DEVELOPMENT OF SIMULATION MODEL OF A HOSPITAL FOR ROBOTIC SIMULATION IN ROS

KOH LING XIAN

UNIVERSITI TUNKU ABDUL RAHMAN

DEVELOPMENT OF SIMULATION MODEL OF A HOSPITAL FOR ROBOTIC SIMULATION IN ROS

KOH LING XIAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Mechatronics Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2022

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ________________________

Name : KOH LING XIAN

ID No. : 1702330

Date : 24/4/2022

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DEVELOPMENT OF SIMULATION MODEL OF A HOSPITAL FOR ROBOTIC SIMULATION IN ROS"** was prepared by **KOH LING XIAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Mechatronics Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	
Supervisor	:	IR. DR. DANNY NG WEE KIAT
Date	:	25/4/22

Signature	:	
Co-Supervisor	:	DR. KWAN BAN HOE
Date	:	25 Apr 2022

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to everyone who had contributed to the successful completion of this project. First and foremost, I would like to thank my research supervisor, Ir. Dr. Danny Ng Wee Kiat for his indispensable guidance, advice and professional knowledge provided throughout the research period. Furthermore, my appreciation also extends to my research co-supervisor, Dr. Kwan Ban Hoe for his invaluable patience and guidance. Both of them had kept me constantly engaged with my project. Their guidance is extremely significant for me.

In addition, my deep gratitude also goes to my family and friends who had never stopped encouraging and enlightening me. Furthermore, I am indebted to my course mates for their continuous support and sharing of ideas that had helped me in the completion of this project. Last but not least, I also thank Dr. Lee Jer Vui for his time and involvement in this project as my research moderator.

ABSTRACT

The outbreak of novel coronavirus disease 2019 (COVID-19) causes the healthcare workers to face an extremely high risk of getting infected as they need to interact with many different patients. Therefore, it indicates the importance of applying autonomous delivery robots within the hospital as it can help to reduce the contact between the healthcare workers and patients. It requires the robots to be tested and debugged in a reliable simulation environment before it can be tested in a real environment. A key problem in this area is the lack of simulation environment for hospital in Malaysia for the purpose of robotic simulation. Therefore, this project aims to develop a simulation model of a hospital for robotic simulation in Robot Operating System (ROS). A hospital inspection has been conducted to collect relevant information. Then, the simulation model of hospital is developed by using Gazebo, and connected with ROS 2 for robotic simulation purpose. In general, it is proven that simple robotic simulation such as mapping can be conducted smoothly in the developed simulation model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

3D	three-dimensional
AGV	automated guided vehicle
COVID-19	coronavirus disease 2019
DART	dynamic animation and robotics toolkit
DAE	digital asset exchange
FYP	final year project
GUI	graphical user interface
LRF	laser range finder
MKR	Muratec Keio robot
MOH	Ministry of Health
ODE	open dynamics engine
OGRE	object-oriented graphics rendering engine
PGM	portable gray map
ROS	robot operating system
RViz2	ROS 2 visualization
SDF	simulation description format
SLAM	simultaneous localisation and mapping
TKB	topography knowledge base
UCSF	University of California San Francisco
XML	extensible markup language

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

There exist more health threats that are spreading quickly around the world nowadays. On 11 March 2020, the outbreak of novel coronavirus disease 2019 (COVID-19) has been declared as a pandemic. According to the Ministry of Health (2021), the number of confirmed cases of the disease mentioned has exceeded 1.81 million cases in September 2021. Many healthcare workers have been infected with COVID-19. On top of that, nearly 17000 of them have lost their lives due to the outbreak of COVID-19. The doctors and nurses who are working in the hospitals face a very high risk of getting infected as they need to interact with many different patients in a closed environment.

Therefore, it explains the importance of applying autonomous logistics system within the hospital. Such system is able to reduce the contact between the healthcare workers and the patients and indirectly it helps to reduce the risk of the doctors and nurses to get infected with viruses such as COVID-19 virus. Autonomous vehicles or robots can be utilized in order to apply the autonomous logistics system and many hospitals have started to implement such a system. There are many different things that can be transported by using autonomous robots such as drugs, foods, documents, medical instruments, specimens, etc. According to Niechwiadowicz and Khan (2008), by implementing autonomous robots in delivering items within hospital, there are many benefits such as it is cost efficient, less mistakes will be made, transportation can be scheduled, and the healthcare workers can pay more attention to patients.

Niechwiadowicz and Khan (2008) stated that there are quite many commercial autonomous robots which have already been introduced and applied in the hospitals. For examples, HelpMate robot, Swisslog's TransCar AGV robot, HOSPI robot, TUG robot, SpeciMinder robot etc. These robots are able to deliver carts, blood samples, medical records, specimens and so on. Some are designed to transport only specific goods, but some are able to able to carry varied goods. At the same time, some of the robots mentioned above have the functionality of task scheduling but some do not (Niechwiadowicz and Khan, 2008).

Nevertheless, nowadays there are still many challenges and problems introducing and installing autonomous robots in hospital for delivery purpose especially when the robots need to operate under a highly populated environment. Firstly, one of the main challenges of the autonomous logistics system is the safety. The designer needs to ensure that collision between robots and human will not take place during the delivery process. Also, the robots need to be designed in a way that it is able to detect obstacle and take necessary steps when obstacle is detected. Moreover, path planning is also another significant challenges to overcome as a good path planning can ensure the system is efficient.

Besides, the navigation and logistics organization are also important challenges for such a system. The position and orientation of the robot needs to be known all the time when it is operating whereas the task ordering and scheduling are also required to be figured out properly. On the other hand, another challenge of implementing autonomous logistics system in hospitals is the installation of other infrastructures such as charging stations, delivery and pickup stations, and parking stations (Niechwiadowicz and Khan, 2008). It is important to overcome all the challenges in order to have an efficient and safe autonomous logistics system.

Overcoming all these challenges and problems requires numerous testings and debugging under different situations. However, it is impractical and dangerous to test the autonomous robot in the hospital at the early stage of developing the robot. Also, it will be both time and cost consuming to specifically build an environment to test the robots. Hence, testing and debugging the robot in a virtual simulation environment become a more practical and useful choice. A reliable simulation environment can help to save cost and time and it is safer too as risk of accidents will be reduced. Researchers stated in his studies that the software used can be applied directly to the real control of robots if the developed simulation environment and robot model are both designed properly during the development stage (Takaya et al., 2016).

According to their studies, the simulation environment developed using Robot Operating System (ROS) and Gazebo shows high usability. ROS is usually known as a set of different libraries, conventions, drivers and tools that can be applied for robotic application whereas Gazebo is a software where robots and sensors application can be applied in a three-dimensional environment. It can also be used to develop a three-dimensional simulation environment for robots testing purpose. Both ROS and Gazebo are able to have direct communication and therefore simulated and real robots can be controlled at the same time.

In short, this project aims to develop a simulation model of a hospital for robotic simulation in ROS. Its purpose is to provide a working environment for the testing purpose of an autonomous robot which is designed for transportation and delivery purpose.

1.2 Importance of the Study

The continuously increasing number of confirmed cases of COVID-19 has caused the medical burden to continue increasing. The workload of the healthcare workers and the number of patients they need to take care of have been increased drastically too. More healthcare workers such as doctors, nurses and volunteers have been infected because of having close contact with the infected patients during their service. Therefore, it indirectly shows that the application of autonomous logistics system within the hospital is important as it can reduce the needs of the healthcare workers especially the nurses to have direct interaction with the patients. The risk of getting infected will be reduced through this. Furthermore, it decreases the workload of the healthcare workers too and this will help the healthcare workers to focus more on other services that they can provide to the patients. In other words, they can focus better on their patients to fulfil their other needs instead of having to spend their time and energy on delivering items such as foods, medicines, specimens, documents etc.

This study is also significant as autonomous robots or autonomous logistics system are not only applicable in the health sector but also useful in other sectors as well. Hospital is a highly populated indoor environment in which offices, schools and factories also have the same characteristics where there are a lot of obstacles and moving people. Therefore, this study can also become a reference when autonomous robots are to be implemented at other places that share the same criteria.

Moreover, the other importance of this study is that it shows that virtual simulation environment is useful for testing and debugging the robot system including its control algorithm, sensors application etc. As mentioned earlier, this project will show how a hospital virtual simulation environment can be created using ROS and Gazebo and this can be applied in other sectors as well. For example, Gazebo and ROS can also be applied together to create other simulation environment such as offices, schools, or factories. Such simulation environment can shorten the time to develop a robot system as the developer team does not need to build a working environment to test the robot system. At the same time, a properly developed virtual simulation environment can also ensure the control algorithm of a robot system can be applied directly to the real robot in real environment.

1.3 Problem Statement

ROS and Gazebo are often applied together for simulation purpose as ROS is a good interface to control the robot while Gazebo is able to conduct the simulation in a three-dimensional (3D) scenario, at the same time providing properties such as gravity, inertia and illumination. However, it is used more often to simulate the robot system only or to create simple environment. It is seldom to see virtual hospital simulation environment created by using ROS and Gazebo. In short, a key problem in this area is the lack of simulation environment for hospital in Malaysia for the purpose of robot development, and the information and challenges of developing such environment is limited too. Thus, it is desirable to develop a simulation model of a hospital for robotic simulation by using ROS and Gazebo.

1.4 Aim and Objectives

The overall aim of this project is to conduct the development of simulation model of a hospital for robotic simulation in Robot Operating System (ROS) using Gazebo. More specifically, the particular objectives of this project are to:

- Collect information on the layout of a hospital and the operation of a hospital.
- Simulate the human and other objects in the simulated environment mimicking the data collected.

- Conduct simulation of the environment for testing the behaviour of the autonomous delivery robot.

1.5 Scope and Limitation of the Study

In this project, the development of the simulation environment will focus on a specific area or floor of the hospital, but not the whole hospital building. This is due to the limited time and knowledge. Besides, the delivery robots that will be used to implement in the simulation environment are assumed to be given tasks to transport food and medicine. Therefore, a kitchen and medicine dispensary should be included in the simulation model. In terms of the types of obstacles, static obstacles will be focused because dynamic obstacles are much unpredictable when compared to static obstacles.

1.6 Contribution of the Study

This project aims to develop a simulation model of a hospital which is capable of conducting robotic simulation. It provides a simulated world for the delivery robots to perform robotic simulation. This allows the robot development team to test and debug their robots within the hospital environment. By doing this, their development time and cost can be reduced as they do not need to perform their simulation at a real environment directly or to build a large-scale space to test their robot. The researchers can perform preliminary tests at the simulated world before they do it in a real world. Furthermore, this study also helps to identify the limitations and problems when developing a simulation world. Other researchers can take this as a reference when developing any other types of simulation world such as offices, kitchens and schools.

1.7 Outline of the Report

This report is divided into five chapters, namely introduction, literature review, methodology and work plan, results and discussions and conclusions and recommendations. The general introduction, importance of the study, contribution of the study, problem statement and aims and objectives are discussed in Chapter 1. On the other hand, literature review conducted on different topics such as classification of hospitals in Malaysia, technical guidelines and specifications for hospital environment, commercial delivery robot products used in hospitals, research on autonomous logistics system under development, challenges faced by delivery robots and the simulation software required for this project are included in Chapter 2. This chapter helps the author to better understand the background knowledge related to the project.

Furthermore, the methodology and work plan such as hospital inspection, software implementation and project activities and scheduling are explained in Chapter 3. The steps and processes needed to conduct this project are discussed in detail. Besides, Chapter 4 discusses results obtained and their relevant discussion. For instances, hospital floor plan, built simulation model and results of simulations are discussed. The important results and findings are presented and explained. Last but not least, the last chapter of this report presents the conclusions and recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to achieve the objectives of this project, there are several aspects of literature reviews required to be done. First of all, it is vital to understand the normal hospital environment so that the developed virtual simulation environment can be more realistic and reliable. Hence, the guideline and specifications of hospital facilities and layout should be studied. The way of classifying hospitals in Malaysia is also required to be given attention.

Furthermore, it is also significant to study on different types of delivery robots. This is to understand more regarding the current robots that have been developed and the robots that are still under development and research. Also, the challenges and problems faced by the autonomous logistics system is worthy to look into as well. Moreover, different types of simulation software such as Robot Operating System, V-REP and Gazebo will be discussed in this chapter too as they will be used to develop the simulation environment of the hospital. Necessary comparisons are also included.

2.2 Classification of Hospitals in Malaysia

According to Namawi (2000), Malaysia Referral System can be divided into three set ups including the primary care, secondary care and tertiary care. Hospitals are belonged to secondary care, and it consists of hospitals of different scales. The hospitals are categorized based on their number of beds provided. In her study, it is stated that the small district hospital provides 50 to 300 beds while the medium district hospital provides 300 to 500 beds. On the other hand, the large district hospitals have 500 to 750 beds. The district hospitals are usually located at the areas that consist of more than 40000 catchment population. Besides, the state or general hospitals usually provide less than 1000 beds whereas the national referral hospital provides more than 1000 beds. The district general hospitals are usually located only in those large states. For instance, Perak, Johore, Sarawak and Selangor (Nawawi, 2000). There are also other types of hospitals serving different purposes, but this subsection briefly discusses the classification of hospitals based on the number of beds they provide.

2.3 Technical Guideline and Specifications for Hospital Environment

The Ministry of Health Malaysia (MOH) has published the "Handbook on Setting Up of Private Hospital in Malaysia: Submission Process and Harmonisation of technical Requirements" in 2019. It is prepared by the Private Medical Practice Control Section of Medical Practice Division under the MOH, in collaboration with Malaysia Productivity Corporation. This handbook specifies some of the technical guideline and specifications that a hospital should follow. For instance, regulatory requirements for corridors, doors, ramps, stairways and elevators or lifts are included. To ensure the simulation model of hospital can fulfil these requirements, literature reviews on these facilities are carried out.

2.3.1 Corridors

According to the handbook published by the Ministry of Health Malaysia (2019), the width of corridors within the hospital shall be at least 2.1 meters. Besides, it is also stated that handrails shall be installed on both sides of corridors so that it can be more convenient for patients with physical disabilities. This should be applied to all the areas which are accessible by the patients for their safety purpose (Ministry of Health Malaysia, 2019).

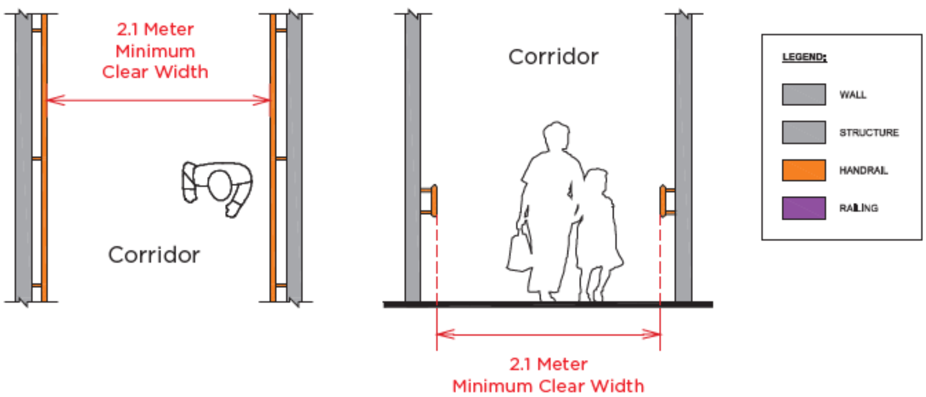


Figure 2.1: Illustration of Minimum Corridor Width (Ministry of Health Malaysia, 2019).

2.3.2 Doors

The doors where patients will be transported through shall have at least 1.2 meters clear opening according to the Ministry of Health Malaysia (2019). This includes doors in operating room, emergency room, X-ray room, patient room, recovery room and so on. However, there is no specified requirements for those doors which are not designed for patient's transportation such as doors for storeroom, receiving entrance and other doors as long as the width is adequate. Also, swinging of doors into the corridors are prohibited, except closet doors (Ministry of Health Malaysia, 2019).

