ENHANCING THE CAPABILITIES OF INTELLIGENT WHEELCHAIR ROBOTS IN APPROACHING AND DOCKING TO SERVICE SCENARIOS

Vadivel Hiroshaan

(168663D)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science in Industrial Automation

Department of Electrical Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

December 2021

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:
V.Hiroshaan

The above candidate has carried out research for the Masters/MPhil/PhD thesis/ dissertation under my supervision.

Signature of the supervisor: Date:
Prof. A.G. Buddhika P. Jayasekara
Senior Lecturer
Department of Electrical Engineering
University of Moratuwa

To all the people who feed us….

ACKNOWLEDGEMENT

This research would have been another dream without the support from the following people.

First of all, I wish to express my sincere thanks to my supervisor, Prof. A.G.Buddhika Jayasekara, for guiding me in the research process and providing feedback and advice over the years.

Next, I wish to thank Prof. D.P Chamdima and Prof. K.T.M.U Hemapala for their valuable feedback. Their feedback was extremely valuable in asking the right research questions and steering the study in the correct direction.

I wish to thank Mr. Sahan Priyanayanna & Mr. Ravindu for mentoring me through the years and helped in the laboratory works. I wish to thank our staff and students from University College of Jaffna.

Finally, I would like to thank my wife, my parents, and my siblings for all the love and support they have given me and for preserving with me during the entire period of the research.

ABSTRACT

The world is presently confronted with the issues of aging and disability as a result of accidents and other events. For the above problem, wheelchairs are the evaded partner in the lives of many differently-abled people to support their day-to-day activities. Many academics are focusing on the usage of wheelchairs to discover better robotics solutions. However, the status of the automated powered wheelchairs is not up to the required level of autonomy in the areas such as docking behavior for a specific task.

Approaching and docking need correct layout information, and identifying the proper degrees of docking in the environment and then assessing them for safety. A machine learning system trained on various docking-level setups is one viable approach. However, retraining is required for different scenarios such as furniture placements or real-time random changes of the layout. Furthermore, vision data is affected by changes in light conditions. The method for detecting docking surfaces, on the other hand, is dependent on geometric information computed from depth data, which makes it invariant to scene or light changes.

As the main aim of this research, a human study was performed to identify the docking behavior of a wheelchair to the table or desk in four different scenarios such as writing, reading, eating, and using a laptop with 3D point cloud data. This research developed a novel method for determining comfortable docking locations based on analyzed ergonomics data. It was gathered from human subjects on actual wheelchair usage. Analyzed data can be applied within a single algorithm to obtain a safe location using 3D point cloud data. While docking with the table, two situations were evaluated. The first is a table with an object on it, while the second is a table with no object on it. If the object is on the table, it will dock based on its location on the table and the availability of open space. If no item is present, it will dock based on the user's desire and the available free space. This wheelchair also has navigation and obstacle avoidance built in to let it travel in a residential setting more independently.

From the human study, the optimized distance between wheelchair back end and table were identified for eating, writing, reading and using a laptop as 29 cm, 27.75 cm, 27.25 cm and 40 cm respectively. The optimized height difference between table surface and wheelchair seat for all scenarios were obtained as the same value as 32.25 cm for a particular table (height = 81 cm). Seat height was not dependent on the scenario. Obtained results were applied to the simulation design for above two situations and validated through fourty test cases.

Keywords—docking behavior, wheelchair, human study, point cloud, ROS, navigation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ROS: Robot Operating System
SLAM: Simultaneous Localization and Mapping
TF: Transform Frame
URDF: Universal Robot Description Format
CAD: Computer-Aided Design
COLLADA: COLLAborative Design Activity
AMCL: Adaptive Monte Carlo localization
SDF: Simulation description format
DOF: Degree of Freedom

CHAPTER 1 INTRODUCTION

With the growing challenge of an older population and rising healthcare demands, the question of ways to improve the healthcare of the disabled and aged people has gained increased attention from both business and the scientific community in recent years. Also a result of accidents and other occurrences, the globe is currently dealing with challenges of aging and handicap. With the increasing growth of social and economic development, as well as the advancement of science and technology, the question of how to improve disabled and elderly people's health care by utilizing ever-advancing technology that is scientific is gaining increasing attention from both the scientific community and industry. In the case of the aforementioned issue, several research projects are focusing on intelligent assistive technology systems for the aged and handicapped.

In current history, a vast variety of assistive devices or systems have been developed to address the requirements of the elderly and the disabled, allowing them to live more freely. Robots are making their way into our lives for a variety of reasons, including toys, friends, guides, and assistants. This is why robots in the service sector were created. After the successful industry robots and experimental humanoids, service robots are a relatively young field of robotics. Service robots are designed to carry out duties that would typically be performed by people in daily operations. They do not, however, have to do it in the same way that people do, nor do they have to appear like humans. They must function in surroundings designed for people, which means they must navigate around unexpected moving items and topics. Service robotics is one of the emerging fields these days and will help people in a variety of ways such as [1], [2], educational [3], health care [4], travel, and entertainment [5], [6]. To accomplish these types of tasks, robots reproduce their behavior or abilities in their daily work.

A great deal of research has gone into designing intelligent wheelchairs that can help disabled users gain independence and enhance their standard of living. The two main types of intelligent wheelchair research are: 1) Intelligent human-machine interactions like speech [7]; 2) Automated navigation like self-planning and navigation [8]. Even though the amount of automation in intelligent wheelchairs has risen. The usage of wheelchairs is critical in addressing the demands of these individuals. People who are unable to use a manual wheelchair owing to strength or handicap can use powered wheelchairs to go around. Using an electric wheelchair, on the other hand, can be a tough, time-consuming, and taxing activity. The operator must be able to perceive their environment properly. Translate their mobility wishes into continuous wheelchair joystick instructions utilizing standard control interfaces. Those without the necessary motor abilities, strength, or peripheral vision are generally unable to use a motorized wheelchair. In a survey of clinicians [9] 10% of patients who receive powered wheelchair training find it extremely difficult or impossible to use, and the number rises to 40% if they are asked about the maneuvering tasks associated with completing activities of daily living. Many academics are focused on the use of wheelchairs to find the greatest robotics solution to their problems, as stated above.

Current powered wheelchair users have several challenges with daily maneuvering chores, and an automated navigation system would be beneficial. Users of future smart wheelchairs would be able to choose from a variety of options autonomously or semi-autonomous ways of operation, depending on their preferred input methods. They will use their smart wheelchair to traverse pre-programmed routes indoors and outdoors, going through doors and up and down elevators, with no physical or cognitive exertion. When it's suitable, the smart wheelchair will interact with the user to alleviate anxiety and develop a customized profile. One of the drawbacks of early work on smart wheelchairs and autonomous driving [10],[11],[12] was that the surroundings had to be permanently altered to allow the wheelchair to move freely.

In the creation of a smart wheelchair (Figure 1.1), the human-smart wheelchair interaction model is critical. The reinforcement learning approach [13] might be used to create an effective user-smart wheelchair interaction model. Sensor feedback from Emotive detectors and other surroundings detecting sensors, such as oculus virtual reality sensors, laser scanners, cameras, and global positioning systems, may be included in the produced model (GPS). These command signals are used by the smart wheelchair to travel and operate its robotic arms. A sophisticated signal processing model will be used to record and evaluate human user thought [10]. The smart wheelchair's activities, including wheel motors and robotic arms, will be controlled by the output signals. A shared control system may be developed to allow for a collaborative effort between the person and the smart wheelchair and to also adapt the help with a smart wheelchair to changes in the participant's performance as well as the environment.

