</b>	0.632	0.014	$\pm 0.016$	cm
<b>Z.13</b>	12.974	0.044	$\pm 0.05$	cm
<b>X.20</b>	1.074	0.024	$\pm 0.063$	cm
<b>Y.20</b>	1.072	0.018	$\pm 0.021$	cm
<b>Z.20</b>	20.000	0.00	0	cm
<b>X.30</b>	0.515	0.034	$\pm 0.0945$	cm
<b>Y.30</b>	1.500	0.011	$\pm 0.066$	cm
<b>Z.30</b>	30.101	0.010	$\pm 0.05$	cm
<b>X.40</b>	-1.107	0.00	$\pm 0.0015$	cm
<b>Y.40</b>	1.901	0.001	$\pm 0.002$	cm
<b>Z.40</b>	39.798	0.014	$\pm 0.05$	cm

### 5.2.5 Conclusion

- a) There are two technical issues which need to be handled outside the elevator, which are detecting and pressing the entry button, and recognizing the elevator door status.
- b) Two kinds of sensors are presented to detect the elevator entry button position in a glassy elevator, which are the Microsoft Kinect sensor and an Intel F200 camera.
  - For the Microsoft Kinect sensor, a pre-processing stage is applied to overcome the influence of the direct sunlight. Two approaches based on the Kinect sensor are presented to detect the entry button, which use RGB and HSL color representation. The RGB has an 88.5% success rate and it gets worse in unstable lighting conditions. Thus, the HSL color representation approach was used, which has stable performance in different lighting conditions. The HSL approach achieved a success rate of 99% for entry button detection. The main limitation of the Kinect depth sensor is that it requires a distance of more than 50cm in order to detect the button. However, within this distance the H20 robot arm fails to press the detected button due to its limited workspace.
  - The Intel F200 camera was used instead of the Microsoft Kinect sensor due to its closer working range between (20-120) cm. This camera does not require a preprocessing stage since it is installed in the robot arm which makes it much closer to the entry button, and thus direct sunlight has no effect. The HSL approach based on the F200 camera is presented for entry button detection. It achieved a success rate of 100% for entry button detection within less than 40cm distance to the elevator door and 96% success at 40cm distance.

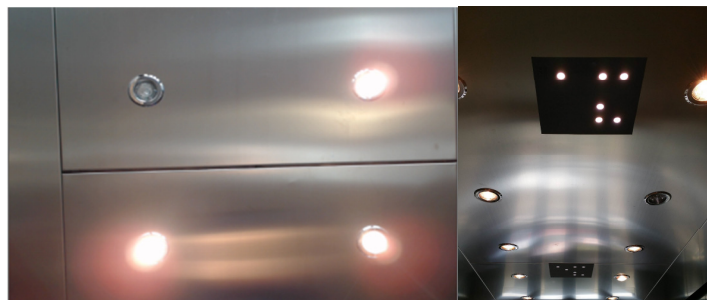
- c) The ultrasonic sensor was utilized as a range-finder to recognize the status of the elevator door and to determine if there is free space for the robot inside the elevator.

## 5.3 Inside Elevator

Inside the elevator, it is important for the mobile robot to localize itself, to move so as to reach the control panel, distinguish between buttons on the control panel, detect the destination floor, and finally to read the current floor and leave the elevator when the door is opened. The robot localization inside the elevator, control panel recognition, internal button detection, and current floor reading methods along with the relevant experiments are explained in detail in this section

### 5.3.1 Localization

This activity is very important for the mobile robot, involving how it estimates its position inside the elevator. Two methods have been developed to handle the problem of localization in the elevator. The first is based on the ceiling light as a natural landmark, as shown in Figure 84. This landmark has unique features and it can be extracted easily using the vision sensor as shown in the experimental section below. Using the ceiling light as the landmark method is not, however, secure since it may break at any time. The second method is based on an artificial passive landmark and SGM to localize the robot inside the elevator as also shown in Figure 84. After performing many experiments, localization inside the elevator based on artificial landmarks was adopted due to its stability.



*Figure 84 Ceiling light natural landmark (left) and artificial landmark (right) inside the elevator*

### 5.3.2 Robot Movement Inside Elevator

The arm's limited workspace and the small size of the cabin add further challenges for the robot movement inside the elevator. To overcome these difficulties, the movement core utilizes the installed landmark with a fine method to reach a predefined position accurately, and then the robot rotates towards the control panel. Finally, the ultrasonic range finder sensor with the motor encoder is employed to reach the exact position in front of the control panel so as to conform to the arm's limited workspace. In this position, the H20 robot has a



working range between 22- 39 cm. These procedures (clarified in Figure 85) are the basis for destination floor button pressing operation. After completion, the movement core returns the robot back towards the starting position so that it can leave the elevator safely when the destination floor is reached.

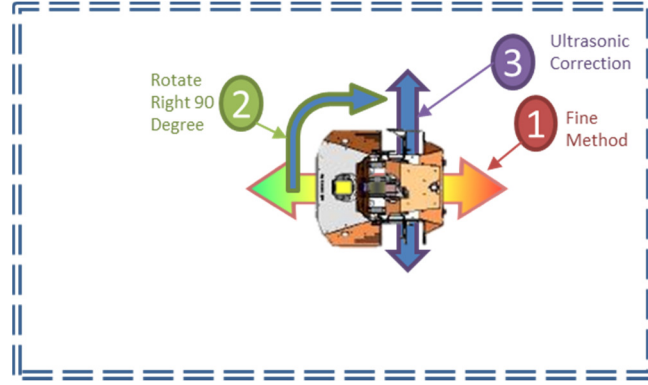


Figure 85 Robot behavior inside elevator

### 5.3.3 Internal Button Detection

As a first stage to detect the destination floor button, the movement core is used to specify the destination floor. At this level the movement core depends on the transportation task status (Grasp Position Done, Place Position Done, and Charge Position Done) to determine the current destination. When the robot enters the elevator, the movement core checks the current destination floor based on the current intermediate goal, as clarified in Table 9. For example, if the grasping operation has been completed, the placing operation floor has to be requested. Then the elevator is directed to go to the required destination floor.

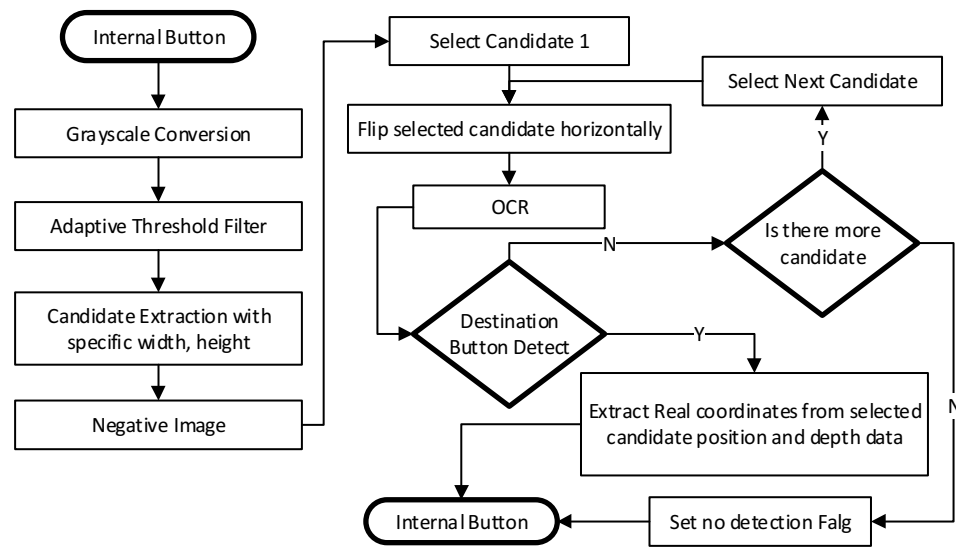
Table 9 Destination floor selection strategy

GraspPosition	PlacePosition	ChargePosition	EntryButton	Internal Button
not yet	not yet	not yet	Current Floor	GraspPosition Floor
Done	not yet	not yet	Current Floor	PlacePosition Floor
Done	Done	not yet	Current Floor	ChargePosition

At this level of the elevator handling system, the OCR is embedded with a new method to find the internal buttons. In this method, it is important to recognize each button label separately, and thus the method developed applies a combination of filters. These include grayscale conversion, which makes the captured image suitable for subsequent stages, stretch contrast to improve the contrast in the image by stretching the domain of intensity, and an adaptive threshold to choose the best threshold under different light conditions for binary image conversion. Next, the search among each button candidate uses specific width and height features and takes the inverse value of the pixel and flips the candidate images horizontally



(as the Kinect image stream has a mirror image) to make them suitable for the Optical Character Recognition (OCR) stage. Each extracted candidate passes to the OCR engine for comparison with the required destination and finally, based on the position of the matching candidate's the image and depth information, the real coordinates are extracted and translated into a robot arm reference. Figure 86 demonstrates the internal button detection method.



*Figure 86 Internal button detection method*

As clarified later in the experimental section, the validation of this method showed that a number of false positives resulted during button '1' detection (44 false, see Table 10). These false readings arose because the OCR system recognized a corner as the character "1", and so it detected another character in another position. In the developed system, a false negative identification merely delays the detection operation, while false positive measurements cause serious errors in mobile robot arm guidance. Thus an initial stage has been applied to minimize the search area by adding special landmarks around the internal button labels as shown in Figure 87.

The detection of the destination floor button and the real position extraction process based on the landmark method has two stages. First the elevator button panel, which represents the region of interest (ROI) for the next stage, must be found. This stage is based on finding a true landmark positioned around the button panel (see Figure 87) to extract the ROI using HSL color representation, and then to apply the HSL filter and correct the image orientation. The second stage is based on the first to find the buttons inside the ROI. After extracting the button shapes, the vision algorithm starts to apply digital OCR on the detected button to find the floor number, which is printed over the button. The position in pixels is extracted from the acquired RGB image and the required mapping is performed with the captured depth data to send the

accurate destination button position to the MFS. The workflow of internal button detection based on the Kinect sensor is illustrated in Figure 88. The internal button detection algorithm is demonstrated in Appendix A.5. The RAKM (details about the developed arm kinematic module are found in [167]) is used to press the destination button based on the extracted position.



Figure 87 Inside elevator BP detection and floor number extraction

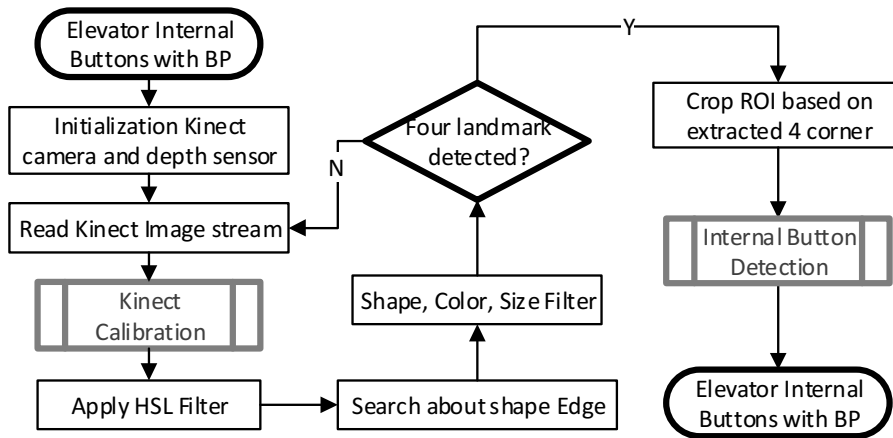


Figure 88 Elevator BP detection and floor number extraction method

### 5.3.4 Internal Button Detection With F200 Hand Camera

Due to the limitations of the Kinect depth sensor, which only works at distances over 50cm, and the limited workspace of H20 robot arm which can only reach 39cm as a maximum distance for the pressing operation, the F200 camera was utilized instead. This has a short working depth of 20-120 cm and was used as the main sensor to detect and extract the internal buttons based on the method described in Figure 86. The F200 camera is installed in the robot's arm, and thus it has a limited view which eliminates the problem of false positives without using the button panel landmarks.

### 5.3.5 Current Floor Estimation

The floor estimation technique is applied to inform the robot about its current position inside the elevator. When the destination matches the current floor and the ultrasonic sensor

recognizes the door status as opened, the robot can leave the elevator. As a first method, a computer vision method was used to read the current floor number indicator in the glassy elevator environment. As explained later in this section, this approach has many limitations and thus a current floor estimation method based on a height measurement system is utilized instead.

### 5.3.5.1 Floor Reader Based On Computer Vision

This system recognizes the current floor number which is installed in the elevator shaft for each floor based on computer vision. As a first step, the image is converted into grayscale to make it suitable for the adaptive binary threshold filter. Secondly, the objects inside the processed binary image are collected to choose the best candidate based on size. Finally, the OCR engine is applied to the candidate object to recognize the floor number. The workflow of the current floor reader based on computer vision is clarified in Figure 89. When the destination matches the current floor number, the robot leaves the elevator. This approach has been validated in normal weather conditions and its functionality proven with a success rate reaching 99%. However, this approach has many limitations. Since the floor number mark is installed over a glassy surface, the robot may not be able to read the number at some times of the day because of sunlight reflecting over the number mark. The robot must also detect the floor number before the door is opened, and in certain situations such as a human forming an obstacle between the robot camera and the floor number indicator the robot will fail to detect the floor number mark. Incorrect current floor estimation can make the robot lose its way to the destination. Thus, an innovative current floor estimation method based on a height measurement system is used instead. The current floor reader algorithm is shown in Appendix A.6. The internal and external button detection and the floor reader was published in [171].

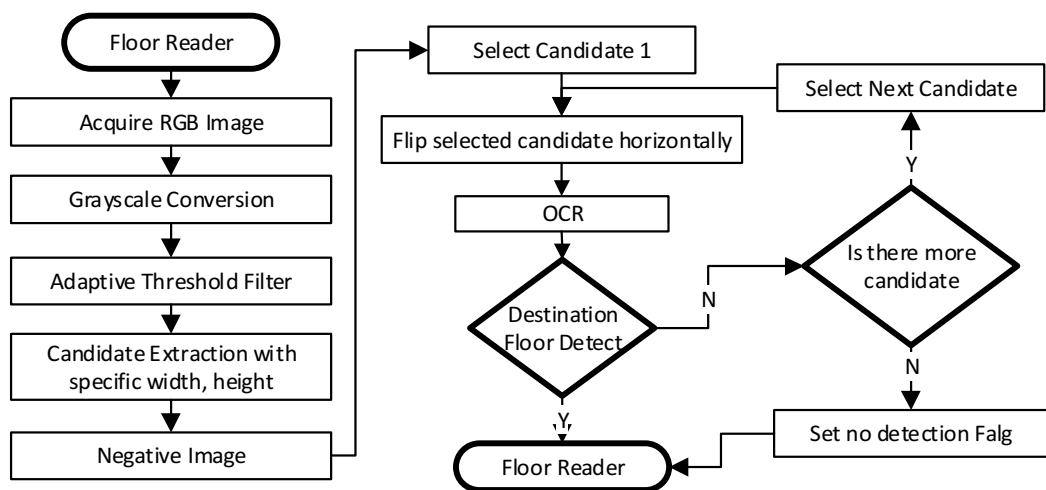


Figure 89 Elevator floor reader

### 5.3.5.2 Floor Estimation System Based On Height Measurement

As a hardware platform for the height measurement system, the LPS25HB pressure sensor and STM32L053 microcontroller were configured and programmed to sense the environment and detect the current floor position. A soldering drift, which is defined as the difference between the accuracy of the sensor before and after soldering, appeared when the pressure sensor was attached to the STM32L053 microcontroller. A one-point calibration technique was used to solve this problem by comparing the pressure sensor readings after attachment with a precision barometer. The difference was calculated and added as an offset value to each pressure sensor reading.

The absolute digital barometer (pressure sensor) readings vary with both weather and altitude. The pressure sensor was examined regarding the possibility for it to be employed as a floor estimation sensor in multi-floor environments, as clearly shown in the next experimental section. However, the oscillation of the output signal and wide variations in readings during the week would reduce the utility of this technique for floor detection. Two methods were applied to deal with the variations in pressure. Firstly, a smoothing filter was used, and a practical analysis was performed to choose the filter order to achieve a balance between the time required and the stability of pressure readings. Secondly, an adaptive calibration method was used to calibrate the sensor readings for the robot's current floor before entering the elevator in order to overcome the wide variation in daily pressure readings. The indoor localization method developed earlier, which is based on passive landmarks for a multi-floor environment, was utilized to identify the robot's current floor number outside the elevator.

Smoothing filters with a finite impulse response (FIR) structure were used to solve the problem of small variations in pressure sensor readings. The FIR filter implemented inside the microcontroller for signal smoothing was chosen due to its simplicity, linearity, and stability. The FIR design diagram is shown in Figure 90, and described in equation 20.

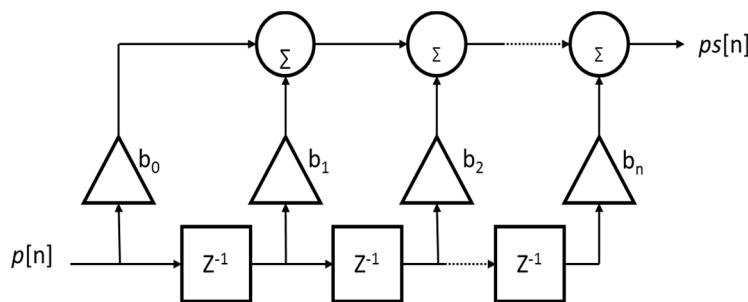


Figure 90 Finite impulse response filter

$$Ps[n] = b_0 p[n] + b_1 p[n-1] + \dots + b_n p[n-N] \quad (20)$$

where:  $p[n]$  is the input pressure reading,  $Ps[n]$  is the output smooth pressure value after filtering,  $b_i$  is the Filter coefficient, and  $N$  is the Filter order.