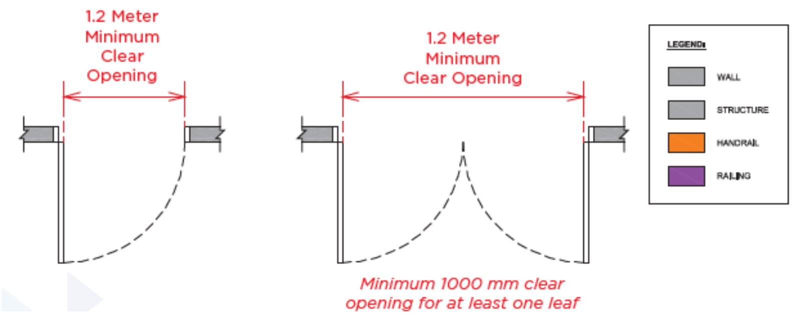


Figure 2.2: Illustration of Minimum Door Opening (Ministry of Health Malaysia, 2019).

2.3.3 Ramps

The Ministry of Health Malaysia (2019) also stated in the handbook that the ramps within the hospital shall ensure that its gradient will not exceed 1:16 and its minimum width shall be 1.1 meters. Besides, it is also mentioned that the minimum landing size shall be 1.8 meters wide to ease the egress or exit of patients (Ministry of Health Malaysia, 2019). Similar to the corridors, handrails shall be installed where necessary too.

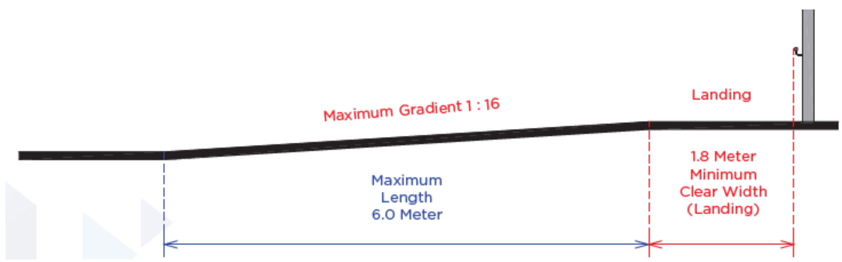


Figure 2.3: Illustration of Maximum Gradient and Length (Ministry of Health Malaysia, 2019).

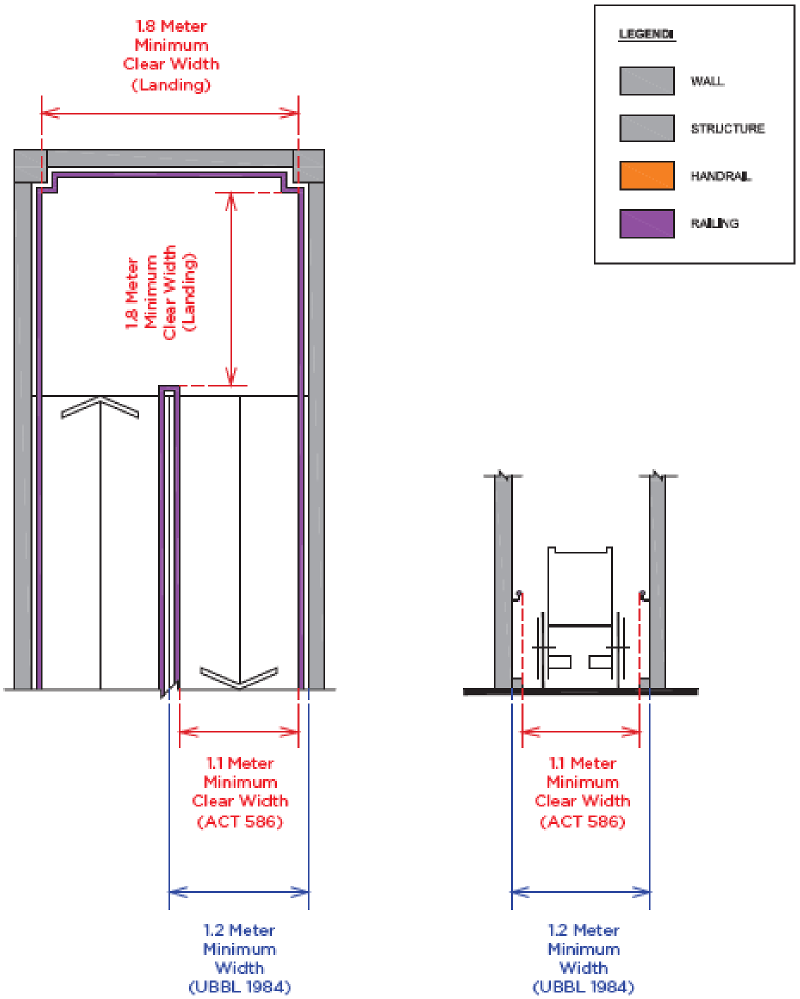


Figure 2.4: Illustration of Minimum Clear Width of Ramp with Side Railing (Ministry of Health Malaysia, 2019).

2.3.4 Stairways

Stairways that will be used for patients' transportation shall have a minimum width of 1.1 meters according to the Ministry of Health Malaysia (2019). Moreover, similar to the ramps' requirements, to ensure the exit or egress of patients can be done smoothly, it shall have a minimum landing size of 1.8 meters. Handrails are also required on both sides for the disabled persons too for safety purpose. Also, the handbook also mentioned that the maximum flights shall not exceed 16 steps (Ministry of Health Malaysia, 2019).

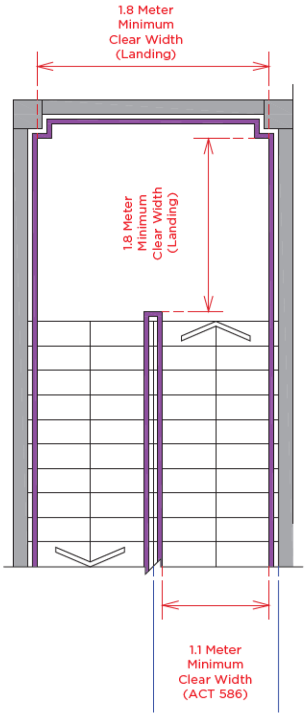


Figure 2.5: Illustration of Minimum Clear Width of Stairways (Ministry of Health Malaysia, 2019).

2.3.5 Elevators

The elevator which is used for patients' transportation shall have a minimum clear size of 1.5 meters by 2.1 meters (Ministry of Health Malaysia, 2019). Its minimum capacity shall be 1500 kilograms. For its door opening, it shall at least have a clear width of 1.2 meters. Also, it is recommended that at least one evacuation lift is provided in the hospital.

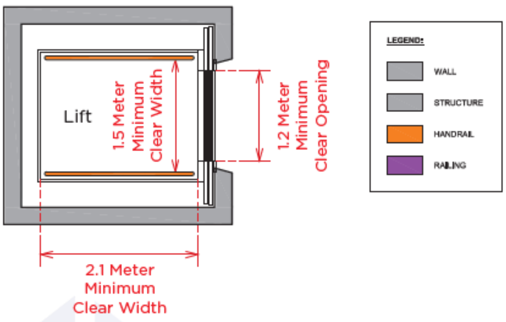


Figure 2.6: Illustration of Minimum Clear Opening and Width of Elevator (Ministry of Health Malaysia, 2019).

2.4 Commercial Delivery Robot Products Used in Hospitals

It is proven that autonomous robot system can help to reduce the cost and increase the efficiency of the hospital. Hence, different types of robots have been applied by some hospitals to help their daily operations. This subsection is presented to briefly discuss the current commercial robot products that have already been introduced to the market and utilized by the hospitals. The robot that will be discussed in this section is mainly used for delivery and transportation purposes. Their functionalities, advantages and weakness will also be analysed in a nutshell. This provides an insight to understand the current situation and challenges faced by autonomous robot system in the real-life situation.

2.4.1 HelpMate

HelpMate is a delivery robot that is able to carry out autonomous operation without supervision (Evans, 1994). It has the ability to work in hospital environment which is uncontrolled continuously without a break. However, Evans (1994) stated in his study that HelpMate is developed in the mid-1980's and the technologies applied can be improved further now. It can conduct its work to transport materials between different departments and nursing stations in the hospital. The user needs to specify the destination of the mission by using the graphical user interface (GUI) and HelpMate will then carry out the mission by planning its path.

A supervisory computer is used to ensure that communication between different HelpMate robots can be achieved so that conflict between robots can be arbitrated. Different mobile robots will compete for limited resources such as elevators and therefore the communication mentioned is important. A computer aided design program is used to construct the topography knowledge base (TKB) which is used by the robot to achieve mission planning. The receiver needs to verify that the materials or goods are successfully delivered. HelpMate is able to deliver lab, pharmacy supplies, meal trays, patient records and so on (Evans, 1994). It is worth mentioning HelpMate when discussing the delivery robot because HelpMate is always perceived as one of the pioneer robots that is able to conduct its mission in a highly populated environment such as hospital.

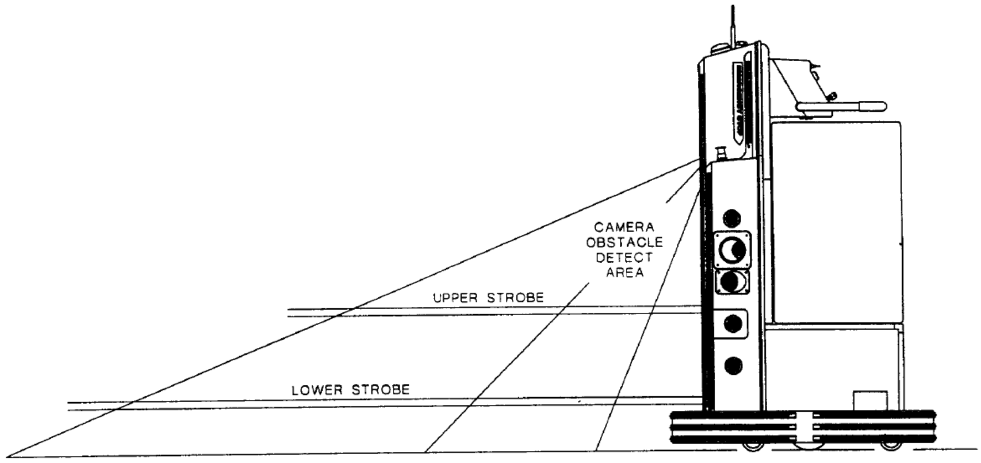


Figure 2.7: Schematic Diagram of HelpMate (Evans, 1994).

2.4.2 HOSPI

Panasonic (2019) has launched an autonomous delivery robot called HOSPI that can carry out its mission within the hospitals in 2013. HOSPI has the capability of self-navigating. It uses the data of the hospital's map, and it is able to coordinate with other physical facilities in hospitals such as automatic doors and elevators through the built network. To ensure that incident like collision will not take place, different types of sensors are installed on HOSPI so that obstacles with various shape and height can be detected properly. It is claimed that even obstacles such as cantilever table, walker, wheelchair, connected seats and IV stand can be detected with the sensors they applied (Panasonic, 2019).

HOSPI is usually used to transport things like medical samples, injections, medicines, etc. The medical staff or the healthcare workers need to scan their authentication tag when they need to move the things out from the robots. This is designed to increase the security level as the medicine or other things being delivered will not be stolen during the delivery process. A control centre is used to observe, record and control the location and trajectory of the delivery robots. The users can always observe the delivery process through the images sent from the HOSPI's front camera. Nowadays, HOSPI robot is not only applied within hospital, but it is also utilized for other applications. For instance, Beverage Delivery HOSPI can be utilized by restaurants or cafes to transport beverage upon order whereas Mobile Signage HOSPI can be used for advertisement or guidance purpose within airports or hospitals (Panasonic, 2019).

Figure 2.8: Authentication Tag is Used for Security Purpose (Panasonic, 2019).

2.4.3 Swisslog's TransCar AGV

Niechwiadowicz and Khan (2008) stated that a TransCar AGV (Automated Guided Vehicle) system has been developed a company called Swisslog. TransCar AGV serves the purpose to transport carts. It is designed to be a flat robot so that it can go under the carts easily. The operator needs to place the materials in the carts at a specific pick-up station. Then, he or she is required to enter the destination through a terminal which is mounted on wall. After that, the TransCar AGV will lift the cart and send it to the destination. Its benefit is that it is able to carry many different types of carts (Niechwiadowicz and Khan, 2008). However, TransCar AGV needs to work under a suitable environment where the hospital needs to make many changes to its facilities in order to install this system. Also, it also cannot handle simple items such as medicines, food and documents.

Figure 2.9: Swisslog's TransCar AGV (Niechwiadowicz and Khan, 2008).

2.4.4 SpeciMinder

Another delivery robot to de discussed in this subsection is the SpeciMinder. As its name suggests, SpeciMinder's main role is to deliver the lab specimens. It is stated that a lab technician spends up to 50% of his or her time to move the specimens around (CCS Robotics, 2008). Therefore, with SpeciMinder's help, the lab technicians can focus more on their study and research. In order to install this system, the hospital or laboratory needs to set up facilities such as central dispatch area, charging dock and delivery station.

The operators only need to press the simple illuminated pushbuttons to choose the destinations when they want to deliver the lab specimens. Then, the robot will pick the route which is most efficient and deliver the specimens to all the selected destination. It will then go back to the central dispatch area when the mission given is completed. When it is not in use, it will go to the charging dock automatically. Similar to most of the other delivery robots mentioned, when SpeciMinder meets any obstacles, it will either stop and wait until the obstacle is cleared or replan its route. It can also utilize the automatic doors and elevators and the communication can be done wirelessly. According to Niechwiadowicz and Khan (2008), SpeciMinder is utilized by Delaware Hospital in the United States, and it works successfully. The disadvantage of SpeciMinder is that the operators cannot schedule the delivery tasks (Niechwiadowicz and Khan, 2008).

Figure 2.10: SpeciMinder (CCS Robotics, 2008).

2.4.5 TUG

UCSF Medical Center (2017) stated in its website that their hospitals have applied 25 TUG robots to help in delivering food, specimens, linens and medications. They claim that this can help to increase the efficiency and they can spend more time in interacting with the patients. According to them, TUG is developed by a company called Aethon and it is an autonomous delivery robot which is designed for the hospital usage. TUG is able to self-navigate with the built-in sensors and map. It can also utilize resources such as automatic doors, fire alarms and elevators and the communication are performed via Wi-Fi.

Its working principle and functionalities are quite similar to Swisslog's Transcar AGV where it is also designed to be flat and able to lift the carts. The operators interact with TUG through simple buttons like "start", "stop" and "lift", and a touch screen is also built in. The touch screen is used to display the drop-off or pick-up task that will be carried on next. Similar to HOSPI, it also has a security functionality where the hospital staff can only open or close the TUG's compartment to take the goods through fingerprint identification (UCSF Medical Center, 2017).

Figure 2.11: TUG (UCSF Medical Center, 2017).

2.4.6 Summary of the Commercial Delivery Robot Products

Table 2.1 shows the summary of the commercial delivery robot products discussed in this subsection. The functionalities such as automatic elevator control, monitoring system, task scheduling, carts carrying, different types of goods transportation, and built on security system are compared.

Table 2.1: Summary of the Commercial Delivery Robot Products.

Functionality	Commercial Delivery Robot Products Used in Hospital				
	HelpMate	**HOSPI**	**TransCar AGV**	**SpeciMinder**	**TUG**
Automatic elevator control	Yes	Yes	Yes	Yes	Yes
Monitoring system	No	Yes	N/A	No	N/A
Task scheduling	No	N/A	N/A	No	N/A
Carry carts	No	No	Yes	No	Yes
Different types of goods transportation	Yes	Yes	Yes	No	Yes
Built on security system	N/A	Yes	N/A	N/A	Yes

2.5 Research on Autonomous Logistic System under Development

Few autonomous logistic systems that are still under development and research will be discussed in this subsection. For examples, i-Merc and Muratec Keio Robot. These systems are not introduced to the market yet and they are still under development. However, systems like this are still worth to be discussed to understand the challenges and problems faced by the delivery robots especially those which are designed for hospitals.