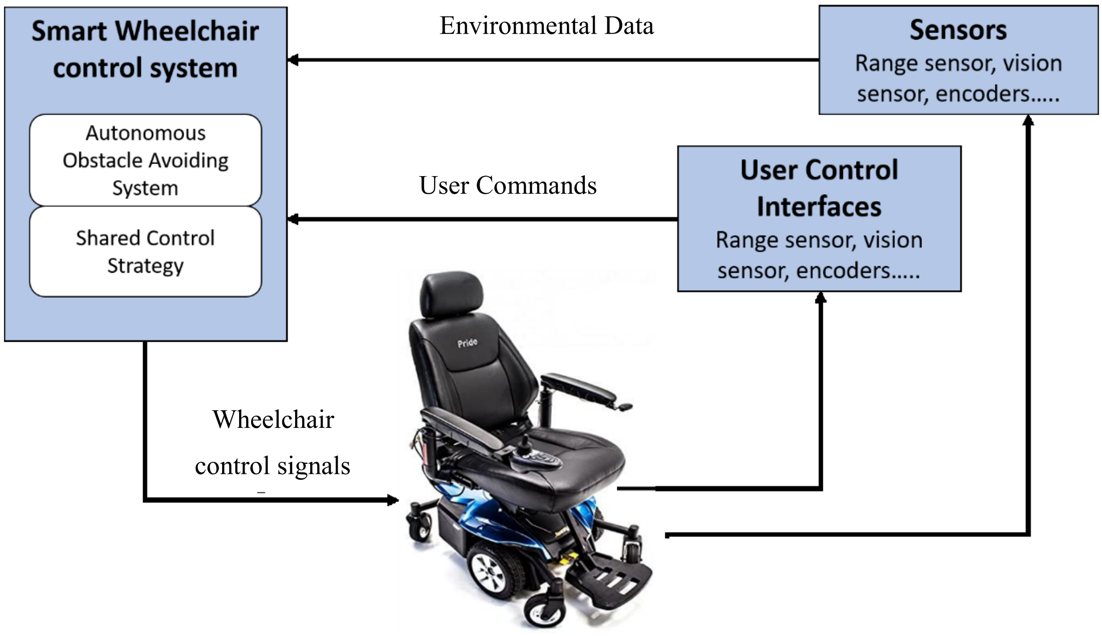


Figure 1.1 A Smart Wheelchair system's overall design architecture

Previous research has proposed numerous "smart" wheelchairs that take advantage of robotic automation to enable an electric wheelchair to drive autonomously as a step toward assisting individuals with needs. Intelligent wheelchairs have the capacity to sense the surroundings, analyze the user's purpose, and apply control techniques that permit complicated operations through smooth and secure motions, to name a few features. Smart wheelchairs are designed to assist in everyday transactional tasks that demand greater expertise to prevent crashes, such as steering and operation in small, narrow, or crowded environments. Door-to-door entrance and docking are two especially restricted aids that receive considerable attention in the literature [14],[15],[16]. Despite the fact that numerous works show the level of autonomy necessary to pass through the door, the concept of completely autonomous docking assistance has been addressed and remains an open research topic.

1.1 Problem Statement

Docking needs correct layout or based information, and identifying the proper degrees of docking in the environment and then assessing them for safety is a difficult visual process. A machine learning system trained on various docking level configurations is one viable approach; however, retraining is required for scenarios such as furniture placements or scene random changes. Furthermore, picture data is impacted by changes in light conditions and vice versa when it is employed. The method for detecting docking surfaces, on the other hand, is dependent on geometric data generated from data, which might be confused by visual or light changes.

Another lack in the current approaching behavior is, there is no auto docking support when the wheelchair is docked near to the table. Users need to manually adjust the wheelchair seat height based on their needs. Also need to adjust the distance between the wheelchair seat and table edge according to the docking purpose with the table. The height difference between the table and wheelchair seat may vary based on the docking purpose. Also, the distance between the seat and table edge will vary. Based on the purpose, users need to adjust manually to fit with the table. It will be a time-consuming task as well as challenging for disabled people. If they are going to manually adjust for docking with the table, it is not autonomous.

Figure 1.2 Wheelchair usage for a specific purpose

The research proposes an approach for determining docking parameters based on the analyzed ergonomics data which is gathered through human subjects on actual wheelchair usage, to be integrated within a single algorithm to get a safe location available in the input scene. The approach involves identifying the table and a sequential search around the bottom of the table and identifying the free space and finding out the safe docking locations and based on the user purpose to dock with the table, applying docking parameters such as distance between wheelchair seat and table edge and height difference between the table and wheelchair gained through analyzed ergonomics data for autonomous docking areas for a wheelchair in the situation (Figure 1.2). The main goal of this research is to develop complete docking support for the intelligent wheelchair, enabling the wheelchair to automatically locate secure docking locations (Figure 1.3) and integrate docking parameters that affect the comfort of the docking for scenarios such as writing, reading, eating and using a laptop.

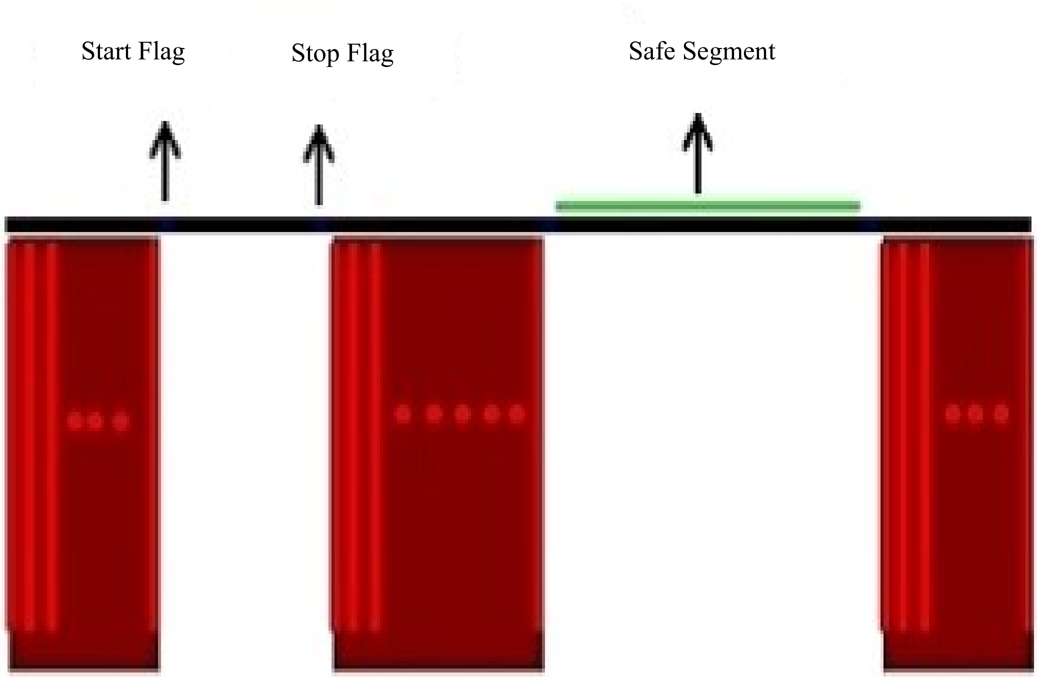


Figure 1.3 The sequential search approach. Adapted from [17]

1.2 Objectives

The objective of this study is to:

To develop complete docking support for the intelligent wheelchair, enabling the wheelchair to automatically locate secure docking locations for scenarios such as writing, reading, eating, and using a laptop.

Considering the complexity of the research problem, the main objective was divided into three sub-objectives as below.

Sub-objective 1: What are the parameters that affect the comfort of the docking

Identify the parameters which are affecting the comfortable docking behavior of the wheelchair for the different purposes to use the table. The height difference between the table and wheelchair and distance between the wheelchair seat and table edge was considered as the parameters which may affect the comfortable docking of the wheelchair to the table for different purposes.

Sub-objective 2: According to user preference wheelchair docks with the table

The following statement is presented to support the importance of the research challenge. This circumstance necessitates the usage of wheelchair docks with tables for the following purposes. Writing, reading, eating, and using a laptop are just a few examples. If no object in the table user needs to input the purpose to use the table for docking. This challenge necessitates determining the most comfortable docking distance for each purpose.

Sub-objective 3: *According to the object on the table wheelchair docks with the table*

Here the initial system identifies the object on the table, according to the object wheelchair docks with the table. The docking distance will be determined using the sub-objective one.