The performance of this filter relies on the filter order. Increasing the filter order improves the performance of the smoothing filter and increases the time needed to receive the output signal. In this application, time is an important aspect to consider since the mobile robot has to estimate its current floor, recognize the door status, and leave the elevator before the door closes again.

The smoothing filter was implemented inside the microcontroller. The MBED development environment was used to program the microcontroller in the C language. As clarified in Figure 91, the processing of the sensor reading begins by initializing the microcontroller and receiving the readings. The sensor data then pass through the smoothing filter to reduce the oscillation in the pressure sensor output signal. The filter queue is initially filled with the same data to make the output signal stable from the sensor's first running operation. Finally, the smoothing output is collected into a specific form to send it via a USB cable to the robot board computer every 1s.

The daily pressure sensor readings are reported in section 5.3.6c. As clearly shown in this section, the pressure readings show considerable changes in their values even during the same day. The variation can reach up to ten hPa (approximately one hundred meters), and thus another stage was added to calibrate the sensor each time before the robot enters the elevator.

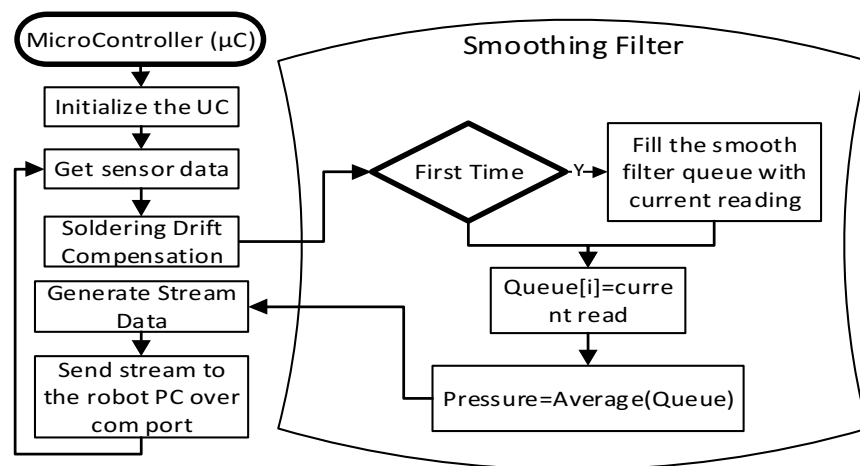


Figure 91 Microcontroller flowchart

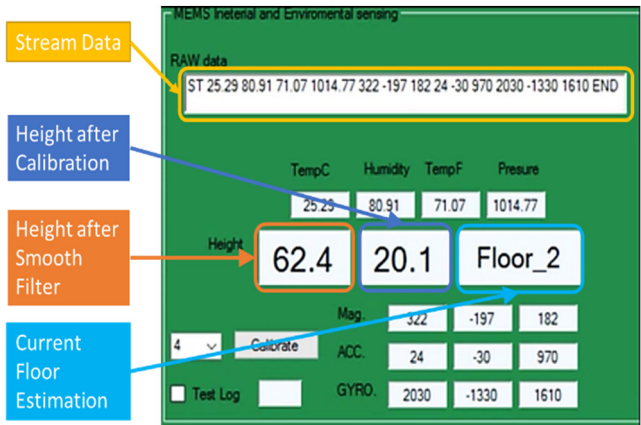
The calibration stage depends on the localization method developed earlier for the MFS to receive the current floor information outside the elevator, then the pressure sensor data is

calibrated with the current floor height to get a constant number which represents the calibration value. The current height converted from the pressure sensor using equation 21 is added to the calibrated value to find the robot height. Then the extracted height is compared with the height range for each floor (as shown in Figure 92) to estimate the robot's current floor number inside the elevator.

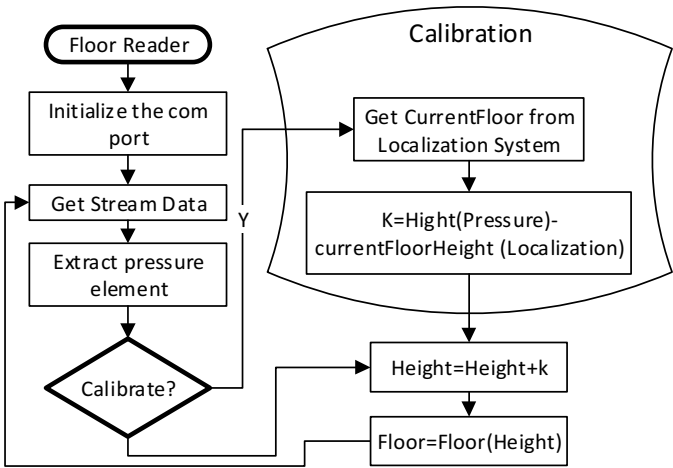


Figure 92 Range of height measurements for each floor (m)

The MFS is embedded with the required GUI and coded with C# to handle the connection to the microcontroller; the incoming data and calibration stage are shown in Figure 93.



(a) Developed graphical user interface



(b) Floor estimation flowchart inside multi-floor system

Figure 93 Embedded floor estimation inside MFS

When the robot reaches the elevator door based on the earlier developed navigation system during multi-floor transportation task, it utilizes the ultrasonic sensor to detect the elevator door's status and the height measurement system to estimate the current floor number inside the elevator. If the estimated current floor matches the destination floor and the door status is detected as “open”, the robot can leave the elevator to complete the transportation process. Otherwise, the MFS will send an elevator call again to the destination floor over Wi-Fi every second until the destination floor is reached.

$$Height(p) = \frac{1 - \left( \frac{p}{1013.25} \right)^{0.190284} * 145366.45}{3.2808} \quad (21)$$

### 5.3.6 Experiments

In this section three experiments with the inside EHS are reported which involve internal destination button detection and the floor reader based on computer vision, and finally two further experiments determined the stability of the height measurement system for current floor estimation.

The first experiment was conducted to verify the internal button detection method. The RGB-D Kinect sensor was placed in front of the buttons under normal weather conditions and the button selected was tested 100 times. Figure 94 shows the captured and processed image for destination floor button recognition with the extracted button position in both images and real coordinates. The real coordinates were measured based on the Kinect position as a reference point. The success rate for destination floor button recognition is shown in Figure 95.

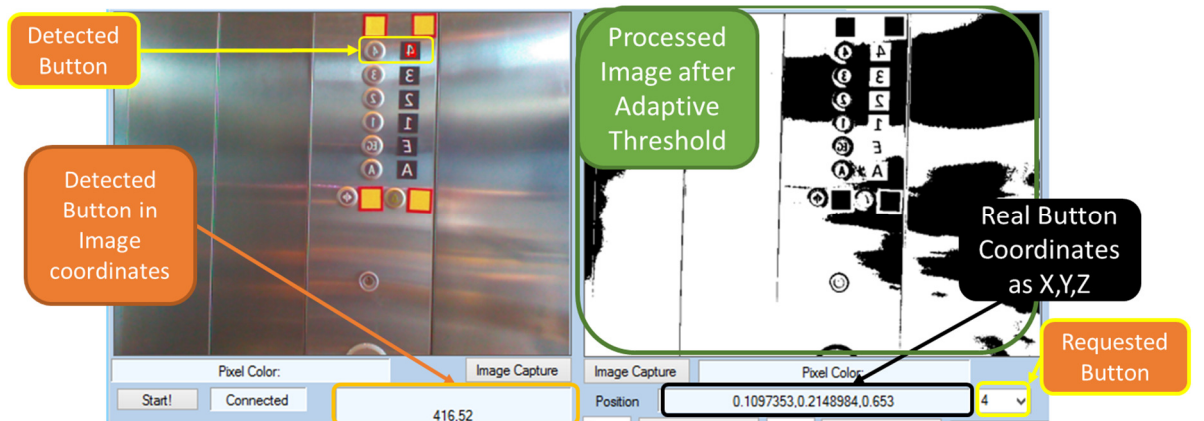


Figure 94 Destination floor button recognition (DFBR): processed image without BP



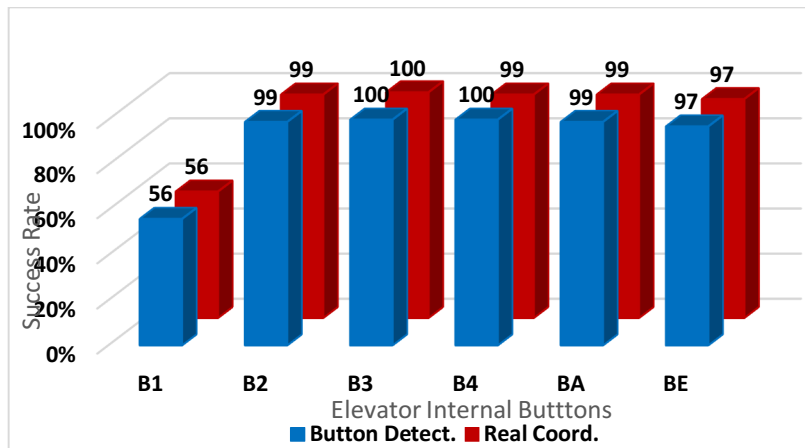


Figure 95 Destination floor button recognition (DFBR): success rate without BP

Table 10 False readings in DFBR without BP

Button	False Positive	False Negative
1	44	0
2	1	0
3	0	0
4	0	0
A	1	0
E	3	0

The same experiment was repeated inside the elevator with the landmark button panel searching method to overcome the false positive problem as clarified in Figure 96. The method developed has proved its efficiency, as shown by the success rates in Figure 97.

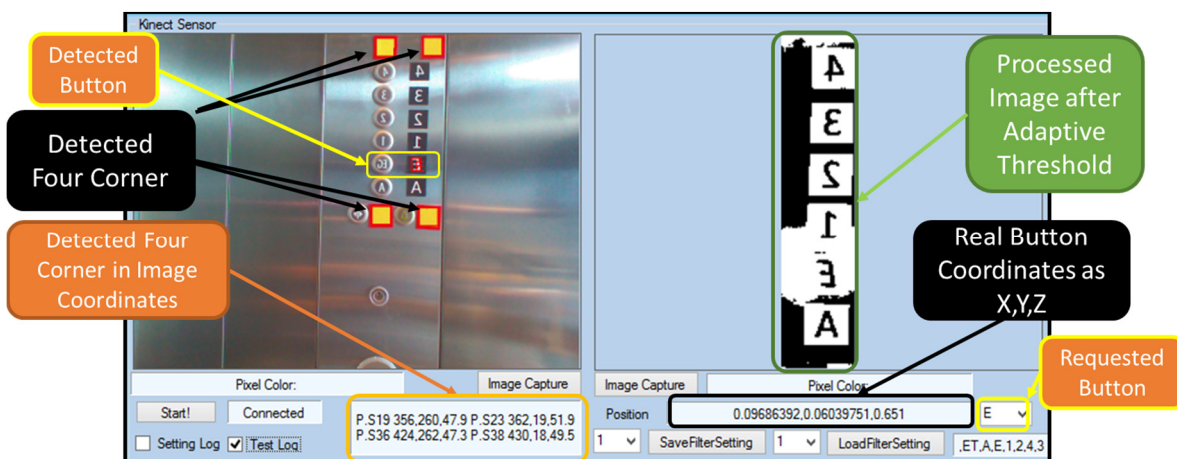


Figure 96 Destination floor button recognition: processed image with BP



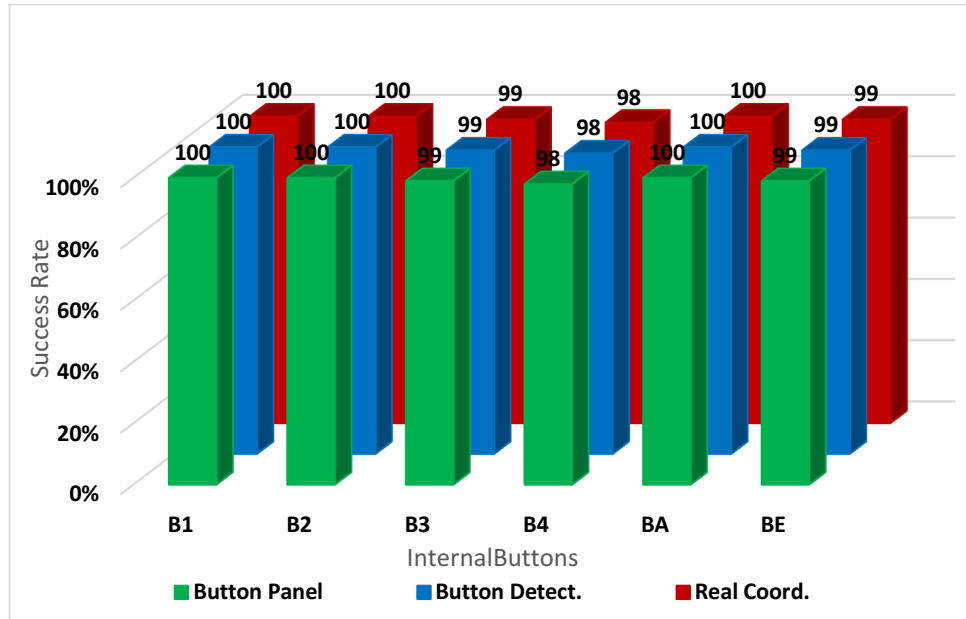


Figure 97 Destination floor button recognition: success rate with BP

The second experiment was conducted on two different floors to validate the performance of the current floor reader method based on computer vision. The Kinect RGB-D vision sensor was positioned facing the elevator door to recognize the current floor number. The processed image for the extraction of the floor number is shown in Figure 98. The results from repeating this method 100 times in normal weather show success rates of 99% for the third floor and 100% for the second floor.

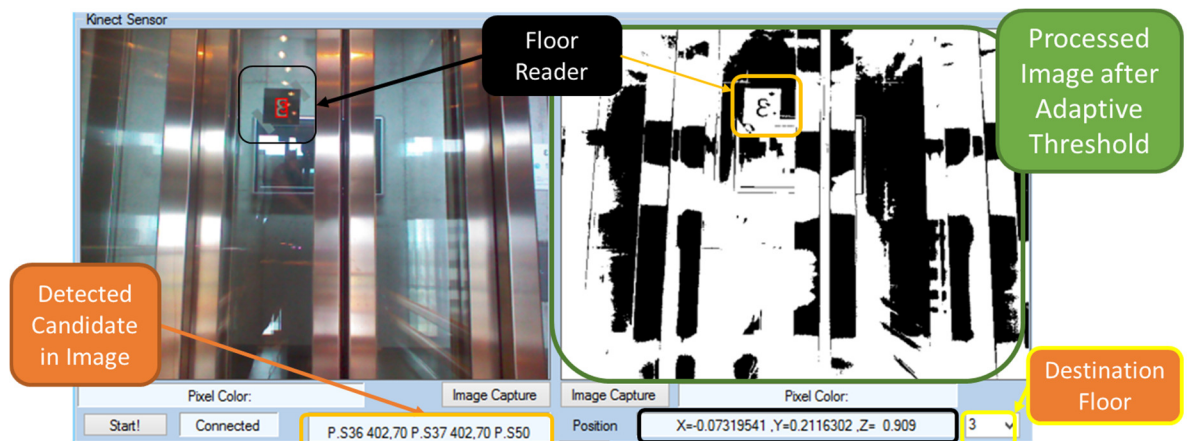


Figure 98 Current floor reader (CFR) processed image

Experiments were then performed at the CELISCA building to validate the method developed for the height measurement system as a robust floor estimation approach. Firstly,

experiments were conducted to check sensor stability without the smoothing filter, then the smoothing filter was applied at the first and second orders in multiple floor environments to choose the best filter order according to the time required and the stability of measurements. A second test was performed to check the variation in daily pressure so as to investigate sensor stability over time.

The first experiment was executed in three steps, starting by receiving the pressure data and converting it into height readings. This was accomplished 2,500 times distributed between five floors, and Figure 99a shows a graph of the data acquired. The second and third steps applied the smoothing filter with different parameters. In the second step the smoothing filter was taken with first order (see Figure 99b), while in the third step the second order was chosen to give greater smoothing behavior (Figure 99c).

Table 11 summarizes the three steps of the first experiment regarding average and tolerance. The first order smoothing filter was chosen to achieve a balance between the stability of the pressure signal and the time required to receive updated values especially when the elevator is moved between floors.