2.5.1 i-Merc

Carreira et al. (2006) stated that their team is developing a mobile robot called i-Merc which is used to transport meals within the hospital and health care centres. A unique point of this robot is that it has a built-in heating system. The built-in system can prevent bacterial growth and ensure the meals temperature is suitable for the consumers. The operators or the users can access the patients' diets information through an integrated personalised diets information system. They can access it via the touch screen to update or read the information anytime.

The robot will move to the kitchen from the park room when they need to deliver the food to the wards. Once i-Merc reaches the kitchen, the persons in-charge will place the meals either normal or personalised inside the compartment of the robot. Then, i-Merc will travel to the patients' ward and the persons in-charge at the wards will pass the food to the patients. Also, the persons in-charge at the wards will also place the soiled dishes back to the compartment so that i-Merc can bring them to the washing room. Finally, i-Merc will move back to the park room. Cleaning, checking and recharging will be carried out at the park room (Carreira et al., 2006).

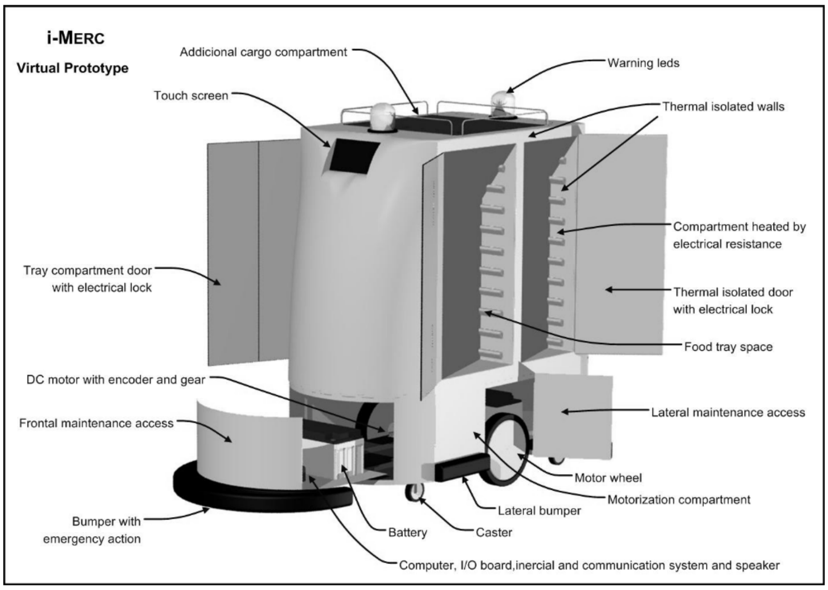


Figure 2.12: Virtual Prototype of i-Merc (Carreira et al., 2006).

2.5.2 Muratec Keio Robot

Takahashi et al. (2010) have been working to develop an autonomous mobile robot which is able to deliver goods such as luggage and specimens. The robot is called Muratec Keio Robot (MKR) and it applies a technique to ensure that collision with either static or dynamic obstacles will not happen. This method also ensures that the robot can move smoothly in a narrow aisle. They implement an omni-directional movement mechanism to achieve this. The sensors installed on MKR include ultrasonic sensors, Laser Range Finder (LRF) and stereo vision system.

Unsimilar to the other robots mentioned earlier in the previous subsection, MKR transports the material by pulling the wagon. The operators will put the goods or materials to be delivered in the wagon and the destination will be entered. MKR will then plan the route from the starting point until the destination point. Its algorithm will help it to avoid both the static and dynamic obstacles. The person in-charge needs to take the materials or goods delivered out of the wagon (Takahashi et al., 2010).

Figure 2.13: Different Models of MKR (Takahashi et al., 2010).

2.5.3 Summary of the Autonomous Logistic System under Development

Table 2.2 shows the summary of the autonomous logistic system under development. The functionalities compared include automatic elevator control, monitoring system, task scheduling, carts carrying, different types of goods transportation, built on security system and other special feature.

Table 2.2: Summary of the Autonomous Logistic System under Development.

Functionality	Autonomous Logistic System	
	i-Merc	**MKR**
Automatic elevator control	Yes	N/A
Monitoring system	N/A	N/A
Task scheduling	N/A	No
Carry carts	No	Carry wagons
Different types of goods transportation	No	Yes
Built on security system	N/A	N/A
Other special feature	Has a heating system	Applies omni-directional movement mechanism

2.6 Challenges Faced by Delivery Robots

Hospital environment is perceived as a highly populated environment and due to its special function to rescue lives, many factors are required to be taken into account when an autonomous logistic system or robot system is to be developed. This may help to provide insight into the potential problems when developing an autonomous logistic system.

2.6.1 Safety

Safety is always one of the most important challenges that cannot be ignored when developing any types of robots. In addition to that, hospital is normally a crowded place, and it has many static and dynamic objects or obstacles too. Therefore, the robot developer must make sure that the robot designed to transport materials or goods within hospital will not be harmful to humans. Also, they need to ensure that the robot can detect both the static and dynamic objects. Once these objects and obstacles are detected, the robots must know how to prevent collision so that unwanted incidents will not happen (Takahashi et al., 2010). The unpredictable behaviours of the moving objects especially human's movement makes the mission become more difficult. The problem becomes more challenging when there are many static and moving obstacles at the same time. Evans (1994) stated in his study that real hospital environment which is uncontrolled and unpredictable causes problems that industrial robots will not face. Hence, modification must be done to ensure the safety of humans.

2.6.2 Path Planning and Navigation

Another factor which can be considered as one of the most crucial challenges in designing an autonomous mobile delivery robot is the path planning and navigation. Path planning is vital because efficiency is important for a robot to complete its mission. Without a good path planning algorithm, the delivery robot might spend more time and resources for it to reach a certain point. Indirectly, it will waste the resources. Therefore, Niechwiadowicz and Khan (2008) mentioned that planning of the path from its starting point to destination point is required. Also, he claimed that it is also equally important to ensure that the robot can follow the pre-planned path calculated by its algorithm. The robot must also be able to generate another path if sudden blocking of path happens. This is to ensure that the robot can reach its destined position.

Moreover, the developer of the robot must focus on the navigation of the robot too. Niechwiadowicz and Khan (2008) also stated that the orientation and position of the mobile robot should always be known especially when it is carrying out its task. He claimed that artificial landmarks can be installed to help identifying the robot's position and orientation. However, in the same study, he also mentioned that it is better to have less extra artificial landmarks in hospitals. Thus, landmarks which are already existed in the hospital is preferable. For instance, lamps and corners. Nevertheless, this is another challenge that the developer team needs to face when designing an autonomous logistics system.

2.6.3 Obstacle Detection

As mentioned earlier, obstacle detection is important for safety reason. Without a good and promising obstacle detection, the robots cannot prevent collision and it might cause accidents which can lead to hospital's losses and injury. The challenging part of the obstacle detection is that the engineers need to take many factors into account. For example, the static objects that might appear in hospitals. More challenging ones are the moving obstacles such as hospital staff, patients, wheelchairs, and other mobile robots. Not only these obstacles are required to be identified, but the developers should also decide the types of sensors which should be installed on the robots (Niechwiadowicz and Khan, 2008). In addition to that, the position of the sensors is also vital to ensure that every area can be covered so that there will not be any blind spots.

2.6.4 Monitoring System

Niechwiadowicz and Khan (2008) stated in his study that it is vital to monitor the tasks carried out by the autonomous logistics system. This is to ensure that all the tasks and missions given to the mobile robot can be completed properly, and without any accidents. To do this, the engineers should make sure that a monitoring system will be developed too. This has become another challenge when developing a mobile robot. Also, the system should be able to ensure that the mission can still be completed if there is any failure happens. Thus, it becomes important for the developers to develop an algorithm so that the system can fix the failure itself and continue to complete the given missions or tasks.

2.6.5 Goods and Materials Handling

According to Niechwiadowicz and Khan (2008), one of the challenges that the mobile robots must face is that the goods and materials which will be transported need a stable handling. For examples, things such as lab specimen and meals. He also stated that people other than those who are in-charge should not be allowed to touch the items during the transportation. Specimens that are polluted might affect the results and more problems will be caused. On the other hand, it is furthered explained that goods like meals need to be given more attention and consideration too (Carreira et al., 2006). This is because contamination might happen during the transportation, and this might affect the patients' health. In short, the developer team needs to find out the ways to ensure the materials that will be transported can be handled in a good manner. Different types of goods have different requirement during transportation, and this is one of the main challenges.

2.6.6 Supporting Facilities

The implementation of autonomous logistics system within the hospital will bring another challenge to the hospital and the robot developer team too, which is some extra facilities are required to be set up. Charging station, parking station, pickup station and delivery station are required for certain autonomous logistic system. For example, according to UCSF Medical Center (2015), the TUG robots will need to return to their charging stations so that they can have sufficient energy for the next mission. Therefore, when designing the autonomous logistics system, locations and requirements of those mentioned facilities should also be identified. This problem is challenging because the system might require certain changes to be done on the existing layout and facility of the hospital.

Furthermore, it is also important to ensure that the system can communicate with the automatic doors and elevators of the hospital. This is to ensure that the mobile robots can travel automatically without human interference. Other than that, telecommunication system is another challenge too (Fanti et al., 2020). The planning of the network that will be utilized by the system should be conducted well so that all the units including humans can communicate well wirelessly.

2.7 Simulation Software

After developing a mobile delivery robot, testing, and modifying of the robot are always crucial to ensure that the robot developed can perform their tasks and missions well. However, creating a large physical simulation environment is time-consuming, and costs a lot too. Furthermore, it is very impractical to build a hospital environment to test the behaviour of the robots because hospital is a huge and populated environment. It is also dangerous to directly conduct the testing of robots at the actual hospital environment because it might be harmful to the hospital staff and patients.

It is stated that a proper environment modelling is required when it involves navigation, SLAM, and mapping for mobile robots (Lavrenov and Zakiev, 2017). Therefore, the use of proper simulation software become more important in this context. Thus, this sub-section will discuss the simulation software that will be used for developing the hospital environment.

2.7.1 Mapping, Localisation and Path Planning

Before discussing about the simulation software and methods to build a simulation model, it is important to understand few concepts. Lavrenov and Zakiev (2017) in their study mentioned that mapping, localisation, and path planning are three main important components when it comes to the autonomous robot's navigation. Firstly, mapping is a process to collect the environment's sensory data for future use. The data collected will be stored in a convenient form. Secondly, localisation means the process of determining the current position of a robot in an environment. Lastly, path planning is route searching process within a given environment, from the starting point to its destination point. Path planning should also take obstacles into account.

2.7.2 Robot Operating System (ROS)

The Robot Operating System (ROS) is an open-source and meta-operating system. It is a framework which consists of the collection of different libraries and tools which are useful for autonomous robots programming purpose (Linz et al., 2014). It is always perceived as a very useful software when a complex robotic system is required to be built as ROS consists of many different services and functions. Furthermore, ROS is also independent in terms of language and platform. For examples, it can be implemented in C++, Python, and LISP (Takaya et al., 2016). In other words, the user can develop their different nodes using different languages and the nodes are still able to communicate with each other without any problems.

One of the ROS' main tasks is to conduct data exchange between the processes. When using or discussing ROS, there are few important keywords that cannot be ignored such as "nodes" and "messages". Nodes are the processes in ROS whereas the messages are the data that will be used during exchange of data. Node can also be known as an individual module that is able to carry out its own computation. Normally, an autonomous robot system consists of multiple nodes. Each single node will be in-charge of different tasks. For the nodes to communicate with each other, they need to do it through "topic". The "publisher" will publish the messages to a topic and the "subscriber" will receive the messages from it. A node can publish and subscribe to more than one topic. On the other hand, a topic can be subscribed or published by more than one publisher or subscriber (Takaya et al., 2016).

As stated by Takaya et al. (2016), a Master Server is used to schedule the execution of process. There are many helpful tools and packages such as navigation tools, sensor drivers, path planning tools, environment mapping, visualization tools and so on. Rviz is one of the vital and useful tools that will be used frequently when it involves navigation. As its name suggests, Rviz is used when three-dimensional visualization is required. It is able to visualize different types of data in ROS in 3D manner. They also mentioned in their studies that when ROS is used together with Gazebo, a reliable robotic control can be achieved, and this will decrease the developing time a lot.

2.7.3 Gazebo

Gazebo is also an open-source software, and it can be supported by ROS. It is stated that the robotic and sensors application can be simulated by using Gazebo (Linz et al., 2014). The nodes developed in ROS and the simulation environment built in Gazebo can communicate well as they are able to send and receive messages between each other. The actuators and sensors can be applied through a plugin system too. As mentioned in the sub-subsection 2.7.2 earlier, Rviz can be used to visualize the position of the mobile robot and also the data collected from the simulated sensors too.

Gazebo has a convenient interface for the users to access. By accessing the Gazebo interface, the user can pick and use the fundamental shapes like cylinder, cube, and sphere to build simple robot or create environment for simulation purpose. Both two-dimensional and three-dimensional design interfaces are provided by Gazebo. On the other hand, Gazebo also supports libraries that are able to provide pre-developed real-life objects such as table, chair and dustbin. This makes it even more user-friendly for the developers to develop a simulation environment to test and debug the robot system. Sensors such as stereo camera, depth camera and LiDAR can also be found in Gazebo. Not only that, physical options like forces, weights, inertia and weights are also provided by Gazebo.

Those simulation objects in Gazebo can be controlled by more than one controller. This eases the process of commanding the objects in Gazebo. “Ifaces”, a Gazebo interface is used so that the processes can access the data generated by the controllers through a shared memory. According to Takaya et al. (2016), Gazebo can provide a proper and reliable rigid body physical simulation because it can access the physics engines such as Open Dynamics Engine (ODE), Simbody, Bullet, Dynamic Animation and Robotics Toolkit (DART) whereas the three-dimensional graphics are rendered by the Object-oriented Graphics Rendering Engine (OGRE).

2.7.4 V-REP

Other than the well-known Gazebo software, another simulation software that is also used for robotic simulation is called V-REP. V-REP is a commercial robotic simulator, and its educational version can be obtained for free. All Windows, Mac and Linux operating systems support V-REP, and seven types of programming languages are possible to be used. This includes C, C++, Java, Lua, Matlab, Octave and Python (Coppelia Robotics, 2021). The features provided by V-REP include mesh editing and ability to interact with the environment when simulation is carried out.

Similar to Gazebo, it can also access to a few physics engines. For examples, Open Dynamics Engine (ODE), Vortex and Bullet. Moreover, the object or model in V-REP can be controlled by ROS node, BlueZero node, embedded script or a plugin.

2.7.5 Comparison between Gazebo and V-REP

Nogueira (2014) had conducted a study to compare both the Gazebo and V-REP. The first aspect he compared is ROS integration. He found out that there exists a large number of community-developed plugins and code that are already available. However, even though V-REP also has some of the capabilities, but its amount is not as much as Gazebo. The ROS official repository has a package specifically for Gazebo which is the Open Source Robotics Foundation. This allows ROS to communicate easily with Gazebo. On the other hand, V-REP can only run alongside ROS because a native ROS node is not provided. Therefore, it cannot run in ROS in a single launch file just like Gazebo. Thus, it can be said that Gazebo performs better in this criterion (Nogueira, 2014).

Next, Nogueira (2014) compared Gazebo and V-REP in terms of the difficulty in creating a world model. He argued that V-REP is more user-friendly when compared to Gazebo because Gazebo requires the user to have deeper knowledge into SDF specifications whereas V-REP does not need a deep understanding of extensible markup language (XML. Nevertheless, he also stressed that it is still possible to develop a complicated simulation using Gazebo after mastering the skill and knowledge. V-REP provides many object models such as walls, doors and furniture while Gazebo does not provide many models. However, many community-developed models are still able to be accessed through online database.