The overall objective can be achieved either by automatically identifying the object in the table, and dock with the table based on that purpose or user selecting the purpose of docking and applying the docking parameters based on the purpose.

1.3 Chapter overview

This study is organized into six chapters. In the first chapter, an introduction to the research, background, problem statement, and objectives of the study were presented. Chapter 2 presents the detailed literature review and background work. Chapter 3 provides details on the methodology of designing and developing the docking system. It is presented as subsections such as Human study design, Algorithm design, simulation design, and hardware design. Chapter 4 describes the experimental parameters that have been identified from the Human study and analyzed based on the experimental results. The final section concludes the research findings together with the recommendations for future work.

1.4 Limitations in this research

There have been many constraints and limitations to the research conducted. This section aims to present delimitations that are of general importance to the research as a whole.

Firstly, this study was conducted in the domestic environment with static objects. It is also assumed that there is no obstacle in the pathway and considered the table height to be greater than jazzy air wheelchair hand rest height. It was considered that the wheelchair moving environment has only one table to dock.

Furthermore, it was assumed that at a time one object was placed on the table for the object detection scenario.

Chapter 2 Literature Review

2.1 Introduction

Embedded computers have become more common in industrial devices during the last decade. Product development, on the other hand, has been concentrated on producing new items for niche markets, rather than the more challenging task of creating things that are generally accessible. This is particularly true when it comes to mobility aids when choosing platforms that are usually unsuitable for people who have physical disabilities and can't perform rapid coordinated movements. Because the Interaction with the embedded controller by the user and sensors affects the functioning of biological devices like wheelchairs, the inputs that are direct that humans provide to intelligent machines are especially inadequate [18]. The study of human-robot interaction (HRI) is becoming increasingly popular. Many elements of HRI have yet to be completely investigated, even though it is not a new field. Because the user supervises the robotic system while it functions independently, bringing in fresh technology with people has been done at the supervisory control level [19].

This section entails the recent advancements in docking behavior when users command a wheelchair with their voice, keyboard, or any other method. Also, the docking behavior of the wheelchair is analyzed based on the detected object in the table.

2.2 Related works in docking behavior based on the user input

There are similar studies on approaching vision in the context of smart wheelchairs and mobile robots in this chapter. Many people see smart wheelchair job docking as a desirable behavior or a vision for the future [19]–[25].

[26] proposed an autonomous docking control for nonholonomic restricted mobile robots and implemented it in an intelligent mobility device or wheelchair to aid the user in approaching resting furniture like a chair or a bed shown in Figure 2.1. A virtual landmark inferred from the desired docking destination is defined here. Then, using a unique definition that permits control over the intended end-pose, address the challenge of maintaining the targeted volume inside the field of vision (FOV) of a tracking camera and docking to the virtual landmark. The depth camera's FOV was used as a constraint in this study, and a nonlinear feedback controller was devised to conduct the docking.

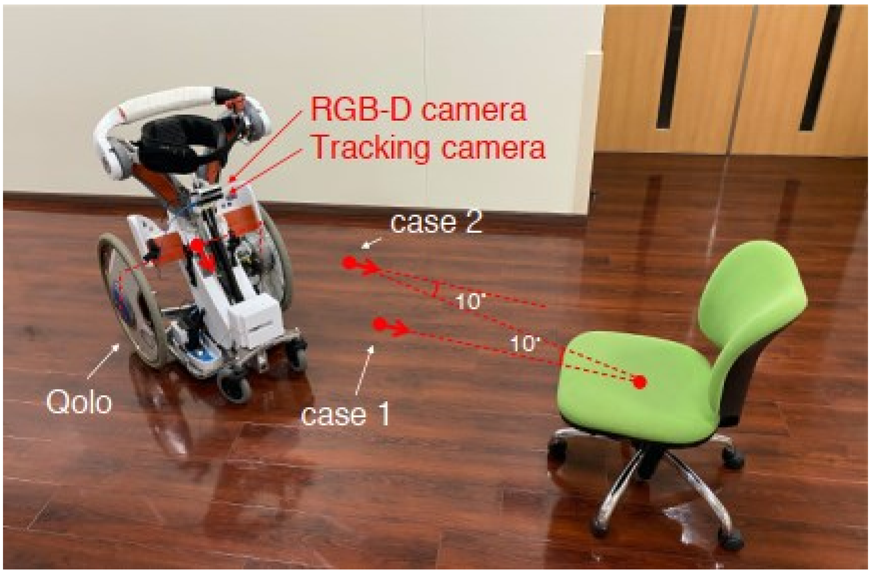


Figure 2.1 Image of the robot Qolo and the desired docking item taken during an experimental evaluation (a chair).

For smart access to scattered charging sites on the map, [27] presented an intelligent path planner. Based on the tasks completed and the remaining battery power, the system determines an overall priority score. Each task's overall power requirements are determined. When the robot's remaining battery power goes below a certain level, the system also handles severe situations. The robots have complete knowledge about the free charging outlets and can plot a course to the next empty charging point right away. If all of the charging points are filled, the algorithm can aggressively empty a charging station occupied by a robot that is completely charged and ready to do the next job. Collision-free pathways are designed in this situation, and the shortest path is delivered to the requesting robot. The locations of the charging stations might be plotted on a map.

A self-reconfigurable mobile robot with an integrated 2-DOF active docking system was demonstrated in [28]. To boost terrain flexibility, the robot has hybrid mobility, which allows it to move using both track-based and wheel-based mechanics. This robot has also been utilized to execute the autonomous docking of several robots, with the active mode of the 2-DOF docking mechanism being shown to couple or decouple with the target robots.

Based on Lidar and a V-shaped artificial landmark, [29] this article developed an autonomous wheelchair/nursing-bed docking approach. The docking mechanism allows the wheelchair and bed to be separated and merged automatically. While maintaining the function of turning over the nursing bed (Figure 2.3), preventing secondary damage to bedridden patients during the transfer process, and considerably enhancing bedridden patients' self-nursing capacity. After successful docking, the distance deviation and yaw angle between the final and ideal positions are regulated between 0.02 m and 2.5 under ambient light conditions.

[30] proposed a sectional auto-docking charge control system for mobile robots. Several ultrasonic sensors are used in conjunction with classic infrared navigation to provide exact relative posture at a cheaper cost than sight or laser guidance. Wall following, coarse positioning, and exact positioning are the three components of the auto-docking procedure, and each section is explained. Two devices are used in the method: one is mounted on a mobile phone and contains two infrared receivers, at least two ultrasonic transceivers, and an RF transceiver, while the other is mounted on a charging pile and contains an infrared transmitter, two ultrasonic transmitters, and an RF transceiver.

Some methods rely on tailored docking systems to make the task easier to solve. Bailey *et al.* [31] used a laser scanner and geometric constraints to measure the relative container position of a rectangular container of known dimensions to docking purposes of the mobile robot. Here Distributed Framework for Mobile Navigation (DAMN) used a modular architecture based on behavior. The raw sensor data is processed to create reliable virtual sensors that feed data to the behavior modules. The behavior outputs are then processed asynchronously by centralized arbiters, who decide the setpoints for the mobile robot drive and steering actuators.