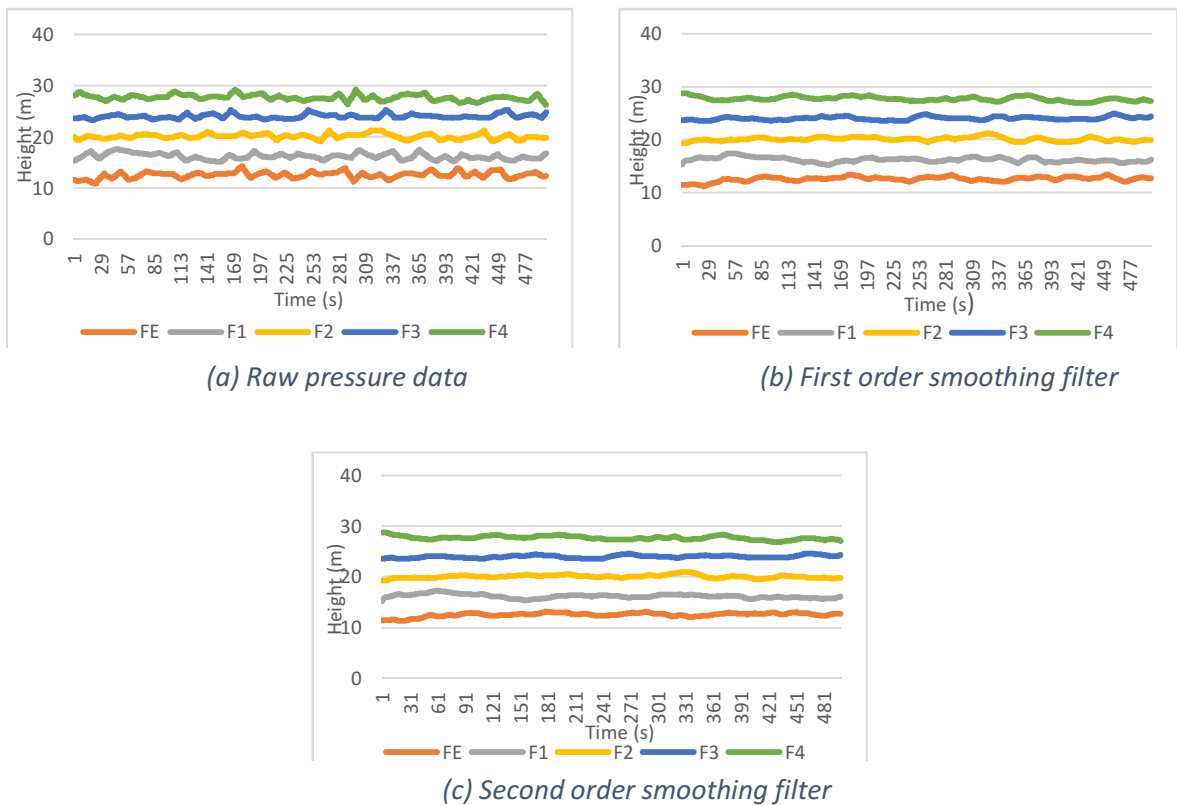


Figure 99 Pressure sensor data and the smoothing filter.

Table 11 Height measurement in meters with different filter orders.

Floor	Raw		First Order		Second Order	
	Averag	+ -	Averag	+ -	Averag	+ -
4	27.696	1.29	27.694	0.81	27.694	0.768
3	24.032	0.965	24.03	0.71	24.038	0.52
2	20.108	1.015	20.11	0.83	20.116	0.73
1	16.211	1.245	16.208	1.097	16.209	0.928
e	12.586	1.59	12.598	1.09	12.609	0.803

A second experiment considering daily pressure variation was executed to determine the stability of the pressure sensor during one working week. The sensor was placed in the same position and three experiments were performed every day at different times during the day, each with 100 readings. The average values are shown in the chart in Figure 100. It is clear from this chart that the sensor gave wide variations in pressure during the same day and different weekdays. Therefore, this sensor cannot be used directly without a calibration stage to add a reference to correct the floor reading.

Finally, the integration of the developed floor estimation approach into a real transportation system was tested. The experimental results show that the proposed method has a 100% success rate, which keeps the robot inside the elevator when it realizes it is on the wrong floor and calls the destination floor periodically until the correct floor is reached and the robot exits the elevator, as clarified in section 5.4.

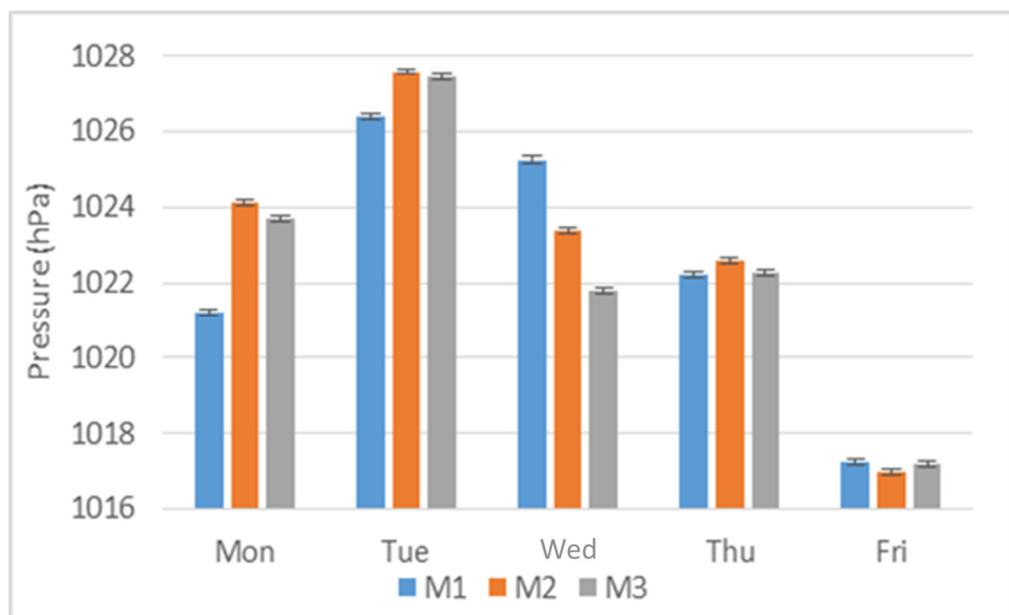


Figure 100 Daily pressure changing over the week

### 5.3.7 Conclusion

- Three technical issues should be handled inside the elevator which are localization, internal button detection, and current floor estimation.
- For inside elevator localization, a passive landmark with SGM as a HEXA reader was used to realize a robust localization method compared with localization based on ceiling lights.
- Two vision sensors were proposed for internal button detection, which are a Microsoft Kinect sensor and an Intel F200 camera.
  - a) An adaptive binary threshold method with a Kinect sensor was utilized for internal button detection. An OCR algorithm was embedded to recognize each button label. First, a number of filters were applied to the captured picture to realize the best candidate for the OCR algorithm. The experimental results for this approach showed a number of false positives due to the OCR algorithm recognizing some elevator edges as the number '1'. Thus, four special landmarks were placed around the internal button labels and an HSL approach developed to recognize these landmarks was employed to minimize the search area. Applying the adaptive binary threshold method with the installed landmarks achieved a success rate of 99%.
  - b) The Kinect sensor requires distances greater than 50cm for button detection, and is thus unsuitable for use with the H20 robot arm that has a limited workspace. An Intel F200 camera was used instead. The approaches explained above were used for installed landmark recognition and internal button detection.
- For current floor estimation, computer vision and an innovative height measurement system approaches were proposed.
  - 1) As a first method, a computer vision method was used to read the current floor number mark which is installed in the shaft in the glassy elevator environment. This method has a high success rate in cloudy weather but requires high speed to read the floor number before the elevator door start opening, and it failed at certain times during sunny days and if there was an obstacle between the robot and the number mark. These limitations make this method unsuitable to be applied in this work, and therefore an innovative height measurement system was developed for current floor estimation.
  - 2) The innovative height measurement system consists of an LPS25HB pressure sensor and STM32L053 microcontroller. When the pressure sensor was attached to the microcontroller, soldering drift appeared and a one-step calibrated method was used

to solve this problem. The pressure sensor readings vary with both weather and altitude. Thus an efficient smoothing filtering and calibration stages were used. The experimental results for current floor estimation based on the new height measurement system showed a 100% success rate.

## 5.4 Automated Elevator Management System

Due to the importance of the elevator for mobile robot transportation in multi-floor environments, it is necessary to develop more than one strategy to guarantee a safe elevator handling strategy. The automated elevator management system was developed as a second method to handle the elevator in case of failures in vision detection methods and robot arm movements. This method depends on the ADAM module, which works as an interface between the robot and elevator controller as shown in Figure 101, and the movement core which is embedded inside the MFS to control the robot's motion for planned path execution. The automated elevator connects to the required floors. The automated elevator management system inside the MFS was developed to control the connection between the automated elevator and the MFS. It receives the required floor information from the movement core, translates it into a specific port and pin number inside the ADAM module, and sends a pulse of 100ms duration to call the elevator to the specified floor.

The movement core calls the elevator to the current floor and then directs it to the destination floor. The elevator is always called to the robot's current floor. The proposed MFS indoor localization sensor is used to find the mobile robot's current floor outside the elevator.

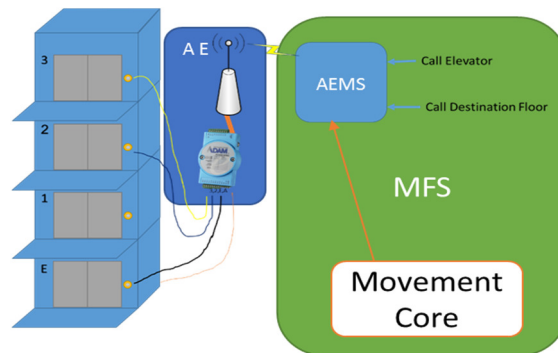
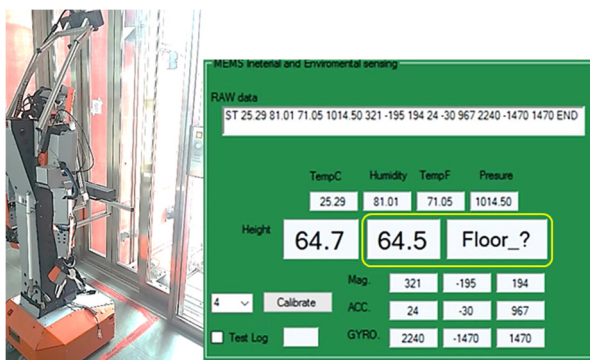


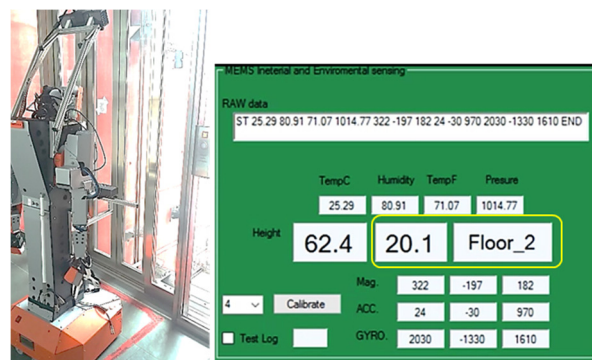
Figure 101 Automated elevator with ADAM socket and EMS

The operation inside the elevator requires a strategy to specify the destination floor. The strategy utilized is the same one as was used for the EHS developed earlier and described in section 5.3.3, which depends on the ITMS status (Grasp Position Done, Place Position Done, and Charge Position Done) to know the current destination. For example, if the robot has completed the grasping operation (Grasp Position Done), then the selected floor should be the placing operation floor.

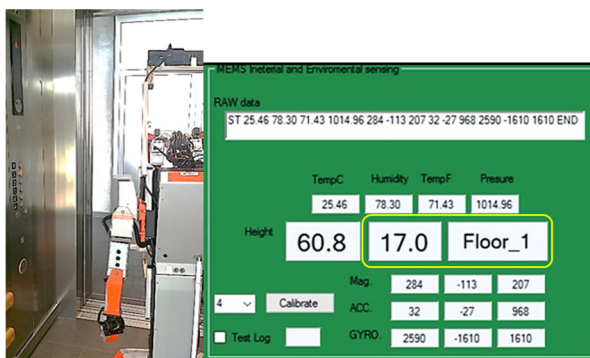
An experiment was repeated 100 times to validate the integration of the automated elevator management system into the multi-floor transportation system. The automated elevator management system over Wi-Fi and floor estimation based on pressure sensors was integrated directly into a real mobile robot multi-floor transportation system in life science automation. Figure 102 shows the GUI developed for automated elevator operation management. In the figure, the robot reaches the elevator area (a) based on the navigation system developed earlier without pressure-based floor estimation calibration. Then the robot calibrates the pressure sensor to the current floor, and calls the elevator (b). The robot enters the elevator and then a wrong floor was called by the elevator passenger, which forces the elevator to execute another call before the robot's destination floor call (c). When the robot reaches the wrong destination floor, it does not leave the elevator and requests the required destination floor again. The robot is then in transition between floors (d), reaches the requested floor, the ultrasonic sensor detects that the door is open, and finally the robot leaves the elevator (e). The automated elevator management system achieved a 100% success rate: for calling the elevator to the current floor and specifying the required destination floor; and then the pressure sensor successfully kept the robot in the elevator when it reached a wrong floor and called the destination floor periodically until the destination floor was reached whereupon the robot left the elevator. This work was published in [172].



(a) Robot reaching elevator before calibration



(b) After calibration



(c) Wrong floor



(d) Elevator in movement





(e) Robot exit on the correct floor

Figure 102 Transportation Experiment

## 5.5 Button Pressing Operation

### 5.5.1 The Complete Scenario

The entry and internal button detection and current floor reading methods have been developed as an elevator handling system. In the EHS the F200 camera is utilized as an RGB-D sensor and the HSL method operates a stable color detection method. The MFS controls the EHS and the arm's kinematic module to execute the pressing operation. The button pressing operation procedures can be explained as follows. The movement core unit is used with a fine correction method and motor encoder to reach the accurate elevator button detection position. The arm is then sent an initial request via a socket to the arm's kinematic module, and the F200 camera is sent an initial request via a socket to the EHS after the arm has reached the initial button position. The destination button label is sent to the EHS to detect it and extract its real coordinates, and then the extracted coordinates are sent back to the MFS. A check is made that the EHS has detected the button, and the destination button detection request is re-sent to the EHS if it fails to detect the button after 3 attempts. Calibration is then performed between the current position of the F200 and the kinematic arm end-effort position. Finally, a pressing order is sent with the extracted x, y, and z coordinate to the arm's kinematic module. The pressing orders are performed using the model developed by our research group, which utilizes the inverse kinematic solution to calculate the required angle of the arm joints based on the destination button position. The calculated joint angles are sent via sockets to the arm's servo motors. The combination of servo motor movements for all joints controls the robot arm in reaching the destination button position accurately. Flowcharts for this working scenario are shown in Figure 103. In the case of failure at any of these stages, the MFS rearranges the transportation task to complete it or returns to the charge station and informs the RRC level about the failure.

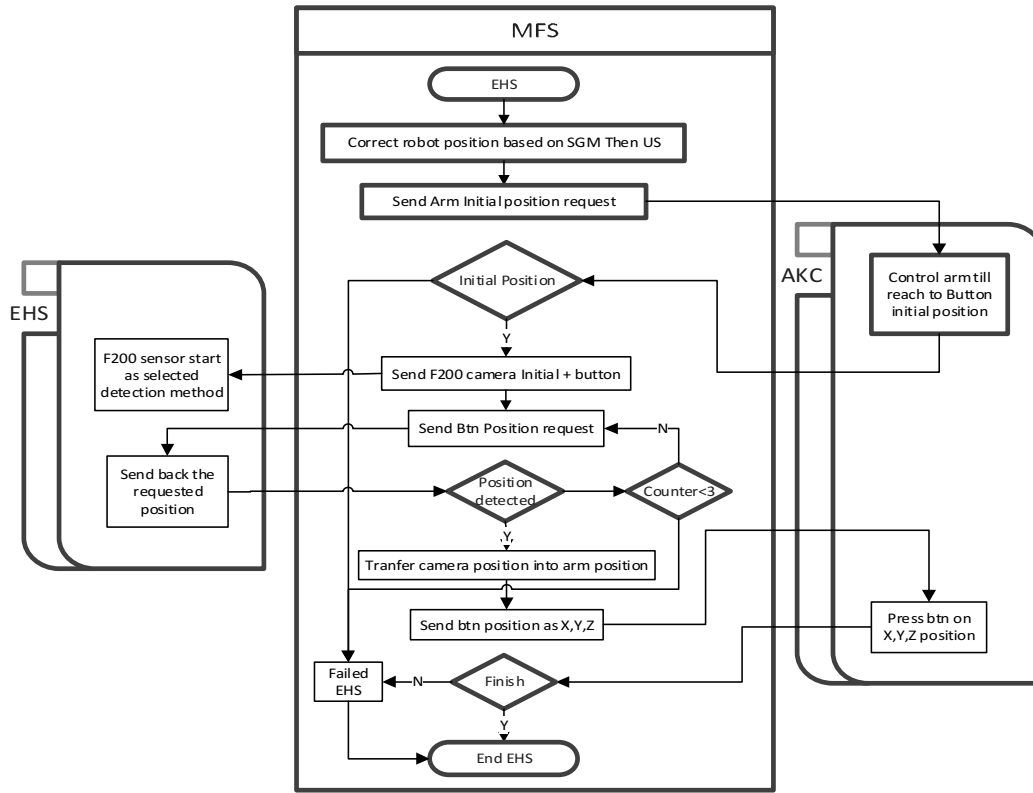


Figure 103 Complete scenario

### 5.5.2 Validation Experiments

In this experiment the MFS started to execute a multi-floor transportation task between the second and third floors at the CELISCA building. When the MFS reached the predefined elevator entry position (the position that allows the robot to enter the elevator directly after door opening), the MFS requested that the EHS initialize the F200 camera and order the RAKM to control the robot arm. The RAKM starts its working process by attaching the finger as a new end effector for button pressing then controlling the arm to move to the initial detection position. After completing the entry button detection method and extracting the button's real x, y, and z coordinates, the RAKC will press the button. This experiment was repeated for ten time. Each time the communication sockets success to transport the MFS orders to the EHS and the RAKM and the EHS detect the entry position and send the x, y, and z coordinate to the RAKM for pressing operation. The RAKM fails to press the entry button from the defined position each time. Thus, a new position is assigned to the robot for detecting and pressing the entry button. In the new position, the robot requires to correct its position after pressing operation. In this stage, the position correction relies on the motor encoder. The experiment has been repeated for ten time to evaluate the RAKM performance in the new position. The robot succeed to reach the elevator entry button position with repeatability range is  $\pm 1.25\text{cm}$  for x axis and  $\pm 1\text{cm}$  for y axis. The RAKM [167] has 10% success rate for pressing operation after receiving the correct x, y, and z coordinates of the detected button, this problem is



related to weak robot arm joints due to servo motors. The pressing stages are shown in Figure 104, while Figure 105 shows the error position reach by robot arm.