In addition to that, Nogueira (2014) also stated that V-REP is more user-friendly in terms of robot model modifications. He claimed that V-REP offers many different sensors that can be inserted easily inside V-REP. However, editing of SDF files is required when there is a need to modify a model in Gazebo. Thus, it is argued that V-REP is more convenient in terms of this criterion. But there exists another problem where Coppelia Robotics does not maintain the vrep_ros_bridge tool. Similar to the previous criterion, even though Gazebo does not have simple way of editing model, it does have many useful plugins that are developed by the community (Nogueira, 2014).

Last but not least, Nogueira (2014) stated that V-REP and Gazebo does not show clear difference in terms of programmatic control. In a nutshell, the study claimed that V-REP is more user-friendly and provide more features while on the other hand, Gazebo has a better integration into ROS framework and due to its open-source nature, Gazebo has more community-developed plugins and models that are available.

Table 2.3: Comparison between Gazebo and V-REP.

Criterion	Simulator	
	Gazebo	**V-REP**
ROS integration	Can integrate better than V-REP	Able to integrate
Difficulty in creating a world model	Require deeper knowledge but many community-developed models are available	More user friendly
Robot model modifications	Editing of SDF files is required	More user friendly but the tool provided is not maintained well
Programmatic control	No clear difference	

2.8 Summary

In short, the literature reviews on hospital environment and layouts, commercial delivery robots, research on autonomous logistics system within hospital and simulation software are conducted and presented in this chapter. It can be said that delivery robots within hospital become more important nowadays in order to increase the efficiency and performance of the hospital. The application of autonomous logistics system within hospital has become more common. However, suitable and reliable simulation model of hospital still lacks in Malaysia. A reliable simulation environment of hospital is crucial for the development of autonomous delivery robot.

It is proved that ROS and Gazebo can help to develop a simulation environment to test and debug the robot behaviour. Although V-REP might be more user-friendly when compared to Gazebo, Gazebo is still preferable after taking other aspects into consideration. This is due to its large number of community-developed plugins and models. Furthermore, according to Nogueira (2014), Gazebo has a better integration into ROS framework.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

Methods that are used to complete the project and achieve the objectives of this project are discussed in this chapter. In addition to that, this chapter also discusses the work plan of this project. The work plan includes and explains the identified activities and tasks that are required to achieve the outcome of this project. In order to clearly illustrate how the work plans are conducted, Gantt charts are included in this chapter too. This project's duration is one year. Thus, the work plans included here are separated into two parts, where the first part is the work plan for part I project whereas the second part is the work plan for part II project.

This project's main aim is to develop a simulation model of a hospital for robotic simulation in ROS. To achieve this main goal, ROS 2 and Gazebo are chosen to build the hospital environment model after having the comparison between Gazebo and V-REP. Before building the three-dimensional environment using ROS 2 and Gazebo, a site visit or inspection has to be conducted in order to gather more information about the chosen hospital. This is to ensure that the developed environment model can be more realistic and closer to real environment so that the simulation that is to be conducted is more reliable.

3.2 Hospital Inspection

In order to develop a realistic virtual hospital model for robotic simulation purpose, a hospital inspection is carried out. This can help to have a better understanding of the real hospital environment including the size of hospital, the obstacles, human traffic and the dimensions of the spaces or rooms in the hospital. Therefore, a hospital has been identified, and approval to conduct site visit has been obtained too. For this project, the hospital that has been chosen for site visit purpose is Sungai Long Specialist Hospital, Kajang. It is a medium-size hospital located in Bandar Sungai Long, Kajang. Sungai Long Specialist Hospital is a six-floor hospital which provides inpatient and outpatient services.

After identifying the hospital, a request to have the hospital layout or floor plan has been conducted. However, after discussing with the management of the hospital, it was informed that the layout or floor plan is considered as confidential information and therefore, the layout or floor plan could not be obtained. Thus, after further discussion with the supervisor and the hospital management, it is decided that Sungai Long Specialist Hospital will be taken as reference. Few site visits have been carried out to study on the hospital environment. The important information that has been collected through the site visit includes the obstacles position, dimensions and sizes of certain spaces and rooms. As mentioned earlier, the scope of this project focuses on a specified area or floor instead of the whole hospital. The site visits conducted mainly focus on observing the situation and condition in rooms and spaces such as general ward, nurses centre, corridor and medicine dispensary etc.

Figure 3.1: Photo Taken during Site Visit at Sungai Long Specialist Hospital.

The information collected from the hospital through site visit is used as the basic reference. Together with the guidelines for the dimensions provided by Ministry of Health Malaysia (MOH) as stated in Chapter 2, this information is used to draft and draw a floor plan of the hospital. As the scope of this project suggests, only a single floor of hospital's floor plan is drawn. The floor plan of the hospital is drawn using AutoCAD.

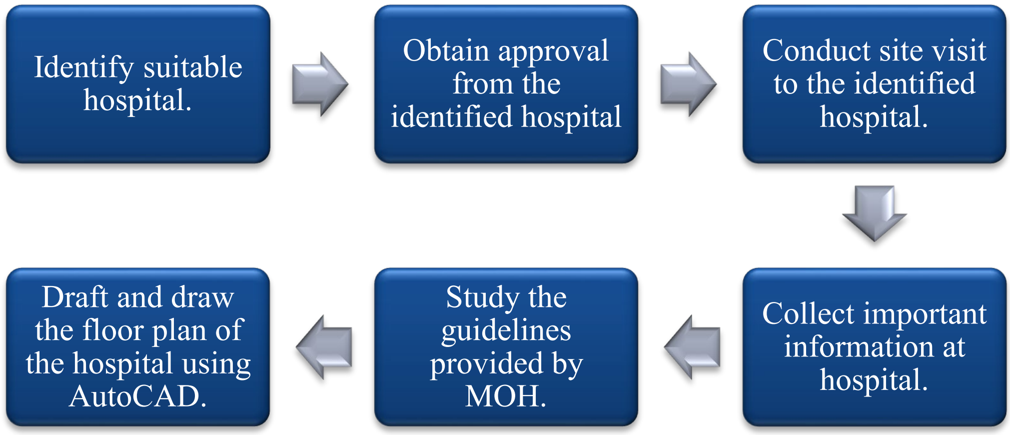


Figure 3.2: Process of Conducting Hospital Inspection.

3.3 Software Implementation

There are a few software which are required to be used to meet the objectives of this project which is to create a simulation model of hospital environment for robotic simulation. The Robot Operating System (ROS) is chosen as the framework so that the autonomous mobile robot, sensors application and the environment for testing purpose can be integrated well. Besides, Gazebo is used to create the simulation environment due to its large number of available plugins and database. These software and framework are to be used on Linux system.

This project is started with the installation of ROS 2 and Gazebo on the author's laptop. Following that, hospital inspection mentioned in the previous subsection is carried out in order to obtain the necessary information of the hospital. Then, Gazebo is used to create the hospital building. After the layout of the hospital is created, other objects that will act as obstacles are also added into the Gazebo world. Once the simulation environment is ready, the delivery robot is implemented into ROS 2 so that it can integrate into Gazebo for simulation purpose. To fulfil the main aim of this project which is to develop a hospital environment that can be used for robotic simulation purpose, simple simulation such as simultaneous localisation and mapping (SLAM) is carried out. Troubleshooting is conducted when problems are found after the simulation is carried out. After that, finalisation of the developed environment is conducted and followed by documentation. A flowchart of the software implementation is shown in Figure 3.3.

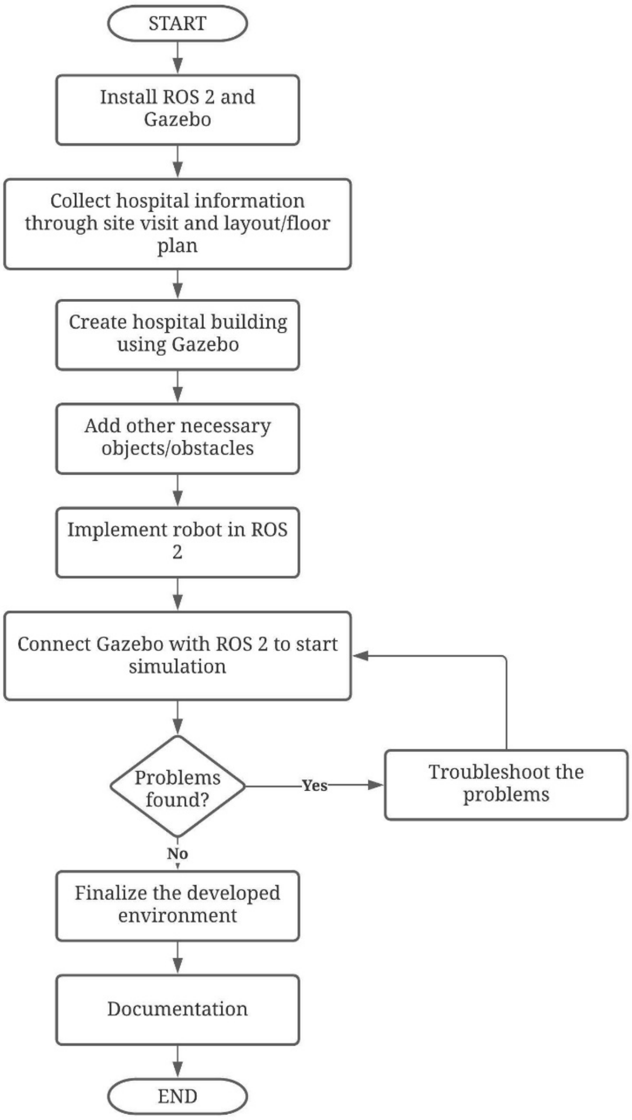


Figure 3.3: Flowchart of the Software Implementation.

3.3.1 Gazebo

The literature review done has compared two simulators that can be used to build the simulation environment. It is decided that Gazebo will be used to develop the simulation model of hospital for this project because Gazebo is a free simulator that consists of many available community-developed plugins and tools. The developed model can be inserted into the Gazebo world as shown in Figure 3.4 to conduct simulation to test the robot's behaviour. This project's scope is limited to develop a simulation model of a specified area or floor of a hospital. To achieve this outcome, there are some fundamental steps or processes which are required to be followed and this sub-subsection will briefly discuss this.

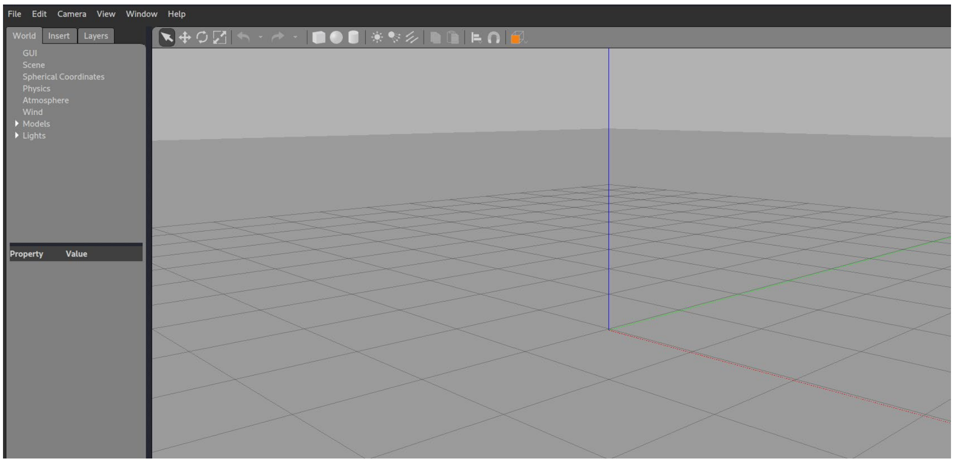


Figure 3.4: Gazebo User Interface.

First of all, there are two ways to add objects into the render window. The first one is by adding simple shapes such as sphere, cylinder and cube. The scale of the simple shapes can be changed. The second mechanism is by accessing the model database. Inside the model database, objects such as tables, building and robots can be selected. The models are to be downloaded from online resources. All these models can be translated and rotated to alter their position. They can be translated along three axes which include the x, y and z axes. On the other hand, the models can be oriented around the x, y and z axes too. The world can be modified too. For example, its scene properties such as ambient, background and shadows can be modified. One of the most important features of Gazebo is that it can provide physics properties which are vital for simulation purpose. Open Source Robotics Foundation (2014) provided the steps to create a building directly through its user interface as shown in Figure 3.4. Simple features like creating walls, windows, doors and stairs can be done. Other than doing it on our own, Gazebo also allows the user to import his or her own floor plan into it and trace over.

In this project, the steps taken in Gazebo are summarised in Figure 3.5. Firstly, Building Editor of Gazebo is used to build the layout. The floor plan drawn is inserted into the building editor to trace over. Once the building model is created, it is saved as a model and the model is inserted into the world. After creating the building model, obstacles should also be added into the world. For simple objects, spheres, cylinders and cubes can be added directly. However, to ensure the virtual environment can be more realistic, more complex objects are added.

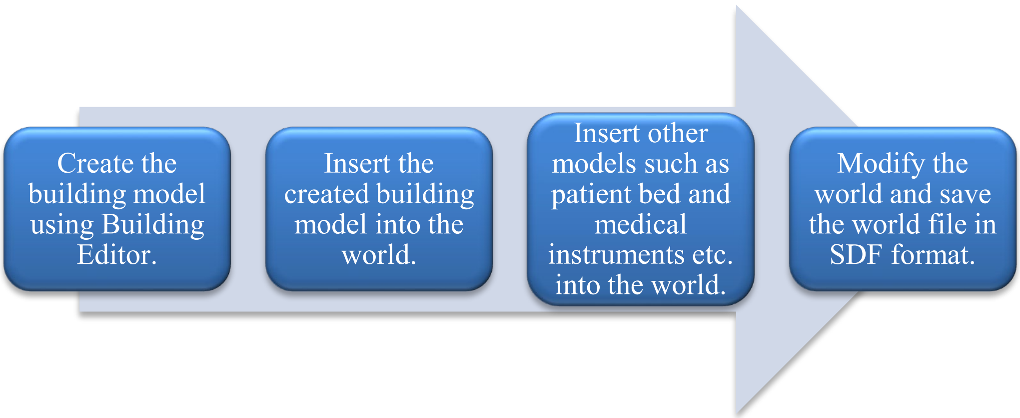


Figure 3.5: Steps Taken to Create the Simulation Model of Hospital.

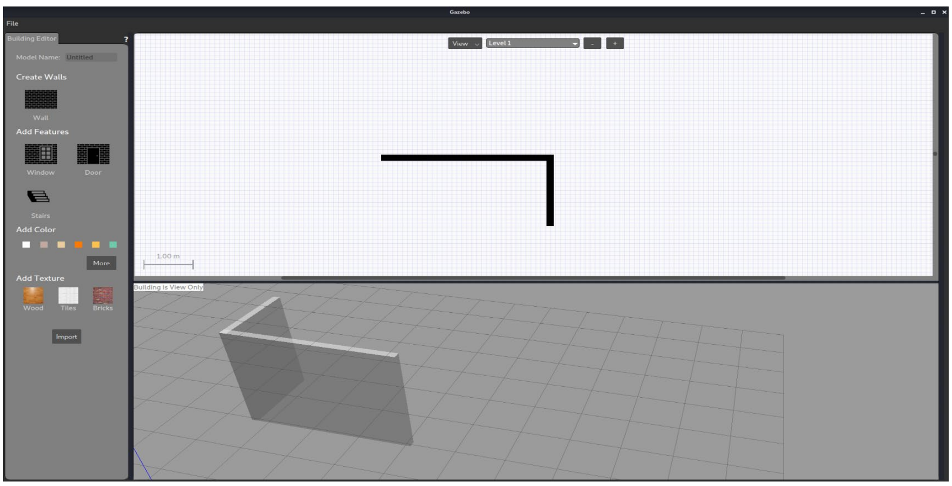


Figure 3.6: Building Editor User Interface.