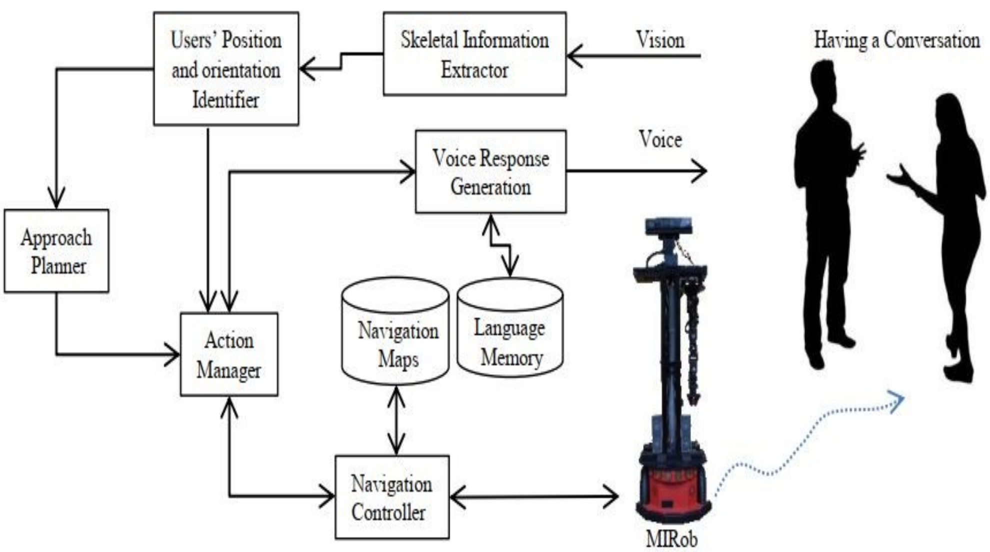


Figure 2.2 System overview of the Human-Robot interaction

Samarakoon *et al.* [32] were concentrating their efforts on understanding how service robots approach humans in conversational contexts. The behavior of an individual meeting two individuals in a conversational environment was replicated in one sample to create a fluid approach behavioral system in a specific situation, or a participant conversed with another person (Figure 2.2). An RGBD camera was used to detect the setting, the user's location, and path identifiers, and an approach planner was used to guide the service robot towards a smooth approach movement in this project.

Figure 2.3 wheelchair's docking process in U shaped bed

Most of the methods listed above use fiducial markers or landmarks to offer a narrow solution to the docking problem [33]–[35] or have generalized the problem by relying on customized docking frameworks [21], [31]. Ren *et al.* [21] suggested an improved approach for controlling the docking of an intelligent wheelchair in a small space. The location and orientation of the wheelchair are calculated using a formula based on specialized symbol identification for U-shaped bed detection (Figure 2.3). A control algorithm for docking, which comprises designing a path and tracking the course of a line, is meant to direct the wheelchair to dock automatically into the bed. The path planning guarantees that the wheelchair may reach the bed's entrance without colliding with it and the algorithm for monitoring line trajectory directs the wheelchair to travel in a docking into the bed with a nice arc trajectory (similar to [33], [34]).

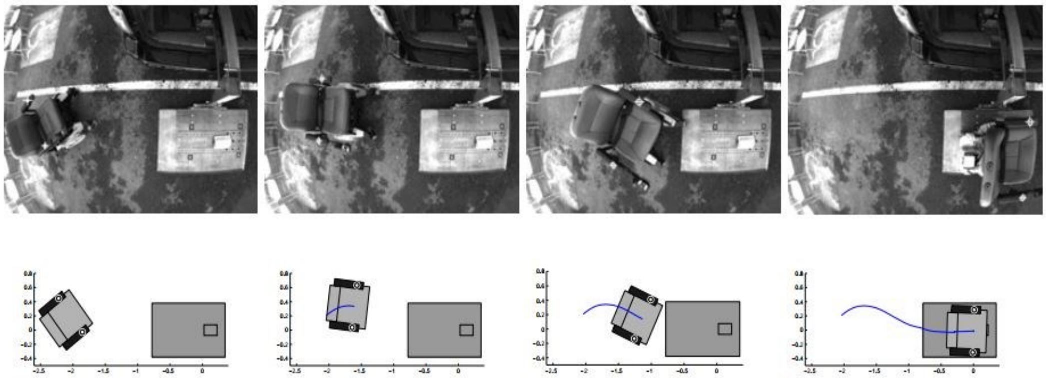


Figure 2.4 Docking a wheelchair into an automobile vehicle lift platform

Rather than docking at tables and chairs, Sermeno-Villalta & Spletzer [22] demonstrated perception control for parking a wheelchair autonomously onto an automotive platform for vehicle raise (Figure 2.4). The Automated Transport and Retrieval System (ATRS), a novel form of transportation for drivers with lower-body disabilities, has this feature. The ATRS is a computer vision, robotic systems, and automation system that can be installed inside a standard minivan or SUV (SUV). The ATRS is propelled by a self-driving "smart" wheelchair system that travels between the driving seat and a motorized lift in the back of the vehicle, removing the need for an attendant.

Nuttin *et al.* [23] used joystick signals to estimate user aim and detect tabletops to help with wheelchair docking. To solve the problem of an identified table with an ambiguous docking position, their algorithm chooses a consensus location halfway between the sides. There are also assistive wheelchair works that rely solely on the person to identify docking sites[19], [24]. Combining a brain-computer interface with a commercial wheelchair via a common control layer can result in a feasible brain–actuated wheelchair. The shared controller combines the user's intelligence and wishes with the machine's accuracy. In [19], they discovered that this allowed both expert and novice users to accomplish a driving assignment that included docking to two different tables along the route.

There are methods for finding supporting planes in a scene that use 3D data, which is close to the approach. Williamson *et al.* [36] proposed using a Mixture of Gaussians probabilistic model constructed with Expectation-Maximization (EM) to find potential docking artifacts by segmenting data from a LIDAR 3D laser range-finder for modeling plane-like data clusters—which, however, involves understanding the number and types of clusters a priori and is prone to discovering low local minimum solutions. For a smart wheelchair, Wei *et al.* [25] used a shared control system based on a 3D semantic map. To increase the wheelchair's sense of its surroundings, a 3D semantic map is used. The wheelchair can recognize unique things in new interior environments and use that information to assist the user with object-related navigation tasks including door passing and furniture docking, but no implementation details are offered.

Some works [16], [25], [34] offer a general solution for identifying docking structures in the scene, but the search and determination of exact safe positions and orientation information are uncertain. Using a table as an example, the raw data collected by Kinect was recovered using the 3D conceptual map work in [25]. A 3D conceptual map contains parts of intermediate discoveries, such as identifying the tabletop and walls of the building and projecting the obstacle on the ground. After that, the object recognized as a PC moved to the desired position according to the wheelchair. Additionally, image-based techniques are vulnerable to issues such as shifting contrast and perception distortions, which have been described as bottlenecks in certain schemes [33].

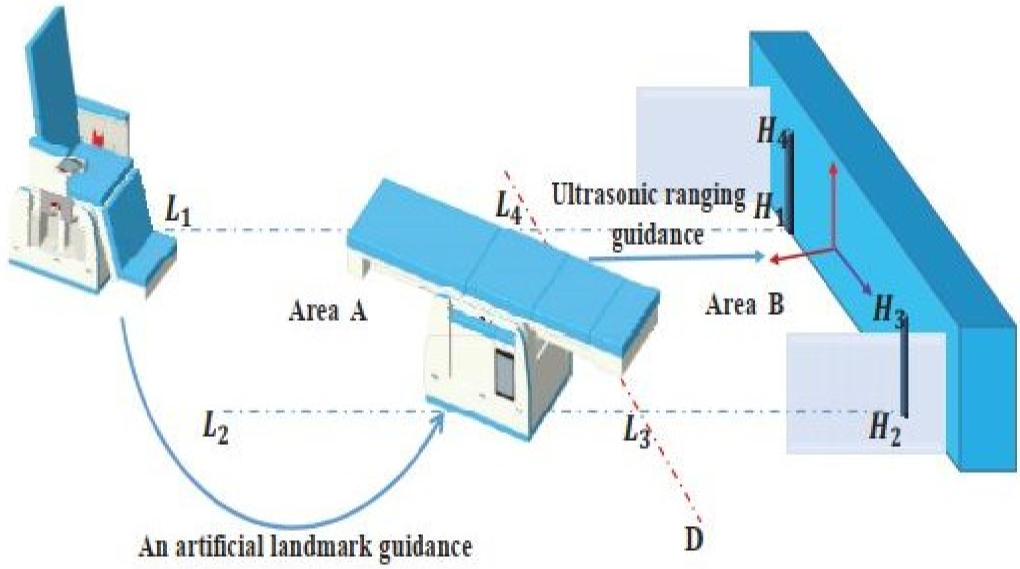


Figure 2.5 Diagram of the automated docking procedure with the docking control region divided.