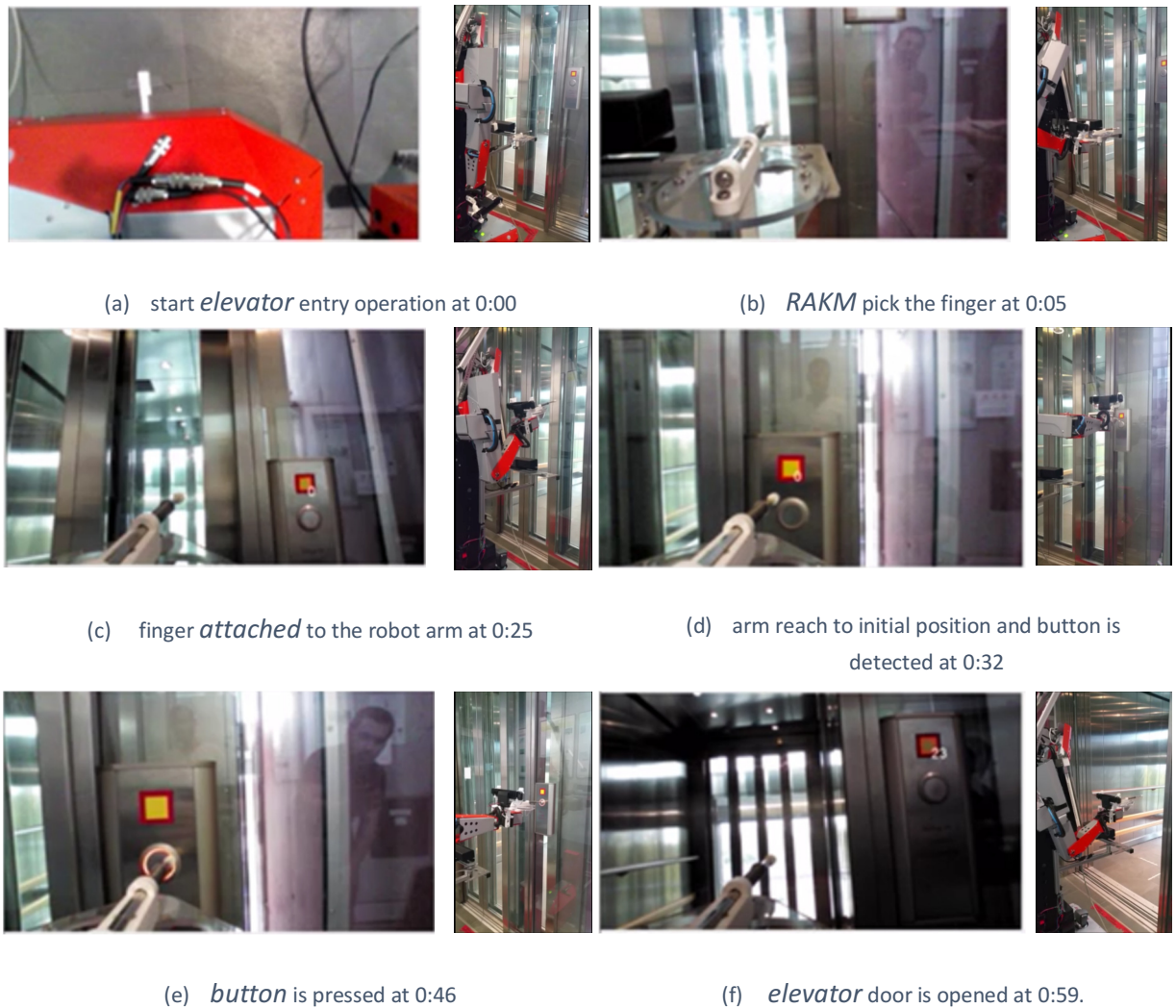


Figure 104 Complete operation scenario

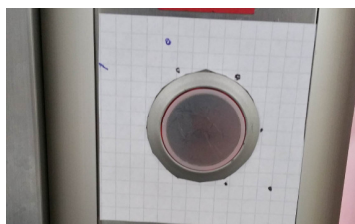


Figure 105 arm pressing position error

## 5.6 Elevator Handler Error Mangement System

The proposed detailed robot-elevator interface control strategy is illustrated in Figure 106. The EHS classifies the error according to its appearance space as either outside or inside elevator error handler. The elevator handling system has two methods for handling outside elevator errors. In case of the failure of the mobile robot to reach the required elevator button pushing

area accurately due to wheel slip, the position and orientation correction function controls the robot so that it reaches the defined position with high-accuracy orientations. This function checks the robot's position after movement and tries to correct it three times. In addition, a hybrid elevator control process starts after the robot has identified the button's position as X, Y, Z. The values are then passed to the kinematic arm module for button pressing. The robot checks the elevator door with the ultrasonic sensor and controls the arm again to press the button when the door is still closed. If previous attempts have failed, the hybrid elevator controller selects the automated elevator controller over the socket method to open the door. If the functions listed above have failed in handling the elevator correctly, the ITMS informs the higher level controller RRC, stops the current transportation operation, and controls the robot until it reaches the charging station.

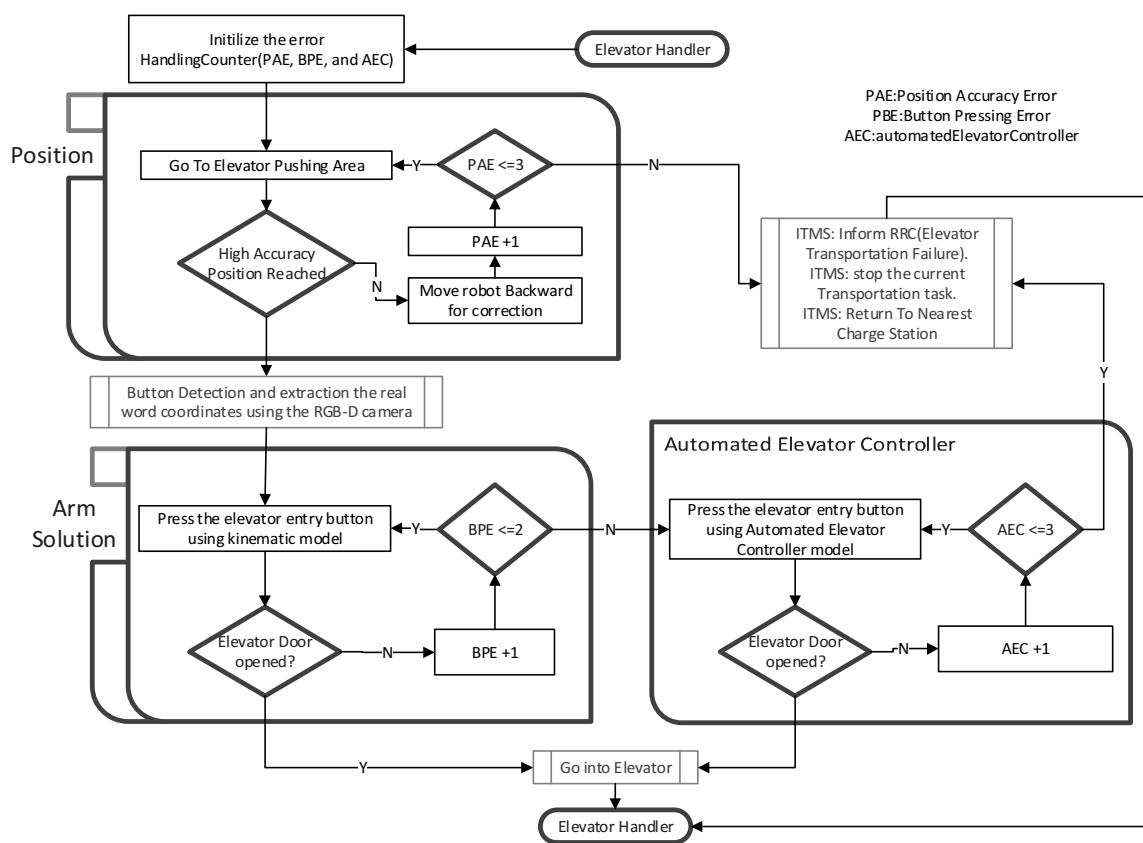


Figure 106 Outside elevator error handling flow chart.

The error handling system inside the elevator starts by monitoring whether or not the destination floor has been reached by continuously checking the current floor reader. If it does not reach the destination floor after a specified number of attempts, the MFS sends a warning alarm to the RRC to initiate the appropriate procedures. After finding the destination floor, the next stage of error handler is enabled to check the elevator door status, and another warning alarm is sent to the RRC if the door is still closed after a defined number of attempts.

The last error handling maneuver is to check if the floor reached if is the required floor based on the land marks installed outside the elevator. If the landmark is missing or a wrong floor number landmark is read, the error handling system returns the robot back to the elevator and chooses the destination floor again. Finally, if the robot fails to reach the destination floor after reaching the maximum allowed number of attempts, an error message is sent to the RRC. These steps are arranged as a flowchart in Figure 107.

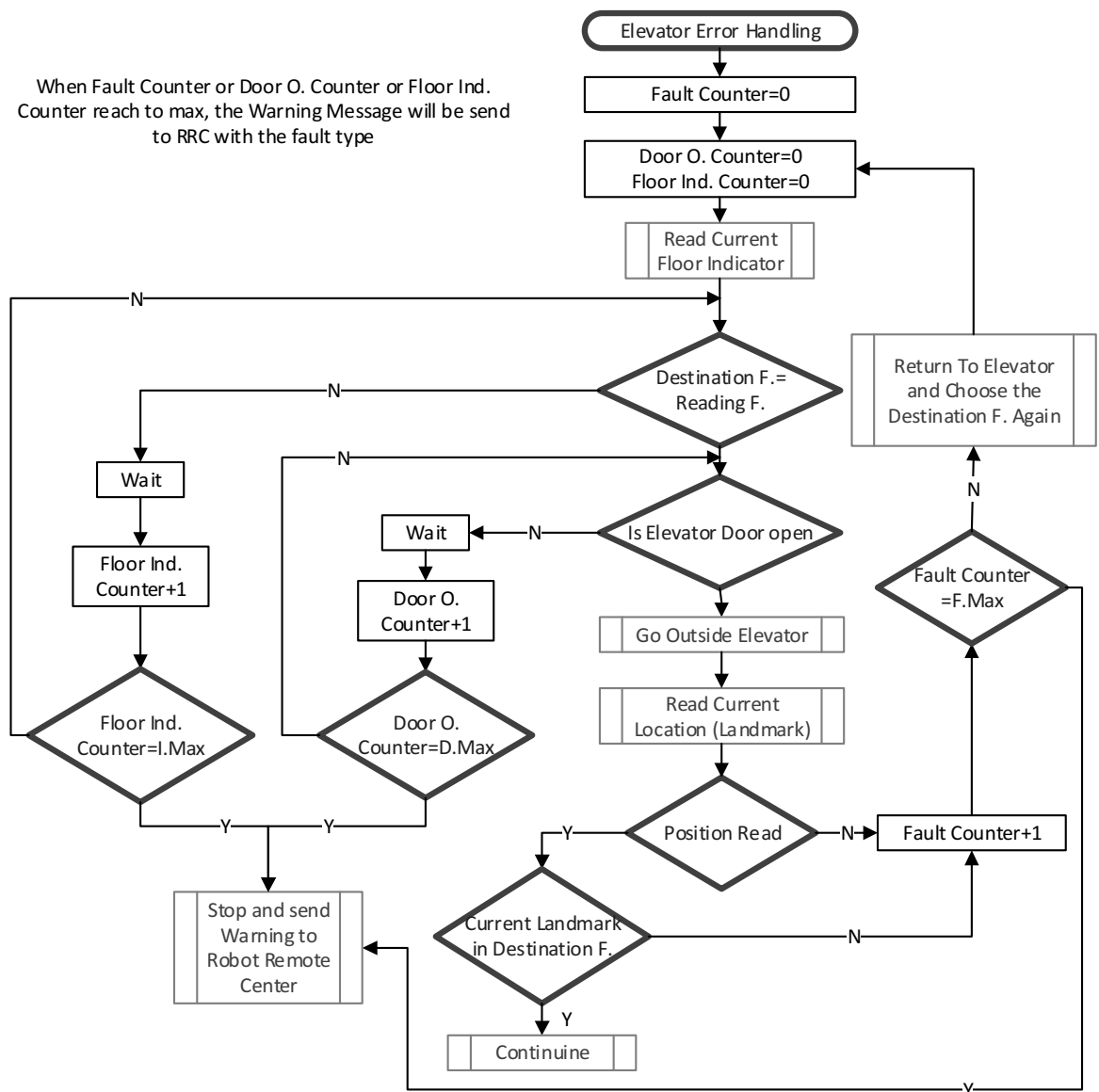


Figure 107 Inside elevator error handling flowchart



## **Chapter 6      SUMMARY AND CONCLUSION**

In the last decade, researchers have shown increasing interest in automated systems development which enable huge advances to be made in life sciences laboratories. In general, automated systems increase productivity, decrease costs and save the time of scientific staff.

The recently developed Hierarchical Workflow Management System (HWMS) has a middle control layer named the Transportation and Assistance Control System (TACS). The Robot Remote Center (RRC) is a control sub-layer between the Robot Board Computer (RBC) and the TACS which manages transportation tasks using processes for receiving transportation tasks, forwarding them to available robots along with the correct planned paths, and reporting real-time transportation results back to the TACS.

The RBC level of the existing system works for transportation tasks on a single floor. In this dissertation, an intelligent multi-floor transportation system is presented to handle movement in a complex building structure with facilities distributed on different floors.

The navigation system consists of mapping, localization and path planning processes. Two methods are developed for mapping. Relative mapping is responsible for generating a global map in a multiple-floor environment with high accuracy and flexibility to be used with any number of floors. This three-dimensional representation has a unique reference for each floor. The mapping methods used employ the SGM as landmark reader to gather the information required. The path map consists of a set of rich information waypoints whose positions are related to the relative map and which are then utilized in path planning to find obstacle-free transportation paths. Each waypoint has information about the incoming door number and the robot's maximum speed, acceptance tolerance, and movement direction, enabling dynamic collision avoidance and fine correction function.

The multi-floor indoor localization stage is used to extract the positions as  $x$ ,  $y$ , and floor number in the global map for any number of robots. The incoming data from the landmark reader is translated into positions based on the relative map to find the current floor from the landmark ID and the relevant position. An error handling system is developed to assist the stable position reader which has the ability to detect wrong readings and correct them based on a hybrid scenario.

An innovative backbone path planning method with flexible goal selection is proposed to optimize planning speed and the number of paths required to reach the destination. A hybrid Backbone-Floyd method is then developed to benefit from the high speed of the Backbone and the flexibility of the Floyd algorithms. The Floyd algorithm is employed to generate a fresh path from the current position to the destination in cases of collision avoidance, blocked paths, defective doors or any other causes of backbone paths being unavailable. The hybrid path planning method saves time in life sciences automation management by sending information to higher levels about transportation problems earlier to allow task rescheduling. The Floyd algorithm is optimized for indirect graph to minimize the time taken by approximately 2.267 times. An intelligent transportation management system (ITMS) is also developed to execute the Backbone tasks and choose the suitable path planning method, to order the robot's arm to perform grasping, placing, and button-pressing operations at specific times, to postpone transportation tasks in case where dynamic obstacles appear and to enable the collision avoidance system to send a movement request for execution. The ITMS then resumes transportation by controlling hybrid path planning so as to generate a new path to the current goal.

The movement core is developed to receive the path generated and to execute it with the required control of automated doors and elevators and an automated charging function until the defined goal is reached with high levels of accuracy.

To enable the navigation of the wheeled mobile robot (H20) between facilities distributed on different floors, elevators must be handled appropriately. There are many critical issues with this process, including the detection of elevator entry and destination buttons and recognizing the current floor. The system proposed in this study must work in a semi-outdoor environment, since the elevator utilized at Celisca is a glassy elevator which adds further difficulties to elevator handler operations due to the effect of direct sunlight on sensors, especially for entry button detection. Therefore, an innovative enhancement function is developed based on hardware camera calibration to eliminate the effects of direct sunlight. The entry button detection method utilizes image segmentation and blob analysis to rapidly extract the button's position. After button recognition, the depth sensor maps its real world x, y, and z coordinates which are extracted for the button-pressing operation.