To add more complex items into the world, other than the built-in model database of Gazebo, other online websites or databases such as Open Robotics and Automatic Addison are used to download the required models as shown in Figure 3.7. For examples, patient bed, instrument cart, nurse table and bedside table. These models downloaded from the online sources usually contain a CONFIG file, SDF file and their meshes. Meshes is a directory where all the DAE files, STL files and other relevant files that contribute to the models are located at. After adding these models into the world, they are all arranged properly at the desired position. This is done by referring to the real hospital environment. Then, the world is further modified and decorated to ensure they look closer to the real hospital environment. Once all these steps are taken, the world file is saved into SDF format.

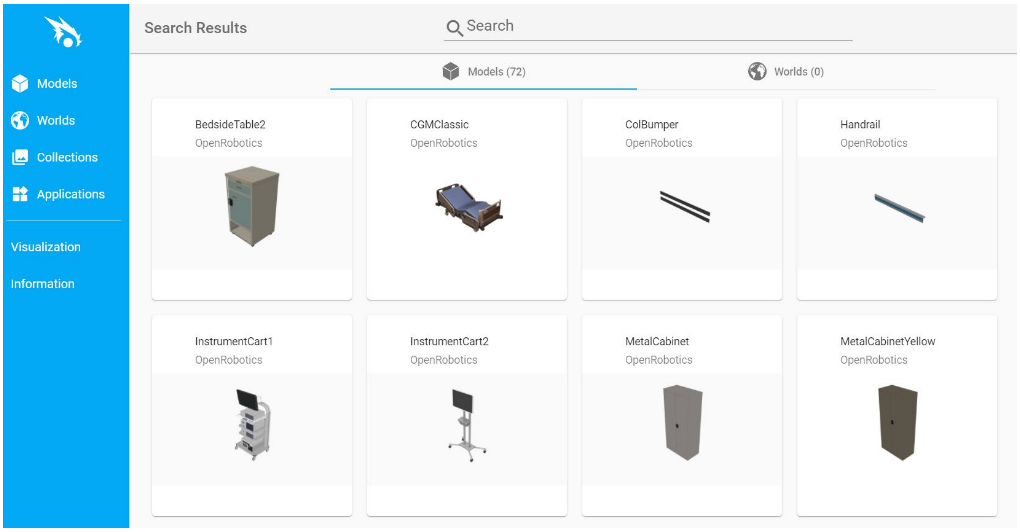


Figure 3.7: Open Robotics Database (Open Robotics, 2022).

3.3.2 ROS 2

ROS is a robotic software framework which consists of many different packages and libraries that are useful for robotic simulation. To be more specific, ROS 2 is used in this project to help developing the hospital simulation model. Even though Gazebo can run independently of ROS, their integration can be done by using correct packages such as gazebo_ros_pkgs. This metapackage consists of other packages such as gazebo_plugins (Open Source Robotics Foundation, 2014).

ROS is mainly used in this project to integrate the robot into the developed simulation environment for testing purpose. Also, ROS is chosen because of its ability to communicate between different software through the topics. ROS is used to connect with Gazebo to control the robot in the created simulation environment. In this project, the ROS 2 version that is applied is ROS Foxy. In order to connect Gazebo and ROS 2 together, few important dependent ROS 2 packages ensured that they are installed too. The packages that are used for simulation purpose in this project include RViz2 package, slam_toolbox package, teleop_twist_keyboard and nav2_map_server package. The RViz2 package and slam_toolbox package is used for SLAM simulation purpose whereas teleop_twist_keyboard is used to control the delivery robot in the hospital simulation model. Last but not least, the nav2_map_server package is used to save the map generated through SLAM process. Steps taken to conduct SLAM simulation are discussed in the sub-subsection below.

3.3.3 SLAM Simulation

Once the development of the simulation model is finished and saved, the next step to be taken is to conduct SLAM simulation. These steps are vital to prove that the developed environment can be applied for robotic simulation purpose. This sub-subsection will first describe the steps to conduct SLAM simulation.

First and foremost, in order to use the ROS 2 and Gazebo packages and nodes, environment setup is carried out by sourcing the following file using the terminal window. The directory is also changed to usr_ws as the source files and other files needed are saved in that directory in the author’s laptop.

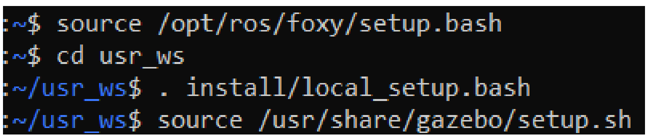


Figure 3.8: Examples of Environment Setup.

After that, ROS 2 is used to launch the simulation model of hospital in Gazebo. On the other hand, a simulated differential drive robot is required to run the SLAM simulation. The simulated differential drive robot used in this project is shown in Figure 3.9. Thus, another terminal window is used to launch a simulated differential drive robot. Once the robot is launched, it can be seen in the world in Gazebo. The command lines used are shown in Figure 3.10 and Figure 3.11. ROS 2 is used to launch the files written in Python language, whereas the “usr_description” refers to the directory where the launch files are located at the author’s computer.

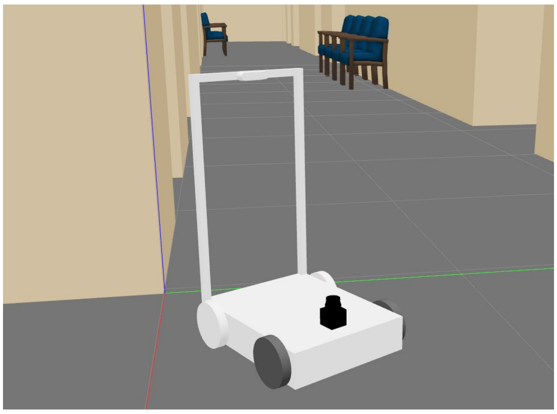


Figure 3.9: Simulated Differential Drive Robot Used.

```
:~$ source /opt/ros/foxy/setup.bash
:~$ cd usr_ws
:~/usr_ws$ . install/local_setup.bash
:~/usr_ws$ source /usr/share/gazebo/setup.sh
:~/usr_ws$ ros2 launch usr_description world.launch.py
```

Figure 3.10: Command Lines to Launch World File.

```
:~$ source /opt/ros/foxy/setup.bash
:~$ cd usr_ws
:~/usr_ws$ . install/local_setup.bash
:~/usr_ws$ source /usr/share/gazebo/setup.sh
:~/usr_ws$ ros2 launch usr_description spawn.launch.py
```

Figure 3.11: Command Lines to Launch Simulated Differential Drive Robot.

In order to conduct the SLAM simulation, slam_toolbox and RViz2 packages are also required. Therefore, another terminal window is opened, and another launch file written in Python language is used to launch RViz2, which is a three-dimensional visualization tool for ROS 2 framework. This launch file utilises both RViz2 and slam_toolbox packages. RViz2 is used to generate the map as the simulated differential drive robot moves and scans by using its installed sensor. Figure 3.12 and Figure 3.13 illustrates how the process is carried out.

```
:~$ source /opt/ros/foxy/setup.bash
:~$ cd usr_ws
:~/usr_ws$ . install/local_setup.bash
:~/usr_ws$ source /usr/share/gazebo/setup.sh
:~/usr_ws$ ros2 launch usr_mapping slamtoolbox.launch.py
```

Figure 3.12: Command Lines to Launch RViz2 and slam_toolbox packages.

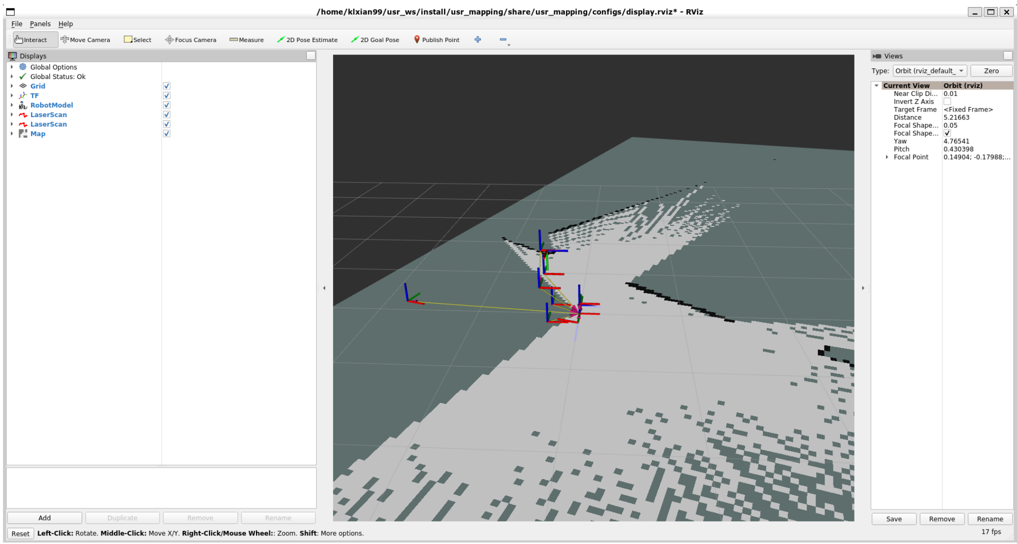


Figure 3.13: RViz2 User Interface.

Furthermore, to move the simulated differential drive robot around the world by using keyboard, teleop_twist_keyboard package is called. Then, the robot is controlled to move around the developed simulation model world to generate a complete map. Once the robot has finished scanning the world and the map is successfully generated, nav2_map_server package is used to save the generated map in Portable Gray Map (PGM) format. PGM file is used to store grayscale image. An example of PGM image is shown in Figure 3.16. Figure 3.14 and Figure 3.15 show the command lines used to carry out these steps. Once the map is generated, the SLAM simulation is considered

```
klxian99@LAPTOP-QCU13IA9:~$ source /opt/ros/foxy/setup.bash
klxian99@LAPTOP-QCU13IA9:~$ cd usr_ws
klxian99@LAPTOP-QCU13IA9:~/usr_ws$ . install/local_setup.bash
klxian99@LAPTOP-QCU13IA9:~/usr_ws$ source /usr/share/gazebo/setup.sh
klxian99@LAPTOP-QCU13IA9:~/usr_ws$ ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args -r cmd_vel:=/usr/cmd_vel

This node takes keypresses from the keyboard and publishes them
as Twist messages. It works best with a US keyboard layout.
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

Figure 3.14: Command Lines Used to Control Robot in Gazebo.