Researchers also have developed the docking process for the wheelchair to be attached to the bed in parallel. Artificial landmark [37] and visual localization [38] were used to identify the bed position orientation with a world frame and coordinated system. These studies were geared at the hospital setting rather than the household context. [37] presented a docking control method based on ultrasonography and a small to large scale reference. The man-made landmark on the supplementary bed is identified initially, and it is then used to build the global control points; finally, the landmark in the visual stage establishes the relative coordinates from the wheelchair to the supplementary bed (Figure 2.5). Finally, the wheelchair goes to the supplementary bed to complete the approaching task, based on the distance determined by the ultrasonography between the wheelchair and the bed. The Robotic Operating System

(ROS) is used to implement the wheelchair docking platform and WINCE platform. [38] presented a position feedback-based visual docking servo control technique. The relative posture link between the wheelchair position and orientation and the world coordinates was first identified using the Intelligent wheelchair visual localization approach. The motion controller was then built utilizing the Mecanum wheeled omnidirectional platform to investigate the connection between where you are now and where you want to go for the wheelchair and the favored stance.

Siddarth and Argall [17] suggested a method for detecting safe docking spots autonomously utilizing data from a 3D point cloud without the need for fiducials visible or surroundings modification. Through assessments on variously shaped docking structures in various combinations, the approach has demonstrated satisfactory accuracy and has been proven to be effective in the discovery of safe and aligned docking sites. In the case of assistive wheelchairs, as well as autonomous mobility robots, finding secure docking places is critical. A path planner may design and execute unique trajectories to perform the docking maneuver by recognizing safe docking sites and alignment information.

2.3 Related works in docking behavior based on the object detection on the table

Some research proposed a robot docking solution focused on neural networks and computer vision, with Orange fruit serving as a visual dependency on the table's edge. Martınez-Marın & Duckett [34] used color segmentation in HSV space to define an orange line on the edge of a table. Weber *et al.* [33] showed that solely machine-learning vision and control techniques can effectively grab an item in a real-world robotic docking issue (Figure 2.7). A greater associative network was created in this brain network, in which mirror neurons will arise. Mirror neurons have been discovered in the motor and language cortex, and they activate when an action is done in one of two ways. The network takes data from a variety of sources and converts it into a secret code. Unsupervised machine learning with sparse coding, similar to the Helmholtz machine, is used to train the vertical connections. The data is gathered via robotic activities that are carried out in real-time in the surroundings. The statistics only include real-time information, thus the entire sequence of events is unknown. As a result, unlike mirror neurons, neurons do not always fire for a long time.

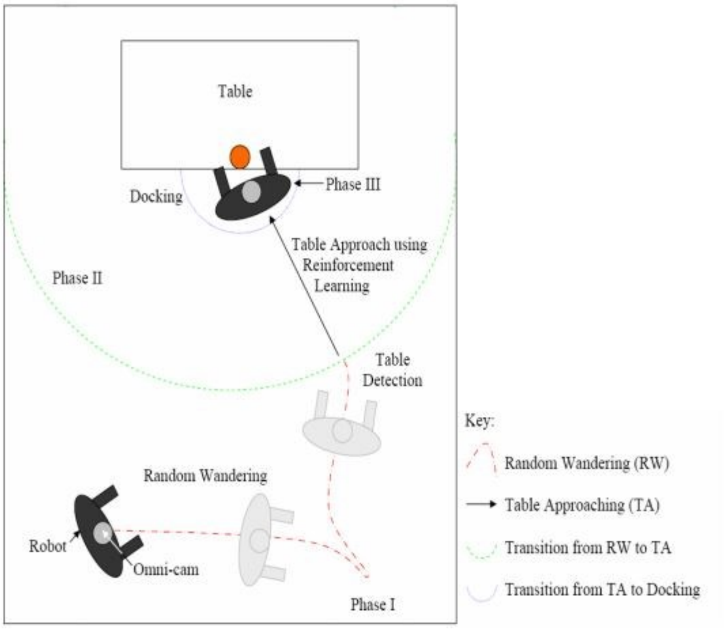


Figure 2.6 PeopleBot robot docking scenario

[39] outlined a research proposal for strategies that would improve the experience and quality of life of wheelchair users by minimizing the cognitive load imposed by repetitive and difficult navigation tasks and movements. This will be accomplished by dividing control between the user and a self-contained controller. To attain this goal, a variety of strategies will be employed. Computer Vision methods will allow the (semi) automated recognition of locations in the user's house, based on the detection and classification of items. The wheelchair will dock with the table using Simultaneous Localization and Mapping (SLAM) and topological mapping.

Muse *et al.* [35] suggested a landmark-based docking approach that uses an omnidirectional camera and image processing to detect a landmark under the table using the PeopleBot robot (Figure 2.6). The actor-critic reinforcement approach was utilized to train the networks to fulfill their goals in the initial docking and extended navigation. The enlarged navigation system has to understand its network for the robot to go to the table. Because there are multiple sub-tasks and different sensors employed, a phased method was adopted to tackle this challenge.

Figure 2.7 Mobile robot docking behavior to grasp the orange fruit

In a visual servoing framework, Tomas Martinez and Tom Duckett [34] developed a reinforcement learning-based technique for mobile robot docking. (Figure 2.7). Using the sweeping of adjacent property and model-based sweeping, a novel Reinforcement Learning(RL) method was proposed that is better suited to actual robots working in spaces of continuous state. They also showed how a simple linear controller may help speed up the learning process even further. The technique does not require any geometric models or calibration, and the response behavior is noise and disturbance-resistant. The closed-loop approach employs a relative coordinate system. [32] implements the docking method for a mobile robot using the Distributed Architecture for Mobile Navigation (DAMN) behavior patterns architecture. Virtual sensors reduce sensor error and give the best estimations in the format necessary. A distributed system of behavior modules gives information on what activities should be done independently, allowing data to be collected without bottlenecks. The information for the behaviors is interpreted by centralized arbiters, who take into account all relevant facts while reaching a judgment. In the presence of other impediments, the architecture proposed in this study enables robust collision-free docking with a container.

Table 2.1 Summary of the literature review

Author	Findings
Chen *et al.* [26]	used a defined virtual landmark inferred from the desired docking destination to approaching resting furniture. Then, using a unique definition that permits control over the intended end-pose, address the challenge of maintaining the targeted volume inside the field of vision (FOV) of a tracking camera and docking to the virtual landmark.
Ravankar *et al.* [27]	a known map consisting of charging points was used by a robot to find the next empty charging point. If all of the charging points are filled, the algorithm can aggressively empty a charging station occupied by a robot that is completely charged. Collision-free pathways are designed in this situation, and the shortest path is delivered to the requesting robot
Sebastian *et al.* [28]	to couple or decouple with the target robots, it has also been utilized to execute the autonomous docking of several robots, with the active mode of the 2-DOF docking mechanism.
Zhu *et al.* [29]	nursing-bed docking approach using LIDAR and a V-shaped artificial landmark by autonomous wheelchair. While maintaining the function of turning over the nursing bed preventing secondary damage to bedridden patients during the transfer process.
Zhang *et al.* [30]	sectional auto-docking charge control system for mobile robots using ultrasonic sensors used in conjunction with classic infrared navigation to provide exact relative posture.
Fearn *et al.* [39]	The wheelchair used computer vision methods that will allow the (semi) automated recognition of locations in the user's house, based on the detection and classification of items. The wheelchair will dock with the table using SLAM and topological mapping.
Bailey *et al.* [31]	used a laser scanner and geometric constraints to measure the relative container position of a rectangular container of known dimensions to docking purposes of the mobile robot.
Samarakoon *et al.* [32]	the behavior of an individual meeting two individuals in a conversational environment was replicated in one sample to create a fluid approach behavioral system in a specific situation, or a participant conversed with another person.
Ren *et al.* [21]	to dock a wheelchair at a U-shaped bed, they use a T-shaped fiducial on the bed to identify it.