After internal button detection, the label of each button is distinguished using the same strategy as for entry buttons using optical character recognition (OCR) in order to select the requested floor.

A robust floor estimation method based on a height measurements system is utilized to avoid the effect of reflected sunlight and human obstacle. The system consists of an LPS25HB

pressure sensor and STM32L053 microcontroller. A one-step calibration method is used to overcome the problem of soldering drift. Efficient smoothing filters and an adaptive calibration stage are utilized to handle oscillations and the wide variations in pressure sensor readings. In addition to external and internal buttons detection, there are many other aspects the elevator handler system must be concerned for successful interaction with the elevator. The EHS employs an ultrasonic sensor to detect the door's open or closed status and a passive landmark is used with the SGM for localization inside the elevator. A kinematic module in the robot's arm is utilized for the pressing operation after the extracted real world position has been sent to it. The arm API translates this position into the required joint movement based on the inverse kinematic solution. Arm joint movements direct the end effector to the button to press it.

To guarantee the safety of the elevator handling system, the elevator is equipped with an interface to control its operations over a Wi-Fi socket. The robot's movement core utilizes the localization method to find the floor it is currently on so as to call the elevator to it. Inside the elevator, the movement core receives information on the status of the transportation task (grasping operation done, placing operation done, or charging operation) in order to specify the destination floor to direct the elevator to it.

In addition to the navigation system and EHS, the robot must deal with some other issues in order to completely automate the transportation task. Firstly, automated door management is considered to control the automatic doors on more than one floor. Multiple network sockets are established to connect the distributed automated doors. This system serves the navigation system by providing the ability to open or close a specific door during a robot transportation task. Secondly, a charge management system is developed to direct the robot to the charging station safely and smoothly when its battery needs charging. The system will deal with many robots, with multiple charging stations installed at different locations with a specific ID for each robot. Each robot has the ability to recognize its charge status. This system continually monitors battery parameters such as voltage and temperature as well as robot position to direct it to or from a charging station.

The MFS has a number of system APIs to connect to the RAKM, the EHS, the collision avoidance controller, and the automated door controller. In general, the MFS controls the execution of the following mobile robot operations during the multi-floor transportation process. The elevator handling system allows the robot to detect aspects of the elevator environment by recognizing the elevator buttons and reading the floor number in real-time. Meanwhile the collision avoidance system detects obstacles in the robot's path and avoids them using human-robot interactions. The robot's arm kinematic module completes grasping,

placing and button-pressing manipulations, while the IMADCS controls the opening and closing of all of the laboratory doors during the robot's dynamic movements. Finally, the robot's motion control system allows it to accomplish different kinds of movement.

A series of experiments has been conducted to validate the performance of the different parts of the multi-floor laboratory transportation system, such as the multiple floor navigation system (including mapping, localization and path planning), Elevator Handling System (button detection algorithms with the utilized arm, the automated elevator management system, and floor estimation based on pressure sensor), automated door management system, communications sockets, battery charging management system, and error handling strategies.

The experimental results show that the new relative global map success to cover a large scale map in multi-floor environment and it has the ability to expansion to n number of floor without changing in the code. It extended to other robots easily since the same map and path work for any H2O mobile robot. A new hybrid Backbone- Floyd method proves its efficiency for path planning under different condition (multiple source, multiple destination, random movement of obstacle avoidance, arm failure in grasping). The Internal Transportation Management System (IMTS) success to manage the transportation tasks and manage various sub-system (elevator handler, automated door automated elevator, movement core, collision avoidance operations, arm operations) till reaches to the goals with higher precision reaches to  $\pm 2$  cm. The multi-floor door controller management system success to call the door in appropriate time during the transportation. The developed elevator handler system proves its higher efficiency to connect the distributed laboratories in different floors based mobile robot as transporter. In spite of high precision of the multi-floor navigation system to reach the entry pressing position and the higher accuracy ( $\pm 2$ cm) of the detected buttons (99%), the weakness of the H2O joint arms makes the elevator handler based computer vision with arm manipulation inefficient thus, the automated elevator management system with a pressure sensor based current floor estimation is employed instead. The developed sockets success to interact with multiple robot management level (RRC) to receive the transportation orders and to control the distributed related sub-system inside the mobile robot. Finally, all these functions works together to achieve a multi-floor transportation tasks smoothly, effectively and efficiently in multi-floor laboratories transportation.



## Chapter 7      OUTLOOK

The positive results obtained in this research also indicate potential areas of improvements, such as:

- A hybrid localization system could be used instead of the current error handler. In cases where the SGM failed to extract the landmark, the hybrid method would use an inertial measurement unit to check the current position based on the last position identified by the multi-floor localization system. The inertial measurement is preferred to motor optical encoding since the latter accumulates error arising from friction with the floor, and when the robot hits an obstacle the encoder continues to count pulses and produces an incorrect position.
- The mobile robot battery could be improved with larger capacity to increase the number of transportation tasks possible for each robot and to allow the robot to work in a larger area without losing its power.
- An efficient multiple robot management system with more flexibility would include:
  - Charger station status: to make the charging station available for any robot, with no specific station for each robot.
  - Robot distributed around work space: distributed charge station near the laboratories and the multiple robot management become responsible to provide information about the next chg. station after transportation to minimize the losing transportation time and make robot ready for next transportation, additionally and after distributed the robot around the laboratories the mobile robot selection must base on both the robot battery value and the nearest robot to the first operation(Grasping) in transportation task.
  - Response directly to the error messages from the MFS to minimize the losses in automation time.
- The design of an efficient method for robot-robot interaction management, especially when the working environment has multiple robot, would allow more than one robot can use the same path, for example in a corridor, while avoiding collision.
- An efficient arm management system with an automatic calibration function would be able to compensate for the error in the arm joints for the weak servo motor.



## REFERENCES:

---

- [1] D. S. Lütjohann, N. Jung, and S. Bräse, "Open source life science automation: Design of experiments and data acquisition via 'dial-a-device,'" *Chemom. Intell. Lab. Syst.*, vol. 144, pp. 100–107, 2015.
- [2] B. Göde, S. Holzmüller-Laue, and K. Thurow, "New approach of standard-based end-to-end process automation in the life science automation," *Chem.-Ing.-Tech.*, vol. 87, no. 5, pp. 518–536, 2015.
- [3] Herbert T Gilkey, "New Air Heating Methods," presented at the BRI fall conferences, Washington: National Research Council (U.S.), 1959, p. 60.
- [4] K. Sakaki *et al.*, "Localized, macromolecular transport for thin, adherent, single cells via an automated, single cell electroporation biomanipulator," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 11, pp. 3113–3123, 2013.
- [5] W. Gecks and S. Pedersen, "Robotics-an efficient tool for laboratory automation," *IEEE Trans. Ind. Appl.*, vol. 28, no. 4, pp. 938–944, Jul. 1992.
- [6] N. Miyata, J. Ota, T. Arai, and H. Asama, "Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 769–780, Oct. 2002.
- [7] J. F. Montgomery, A. H. Fagg, and G. A. Bekey, "The USC AFV-I: a behavior-based entry in the 1994 International Aerial Robotics Competition," *IEEE Expert*, vol. 10, no. 2, pp. 16–22, Apr. 1995.
- [8] J. G. Petersen, S. A. Bowyer, and F. R. y. Baena, "Mass and Friction Optimization for Natural Motion in Hands-On Robotic Surgery," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 201–213, Feb. 2016.
- [9] T. Chapman, "Lab automation and robotics: Automation on the move.," *Nature*, vol. 421, no. 6923, pp. 661, 663, 665–666, 2003.
- [10] J. Lemburg *et al.*, "AILA - design of an autonomous mobile dual-arm robot," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5147–5153.
- [11] C. Ramer, J. Sessner, M. Scholz, X. Zhang, and J. Franke, "Fusing low-cost sensor data for localization and mapping of automated guided vehicle fleets in indoor applications," in

2015 *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2015, pp. 65–70.

[12] “Statistics - IFR International Federation of Robotics.” [Online]. Available: <http://www.ifr.org/industrial-robots/statistics/>. [Accessed: 11-Oct-2016].

[13] F. Hua, G. Li, F. Liu, and Y. Liu, “Mechanical design of a four-wheel independent drive and steering mobile robot platform,” in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, 2016, pp. 235–238.

[14] B. Li, Y. Fang, and X. Zhang, “Visual Servo Regulation of Wheeled Mobile Robots With an Uncalibrated Onboard Camera,” *IEEEASME Trans. Mechatron.*, vol. 21, no. 5, pp. 2330–2342, Oct. 2016.

[15] T. Fujita, T. Sasaki, and Y. Tsuchiya, “Hybrid motions by a quadruped tracked mobile robot,” in *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2015, pp. 1–6.

[16] C. J. G. Aliac and E. A. Maravillas, “Design analysis and implementation of a two legged robot system with dynamics and locomotion using various A.I. algorithms,” in *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2015, pp. 1–10.

[17] B. Kim, H. Shim, S.-Y. Yoo, B.-H. Jun, S.-W. Park, and P.-M. Lee, “Operating software for a multi-legged subsea robot CR200,” presented at the OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension, 2013.

[18] H. Qin *et al.*, “Design and implementation of an unmanned aerial vehicle for autonomous firefighting missions,” in *2016 12th IEEE International Conference on Control and Automation (ICCA)*, 2016, pp. 62–67.

[19] A. Ridolfi *et al.*, “FeelHippo: A low-cost autonomous underwater vehicle for subsea monitoring and inspection,” in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, 2016, pp. 1–6.

[20] L. Ribas-Xirgo, “A virtual laboratory of a manufacturing plant operated with mobile robots,” in *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–4.

[21] H. S. Hong *et al.*, “Development of work assistant mobile robot system for the handicapped in a real manufacturing environment,” in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005*, 2005, pp. 197–200.

- [22] B. Horan, Z. Najdovski, T. Black, S. Nahavandi, and P. Crothers, "OzTug mobile robot for manufacturing transportation," in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 3554–3560.
- [23] M. Uhart, O. Patrouix, and Y. Aoustin, "Improving manufacturing of aeronautical parts with an enhanced industrial Robotised Fibre Placement Cell using an external force-vision scheme," *Int. J. Interact. Des. Manuf.*, vol. 10, no. 1, pp. 15–35, 2016.
- [24] B. Sulle, A. Adriansyah, and S. S. Dewi, "Coordination of mobile-robot system with behavior based architecture," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 16, pp. 7179–7183, 2015.
- [25] Y. Ishizuka and K. Hidaka, "Motion control for an automated guided vehicle in the factory with a stereo camera," *IEEJ Trans. Ind. Appl.*, vol. 135, no. 8, pp. 836–846, 2015.
- [26] D. Fontanelli, D. Macii, and T. Rizano, "A fast and low-cost vision-based line tracking measurement system for robotic vehicles," *Acta IMEKO*, vol. 4, no. 2, pp. 90–99, 2015.
- [27] "Hospital within an autonomous transport robot HOSPI<sup>®</sup>." [Online]. Available: <http://news.panasonic.com/press/news/official.data/data.dir/2013/10/jn131024-1/jn131024-1.html>. [Accessed: 04-Oct-2015].
- [28] "HOSPI-R drug delivery robot frees nurses to do more important work." [Online]. Available: <http://www.gizmag.com/panasonic-hospi-r-delivery-robot/29565/>. [Accessed: 04-Oct-2015].
- [29] R. Murai *et al.*, "A novel visible light communication system for enhanced control of autonomous delivery robots in a hospital," in *2012 IEEE/SICE International Symposium on System Integration (SII)*, 2012, pp. 510–516.
- [30] F. Carreira, T. Canas, A. Silva, and C. Cardeira, "i-Merc: A Mobile Robot to Deliver Meals inside Health Services," in *2006 IEEE Conference on Robotics, Automation and Mechatronics*, 2006, pp. 1–8.
- [31] H. H. Nguyen, T. N. Nguyen, R. Clout, A. Gibson, and H. T. Nguyen, "Development of an assistive patient mobile system for hospital environments," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013, pp. 2491–2494.
- [32] R. R. Cazangi, C. Feied, M. Gillam, J. Handler, M. Smith, and F. J. Von Zuben, "An evolutionary approach for autonomous robotic tracking of dynamic targets in healthcare environments," in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, 2007, pp. 3654–3661.

- [33] G. Dagnino, I. Georgilas, P. Tarassoli, R. Atkins, and S. Dogramadzi, "Vision-based real-time position control of a semi-automated system for robot-assisted joint fracture surgery," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 3, pp. 437–455, 2016.
- [34] K. Ito, S. Sugano, R. Takeuchi, K. Nakamura, and H. Iwata, "Usability and performance of a wearable tele-echography robot for focused assessment of trauma using sonography," *Med. Eng. Phys.*, vol. 35, no. 2, pp. 165–171, 2013.
- [35] M. Takahashi *et al.*, "Development of a mobile robot for transport application in hospital," *J. Robot. Mechatron.*, vol. 24, no. 6, pp. 1046–1053, 2012.
- [36] J. López, D. Pérez, R. Pinillos, S. Domínguez, E. Zalama, and J. Gómez-García-Bermejo, "Design and development of a reconfigurable transport system for hospital environments," *RIAI - Rev. Iberoam. Autom. E Inform. Ind.*, vol. 9, no. 1, pp. 57–68, 2012.
- [37] T. Kohtsuka, T. Onozato, H. Tamura, S. Katayama, and Y. Kambayashi, "Design of a Control System for Robot Shopping Carts," in *Knowledge-Based and Intelligent Information and Engineering Systems*, A. König, A. Dengel, K. Hinkelmann, K. Kise, R. J. Howlett, and L. C. Jain, Eds. Springer Berlin Heidelberg, 2011, pp. 280–288.
- [38] Vladimir A. Kulyukin and Chaitanya Gharpure, "Ergonomics-for-one in a robotic shopping cart for the blind," in *Proceedings of the 1st ACM SIGCHI/SIGART*, New York, NY, USA, 2006, pp. 142–149.
- [39] N. Matsuhira *et al.*, "Development of robotic transportation system - Shopping support system collaborating with environmental cameras and mobile robots -," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–6.
- [40] J. Sales, J. V. Martí, R. Marín, E. Cervera, and P. J. Sanz, "CompaRob: The Shopping Cart Assistance Robot," *Int. J. Distrib. Sens. Netw.*, vol. 2016, 2016.
- [41] Y. L. Ng, C. S. Lim, K. A. Danapalasingam, M. L. P. Tan, and C. W. Tan, "Automatic human guided shopping trolley with smart shopping system," *J. Teknol.*, vol. 73, no. 3, pp. 49–56, 2015.
- [42] V. Flores, J. F. Villa, M. A. Porta, and J. Gutiérrez, "Shopping Market Assistant Robot," *IEEE Lat. Am. Trans.*, vol. 13, no. 8, pp. 2559–2566, 2015.
- [43] N. Doering, S. Poeschl, H.-M. Gross, A. Bley, C. Martin, and H.-J. Boehme, "User-Centered Design and Evaluation of a Mobile Shopping Robot," *Int. J. Soc. Robot.*, vol. 7, no. 2, pp. 203–225, 2015.

- [44] M. Wojtczyk and A. Knoll, "Utilization of a mobile manipulator for automating the complete sample management in a biotech laboratory. A real world application for service Robotics," in *6th International Symposium on Mechatronics and its Applications, 2009. ISMA '09*, 2009, pp. 1–9.
- [45] T. Scherer *et al.*, "A service robot for automating the sample management in biotechnological cell cultivations," in *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, 2003, vol. 2, pp. 383–390 vol.2.
- [46] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "Mobile robotic transportation in laboratory automation: Multi-robot control, robot-door integration and robot-human interaction," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2014, pp. 1033–1038.
- [47] N. M. Thamrin *et al.*, "A turning scheme in the headland of agricultural fields for autonomous robot," *ARPJ. Eng. Appl. Sci.*, vol. 11, no. 5, pp. 3265–3269, 2016.
- [48] Q. Shi, H. Ishii, Y. Sugahara, A. Takanishi, Q. Huang, and T. Fukuda, "Design and Control of a Biomimetic Robotic Rat for Interaction with Laboratory Rats," *IEEEASME Trans. Mechatron.*, vol. 20, no. 4, pp. 1832–1842, 2015.
- [49] N. Le, C. Caillet, D. Morin, C. Baudiment, Y. Pegon, and A. Truchaud, "Development and qualification of an automated workcell that includes a multicapillary zone electrophoresis instrument," *JALA - J. Assoc. Lab. Autom.*, vol. 10, no. 1, pp. 54–58, 2005.
- [50] S. Holzmüller-Laue, B. Gode, and K. Thurow, "Model-driven complex workflow automation for laboratories," in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, 2013, pp. 758–763.
- [51] F. Weisshardt, U. Reiser, C. Parlitz, and A. Verl, "Making High-Tech Service Robot Platforms Available," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–6.
- [52] K. S. Fu, R.C. Gonzalez, and C.S.G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company, 1987.
- [53] R. Li, Z. Du, Y. Zhao, and S. Liu, "Design and implementation of mobile robot ultrasonic localization system," in *2016 Chinese Control and Decision Conference (CCDC)*, 2016, pp. 5347–5352.