```
klxian99@LAPTOP-QCU13IA9:~$ source /opt/ros/foxy/setup.bash
klxian99@LAPTOP-QCU13IA9:~$ cd usr_ws
klxian99@LAPTOP-QCU13IA9:~/usr_ws$ . install/local_setup.bash
klxian99@LAPTOP-QCU13IA9:~/usr_ws$ source /usr/share/gazebo/setup.sh
klxian99@LAPTOP-QCU13IA9:~/usr_ws$ ros2 run nav2_map_server map_saver_cli -f ~/map
[INFO] [1650384936.479953200] [map_saver]:
        map_saver lifecycle node launched.
        Waiting on external lifecycle transitions to activate
        See https://design.ros2.org/articles/node_lifecycle.html for more information.
[INFO] [1650384936.481633935] [map_saver]: Creating
[INFO] [1650384936.485303897] [map_saver]: Saving map from 'map' topic to '/home/klxian99/map' file
[WARN] [1650384936.485542460] [map_saver]: Free threshold unspecified. Setting it to default value: 0.250000
[WARN] [1650384936.485612576] [map_saver]: Occupied threshold unspecified. Setting it to default value: 0.650000
[WARN] [map_io]: Image format unspecified. Setting it to: pgm
[INFO] [map_io]: Received a 335 X 408 map @ 0.05 m/pix
[INFO] [map_io]: Writing map occupancy data to /home/klxian99/map.pgm
[INFO] [map_io]: Writing map metadata to /home/klxian99/map.yaml
[INFO] [map_io]: Map saved
[INFO] [1650384936.882459031] [map_saver]: Map saved successfully
[INFO] [1650384936.883045802] [map_saver]: Destroying
```

Figure 3.15: Command Lines Used to Save the Generated Map.

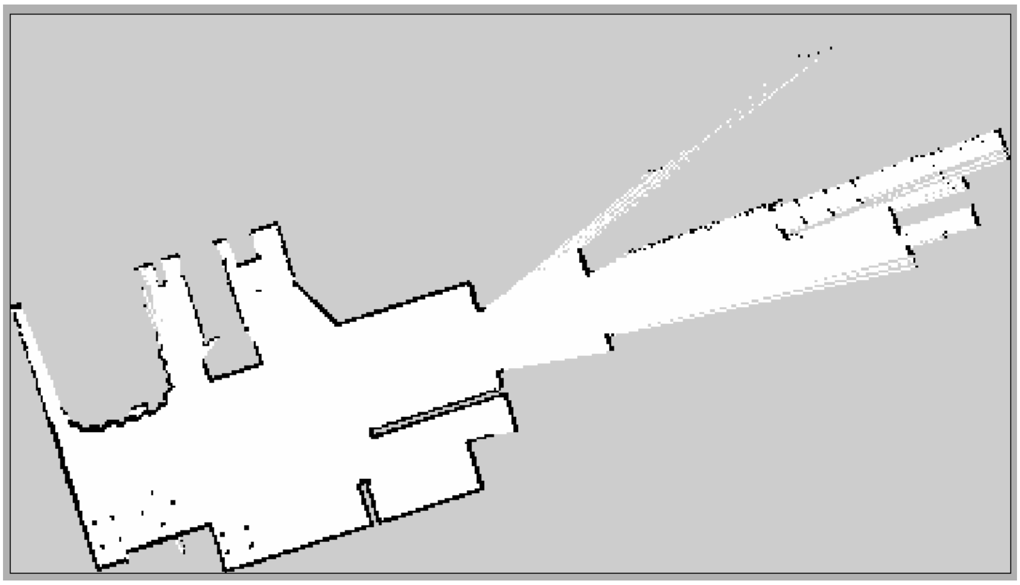


Figure 3.16: Example of PGM Image.

3.4 Project Activities and Scheduling

Project planning is perceived as one of the important criteria to ensure the project can be completed on time. A Gantt chart is always a good way of scheduling the project activities that need to be carried out in order to meet the project objectives. It also helps to ensure that time can be allocated well. This project is separated into two parts. The first part of the project is conducted in May trimester 2021 while the second part of the project is conducted in January trimester 2022.

Table 3.1 shows the planning of project activities for May trimester 2021. The project planning stage roughly took two weeks to complete. It includes the project title and objectives identification, simple project research and discussion with supervisor to determine the project scope. On the other hand, the research and analysis started from week 3 and it took around 9 weeks to complete. Report writing was started in week 5 after the part of literature review had been conducted. Besides, discussion with supervisor was conducted almost every week until week 13. Last but not least, the oral presentation was conducted in the last week of the trimester.

Table 3.1: Gantt chart for May trimester 2021.

No.	Project Activities	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**Project Planning**														
1.1	Identify FYP title and objectives														
1.2	Simple research on FYP title														
1.3	Discussion with supervisor to determine project scope														
2	**Research and Analysis**														
2.1	Literature review on hospital related information														
2.2	Literature review on delivery robots														
2.3	Literature review on software needed														
3	**Documentation**														
3.1	Report writing														
3.2	Report finalizing and checking														
4	**Discussion and Presentation**														
4.1	Discussion with supervisor														
4.2	Oral presentation														

For the second part of the project, it was started by conducting hospital inspection so that important information required to build the simulation model can be collected beforehand. The development of environment was started in week 2 and continued until week 11. Besides, testing and troubleshooting of the environment was started around week 8 to ensure there would be sufficient time. For documentation, the report writing started since week 3 by modifying the previous progress report, and it continued until everything is finalised. Report finalising was carried out in early week 13 for supervisors' checking and amendment. Similar to the first part of project, a continuous discussion with supervisor was carried out too to ensure the project is always on track. Lastly, an oral presentation shall be conducted in the last week of the trimester too.

Table 3.2: Gantt chart for January trimester 2022.

No.	Project Activities	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**Hospital Inspection**														
1.1	Identify suitable hospital and obtain approval														
1.2	Obtain hospital layout or floor plan														
1.3	Conduct site visit														
2	**Development of Environment**														
2.1	Create simulation model														
2.2	Preliminary testing and troubleshooting														
2.3	Finalize developed environment														
3	**Documentation**														
3.1	Report writing														
3.2	Report finalizing and checking														
4	**Discussion and Presentation**														
4.1	Discussion with supervisor														
4.2	Oral presentation														

3.5 Summary

In a nutshell, Gazebo and ROS are used together to develop a simulation model of hospital for robotic simulation purpose. Before the development of the environment, hospital inspection should be carried out first. All the necessary and important information should be gathered properly in order to build a realistic, reliable and promising virtual simulation environment. Then, the developed environment is tested, and troubleshooting will be done to ensure the built environment can perform its function properly. SLAM simulation and is also conducted to ensure the developed simulation model of hospital is able to function well to meet the objectives of this project.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This project aims to develop a simulation model of a hospital for robotic simulation in Robot Operating System (ROS) using Gazebo. The particular objectives of this project include collecting information on the layout of an hospital and the operation of a hospital, simulating the human and other objects in the simulated environment mimicking the data collected, and conducting simulation of the environment for testing the behaviour of the autonomous delivery robot. Therefore, this chapter includes the hospital floor plan, built simulation model, results of simulation, and their discussion too.

4.2 Hospital Floor Plan

As mentioned in Chapter 3, a hospital inspection has been conducted to collect the required information for the purpose of building the simulation model of a hospital. Following the methodology mentioned in Chapter 3, a hospital near to Universiti Tunku Abdul Rahman, Sungai Long Campus is selected which is Sungai Long Specialist Hospital, a six-floor medium size hospital that is located in Bandar Sungai Long, Kajang.

After several times of site visit and meeting with the management of the hospital, it was informed by the management that the floor plan of the hospital cannot be provided due to confidential issue. Therefore, a floor plan has to be designed first and the selected hospital is taken as a reference only while modifications and assumptions have been made. Since autonomous logistics system is usually applied in hospital for delivering food and medicine, it is decided that the simulation model of hospital will mostly focus on spaces such as ward and medicine dispensary. Basically, the floor plan has been designed to consist of general ward, office, kitchen, medicine dispensary and nurses centre. The size and measurements of the rooms, doors and corridors are decided based on the guidelines provided by the Ministry of Health Malaysia (MOH) and the measurements taken at the hospital, with assumptions made. For examples, all the doors where patients will be transported through have at least 1.2 meters, whereas the width of the corridor is 2.2 meters. Figure 4.1 shows the two-dimensional hospital floor plan designed by the author. It has been drafted and drawn using AutoCAD.

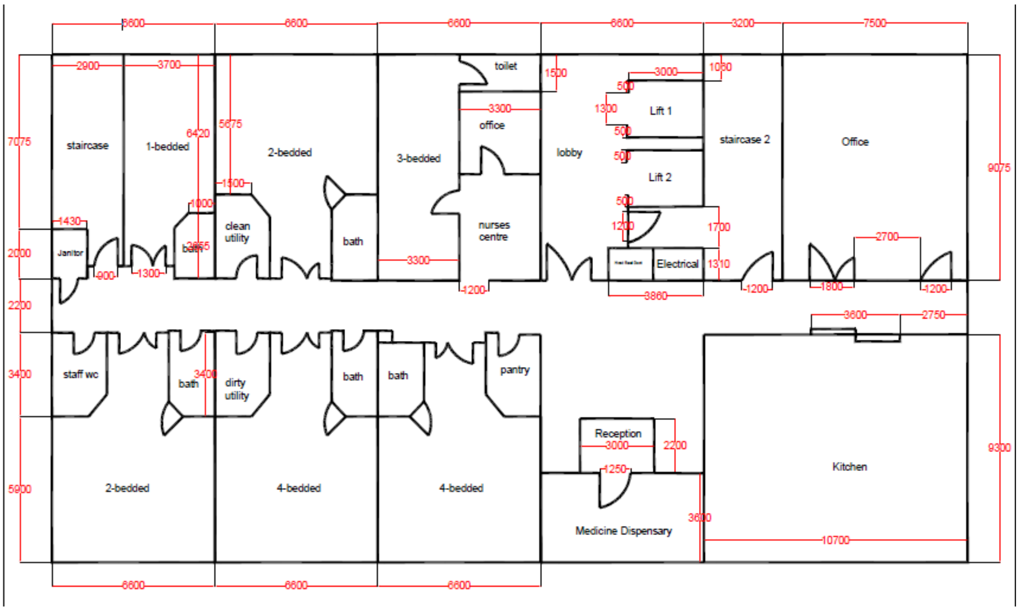


Figure 4.1: Drawn Floor Plan of Hospital.

4.3 Built Simulation Model

As stated earlier, Gazebo is used to develop the simulation model of the hospital. The building model of the hospital has been built by using Building Editor of Gazebo based on the drawn floor plan as shown in Figure 4.1. The building model has been saved in SDF format and it has also been inserted into the world. The result is shown in Figure 4.2.

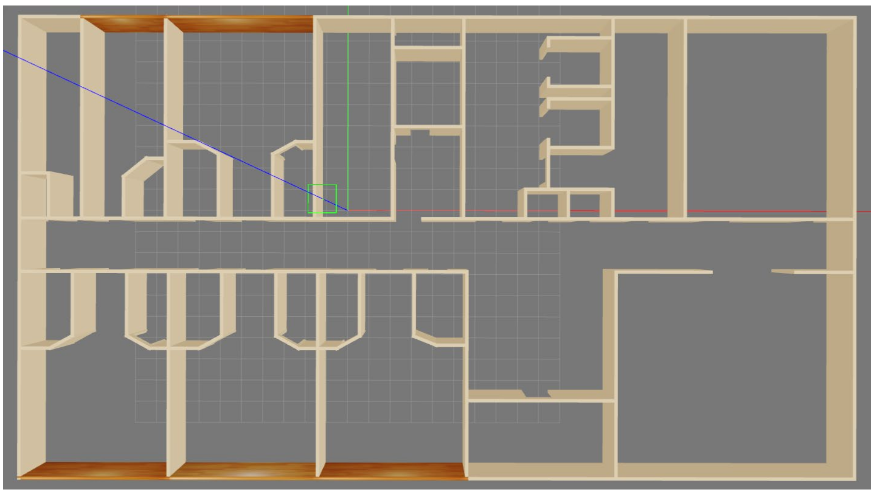


Figure 4.2: Developed Building Model of Hospital.

Following that, the other obstacles have been inserted into the world too. The models required have been downloaded from online database such as Open Robotics and Automatic Addision websites as mentioned in Chapter 3. Figure 4.3 shows the top view of the developed simulation model of the hospital. From the left, there is one single-bedded ward, two 2-bedded wards, two 4-bedded wards and one 3-bedded ward. Also, beside the three-bedded ward, there is an office and a nurses centre. From the right, there is an office and a kitchen. Last but not least, the medicine dispensary and reception are located in front of the lifts's area.

Figure 4.3: Top View of the Developmed Simulation Model.

The most important part of this simulation model is the ward as delivery robots used in hospital are mainly for delivering items to the patients. The general items included in the wards are patient beds, curtains, sofas, tables, bedside table and medical equipment etc. Patients and visitors have also been included randomly at some rooms as they are also considered as obstacles. These items and models are included to ensure the environment built is closer to the real-life environment. The items' arrangement and position have been decided based on the observation at the hospital during site visit. Figure 4.4 shown below is a photo taken at a ward in Sungai Long Specialist Hospital whereas Figure 4.5 and Figure 4.6 show the internal views of the built wards.

Figure 4.4: Photo Taken at a Ward in Sungai Long Specialist Hospital, Kajang.

Figure 4.5: Interior View of a 2-Bedded Ward.

Figure 4.6: Interior View of a Single-bedded Ward.

On the other hand, Figure 4.7 and Figure 4.8 show the internal view of kitchen and medicine dispensary. The items that have been added into the kitchen include kitchen cabinets, dish washers, refrigerators, microwave, table, racks and boxes. However, items located in the kitchen are limited as the models provided from the open-source database are limited especially items for a huge central kitchen because most of the kitchen items available from online resources are home kitchen items. Nevertheless, the simulation results will not be affected as the autonomous robots normally will not travel around kitchen but reach at a fixed position to collect items only. Other than that, the models that have been placed at the medicine dispensary include chairs, counter, nurses, patients, visitors and some boxes have been placed to illustrate the storage area.

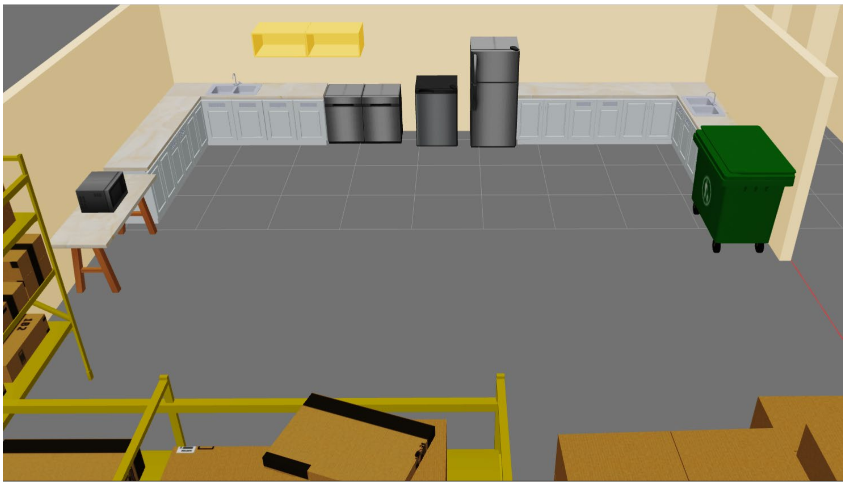


Figure 4.7: Interior View of the Kitchen.

Figure 4.8: Interior View of the Medicine Dispensary.

Moreover, an office has been designed and included in the simulation model of the hospital too. This is because delivery robots can also be used to deliver items such as documents. By including this into the simulation model, such simulation can be conducted in the future too. The simulated office includes conference table, office chairs, office tables, shelfs, tables, whiteboard, television and television table. Two office staff have been added into the office environment too.

Figure 4.9: Interior View of the Office.

Furthermore, other than these main areas or rooms, there are also other spaces which have been included in the simulation model of the hospital. For instances, nurses centre, pantry, janitor room, toilets, lifts, utility rooms and others. Nevertheless, the delivery robots normally will not enter these areas except for nurses centre. Thus, most rooms or areas mentioned in this paragraph have a simpler design and models of items included are less complex than other rooms.

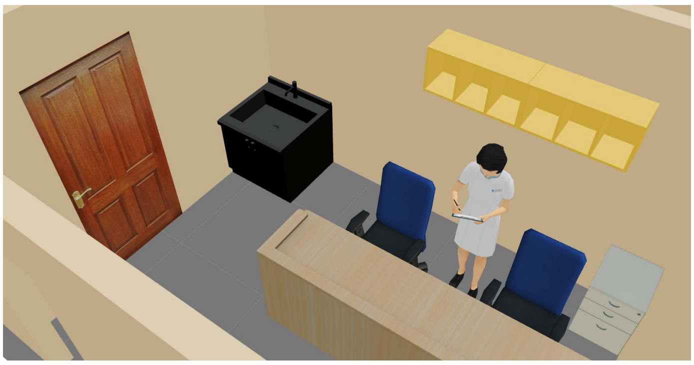


Figure 4.10: Interior View of Nurses Centre.

Figure 4.11: Interior View of Pantry.

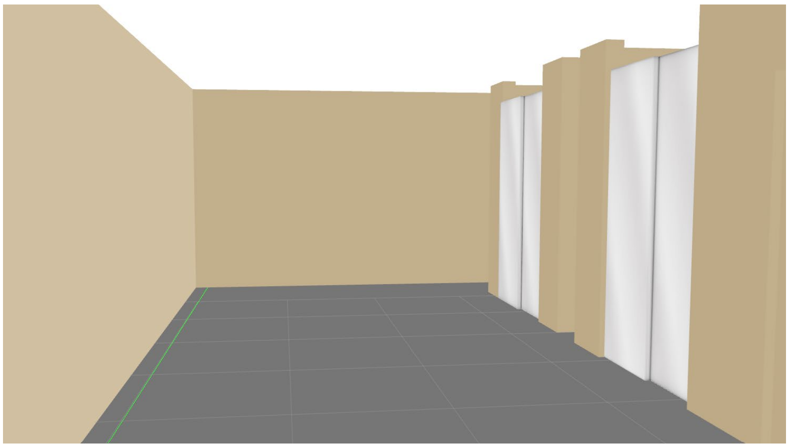


Figure 4.12: Lift Area.

There are some assumptions and neglections made when building the simulation model of the hospital. Firstly, the handrails are not included in this hospital environment, however it is specified in the MOH guidelines that handrails are to be installed at the corridors where patients will pass through. Nevertheless, this will not affect the simulation of the autonomous robots because they are installed at the walls, and they will not block the path where the robots will travel.

Besides, some spaces or rooms where the robots will not be tasked to pass through are left empty to reduce the world file's size and speed up the development process. For examples, the electrical room, lift, janitor and staircase etc. Moreover, another limitation of this project is that moving human is not included in this simulation model too. In addition to that, most of the doors are not installed in the developed simulation model of hospital. This is because installing the door will make the simulation process to be more difficult and complex even though the simulated robot can be programmed or commanded to carry out the door opening and closing task. Thus, the doors are left opened in this simulated world. If the researchers are interested to carry out such simulation, the world can be modified by adding doors.

Lastly, after every model required have been added into the world and the positions of the models are arranged properly, the developed hospital world has been saved in SDF format. At this stage, the first two objectives of this project which is to collect information on the layout of an hospital and the operation of a hospital and to simulate the human and other objects in the simulated environment mimicking the data collected are considered achieved.

4.4 Results of SLAM Simulation

A simulated differential drive robot has been used to conduct simulation in this project. In fact, any functionable simulated robots can be applied to conduct simulation in this developed simulation model of hospital. Robotic simulation has been carried out to ensure that the developed simulation model can be applied properly to meet the project's objectives. The simulation task that has been selected to be conducted in this project is SLAM simulation. SLAM is a process that allows the autonomous vehicles or robots to generate a map and at the same time, the robot or vehicle is also localised in the map. A SLAM simulation has been conducted in the hospital world that has been developed. This step has been taken to confirm that the simulation model built is capable to be applied for simulation purpose.