Sermeno-Villalta & Spletzer [22]	using visual input control and fiducials for image recognition and monitoring to dock a wheelchair onto a vehicle lift pad.
Nuttin *et al.* [23]	used joystick signals to estimate user aim and detect tabletops to help with wheelchair docking. To solve the problem of an identified table with an ambiguous docking position their algorithm chooses a consensus location halfway between the sides.
Williamson *et al.* [36]	using a Mixture of Gaussians probabilistic model constructed with Expectation-Maximization (EM) to find potential docking artifacts by segmenting data from a LIDAR 3D laser range-finder for modeling plane-like data clusters.
Wei *et al.* [25]	used RGB-D data and the Random Sample Consensus (RANSAC) algorithm to extract planes in the scenario, which are then compared to entries in a model library to classify objects. The research suggests that feature extraction be used to find free docking positions and alignment data, but no implementation specifics are given.
Xingnan *et al.* [37]	A docking controlling technique that is based on ultrasound and a simple artificial marker. The artificial marker on the supplementary bed is identified initially, and it is then used to build the global coordinates; finally, the landmark in the visual stage establishes the relative location from the wheelchair to the supplementary bed.
Xiuzhi *et al.* [38]	visual localization was used to identify the bed position orientation with a world frame and coordinated system. This study was geared at the hospital setting rather than the household context.
Siddarth and Argall [17]	The system for docking approach for general table and canteen table with utensils was suggested. It provides a solution for wheelchair docking systems such as tables and desks that uses point cloud data to provide correct orientation information.
Martınez-Marın & Duckett [34]	used color segmentation in HSV space to define an orange line on the edge of a table to identify the table docking position.
Weber *et al.* [33]	robot docking solution focused on neural networks and computer vision, with Orange fruit serving as a visual dependency on the table's edge
Muse *et al.* [35]	landmark-based docking technique using an omnidirectional camera, in which image processing is used to identify a landmark located under the table.

Table 2.2 Analysis of research gap

Objectives	**Limitations in existing systems**	**Developed in this research**
Identify the parameters affecting the docking behavior	In the existing researchers not focused on this objective	Identified the parameters
Docking based on the user preferences	• [22,23,24,29] focused only on docking with the table facing side based on the free space availability. • [32] implemented to dock with the table on all sides. • [20,21,30,31] developed to move the wheelchair near to the table for the docking purpose.	Implemented dock with the table based on the preferences and also checked the availability on all four sides of the table.
Docking based on the object on the table	• [26,27,28] used object detection to identify the docking edges of the table.	Developed to dock with the table based on the identified object position on the table and obstacle identification also implemented.

As a summary of all the above research projects mentioned in above Table 2.1, they were focusing on docking without considering the purpose of docking. They didn't focus on the service scenarios. Therefore there is a lacking part after the docking is there. That is wheelchair needs to be then adjusted furthermore by manually or using a joystick or any other means based on their purpose of using the table. This is not an autonomous behavior. There is no research focused on finding out the user satisfied distance for various purposes in the table. Based on the above research gap shown in Table 2.2, this thesis emphasizes that the performed study focuses on autonomous docking interpretation as well as determining the safe and comfortable location and coordination for the different scenarios in the table, which was absent in the previous studies, which did not have a clear autonomous docking.

2.4 Robot Operating System (ROS)

Any robot must be equipped with a variety of gadgets, sensors, and activators to function. According to [40], establishing a Robot Operational System (ROS), which offers both the communication protocol and its implementation for the robot, is one of the best ways to combine these characteristics.

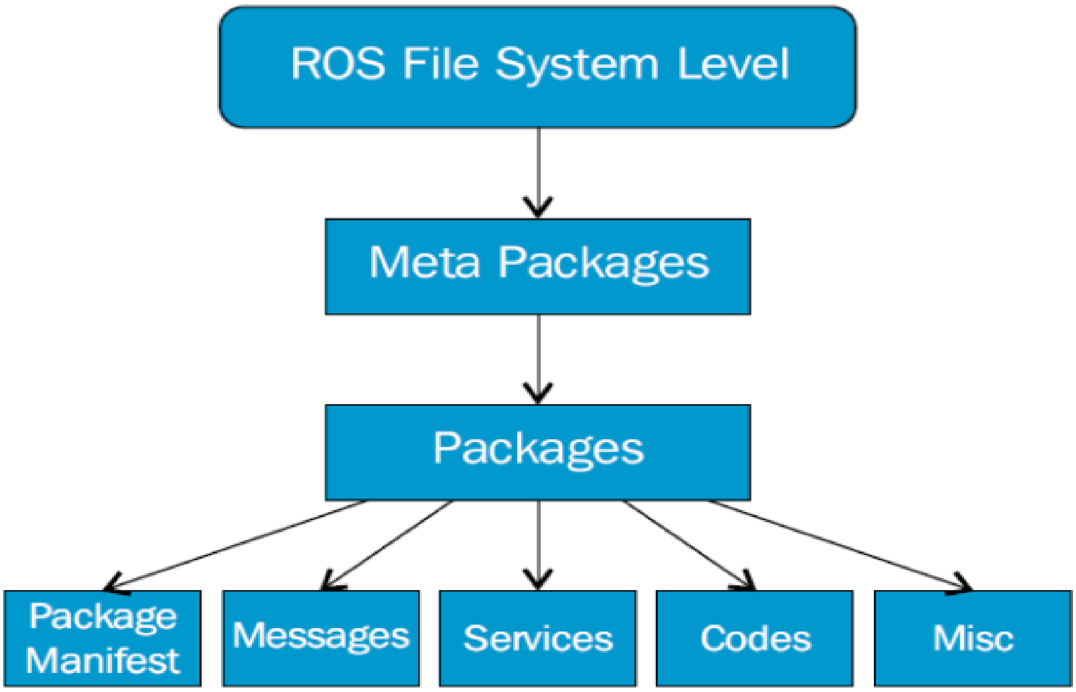


Figure 2.8 ROS File system. Adapted from [40]

The system is popular among students and academics working on robotics projects of various scales because it offers many major advantages over other robotic platforms. ROS has several ready-to-use packages that may be used to create localization, navigation, and motion planning for a variety of robots. The system is jam-packed with robotics device drivers and sensor and actuator packages. Because various jobs are performed individually, the modularity of ROS substantially improves the system's stability throughout the operation. The platform allows scripts created in the C++, C, Java, and Python computer languages to be combined into a single project. Furthermore, ROS comes with several helpful tools for visualizing, troubleshooting, and simulating robotic systems. Another benefit of the platform is that it has a vibrant online community where users may seek assistance and discuss their experiences [40]. The following explanations can be used to describe the ROS file system:

- The primary structural element of the ROS file system is a package, as shown in Figure 2.8. Packages are the simplest level of the ROS software hierarchy. They may include libraries, utilities, executable files, and other items.
- To make code perception easier, packages that need to execute specific actions with distinct parts of the robot hardware can be grouped into Meta Packages.

- A package manifest is a text file that describes the contents of a package. The manifest's primary purpose is to declare package dependencies.
- A message is a piece of data that is sent from one ROS process to the next.
- The service is a request-response communication system for processes.

ROS uses a network of processes called ROS nodes to do computing. A node is a piece of code that is distinct from the rest of the system and is responsible for functions such as hardware initialization or control, processing incoming data or operator signals, and so on. The nodes can publish messages about a subject to communicate information. If a node needs further information, it might subscribe to a topic. A master is in charge of the communication between nodes.