- [54] D. Vivet, P. Checchin, and R. Chapuis, "Radar-only localization and mapping for ground vehicle at high speed and for riverside boat," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2618–2624.
- [55] Gerry, B.A., *Robot Design Handbook*. McGraw-Hill, 1988.
- [56] "ROTARY ENCODERS FA-CODER® | TAMAGAWA SEIKI CO.,LTD." [Online]. Available: <http://tamagawa-seiki.com/english/encoder/>. [Accessed: 17-Nov-2016].
- [57] "Potentiometer » Resistor Guide." [Online]. Available: <http://www.resistorguide.com/potentiometer/>. [Accessed: 04-Oct-2016].
- [58] S. J. Arif, M. S. J. Asghar, and A. Sarwar, "Measurement of Speed and Calibration of Tachometers Using Rotating Magnetic Field," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 4, pp. 848–858, Apr. 2014.
- [59] E. A. Bekirov, E. A. Ebubekirov, M. M. Asanov, and M. Y. Zaliskyi, "Compact electronic tachometer for unmanned aerial vehicles," in *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2015 IEEE International Conference*, 2015, pp. 63–65.
- [60] Milan Sonka, Vaclav Hlavac, and Roger Boyle, *Image Processing, Analysis, and Machine Vision*, 4th ed. Cengage Learning, 2013.
- [61] "CCD vs CMOS: What's the Difference? - Steve's Digicams." [Online]. Available: <http://www.steves-digicams.com/knowledge-center/how-tos/digital-camera-operation/ccd-vs-cmos-whats-the-difference.html#b>. [Accessed: 09-Oct-2016].
- [62] B. Lurstwut and C. Pornpanomchai, "Application of Image Processing and Computer Vision on Rice Seed Germination Analysis," *International Journal of Applied Engineering Research*, vol. 11, no. 9, pp. 6800–6807, 2016.
- [63] J. Cai *et al.*, "A normal method about separating leukocytes based on color features," in *2012 5th International Conference on Biomedical Engineering and Informatics (BMEI)*, 2012, pp. 151–155.
- [64] Y. Zou, W. Chen, and J. Zhang, "Edge map guided stereo matching in HSL color space for mobile robot navigation," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011, pp. 841–846.
- [65] R. Pan, W. Gao, and J. Liu, "Color Clustering Analysis of Yarn-dyed Fabric in HSL Color Space," in *WRI World Congress on Software Engineering, 2009. WCSE '09*, 2009, vol. 2, pp. 273–278.



- [66] M. Mete and U. Topaloglu, "Statistical comparison of color model-classifier pairs in hematoxylin and eosin stained histological images," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2009. CIBCB '09*, 2009, pp. 284–291.
- [67] N. J. Nilsson, "A Mobius Automation: An Application of Artificial Intelligence Techniques," in *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 1969, pp. 509–520.
- [68] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 126–148, Feb. 2000.
- [69] S. M. Jeong, T. H. Song, J. H. Park, and J. W. Jeon, "The local minimum escape using the grid map," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008*, 2008, pp. 378–383.
- [70] H. Jo, H. Choi, S. Jo, and E. Kim, "Grid mapping adaptive to various map sizes for Sbot," in *2013 13th International Conference on Control, Automation and Systems (ICCAS)*, 2013, pp. 1678–1680.
- [71] Y. Endo and R. C. Arkin, "Anticipatory robot navigation by simultaneously localizing and building a cognitive map," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings*, 2003, vol. 1, pp. 460–466 vol.1.
- [72] S. W. Yoon, S.-B. Park, and J. S. Kim, "Kalman filter sensor fusion for Mecanum wheeled automated guided vehicle localization," *J. Sens.*, vol. 2015, 2015.
- [73] J. López, C. Watkins, D. Pérez, and M. Díaz-Cacho, "Evaluating different landmark positioning systems within the RIDE architecture," *J. Phys. Agents*, vol. 7, no. 1, pp. 3–11, 2013.
- [74] S. Jang, K. Ahn, J. Lee, and Y. Kang, "A study on integration of particle filter and dead reckoning for efficient localization of automated guided vehicles," in *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2015, pp. 81–86.
- [75] G. Dudek and M. Jenkin, "Inertial Sensors, GPS, and Odometry," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 477–490.
- [76] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 572–578, Jun. 1999.

- [77] R. Mazl and L. Preucil, "Sensor data fusion for inertial navigation of trains in GPS-dark areas," in *IEEE Intelligent Vehicles Symposium, 2003. Proceedings*, 2003, pp. 345–350.
- [78] A. Martinelli, N. Tomatis, and R. Siegwart, "Simultaneous localization and odometry self calibration for mobile robot," *Auton. Robots*, vol. 22, no. 1, pp. 75–85, 2007.
- [79] J. Hidalgo-Carrio, A. Babu, and F. Kirchner, "Static forces weighted Jacobian motion models for improved Odometry," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 169–175.
- [80] M. Mazzucato, A. Valmorbida, S. Tronco, M. Costantini, S. Debei, and E. Lorenzini, "Development of a camera-aided optical mouse sensors based localization system for a free floating planar robot," in *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*, 2016, pp. 388–392.
- [81] H. Chung, L. Ojeda, and J. Borenstein, "Sensor fusion for mobile robot dead-reckoning with a precision-calibrated fiber optic gyroscope," in *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA*, 2001, vol. 4, pp. 3588–3593 vol.4.
- [82] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2520–2530, Dec. 2005.
- [83] M. Maimone, Y. Cheng, and L. Matthies, "Two years of Visual Odometry on the Mars Exploration Rovers," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, Mar. 2007.
- [84] Z. Qingxin, W. Luping, and Z. Shuaishuai, "Strap-down inertial navigation system applied in estimating the track of mobile robot based on multiple-sensor," in *Control and Decision Conference (CCDC), 2013 25th Chinese*, 2013, pp. 3215–3218.
- [85] M. L. Anjum *et al.*, "Sensor data fusion using Unscented Kalman Filter for accurate localization of mobile robots," in *2010 International Conference on Control Automation and Systems (ICCAS)*, 2010, pp. 947–952.
- [86] Y. Mori, M. Uchiyama, and A. Goto, "Position estimation for a mobile robot using a novel accelerometer: cantilever-type accelerometer with impedance detector," in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005, pp. 946–951.
- [87] J. Kim, M. Choi, and S. Lee, "In-situ sensor information filtering for pedestrian localization," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012, pp. 639–640.

- [88] X. Li, Q. Wang, and X. Zhang, "Application of Electronic Compass and Vision-Based Camera in Robot Navigation and Map Building," in *2013 IEEE Ninth International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2013, pp. 546–549.
- [89] K. Otsuka, K. Tanaka, N. Okada, and E. Kondo, "Image based rendering for a mobile robot using visual landmarks," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, 2005, pp. 875–880.
- [90] X. Chai, F. Wen, and K. Yuan, "Fast vision-based object segmentation for natural landmark detection on Indoor Mobile Robot," in *2011 International Conference on Mechatronics and Automation (ICMA)*, 2011, pp. 2232–2237.
- [91] W. K. Fung *et al.*, "Development of a hospital service robot for transporting task," in *2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings*, 2003, vol. 1, pp. 628–633 vol.1.
- [92] H. Hu and D. Gu, "Landmark-based navigation of mobile robots in manufacturing," in *1999 7th IEEE International Conference on Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99*, 1999, vol. 1, pp. 121–128 vol.1.
- [93] S. Atiya and G. D. Hager, "Real-time vision-based robot localization," *IEEE Trans. Robot. Autom.*, vol. 9, no. 6, pp. 785–800, Dec. 1993.
- [94] Y. Duan, R. Ding, and H. Liu, "A Probabilistic Method of Bearing-only Localization by Using Omnidirectional Vision Signal Processing," in *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, 2012, pp. 285–288.
- [95] C.-P. Lu, T.-H. Yeh, and W. Huang, "Mobile Care Robot Localization and Guidance Using Image Structure Map," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 3796–3801.
- [96] I. Hadda and J. Knani, "Global mapping and localization for mobile robots using stereo vision," in *2013 10th International Multi-Conference on Systems, Signals Devices (SSD)*, 2013, pp. 1–6.
- [97] K. Qian, X. Ma, F. Fang, and H. Yang, "3D environmental mapping of mobile robot using a low-cost depth camera," in *2013 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2013, pp. 507–512.

- [98] R. Sim and G. Dudek, "Mobile robot localization from learned landmarks," in , 1998 *IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998. Proceedings*, 1998, vol. 2, pp. 1060–1065 vol.2.
- [99] Y.-C. Lee, W. Yu, J.-H. Lim, W.-K. Chung, and D.-W. Cho, "Sonar Grid Map Based Localization for Autonomous Mobile Robots," in *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, 2008. MESA 2008, 2008*, pp. 558–563.
- [100] D. Wang, Y. Jia, and Z. Cao, "A Unique Multi-functional Landmark for Autonomous Navigation," in *Fifth International Conference on Natural Computation, 2009. ICNC '09, 2009*, vol. 5, pp. 269–273.
- [101] J. Choi, K. Lee, S. Ahn, M. Choi, and W. K. Chung, "A Practical Solution to SLAM and Navigation in Home Environment," in *SICE-ICASE, 2006. International Joint Conference, 2006*, pp. 2015–2021.
- [102] H. Li, L. Zheng, J. Huang, C. Zhao, and Q. Zhao, "Mobile Robot Position Identification with Specially Designed Landmarks," in *Fourth International Conference on Frontier of Computer Science and Technology, 2009. FCST '09, 2009*, pp. 285–291.
- [103] Y. Takahashi *et al.*, "Mobile robot self localization based on multi-antenna-RFID reader and IC tag textile," presented at the Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO, 2013, pp. 106–112.
- [104] F. Martinelli, "A Robot Localization System Combining RSSI and Phase Shift in UHF-RFID Signals," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1782–1796, 2015.
- [105] J.-H. Kim, G. Sharma, N. Boudriga, S. S. Iyengar, and N. Prabakar, "Autonomous pipeline monitoring and maintenance system: a RFID-based approach," *Eurasip J. Wirel. Commun. Netw.*, vol. 2015, no. 1, pp. 1–21, 2015.
- [106] A. Abdelgawad, "Auto-localization system for indoor mobile robot using RFID fusion," *Robotica*, vol. 33, no. 9, pp. 1899–1908, 2015.
- [107] E. Digiampaolo and F. Martinelli, "Mobile robot localization using the phase of passive UHF RFID signals," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 365–376, 2014.
- [108] E. McCann, M. Medvedev, D. J. Brooks, and K. Saenko, "Off the grid: Self-contained landmarks for improved indoor probabilistic localization," in *2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), 2013*, pp. 1–6.

- [109] J. Bae, S. Lee, and J.-B. Song, "Use of coded infrared light for mobile robot localization," *J. Mech. Sci. Technol.*, vol. 22, no. 7, pp. 1279–1286, 2008.
- [110] Y. Nakazato, M. Kanbara, and N. Yokoya, "Wearable augmented reality system using invisible visual markers and an IR camera," in *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, 2005, pp. 198–199.
- [111] E. Brassart, C. Pegard, and M. Mouaddib, "Localization using infrared beacons," *Robotica*, vol. 18, no. 2, pp. 153–161, 2000.
- [112] H. B. Wang and L. L. Zhang, "Path Planning Based on Ceiling Light Landmarks for a Mobile Robot," in *IEEE International Conference on Networking, Sensing and Control, 2008. ICNSC 2008*, 2008, pp. 1593–1598.
- [113] H. Wang, H. Yu, and L. Kong, "Ceiling Light Landmarks Based Localization and Motion Control for a Mobile Robot," in *2007 IEEE International Conference on Networking, Sensing and Control*, 2007, pp. 285–290.
- [114] S. Lee and I. Kweon, "Color landmark based self-localization for indoor mobile robots," in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, 2002, vol. 1, pp. 1037–1042.
- [115] A. Bais, R. Sablatnig, Y. M. Khawaja, and G. M. Hassan, "A hybrid approach towards vision based self-localization of autonomous mobile robots," in *International Conference on Machine Vision, 2007. ICMV 2007*, 2007, pp. 1–6.
- [116] T. Lee, W. Bahn, B. Jang, H.-J. Song, and D. D. Cho, "A new localization method for mobile robot by data fusion of vision sensor data and motion sensor data," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012, pp. 723–728.
- [117] H. Chae and S.-H. Kim with Y.-C. Lee, Christiad, "Applications of robot navigation based on artificial landmark in large scale public space," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011, pp. 721–726.
- [118] R. M. K. Chetty, M. Singaperumal, and T. Nagarajan, "Distributed formation planning and navigation framework for wheeled mobile robots," *J. Appl. Sci.*, vol. 11, no. 9, pp. 1501–1509, 2011.
- [119] A. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 969–978, 1998.

- [120] C.-J. Yu, Y.-H. Chen, and C.-C. Wong, "Path planning method design for mobile robots," in *2011 Proceedings of SICE Annual Conference (SICE)*, 2011, pp. 1681–1686.
- [121] J. K. Goyal and K. S. Nagla, "A new approach of path planning for mobile robots," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 863–867.
- [122] J. Guo, L. Liu, Q. Liu, and Y. Qu, "An Improvement of D\* Algorithm for Mobile Robot Path Planning in Partial Unknown Environment," in *Second International Conference on Intelligent Computation Technology and Automation, 2009. ICICTA '09*, 2009, vol. 3, pp. 394–397.
- [123] dchamber, "Shortest Path Algorithm Comparison." [Online]. Available: <http://rebusotechnologies.com/shortest-path-algorithm-comparison/>. [Accessed: 13-Oct-2015].
- [124] B. Park, J. Choi, and W. K. Chung, "An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 180–186.
- [125] J. Wang, Y. Sun, Z. Liu, P. Yang, and T. Lin, "Route Planning based on Floyd Algorithm for Intelligence Transportation System," in *IEEE International Conference on Integration Technology, 2007. ICIT '07*, 2007, pp. 544–546.
- [126] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "A floyd-genetic algorithm based path planning system for mobile robots in laboratory automation," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012, pp. 1550–1555.
- [127] F. K. Purian, F. Farokhi, and R. S. Nodoshan, "Comparing the Performance of Genetic Algorithm and Ant Colony Optimization Algorithm for Mobile Robot Path Planning in the Dynamic Environments with Different Complexities," *J. Acad. Appl. Stud.*, vol. 3, no. 2, Feb. 2013.
- [128] S.-H. Chia, K.-L. Su, J.-H. Guo, and C.-Y. Chung, "Ant Colony System Based Mobile Robot Path Planning," in *2010 Fourth International Conference on Genetic and Evolutionary Computing (ICGEC)*, 2010, pp. 210–213.
- [129] C. Arismendi, F. Martin, S. Garrido, and L. Moreno, "Smooth anytime motion planning using Fast Marching," in *2013 IEEE EUROCON*, 2013, pp. 1972–1979.
- [130] A. Benmachiche, B. Tahar, L. M. Tayeb, and Z. Asma, "A dynamic navigation for autonomous mobiles robots," *Intell. Decis. Technol.*, vol. 10, no. 1, pp. 81–91, 2016.

- [131] Y.-S. Park and Y.-S. Lee, "Path planning and obstacle avoidance algorithm of an autonomous traveling robot using the RRT and the SPP path smoothing," *J. Inst. Control Robot. Syst.*, vol. 22, no. 3, pp. 217–225, 2016.
- [132] N. Morales, J. Toledo, and L. Acosta, "Path planning using a Multiclass Support Vector Machine," *Appl. Soft Comput. J.*, vol. 43, pp. 498–509, 2016.
- [133] P. K. Mohanty and D. R. Parhi, "Optimal path planning for a mobile robot using cuckoo search algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 28, no. 1–2, pp. 35–52, 2016.
- [134] D. K. Jha, P. Chattopadhyay, S. Sarkar, and A. Ray, "Path planning in GPS-denied environments via collective intelligence of distributed sensor networks," *Int. J. Control*, vol. 89, no. 5, pp. 984–999, 2016.
- [135] S.-W. Ma, X. Cui, H.-H. Lee, H.-R. Kim, J.-H. Lee, and H. Kim, "Robust elevator door recognition using LRF and camera," *J. Inst. Control Robot. Syst.*, vol. 18, no. 6, pp. 601–607, 2012.
- [136] J.-G. Kang, S.-Y. An, W.-S. Choi, and S.-Y. Oh, "Recognition and path planning strategy for autonomous navigation in the elevator environment," *Int. J. Control Autom. Syst.*, vol. 8, no. 4, pp. 808–821, Aug. 2010.
- [137] J. A. Delmerico, J. J. Corso, D. Baran, P. David, and J. Ryde, "Ascending stairway modeling: A first step toward autonomous multi-floor exploration," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4806–4807.
- [138] T. He, M. Bando, M. Guarnieri, and S. Hirose, "The development of an autonomous robot system for patrolling in multi-floor structured environment," *Int. J. Autom. Technol.*, vol. 6, no. 1, pp. 13–21, 2012.
- [139] J. Baek and M. Lee, "A study on detecting elevator entrance door using stereo vision in multi floor environment," in *ICCAS-SICE, 2009*, 2009, pp. 1370–1373.
- [140] J.-G. Kang, S.-Y. An, and S. Oh, "Navigation strategy for the service robot in the elevator environment," in *International Conference on Control, Automation and Systems, 2007. ICCAS '07*, 2007, pp. 1092–1097.
- [141] X. Yu, L. Dong, L. Li, and K. E. Hoe, "Lift-button detection and recognition for service robot in buildings," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 313–316.