In the first place, the world has been launched successfully and this confirmed that the world built has been saved properly and in a correct format. Next, the simulated differential drive robot has been launched in the same world too. It can be seen from Figure 4.13 that both the world and the robot have been launched successfully in Gazebo.

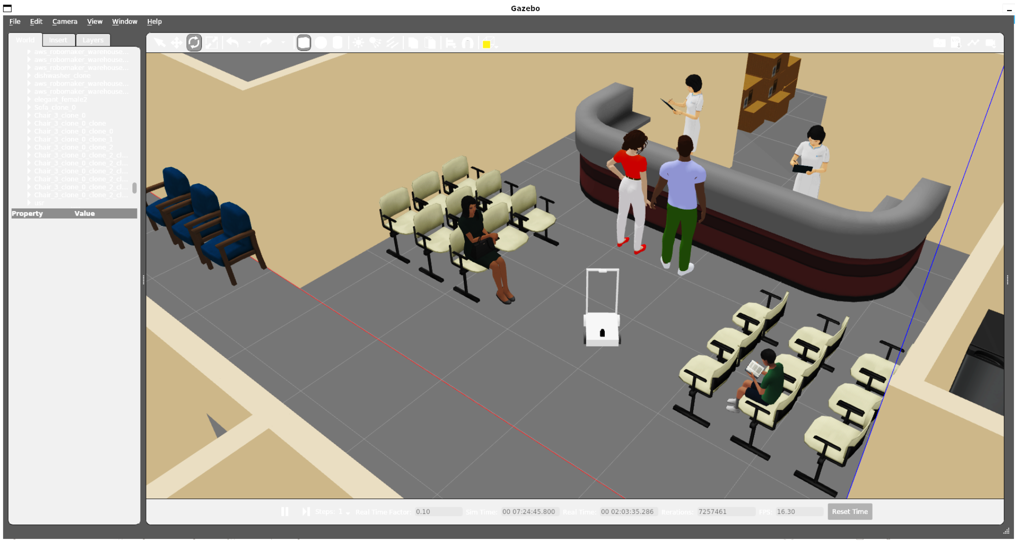


Figure 4.13: Robot Launched Successfully in the Simulated World.

After launching the simulated differential drive robot into the world, a RViz window has also been called and the slam_toolbox package has been utilised too. This process has been taken so that SLAM simulation could be carried out and the map could be generated. It can be noticed from Figure 4.14 that the robot has generated part of the map by using the sensor installed on it. When the robot detects any obstacles or walls, black colour lines or dots are formed. In contrast, if it does not detect any obstacles or walls, the area scanned will be shown in white colour. Besides than that, it is also noticeable from Figure 4.14 that the robot has been localised in the map at the same time too.

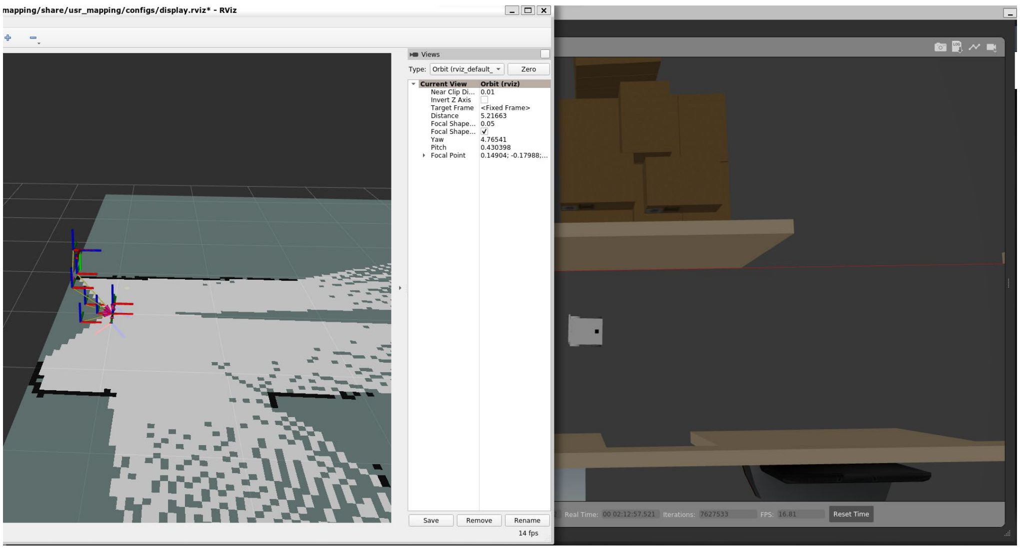


Figure 4.14: Part of the Map Generated during SLAM Process.

To further continue the simulation, teleop_twist_keyboard package has been applied to control the robot so that it could move around the world to generate a full map of the simulation model. Figure 4.15 and Figure 4.16 are used to illustrate that as the robot continued to move, more parts of the simulation environment have been scanned and the map has captured more results too. The keyboard launched has been used to keep moving the robot until the full map has been captured.

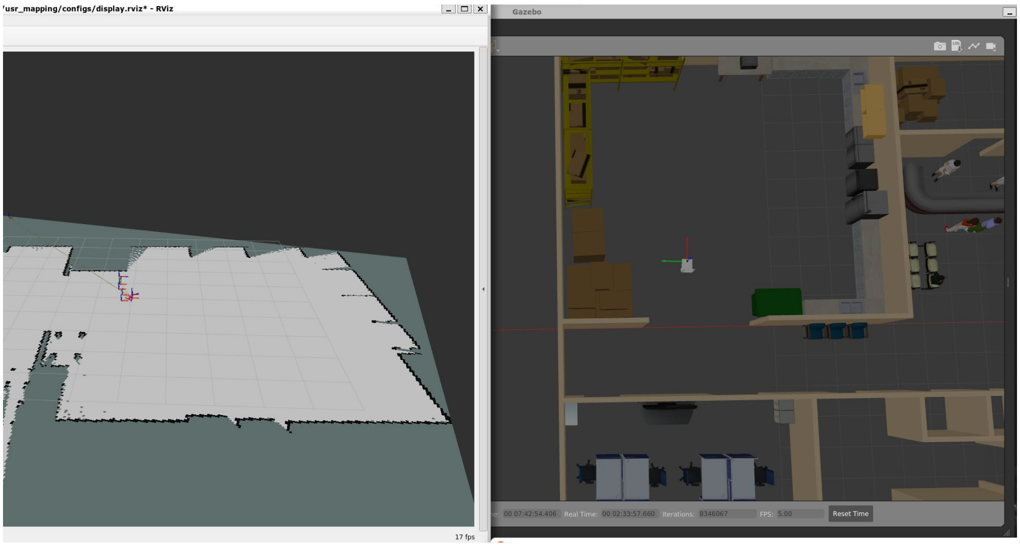


Figure 4.15: More Parts of Map have been Captured as Robot Moved.

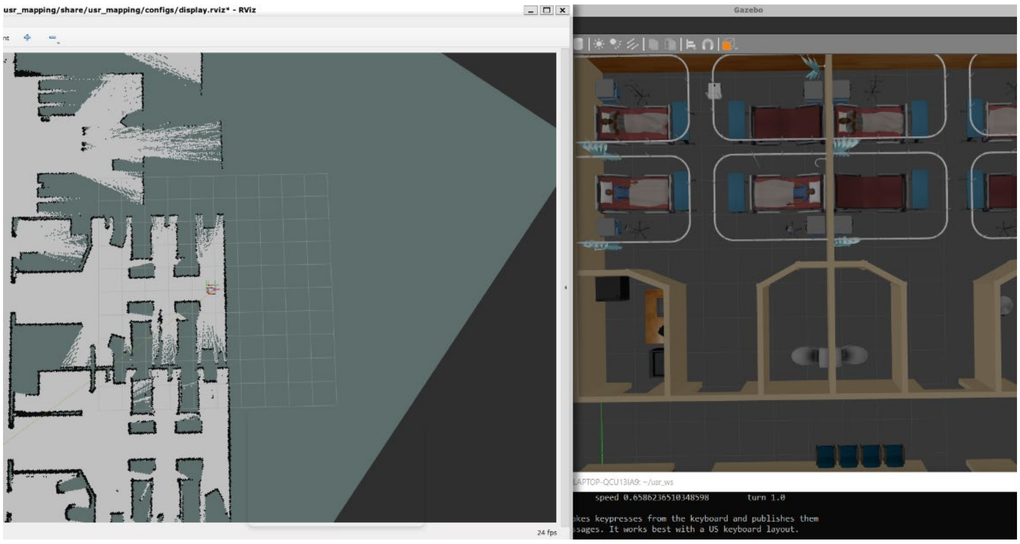


Figure 4.16: Half of the Map has been Scanned and Captured.

Figure 4.17 illustrates the whole complete map that has been generated successfully through SLAM simulation. The generated map has been saved in

PGM format once the process has completed. It can be seen from Figure 4.17 that most of the edges of the walls and obstacles are very clear because the mapping process was conducted through simulation. SLAM process which is conducted at a real-life environment might not be able to deliver such results at most of the time.

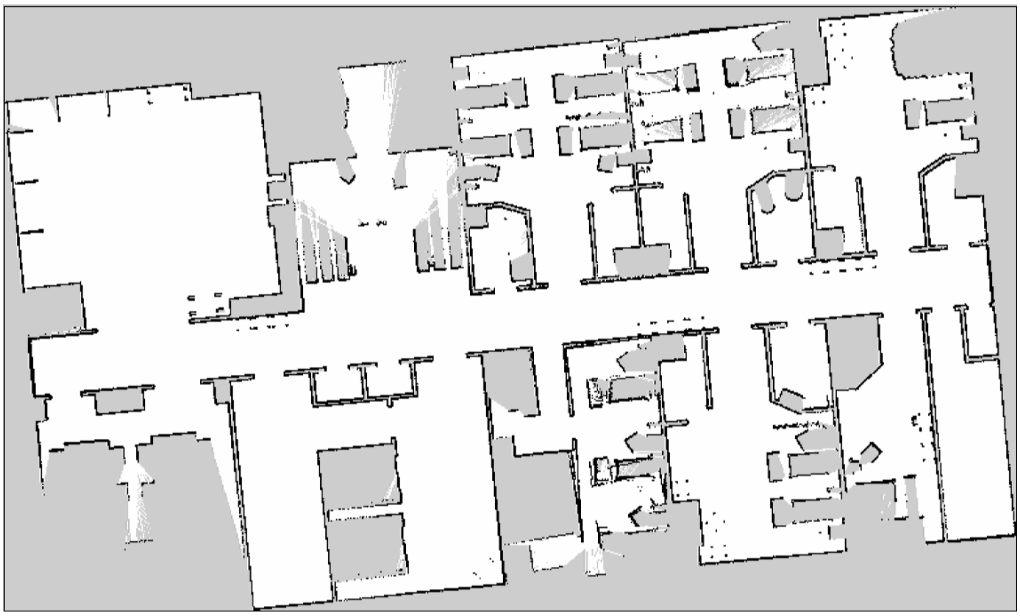


Figure 4.17: Complete Map Generated.

Figure 4.18 is the comparison between the map and the simulated environment of the 2-bedded ward. The 2-bedded ward's walls and the obstacles are well scanned and captured as shown in Figure 4.18. The dots captured at the top left corner are the chairs' legs and coffee table's base because the sensor installed on the robot is placed at a low height and they could only scan the bottom parts of the chairs and legs. Patient bed, patient table, bedside table and the instrument cart placed on top right corner could not be scanned as they are blocked by the closed curtain.

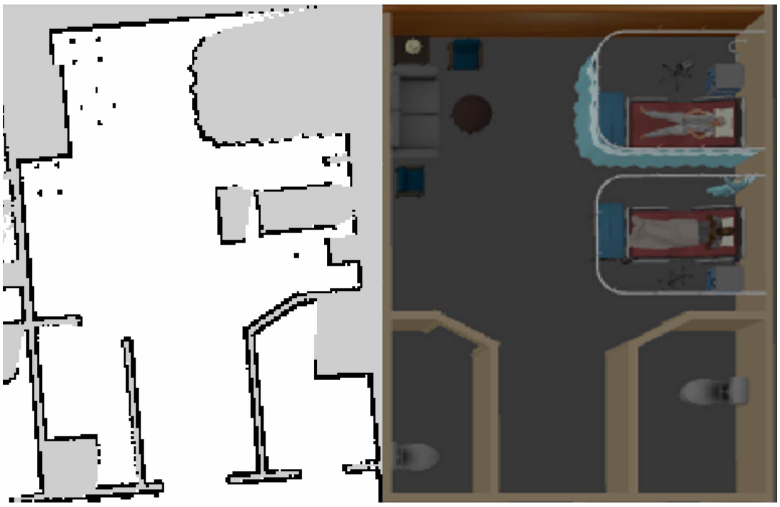


Figure 4.18: Map Captured at the 2-Bedded Ward.

In addition, map captured at areas such as kitchen and lift show very clear and neat lines. For instance, a comparison of the map and simulated environment of the kitchen is shown in Figure 4.19. The map captured at the kitchen and lift is very neat and tidy because there are very little obstacles and most of the obstacles are placed near to the walls. The dots at the bottom right corner are the chairs' legs located outside the kitchen.

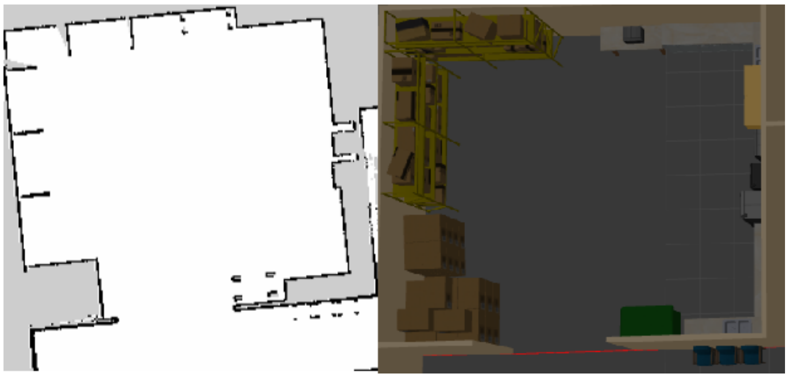


Figure 4.19: Map Captured at the Kitchen.

On the other hand, there are some places where the robot could not enter due to the narrowness of the path. For example, the office is very narrow and some of the paths are blocked by obstacles such as office chairs. Thus, robot could not travel inside and the information there were not captured, and they are shown in grey colour because the sensor could not scan through.

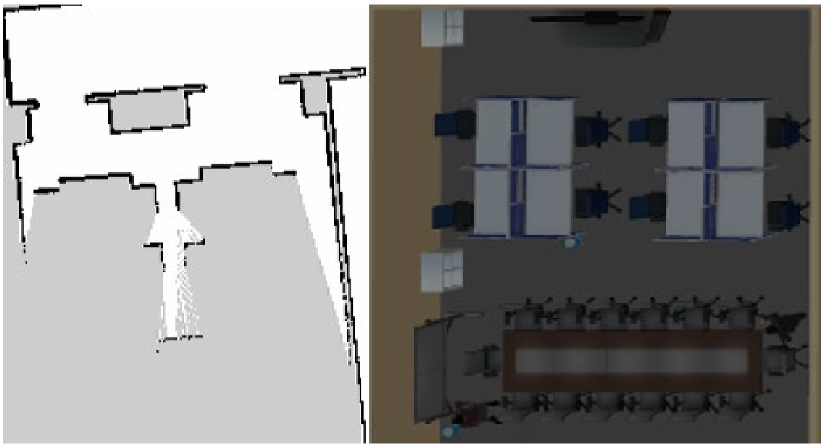


Figure 4.20: Information Loss due to Narrowness of Path.

Furthermore, it is important to ensure that collision does not happen during the SLAM process. This is because the map generated will be distorted if collision happened. An attempt to prove that collision should be avoided during SLAM simulation has been carried out and the result is shown in Figure 4.21. It is noticeable that the map generated cannot show the edges clearly and the map is distorted. Figure 4.22 is also attached to demonstrate the condition when collision occurred. This has also proven that the simulated environment is reliable because a simulation model should be able to simulate the collision situation too. Nevertheless, after the SLAM simulation has been performed, some improvements that can be taken in the future are found too. Firstly, there are some models downloaded from the online resources are lack of certain meshes or correct physical properties that caused it to be unable to simulate properly when collision is occurred. For example, when the robot collided with the standing visitor, the standing visitor remained static instead of moving away or falling down.

Figure 4.21: Map is Distorted due to Collision.

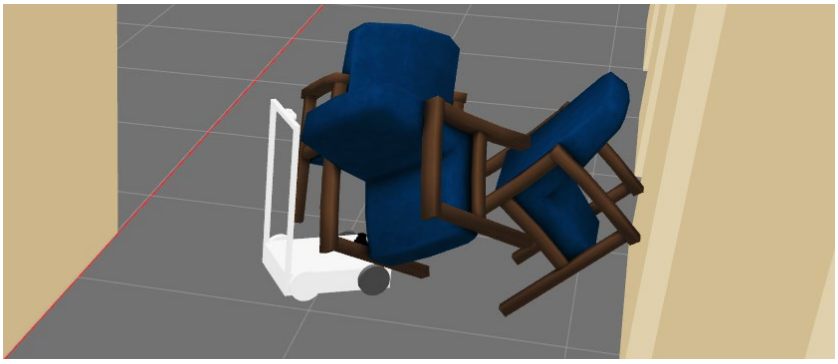


Figure 4.22: Collision between the Robot and Chairs.

The SLAM simulation has been performed successfully and the obstacles have been detected properly. This has led to a correctly generated map.

4.5 Summary

In general, the hospital floor plan has been designed and drawn according to the information collected from site visits and the MOH guidelines. Also, the simulation model of the hospital has been developed using Gazebo. Following that, robotic simulation task such as SLAM simulation has been carried out properly. The results of the simulations are presented and discussed in detail too. It has been proven that robotic simulation can be performed in the simulation model developed by the author.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In a nutshell, the aim to conduct the development of simulation model of a hospital for robotic simulation in Robot Operating System (ROS) using Gazebo has been achieved. The simulation model of a hospital has been successfully developed, and it has proven that the robotic simulation can be carried out smoothly. Information on the layout of the hospital and the operation of the hospital was collected through the site visits conducted at Sungai Long Specialist Hospital. Furthermore, the objective to simulate the human and other objects in the simulated environment mimicking the data collected has been accomplished too. In addition, the last objective which is to conduct simulation of the environment for testing the behaviour of the autonomous delivery robot has also been attained by performing SLAM simulation within the simulated environment. A complete map of the hospital environment was generated by conducting SLAM simulation. The map shows clear, tidy and neat edges and it has shown that the robot is able to detect the obstacles being placed in the environment.

This project has provided a basic framework and structure for the researchers who are interested to continue developing a more complete, reliable and comprehensive simulation model of a hospital. It is significant as there is lack of reliable hospital simulation model which can be applied for robotic simulation especially in Malaysia. One key advantage of this project is that this project has taken the MOH guidelines into account where the sizes and dimensions of the rooms within the hospital are in line with the MOH's requirements. Nevertheless, this study did not cover the development of the whole hospital building as the developed simulation model is a single-floor environment. Another limitation of this project is that it is lack of moving human and object obstacles due to the complexity of the information. Moreover, one issue that was not expected is that few models of items downloaded from online resources contain incomplete meshes and this led to some information loss.

In conclusion, the aim and objectives of this project has been attained. A simulation model of hospital has been built and simulation tasks have been performed properly. Limitations of this project are identified and specified too, whereas recommendations and suggestions will be discussed in the next subsection.

5.2 Recommendations for Future Work

To further improve and extend this work, a few recommendations are discussed in this subsection. First and foremost, future research should attempt to develop and build a hospital simulation model which consists of more floors. This attempt is recommended because this will allow more simulation to be conducted in the hospital environment as some autonomous logistics system requires the robots to travel across different floors. However, such simulation cannot be performed in this built hospital environment as it only consists of one floor. Furthermore, door models which can be opened and closed should be considered to be added into the simulation model in future because some robots might be required to perform door opening and closing simulation even though it might be more complicated and time consuming.

Additionally, to address the problem of missing meshes or physical properties of certain models, future research should consider developing their own obstacle models. It would be interesting if more research can be done on collecting information on the obstacles' physical properties such as their weight and centre of gravity etc as this information can help the researchers to create their own models which are closer to real-life situation. This would lead to a more reliable simulation. For instance, collision occurred will be closer to the reality. Moreover, to ensure collision avoiding algorithm can be better developed for a robot, a direct extension of this work is by including the moving human and moving objects such as trolley and moving patient bed in the simulation model of hospital. The future studies can consider adding handrails at the walls where patients will pass through to make the hospital environment closer to reality look.

Last but not least, factors such as position of charging stations, collection points and delivery points should be considered by the researchers too. The researchers are recommended to communicate with the development team of the autonomous robot to find out the specifications and requirements for different robots so that changes and modifications of the hospital environment can be done.

REFERENCES

Carreira, F., Canas, T., Silva, A. and Cardeira, C., 2006. i-Merc: A mobile robot to deliver meals inside health services. In: *2006 IEEE Conference on Robotics, Automation and Mechatronics*.

CCS Robotics, 2008. *Flexible, On-demand Specimen Transporter*. [online] Available at: <http://www.ccsrobotics.com/products/speciminder.html> [Accessed 4 Sep. 2021].

Coppelia Robotics, 2021. *CoppeliaSim from the creators of V-REP*. [online] Available at: <https://www.coppeliarobotics.com/> [Accessed 1 Sep. 2021].

Evans, J.M., 1994. HelpMate(R): an autonomous mobile robot courier for hospitals. In: *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*. pp.1695–1700.

Fanti, M.P., Mangini, A.M., Roccotelli, M. and Silvestri, B., 2020. Hospital Drugs Distribution with Autonomous Robot Vehicles. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. pp.1025–1030.

Lavrenov, R. and Zakiev, A., 2017. Tool for 3D Gazebo Map Construction from Arbitrary Images and Laser Scans. In: *2017 10th International Conference on Developments in eSystems Engineering (DeSE)*. pp.256–261.

Linz, A., Ruckelshausen, A., Wunder, E. and Hertzberg, J., 2014. Autonomous service robots for orchards and vineyards: 3D simulation environment of multi sensor-based navigation and applications. In: *12th International Conference on Precison Agriculture (ICPA)*. pp.2327–2334.

Ministry of Health Malaysia, 2019. *Handbook on setting up of private hospital in Malaysia Submission Process & Harmonisation of Technical Requirement*. [online] Malaysia Productivity Corporation. Available at: <https://www.moh.gov.my/moh/resources/Penerbitan/Garis Panduan/Garis panduan Umum (Awam)/Submission_Process_Harmonisation_of_Technical_Requirements.pdf>.

Nawawi, N.M., 2000. Master Planning of 10 Existing Hospitals in Malaysia under the Funding of the Asian Development Bank Loan 980 MAL - A retrospective Study. In: *2020 Public Health Group Seminar*.

Niechwiadowicz, K. and Khan, Z., 2008. Robot Based Logistics System for Hospitals-Survey. In: *IDT Workshop on Interesting Results in Computer Science and Engineering*. [online] Available at: <http://www.idt.mdh.se/kurser/ct3340/archives/ht08/papersRM08/30.pdf>.

Nogueira, L., 2014. *Comparative Analysis Between Gazebo and V-REP Robotic Simulators*.

Open Robotics, 2022. *Search Results Search Results*. [online] Available at: <https://app.ignitionrobotics.org/search;q=hospital> [Accessed 10 Apr. 2022].
Open Source Robotics Foundation, 2014. *Gazebo : Tutorial : ROS 2 overview*. [online] Available at: <http://gazebosim.org/tutorials?tut=ros2_overview&cat=connect_ros> [Accessed 27 Aug. 2021].

Panasonic, 2019. *Autonomous Mobility Robot for More Secure and Comfortable Life | Panasonic Key Technologies | Panasonic Newsroom Global*. [online] Available at: <https://news.panasonic.com/global/stories/2019/69861.html> [Accessed 27 Aug. 2021].

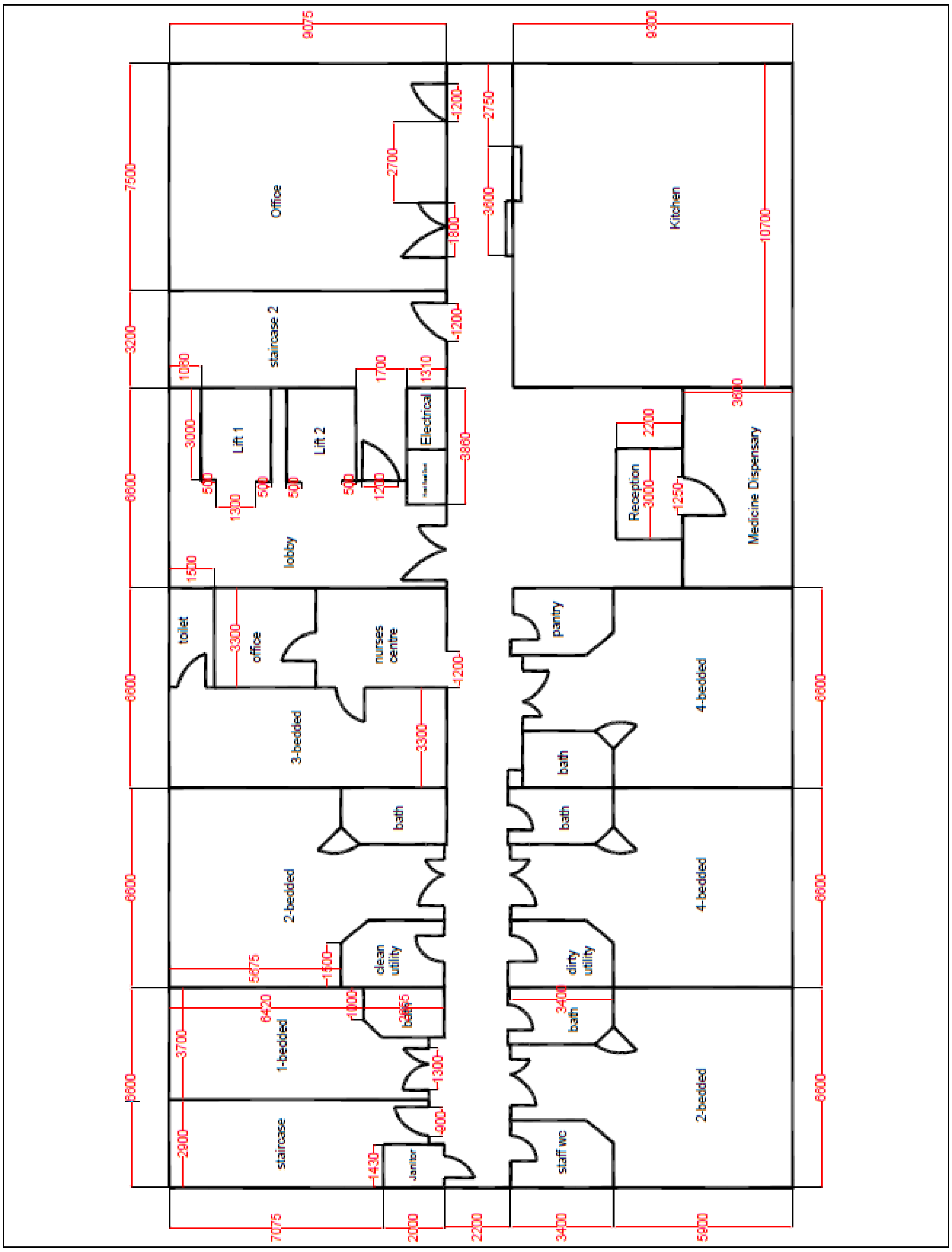
Takahashi, M., Suzuki, T., Shitamoto, H., Moriguchi, T. and Yoshida, K., 2010. Developing a mobile robot for transport applications in the hospital domain. *Robotics and Autonomous Systems*, [online] 58(7), pp.889–899. Available at: <http://dx.doi.org/10.1016/j.robot.2010.03.010>.

Takaya, K., Asai, T., Kroumov, V. and Smarandache, F., 2016. Simulation environment for mobile robots testing using ROS and Gazebo. *2016 20th International Conference on System Theory, Control and Computing, ICSTCC 2016 - Joint Conference of SINTES 20, SACCS 16, SIMSIS 20 - Proceedings*, pp.96–101.

UCSF Medical Center, 2017. *High-Tech 'TUG' Robots Will Do Heavy Lifting at Mission Bay*. [online] UCSF Medical Center. Available at: <http://www.ucsfmissionbayhospitals.org/articles/high-tech-tug-robots-do-heavy-lifting-at-mission-bay.html>.

APPENDICES

Appendix A: Drawn Floor Plan of Hospital



Appendix B: Python Codes of World Launch File