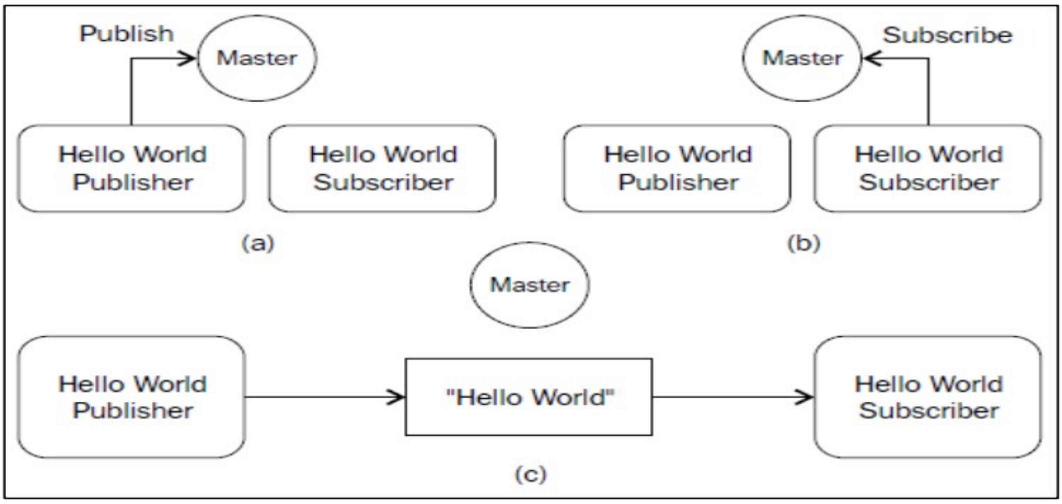


Figure 2.9 Communication between ROS nodes. Adapted from [40]

Figure 2.9 depicts the process of nodes delivering information through ROS. To begin, the node publishes the topic and tells the Master that the data has been published through this subject (step "a" in Figure 2.9). Following then, the Master waits for any node to subscribe to this subject (step “b”). Following a subscription, the master establishes a peer-to-peer connection between the nodes, which allows data to be transferred (step "c").

2.5 Gazebo

The Gazebo program gives the simulator package 3D capabilities. When it comes to indoor robot simulations, a two-dimensional simulator is usually sufficient, but when it comes to outside mobility robots, a three-dimensional simulator is required. The Gazebo was created to precisely replicate the changing situations that a robot could face. I Simulated objects that have mass, velocity, and a variety of other properties that let them interact with the environment realistically. As a result, these items may be pushed, dragged, carried, and so on. The design of Gazebo is founded on the concept that creating new robots, actuators, sensors, and other things should be simple. The general structure design of Gazebo components is shown in Figure 2.10. It uses two external libraries for rigid body dynamics and collision modeling, as well as OpenGL/GLUT for display. The usage of external libraries simplifies simulator development, while internal abstraction allows them to be replaced if better options become available. Gazebo works in the same way as Stage does with Player. The gazebo is treated by the Player device server in the same way that all other devices capable of transmitting and receiving data are treated [41].

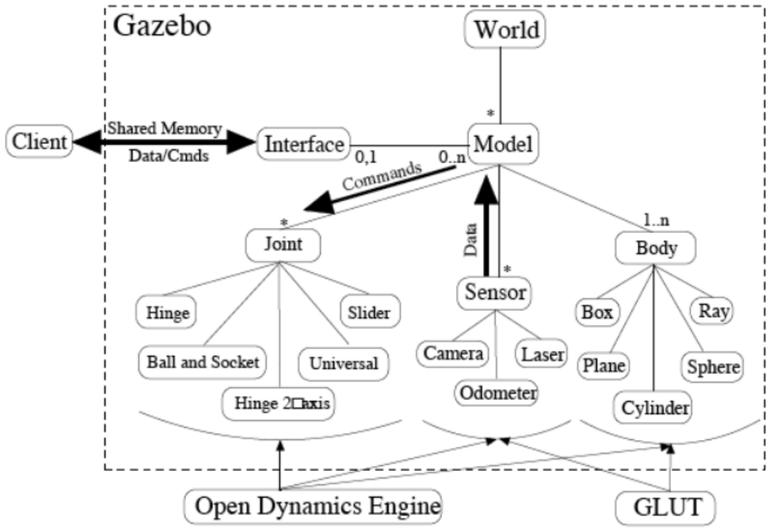


Figure 2.10 Gazebo components

2.6 Overview of the different sensors used in this research

Artificial sensors play a significant part in service robot applications by giving information about the surrounding environment to the control system. In both static and dynamic settings, the mobile robot uses this information for a variety of sensors, including obstacle detection and collision avoidance. Range sensors, including sonar, ultrasonic, infrared, and laser rangefinders, are the most common artificial sensors used in autonomous systems. The characteristics of each sensor will differ, and each sensor will have its own set of benefits and drawbacks, but the type of autonomous system, the application, and the related budgets will all influence the sensor selection. The below subsection mentioned what are the different sensors used in this research.

2.6.1 Laser Range Finders

Laser rangefinders shown in Figure 2.11 can offer precise range data, as well as high resolution and high-speed data streaming. They are made up of a laser transmitter and a receiver and work similarly as ultrasonic sensors. A low-power narrow laser beam is emitted, reflected off objects, and returned along the same path. The laser signal's time of flight (TOF) is recorded and utilized to determine the distance to the item based on the laser light's known speed. Many high-end service robotics projects use laser rangefinders for applications including localization [42], dynamic map construction [43], and collision avoidance [44].

Figure 2.11 Laser Range Finders - SICK LMS 291 and Hokuyo URG 04 LX

Rather than using an array of sensors like those used in ultrasonic systems, these sensors employ a spinning mirror to allow readings from various angles. The advantages of this sensor over many other sensor types are its quick reaction, excellent distance measurement precision, and the fact that it is not affected by environmental noise or illumination conditions. The drawbacks of this sort of sensor include the fact that they normally only put out a beam in a 2D plane, have trouble recognizing glass because the laser tends to travel right through, and are more expensive than any of the other sensor types discussed in this section.

2.6.2 Kinect Sensor

The Kinect sensor has numerous sophisticated sensing components. It features a depth sensor, a color vision, and a four-microphone array, allowing it to capture three-dimensional motion, identify faces, and recognize speech (Figure 2.12). In Figure 2.13, the infra-red (IR) projector, RGB camera, and Infrared camera are all positioned. The depth sensor is made up of an infrared projector and an infrared camera, which is a grayscale complementary metal-oxide-semiconductor (CMOS) sensor.

Figure 2.12 Kinect sensor

With an infrared projector, an infrared camera, and a color camera, the Kinect is a superb imaging tool for both humans and robots. To enhance their range and autonomy, robots must be able to sense the surroundings around them. SLAM is one technique to do this.

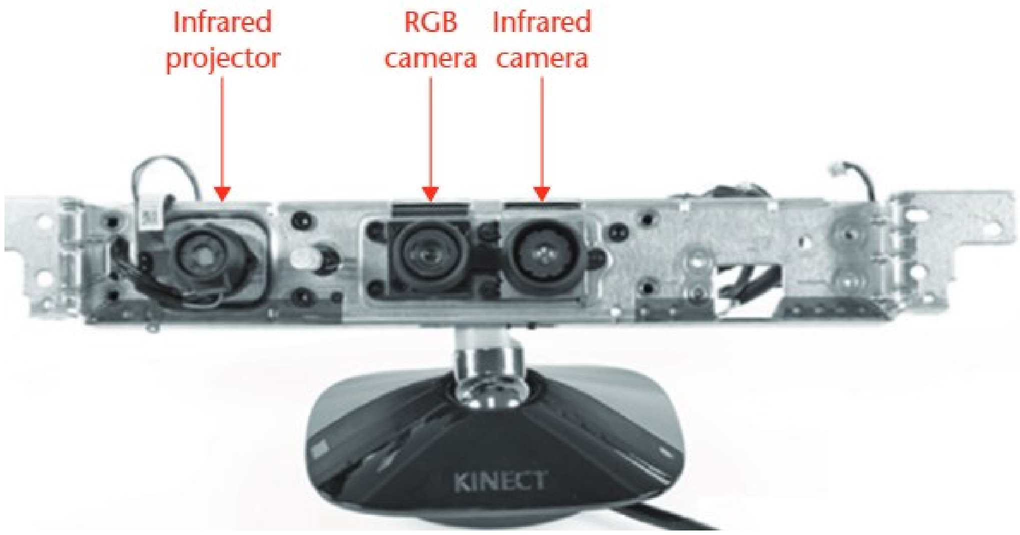


Figure 2.13 Components of Kinect sensor

The Kinect sensor and laser sensors were chosen to achieve our goal based on the above discussions about sensors. The Kinect sensor was chosen because it is capable of capturing the depth of the environment and observing the area around them. It was also chosen because of its availability and pricing constraints. Because of the benefits of the laser sensor over many other sensor types, it was chosen. This sensor was chosen because of its rapid response, high distance measurement precision, and the fact that it is not affected by environmental noise or illumination conditions.