- [142] X. Liu and J. Samarabandu, "An edge-based text region extraction algorithm for indoor mobile robot navigation," in *Mechatronics and Automation, 2005 IEEE International Conference*, 2005, vol. 2, pp. 701–706 Vol. 2.
- [143] S. Kurogi, Y. Fuchikawa, T. Ueno, K. Matsuo, and T. Nishida, "A method to measure 3D positions of elevator buttons from a mobile robot using a 2D artificial landmark, a laser navigation system and a competitive neural net," in *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02*, 2002, vol. 4, pp. 2122–2126 vol.4.
- [144] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2751–2757.
- [145] D. Troniak *et al.*, "Charlie Rides the Elevator – Integrating Vision, Navigation and Manipulation towards Multi-floor Robot Locomotion," in *2013 International Conference on Computer and Robot Vision (CRV)*, 2013, pp. 1–8.
- [146] Chen, L.-K. and Hsiao, M.-Y., "Control of service robot by integration of multiple intermittent sensors," in *AMPT 2013*, Taipei; Taiwan, 2014, vol. 939, pp. 609–614.
- [147] K. Maneerat, C. Prommak, and K. Kaemarungsi, "Floor estimation algorithm for wireless indoor multi-story positioning systems," in *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, pp. 1–5.
- [148] L. Chen, B. Sun, and X. Chang, "MS5534B Pressure Sensor and Its Height Measurement Applications," in *2011 International Conference on Information Technology, Computer Engineering and Management Sciences (ICM)*, 2011, vol. 1, pp. 56–59.
- [149] J. H. Guo, K. L. Su, and S. H. Chia, "Development of the multi-level fusion based security system," *Appl. Math. Inf. Sci.*, vol. 9, no. 3, pp. 1521–1527, 2015.
- [150] J.-G. Juang, C.-L. Yu, C.-M. Lin, R.-G. Yeh, and I. J. Rudas, "Real-time image recognition and path tracking of a wheeled mobile robot for taking an elevator," *Acta Polytech. Hung.*, vol. 10, no. 6, pp. 5–23, 2013.
- [151] S.-W. Ma, X. Cui, H.-H. Lee, H.-R. Kim, J.-H. Lee, and H. Kim, "Robust elevator door recognition using LRF and camera," *J. Inst. Control Robot. Syst.*, vol. 18, no. 6, pp. 601–607, 2012.



- [152] J.-G. Kang, S.-Y. An, W.-S. Choi, and S.-Y. Oh, "Recognition and path planning strategy for autonomous navigation in the elevator environment," *Int. J. Control Autom. Syst.*, vol. 8, no. 4, pp. 808–821, 2010.
- [153] X. Gu, S. Neubert, N. Stoll, and K. Thurow, "Intelligent Scheduling Method for Life Science Automation Systems," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016) IEEE International*, Baden-Baden, Germany, 2016, pp. 156–161.
- [154] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "Mobile robot for life science automation," *Int. J. Adv. Robot. Syst.*, vol. 10, pp. 1–14, 2013.
- [155] I. Ul-Haque and E. Prassler, "Experimental Evaluation of a Low-cost Mobile Robot Localization Technique for Large Indoor Public Environments," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–7.
- [156] M. M. Ali, H. Liu, R. Stoll, and K. Thurow, "Arm grasping for mobile robot transportation using Kinect sensor and kinematic analysis," in *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*, 2015, pp. 516–521.
- [157] "Dr Robot Inc.: WiFi 802.11 robot, Network-based Robot, robotic, robot kit, humanoid robot, OEM solution." [Online]. Available: [http://www.drrobot.com/products\\_H20.asp](http://www.drrobot.com/products_H20.asp). [Accessed: 19-Jan-2015].
- [158] M. Clark, D. Feldpausch, and G. S. Tewolde, "Microsoft kinect sensor for real-time color tracking robot," in *2014 IEEE International Conference on Electro/Information Technology (EIT)*, 2014, pp. 416–421.
- [159] N. M. DiFilippo and M. K. Jouaneh, "Characterization of Different Microsoft Kinect Sensor Models," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4554–4564, Aug. 2015.
- [160] N. A. Zainuddin, Y. M. Mustafah, Y. A. M. Shawgi, and N. K. A. M. Rashid, "Autonomous Navigation of Mobile Robot Using Kinect Sensor," in *2014 International Conference on Computer and Communication Engineering (ICCCE)*, 2014, pp. 28–31.
- [161] "Intel RealSense SDK Architecture," *ProDigitalWeb*, 14-Nov-2014. .
- [162] "What is network socket? - Definition from WhatIs.com." [Online]. Available: <http://whatis.techtarget.com/definition/sockets>. [Accessed: 21-Oct-2015].
- [163] Z. Mi, Y. Yang, and J. Y. Yang, "Restoring Connectivity of Mobile Robotic Sensor Networks While Avoiding Obstacles," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4640–4650, Aug. 2015.

- [164] A. Birk, S. Schwertfeger, and K. Pathak, "A networking framework for teleoperation in safety, security, and rescue robotics," *IEEE Wirel. Commun.*, vol. 16, no. 1, pp. 6–13, Feb. 2009.
- [165] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "Multi-floor navigation method for mobile robot transportation based on StarGazer sensors in life science automation," in *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*, 2015, pp. 428–433.
- [166] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A New Robust Method for Mobile Robot Multifloor Navigation in Distributed Life Science Laboratories," *J. Control Sci. Eng.*, vol. 2016, Jul. 2016.
- [167] M. M. Ali, H. Liu, R. Stoll, and K. Thurow, "Intelligent Arm Manipulation System in Life Science Labs Using H2O Mobile Robot and Kinect Sensor," in *IEEE International Conference on Intelligent Systems (IS'16)*, Sofia, Bulgaria, 2016, pp. 382–387.
- [168] M. Ghandour, H. Liu, N. Stoll, and K. Thurow, "A Hybrid Collision Avoidance System for Indoor Mobile Robots based on Human-Robot Interaction," in *2016 17th International Conference on Mechatronics (ME2016)*, Prague, Czech Republic, 2016.
- [169] M. M. Ali, H. Liu, R. Stoll, and K. Thurow, "Multiple Lab Ware Manipulation in Life Science Laboratories using Mobile Robots," in *2016 17th International Conference on Mechatronics (ME2016)*, Prague, Czech Republic, 2016.
- [170] 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 *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016) IEEE International*, Baden-Baden, Germany, 2016, pp. 406–411.
- [171] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A Robust Method for Elevator Operation in Semioutdoor Environment for Mobile Robot Transportation System in Life Science Laboratories," in *International Conference on Intelligent Engineering Systems (INES), 2016 IEEE International*, Budapest, Hungary, 2016, pp. 45–50.
- [172] 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 (ME2016)*, Prague, Czech Republic, 2016.

## APPENDICES

### Appendix A. Algorithms

#### *Algorithm 1: Localization Method in Multi-Floor Environment*

Input: SGM Raw data in alone mode (IDSGM, XSGM, YSGM,  $\theta$ SGM).  
Input: Relative Map [(ID, XRe, YRe,  $\theta$ Re)1,1,..., (ID, XRe, YRe,  $\theta$ Re)m,n]  
m: Number of Floor.  
n: Number of landmark for each floor  
CF: Current Floor  
CLP: Current Landmark Position

- 1) **For** i=1:m ;Floor Selection
- 2)     **For** j=1:n ;landmark Selection
- 3)         **If** IDSGM =RelativeMap[i,1,j] ; search about matched Landmark ID
- 4)             CF = i
- 5)             CLP= j
- 6)         **End IF**
- 7)     **End For**
- 8) **End For**
- 9) Apply Equation (8), (9),(10) to correct the landmark orientation ( $x_{SGO}, y_{SGO}, \theta_G$ )
- 10) Apply Eq. (11) as following:  $X_G = XF_{CF,CLP} + X_{Re}F_{CF} + x_{SGO}$
- 11) Apply Eq. (12) as following:  $Y_G = YF_{CF,CLP} + Y_{Re}F_{CF} + y_{SGO}$

#### *Algorithm 2: Floyd searching process*

- 1) Initialize DM(i,j) and SM(i,j)
- 2) **Loop** k=1:n
- 3)     **Loop** i=1:n
- 4)         **Loop** j=1:n
- 5)             **If** DM(i,j)>DM(i,k)+DM(k,j)
- 6)                 DM(i,j)=DM(i,k)+DM(k,j);
- 7)                 SM(i,j)=SM(i,k);
- 8)             **End If**
- 9)     **End Loop**
- 10) **End Loop**
- 11) **End Loop**

*Algorithm 3 Optimized Floyd for indirect graph algorithm*

```
1) Initialize DM(i,j) with Inf for undefined vertex weight else with distance
2) Initialize SM(i,j) with -1 for undefined vertex weight else with connection
3) Loop k=1:n
4)   Loop i=1:n
5)     Loop j=1:n
6)       If i<j
7)         If DM(i,j)>DM(i,k)+DM(k,j)
8)           DM(i,j)=DM(i,k)+DM(k,j);
9)           DM(j,i)= DM(i,j);
10)          SM(i,j)=SM(k,j);
11)          SM(j,i)=SM(k,j);
12)        End If
13)      End If
14)    End Loop
15)  End Loop
16) End Loop
```

*Algorithm 4: A Reactive Path Planning Strategy*

```
Input: waypoints  $(x_1, y_1), \dots, (x_n, y_n)$ .
Input: Current robot position (X,Y, Floor Number)
Input: Backbone tasks error (e1-e9)
1) ShortestPath= {}
2) If (e1|e2|e3) & !Position1Done
3) ShortestPath =Apply Floyd algorithm to find the shortest path to the Position1 (Grasp).
4) Set Position1Done
5) Else if (e4|e5|e6) & Position1Done & !Position2Done
6) ShortestPath = Apply Floyd algorithm to find the shortest path to the Position2 (Placing).
7) Set Position2Done
8) Else if (e7|e8|e9)
9) ShortestPath = Apply Floyd algorithm to find the shortest path to the Position3 (Charging).
10) Set ChargeDone
11) End If
12) Return ShortestPath
```

*Algorithm 5: Elevator internal button detection*

**Input:** Captured Image  
DC =No of Detected Corner  
DF = destination floor  
DFB = Detected Floor button  
X, Y, Z =Real coordinates of DFB

- 1) **If** the Captured Image required correction
- 2) Image Enhancement
- 3) **End IF**
- 4) Remove selected background using HSL Filter
- 5) Search about connected pixels in image
- 6) Apply a filter to remove the selected objects based on (shape, size, and colour)
- 7) **If** DC=4
- 8) Selected the Button Panel as ROI
- 9) Grayscale conversion
- 10) Adaptive binary image threshold
- 11) Search about connected pixels inside the ROI
- 12) **Do**
- 13) Select object
- 14) Inverse object value
- 15) Horizontal image flip
- 16) DFB=OCR (Object)
- 17) **While** DF != DFB
- 18) X,Y,Z=Mapping into real coordinates (DFB)
- 19) **Return** X,Y,Z
- 20) **Else return** failure
- 21) **End IF**

*Algorithm 6: Elevator floor detection*

**Input:** Captured Image  
DF = destination floor  
DFR = Detected Floor Reader

- a) Grayscale conversion
- b) Adaptive binary image threshold
- c) Search about objects inside the captured Image
- d) Apply a filter to remove the selected objects based on (size)
- e) Inverse image
- f) **DO**
- g) Select object
- h) Horizontal object flip
- i) DFR = OCR (object)
- j) **While** (DF != DFR) || (!Object search end)
- k) **Return** DFR



## Appendix B. Network Protocol

### B.1. Communication Protocol between MFS & RRC

**Table B.1.1** RRC -> MFS

Order	Command Format
Transportation start	"GoTransport Position1 Position2 Info"
Reply Power Data	"RobotName Monitor FloorNo. Room IDSG Power1 Power2 Power3 PowerStatus1 PowerStatus2 PowerStatus3"

**Table B.1.2** MFS -> RRC (Currently)

Reply	Command Format
Robot Idle	"MONITOR 0.01 0.01 0.01 13.27v 13.56v 0.00 Using Using Using 2.55m 2.55m 2.55m 2012 2683 4 P2PWait"
Robot Busy	"MONITOR 0.01 0.01 0.01 13.27v 13.56v 0.00 Using Using Using 2.55m 2.55m 2.55m 2012 2683 4 P2PGo"
Robot Finish	"MONITOR 0.01 0.01 0.01 13.27v 13.56v 0.00 Using Using Using 2.55m 2.55m 2.55m 2012 2683 4 P2POver"

**Table B.1.3** MFS -> RRC (Possible)

Reply	Command Format
Robot is Busy with Another Transportation Task	RobotName+ " RobotNotReady"
Reply Transportation Start	RobotName+ " TransportStart"
Reply Error Start/End Position	RobotName + " Wrong Path/Parameter"
Reply no Path File	RobotName + " Missing Path File"
Reply Transportation Complete	RobotName + " TransportComplete"
Reply Transportation Task Stop	RobotName + " TransportProblem "

## B.2. Communication Protocol between MFS & RAKC

**Table B.2.1** MFS -> RAKC

Order	Command Format
Blind Grasp/Place Operation	(GRASP/ PLACE)_(RIGHTARM/ LEFTARM)_Value of US
Grasp/Place Operation based on	(GRASP/ PLACE)_Labware Number Value of US Distance
Button Pressing based on the detected Button Position	PUSH_(X Y Z)
Move Arm to initial button position	MoveArmInitial_Button1
Close the socket connection	CloseConnection

<sup>1</sup>: EntryButton, 1, 2, 3, 4, A, E

**Table B.2.2** RAKC -> MFS

Reply	Command Format
Current operation status	Finish/ Problem
Move mobile robot to the next labware group on the same station	(Go Right/ GoLeft)_Value of Position shift

## B.3. Communication Protocol between MFS & EHS

**Table B.3.1** MFS -> EHS

Order	Command Format
Start the entry button detection process	StartEntryButtonPosition
Start detection of a specific button	StartDestinationButtonDetection DestinationFloor1
Read the current floor based on vision	StartCurrentFloorDetection DestinationFloor1
Stop the last detection operation	StopDetection
Close Socket Connection	CloseConnection

<sup>1</sup>: 1, 2, 3, 4, A, E

**Table B.3.2** EHS -> MFS

Reply	Command Format
The detected entry button coordinates	ReplyEntryButtonPosition (X Y Z)1
The detected internal button coordinates	ReplyDestinationButtonPosition (X Y Z)1
The robot in a correct floor	ReplyCurrentFloorDetection

<sup>1</sup>: coordinates from camera reference



#### B.4. Communication Protocol between MFS & CAS

**Table B.4.1** MFS -> CAS (Gesture)

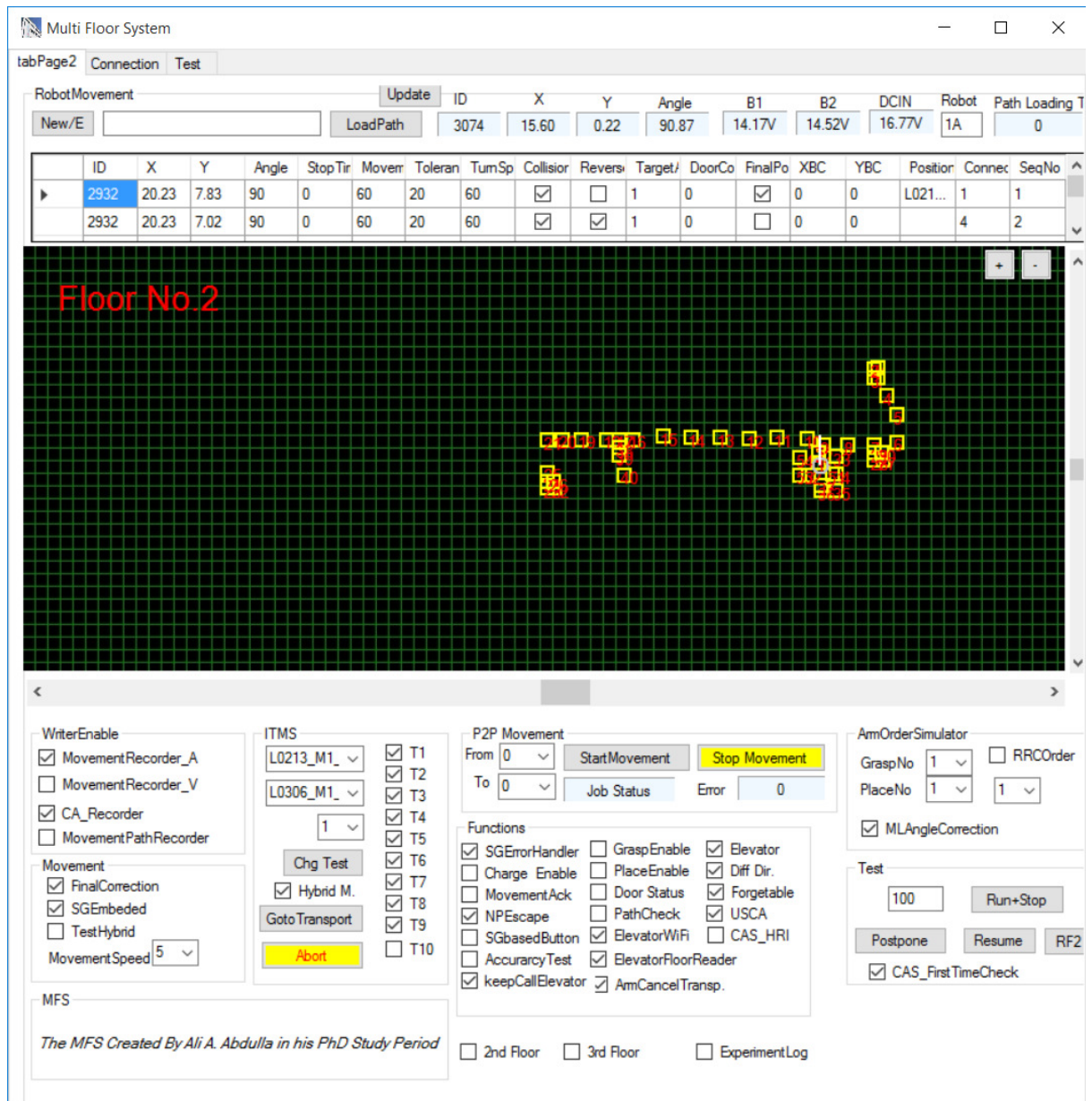
Order	Command Format
enable the CAS	StartCA
Impossible to complete the operation	CAParameters_False
Estimated time for the current movement	CAParameters_Time
Heartbeat monitor Signal	KeepOnline
Current operation is finished	CAParameters_Success
Start the avoidance process, the localization information and the current waypoint position	StartAvoidance_CurrentX_CurrentY_WaypointX_WaypointY

**Table B.4.2** CAS -> MFS (Gesture)

Reply	Command Format
Heartbeat monitor signal	KeepOnline
StartCA acknowledge	StartCA
Obstacle is detected	ObstacleDetection
Obstacle avoidance process is finished	ObstacleFree
Request to move the robot forward/backward or rotation in a specific time	CAParameters_Distance_DistanceTime_Rotation_RotationTime



## Appendix C. Developed GUI



Where:

- 1) WriteEnable is to enable/disable various groups of log recorder,
- 2) Movement Group:
  - a. Final Correction: to enable the high precision movement for the important transportation positions.
  - b. SGEmbedded: to enable the SG socket directly inside the MFS.
  - c. TestHybrid: to check the hybrid path planning method.
  - d. MovementSpeed: to select the required time to complete 1m of movement.