```
import os
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import IncludeLaunchDescription,
DeclareLaunchArgument
from launch.substitutions import LaunchConfiguration
from launch.launch_description_sources import
PythonLaunchDescriptionSource

def generate_launch_description():
    share_path = get_package_share_directory("usr_description")
    world = LaunchConfiguration('world', default=os.path.join(
        share_path, 'worlds', 'hospital_world.sdf'))

    pkg_gazebo_ros = get_package_share_directory('gazebo_ros')

    return LaunchDescription([
        DeclareLaunchArgument('world', default_value=world,
                              description='Path to world file'),
        IncludeLaunchDescription(
            PythonLaunchDescriptionSource(
                os.path.join(pkg_gazebo_ros, 'launch', 'gzserver.launch.py')
            ),
            launch_arguments={'world': world}.items(),
        ),
        IncludeLaunchDescription(
            PythonLaunchDescriptionSource(
                os.path.join(pkg_gazebo_ros, 'launch', 'gzclient.launch.py')
            ),
        ),
    ])
```

Appendix C: Python Codes of Spawn Launch File

```
import os
import xacro
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch.substitutions import LaunchConfiguration
from launch_ros.actions import Node


def generate_launch_description():

    share_path = get_package_share_directory("usr_description")
    xacro_path = os.path.join(share_path, 'urdf/robot.xacro.xml')
    xml = xacro.process(xacro_path)

    use_sim_time = LaunchConfiguration('use_sim_time', default=True)

    name = LaunchConfiguration('name', default='usr')
    namespace = LaunchConfiguration('namespace', default='usr')
    x = LaunchConfiguration('x', default=0)
    y = LaunchConfiguration('y', default=0)
    z = LaunchConfiguration('z', default=0.2)

    return LaunchDescription([
        DeclareLaunchArgument(
            'use_sim_time',
            default_value=use_sim_time,
            description='Use simulation time'),

        DeclareLaunchArgument(
            'name',
```

```
        default_value=name,
        description='Robot name'),

    DeclareLaunchArgument(
        'namespace',
        default_value=namespace,
        description='Robot namespace'),

    DeclareLaunchArgument(
        'x',
        default_value=x,
        description='X position'),

    DeclareLaunchArgument(
        'y',
        default_value=y,
        description='Y position'),

    DeclareLaunchArgument(
        'z',
        default_value=z,
        description='Z position'),

    Node(
        package='usr_description',
        executable='spawn_usr',
        name='spawn_usr',
        output='screen',
        parameters=[{
            'name': name,
            'namespace': namespace,
            'x': x,
            'y': y,
```

```
            'z': z,
        }]),

    Node(
        package='robot_state_publisher',
        namespace=namespace,
        executable='robot_state_publisher',
        name='robot_state_publisher',
        output='screen',
        parameters=[{
            'use_sim_time': use_sim_time,
            'robot_description': xml
        }])
])
```

Appendix D: Python Codes of slam_toolbox Launch File

```
import os
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch.substitutions import LaunchConfiguration
from launch.actions import ExecuteProcess
from launch_ros.actions import Node


def generate_launch_description():
    share_path = get_package_share_directory("usr_mapping")
    namespace = LaunchConfiguration('namespace', default='usr')
    use_sim_time = LaunchConfiguration('use_sim_time', default=True)
    return LaunchDescription([

        DeclareLaunchArgument('use_sim_time', default_value=use_sim_time,
                              description='Use simulation time'),
        DeclareLaunchArgument('namespace', default_value=namespace,
                              description='Robot namespace'),
        ExecuteProcess(
            cmd=['rviz2', '-d', os.path.join(share_path, 'configs', 'display.rviz')],
output='screen'),
        Node(
            parameters=[
                os.path.join(share_path, 'configs',
                             'mapper_params_online_async.yaml'),
                {'use_sim_time': True}
            ],
            package='slam_toolbox',
            namespace=namespace,
            executable='async_slam_toolbox_node',
```

```
        name='slam_toolbox',
        output='screen')
])
```