- 3) ITMS group as a human robot interaction to let the operator choose the start and end position in single/repeated transportation task, enabling the hybrid path planning method, and shows the enabled backbone task inside ITMS.
- 4) P2P Movement which is used to check the created path from single source to single destination.
- 5) Functions group which includes:
  - a. SGEError Handler to enable the wrong reading handling.
  - b. ChargeEnable: to enable the IBCMS.
  - c. MovementAck for a secure communication of movement orders.
  - d. NPEscape to overcome the nearest target problem which makes the robot unable to direct himself towards the next waypoint. After ten failure attempts, it controls the robot to move in back direction for specific distance then trying again to direct the robot.
  - e. Elevator: to enable the elevator handler.
  - f. ElevatorWiFi: to enable the automated elevator management system.
  - g. ElevatorFloorReader: to enable the pressure sensor as floor estimation.
  - h. ElevatorVA: using the computer vision and robotic arm to manipulate the elevator
  - i. AccuracyTest: to add delay in the Grasp/Place position for position recording.
  - j. KeepCallElevator: this function keeps calling the elevator when the robot inside the elevator until the elevator door is opened in the required floor.
  - k. GraspEnable/PlaceEnable: to enable the arm manipulation in these position.
  - l. Door status: to estimate the automated doors (opened/ closed).
  - m. PathCheck: to enable P2P Movement.
  - n. ArmCancelTransp.: to direct the robot back to charging station in case of arm failure.
  - o. Diff Direction: to add the correction waypoint value into reverse path.
  - p. USCA: simple avoidance based on ultrasonic which used as door opening status in elevator and automated doors.

## LIST OF PUBLICATIONS

- **A. Abdulla**, H. Liu, N. Stoll, and K. Thurow, "Multi-floor navigation method for mobile robot transportation based on StarGazer sensors in life science automation," in *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*, **2015**, pp. 428–433.
- **A. Abdulla**, H. Liu, N. Stoll, and K. Thurow, "A New Robust Method for Mobile Robot Multifloor Navigation in Distributed Life Science Laboratories," *J. Control Sci. Eng.*, vol. **2016**, Jul. 2016.
- **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 *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016) IEEE International*, Baden-Baden, Germany, **2016**, pp. 406–411.
- **A. Abdulla**, H. Liu, N. Stoll, and K. Thurow, "A Robust Method for Elevator Operation in Semioutdoor Environment for Mobile Robot Transportation System in Life Science Laboratories," in *International Conference on Intelligent Engineering Systems (INES), 2016 IEEE International*, Budapest, Hungary, **2016**, pp. 45–50.
- **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 (ME2016)*, Prague, Czech Republic, **2016**, [accepted].



## **DECLARATION**

This dissertation 'An Intelligent Multi-Floor Mobile Robot Transportation System in Life Science Laboratories' is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. The work of this dissertation has been done by myself under the guidance of Prof. Dr.-Ing. habil. Kerstin Thurow and Prof. Dr.-Ing. Norbert Stoll at the University of Rostock, Germany. Also the dissertation has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Rostock, 28 November, 2016

Ali A. Abdulla





# CURRICULUM VITAE (CV)

## **A- Personal:**

**Full name:** Ali Abduljalil Abdulla

**Surname:** Abdulla

**Designation:** Since **04/2014** doctoral studies University of Rostock, Celisca: Research Group "Life Science Automation – Mobile Robotics", Topic: "An Intelligent Multi-Floor Mobile Robot Transportation System in Life Science Laboratories".

Since **1/2009 – 10/2013** Assistant lecturer in mechatronics Engineering Dept. College of Engineering, Mosul University.

**Date of birth:** 25 / 1 / 1984

**Nationality:** Iraqi

**Marital status:** Married

**No. of Children** one

**Email:** alialtaee2008@gmail.com

Ali.Abdulla@celisca.de

## **B-Educational:**

### **Scientific Degrees:**

No.	Degree	University and College	Date	Grade	Field of Specialization
1.	B.Sc.	Foundation of Technical Education/College of Electrical and Electronic Techniques	2005	Good (3 out of 72)	Computer Engineering Technology
2.	M.Tech.	Foundation of Technical Education/College of Electrical and Electronic Techniques	2008	Very Good	Computer Engineering Technology

### ***Master Thesis***

“Design and Implementation of an autonomous robot control system”

### ***Teaching Experience:***

**Assistant Lecturer** in Mechatronics Engineering department, College of Engineering, Mosul University, Iraq 2008-2013

### ***Other Scientific and Social Activities:***

\* Member of the Iraqi Engineers Union.

### ***Other Skills:***

- Lecturing and supervising in the following laboratories in mechatronics engineering department through period 2009 ~ 2011: Microprocessor Lab., Electronic Lab., Robot Lab.(2010 ~ 2011), Automation Lab.(2011)
- Supervising on 4th Class Project in mechatronics engineering department as follows:
  - Project name in 2009 – 2010: Line Following Mobile Robot.
  - Project name in 2010 – 2011: Implementation of CNC Machine.
  - Project name in 2011 – 2012: Autonomous Lawn Mower.
  - Project name in 2011 – 2012: Mechanical structure Improvement for CNC Machine.
- Worked in a team to design and manufacture the following laboratory apparatuses in the Electrical and Electronic Technical College in Baghdad at 2004 ~ 2005: 8086 KIT, Real time KIT, Electronic kit, Electrical KIT, Measurement KIT.
- Working in some managerial committees like Examination Com., and Laboratories Apparatuses Importing in Mechatronics engineering department through 2008 ~ 2011.
- Good in Autonomous Mobile Robot.
- Good in Computer Interfacing, Microcontrollers, PLC, and CNC.
- V. good in Arabic (Mother language) and good in English.

### ***Awards:***

- Award due to post in the Festival of sovereignty from Ministry of Youth, Sport & Youth Directory in Nineveh, and Scientific Care Center in 2011.
- Award for third place on the sovereignty of the country at the Festival of the Fifth Scientific from College of Engineering University of Mosul in 2011.
- Letter of appreciation because of the efforts in the examination committee for the year 2010-2011.

## THESES

1. An intelligent multi-floor transportation system is developed to handle movement among automated laboratories distributed on different floors.
2. The presented transportation system is integrated in the hierarchal workflow management system which organizes the whole process of the automated laboratories.
3. New method is presented for mobile robot mapping in multi-floor environments which employs two kinds of mapping. Relative mapping generates a global map while a path map consists of a set of information waypoints positions which are related to the global map.
4. The presented mapping method can be extended into any number of floors with high accuracy and flexibility to be extended into any number of mobile robot without any changing in the code thus it minimizes the map generation time for multiple mobile robot.
5. A new method for mobile robot indoor localization is proposed which evaluates the robot's position in the global map. The localization method relies on passive landmarks and on a stargazer module. Each landmark has a unique id thus; all the laboratory zones can be correctly identified.
6. The ceiling landmarks are used to realize a low-cost solution capable of use in laboratories of any size. This method can be employed to localize any kind of robot inside a multi-floor environment.
7. An error handling system is developed to assist the stable position reader which has the ability to detect wrong readings and correct them based on a hybrid scenario.
8. A new hybrid method is developed for path planning to plan a path from multiple source points to the multiple destinations. This method has two working modes: path mapping (static paths) and an artificial intelligence algorithm.
9. A new method named as backbone method is used as an optimized static path planning method. The new Backbone path planning method uses a flexible goal selection method which provides innovations in terms of the optimization of planning time and the number of paths required to reach the destination.
10. The Floyd algorithm is employed to re-plan the path from any robot current position to the destination if backbone paths become unavailable due to complications such as collision avoidance, blocked paths, or failure in backbone tasks.
11. The Floyd algorithm is optimized for indirect graph (bi-directional path) to reduce the required time by approximately -56%.

12. The hybrid Backbone-Floyd method is developed to realize a robust, high-speed and flexible path planning approach.
13. The movement core is developed to receive the path generated and to execute it with the required control of automated doors and elevators and an automated charging function until the defined goal is reached with high levels of accuracy.
14. An intelligent transportation management system (ITMS) is also developed to execute the Backbone tasks and choose the suitable path planning method, to order the robot's arm to perform grasping, placing, and button-pressing operations at specific times, to postpone transportation tasks in case where dynamic obstacles appear and to enable the collision avoidance system to send a movement request for execution. The ITMS then resumes transportation by controlling hybrid path planning so as to generate a new path to the current goal.
15. An accurate position error management system is developed to achieve highly accurate robot positioning suitable for the limited H2O robot arm workspace and to adapt with multiple-labwares arrangement transportation task.
16. A new robust method for elevator operations for mobile robots is presented which allows the robots to move between different floors in semi-outdoor environments. Two different working strategies are developed to guarantee secure glass elevator handling,
17. Vision detection algorithms with robotic arm used for detection of entry button, internal buttons, current floor number, and control of the kinematic arm for the button-pressing operation by the mobile robot.
18. Automated elevator management system is developed to send an elevator call orders over Wi-Fi signals at appropriate time.
19. A robust floor estimation method based on a height measurements system is utilized to avoid the effect of reflected sunlight and to overcome the limitation of vision based floor reader. The system consists of an LPS25HB pressure sensor and STM32L053 microcontroller. A one-step calibration method is used to overcome the problem of soldering drift. Efficient smoothing filters and an adaptive calibration stage are utilized to handle oscillations and the wide variations in pressure sensor readings.
20. Two sub-networks are presented for the communication of the transportation. The robot external network is used to connect the multi-floor transportation system with the robot remote center level which sends the transportation orders. Meanwhile the robot internal networks are adopted for the communication of the multi-floor transportation system with the robotic hardware modules, the automated elevator, robot arm kinematic module, collision avoidance system, and with the automated doors.

21. The TCP/IP protocol is used to add the reliability and the expandability to the communication networks. The XML/UTF8 code is used to format the APIs control message among various software centers.
22. A charge management system is developed to direct the robot to the charge station smoothly for battery charging. To adapt this for a multi-robot system, a multiple charging station has been initialized with a unique ID for each robot.
23. A series of experiments have been executed to validate the performance of the developed system. The experimental results show that the proposed methods and sub-systems developed for the mobile robot are effective and efficient in providing a transportation service in multiple-floor life sciences laboratories.



## ZUSAMMENFASSUNG

In den letzten Jahren wurden immer mehr mobile Roboter für Transportaufgaben innerhalb von Gebäuden eingesetzt. In dieser Dissertation wird ein Mehrstockwerk - Transportsystem für mobile Roboter in Life Sciences Laboren vorgestellt. Dieses System enthält mehrere neue und innovative Aspekte. Zunächst wird eine neue Methode für die Kartierung von mobilen Robotern in mehrstöckigen Umgebungen präsentiert, die zwei Arten von Mapping verwendet. Relatives Mapping erzeugt eine globale Karte mit hoher Genauigkeit und Flexibilität für die Erweiterung auf eine beliebige Anzahl von Stockwerken, während eine Pfadkarte aus einer Reihe von Informationspfad-Positionen besteht, die mit der globalen Karte in Beziehung stehen. Diese Wegpunkte werden dann in der Pfadplanungsphase verwendet, um einen hindernisfreien Pfad zu identifizieren. Die Mapping-Methoden nutzen ein Vision basiertes System als Landmarkenleser, um die benötigten Informationen zu erhalten.

Außerdem wird eine neue Methode für die Lokalisierung von mobilen Robotern in Innenräumen vorgeschlagen, welche die Position des Roboters in der globalen Karte bewertet. Die Lokalisierungsmethode beruht auch auf passiven Landmarken, um eine kostengünstige Lösung zu realisieren, die in Labors jeder Größe eingesetzt werden kann. Diese Methode kann verwendet werden, um jede Art von Roboter in einer mehrstöckigen Umgebung zu lokalisieren.

Eine Hybridmethode für die Pfadplanung wird verwendet, um einen Pfad vom Quellpunkt zum Ziel zu planen. Diese Methode hat zwei Arbeitsmodi: Pfad-Mapping und ein künstlicher Intelligenz-Algorithmus. Das Backbone der Pfadplanung verwendet eine flexible Zielauswahlmethode, die Innovationen hinsichtlich der Optimierung der Planungszeit und der Anzahl der erforderlichen Pfade zum Erreichen des Ziels bereitstellt. Der Floyd-Algorithmus wird verwendet, um den Pfad von der aktuellen Position zum Ziel neu zu planen, wenn Backbone-Pfade nicht verfügbar sind. (Kollisionsvermeidung, blockiert Pfade oder fehlender Zugriff auf Backbone-Pfade) Dieser Algorithmus ist für indirekte Graphen (bidirektionaler Pfad) optimiert, um die benötigte Zeit um ca. -56% zu reduzieren. Die hybride Backbone-Floyd-Methode wurde entwickelt, um einen robusten, schnellen und flexiblen Pfadplanungsansatz zu realisieren. Ein Internes Transportation Management System erleichtert die entsprechende Pfadplanungsmethode basierend auf der aktuellen Situation.

Es wird eine neue robuste Methode für Aufzugsbetrieb eines mobilen Roboters vorgestellt, die es den Robotern ermöglicht, sich in halboffenen Umgebungen zwischen verschiedenen

Stockwerken zu bewegen. Zwei unterschiedliche Arbeitsstrategien werden entwickelt, um eine sichere Handhabung von verglasten Aufzügen zu gewährleisten, wobei automatische Aufzugs- und Sichtdetektionsalgorithmen verwendet werden. Bei der ersten Strategie werden Befehle über WLAN-Signale an den Aufzug gesendet, während die zweite sich mit der Eingabe, der Erkennung von internen Knöpfen und Aufzugstürstatus, aktueller Stockwerk-Schätzung und der Steuerung des kinematischen Arms für den Knopfdruckvorgang durch den mobiler Roboter befasst.

Ein Drucksensor verbunden mit einer Mikrocontroller-Platine und einem Glättungsfilter mit Auto-Kalibrierungsstufen werden präsentiert, die gemeinsam eine stabile Stockwerk-Ermittlung sowohl für den automatischen Aufzug als auch für das Vision-basierte Aufzugs Handler System verwendet wird.

Ein Software Kommunikationsinterface wurde eingerichtet, um das Kollisionsvermeidungssystem für die Erkennung und Umfahrung von menschlichen Objekten zu steuern, das Hindernisse erkennt und einen kollisionsfreien Pfad für den Roboter unter Verwendung einer Mensch-Roboter-Interaktion bestimmt. Währenddessen führt ein kinematisches Armmodul Handgriff-, Platzierungs- und Aufzugsknopfdruckvorgänge durch, und ein internes automatisiertes Türsteuersystem steuert das Öffnen und Schließen von Labortüren während der Transportaufgabe.

Ein Batterieladesteuersystem wurde entwickelt, um den Roboter reibungslos zur Ladestation zu führen. Um dies für ein Multirobotersystem anzupassen, wurde eine Mehrfachladestation mit einer eindeutigen ID für jeden Roboter initialisiert.

Es wurden zahlreiche Experimente zur Bewertung und Überprüfung der Leistung des Mehrstockwerk - Transportsystems durchgeführt und ausgewertet. Die experimentellen Ergebnisse zeigen, dass die vorgeschlagenen und entwickelten Methoden und Subsysteme für den mobilen Roboter effektiv und effizient geeignet sind, einen autonomen mobile Transport-Service in mehrstöckigen Life Science Laboren zu etablieren.