CHAPTER 3 METHODOLOGY & SYSTEMS DESIGN

3.1 Introduction

This chapter provides a summary of the research approach, including the software, hardware, and experimental sets used to investigate docking behavior. Human study design, algorithm design, simulation design, and hardware design are the key subparts of this chapter. The design of the human study and how to perform the human study were covered in the human study paragraph. The next section discussed algorithm design. The simulation design and how we implemented the algorithm were covered in the next part. The final part discussed the transformation of simulation design into a hardware implementation.

3.2 Methodology

This research ended up with several iterations in the process of refining the research strategies and conceptual design as a response to new observations, circumstances, and due to comments brought about during the progress reviews.

To identify the parameters which affect the docking behavior of the wheelchair in different service scenarios, a human study was carried out. Affecting parameters and their values were gained from the analysis based on the human study.

The Algorithm was designed using the analyzed ergonomics data to develop the docking behavior of the wheelchair in different service scenarios. Two conditions were considered to dock with a table such as an object detection and docking and docking based on the user input. The Algorithm was developed to identify the surface of the table, then identify the corners and edges of the table, after that free space identification under the table, finally docking of the wheelchair based on the point found from the free space identification.

The developed algorithm was tested using the simulation design implemented in the ROS platform. Simulation of the intelligent wheelchair was designed in the domestic environment like a bedroom. It was developed to come from the entrance of the room and then identify the table in the room. The rest of the docking behavior was implemented based on the algorithm design.

Hardware design contains the details of the input connection, output connection, and wiring diagram of the wheelchair.

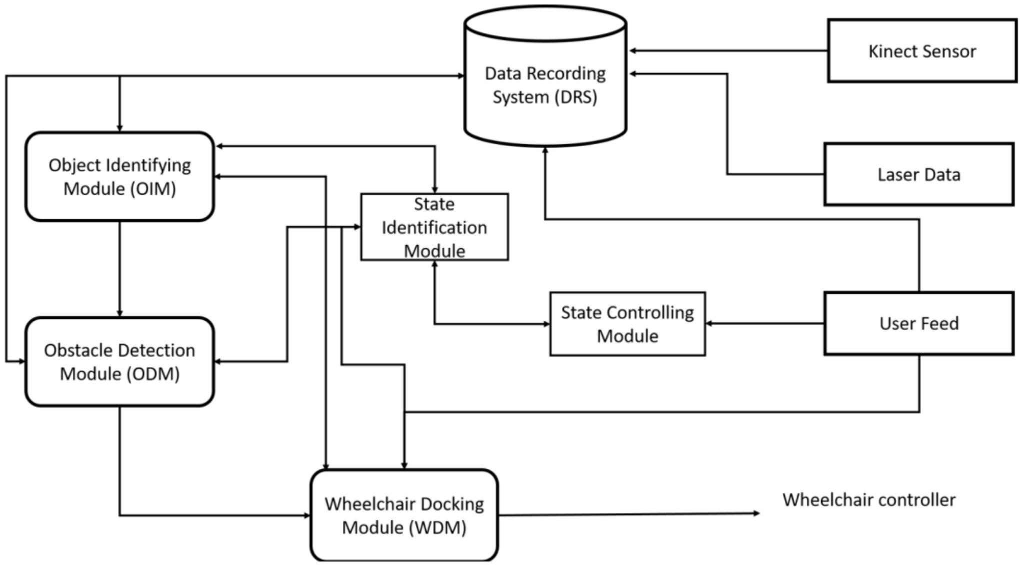


Figure 3.1 System Overview

The above Figure 3.1 shows the overview of the proposed system. The Kinect sensor was used for free-space identification. Laser data was output from the laser sensor to detect the obstacles. User feed was used to capture the user input. The Data Recording System (DRS) was designed to process the sensor data and user input. Object Identifying Module (OIM) was designed to detect the object availability on the table. Obstacle Detection Module (ODM) was implemented to identify the obstacles under the table. Wheelchair Docking Module (WDM) was developed to control the movement of the wheelchair, based on the output from the above modules.

The outline of the prototype design is given below:

1. Comprehensive literature review & background study
2. Research gap identification
3. Human study
4. Analyze human study data
5. Parameters identification
6. 3D model design of wheelchair
7. Environment design
8. Determine the docking area
9. Table surface detection, and corners and edges identification
10. Object detection
11. Free space identification
12. Design simulation setups to check the test cases for wheelchair docking based on the user preferences
13. Design simulation setups to check the test cases for wheelchair docking based on the object detection on the table
14. Integration of obstacle avoidance and navigation for a known environment

15. Validate the docking behavior of the intelligent wheelchair

3.3 Human Study Design

The task of the human study was to identify the docking behavior of the wheelchair to the table. The main intention of the study was to gather information necessary for implementing a human-like docking mechanism for an autonomous wheelchair. For this purpose, four types of scenarios were considered in this work.

Here, the office table which can be considered as the standard table specifications in the American with Disabilities Act (ADA), where a wheelchair can be compatible fit to it and the manual wheelchair was used in this study to collect data by replicating human-like behavior in the docking control for the wheelchair and table. For the data collection, thirty (30) human subjects (male 15 & female 15), whose age was between 20-26 years (for the age Mean value is 22.23 and the standard deviation is 1.498), had participated as the manual wheelchair riders or users. All the participants were normal students of the University College. Furthermore, before starting the study, the participants were asked whether they have experience in manual wheelchair operation and how to be attached to the particular table for some scenarios performed at the table.

For the four scenarios tested in manual wheelchair users (eating, writing, reading, and using a laptop), the wheelchair users need the table to complete this task. The following data were collected from each human subject selected for the study, to analyze and find the parameters that are affecting the docking of the wheelchair. They were human height, height above the hip, the distance between the table and wheelchair back end (DBWT), the height difference between the table surface and wheelchair seat (HDBTW), and orientation. These parameters were decided according to the measurements taken from the manual wheelchair users.

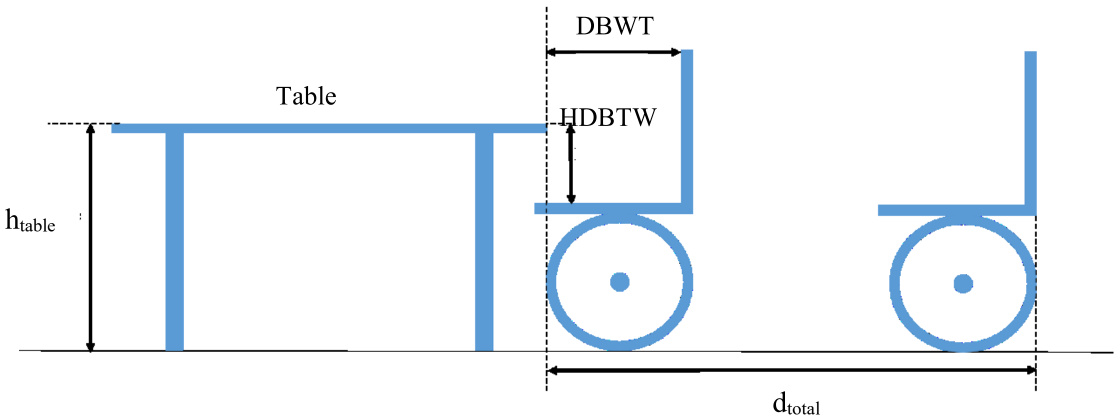


Figure 3.2 Diagram for the human study design, marked with all the parameters measured in cm.

Figure 3.3 Actual image of the human study for the scenario of using a laptop.

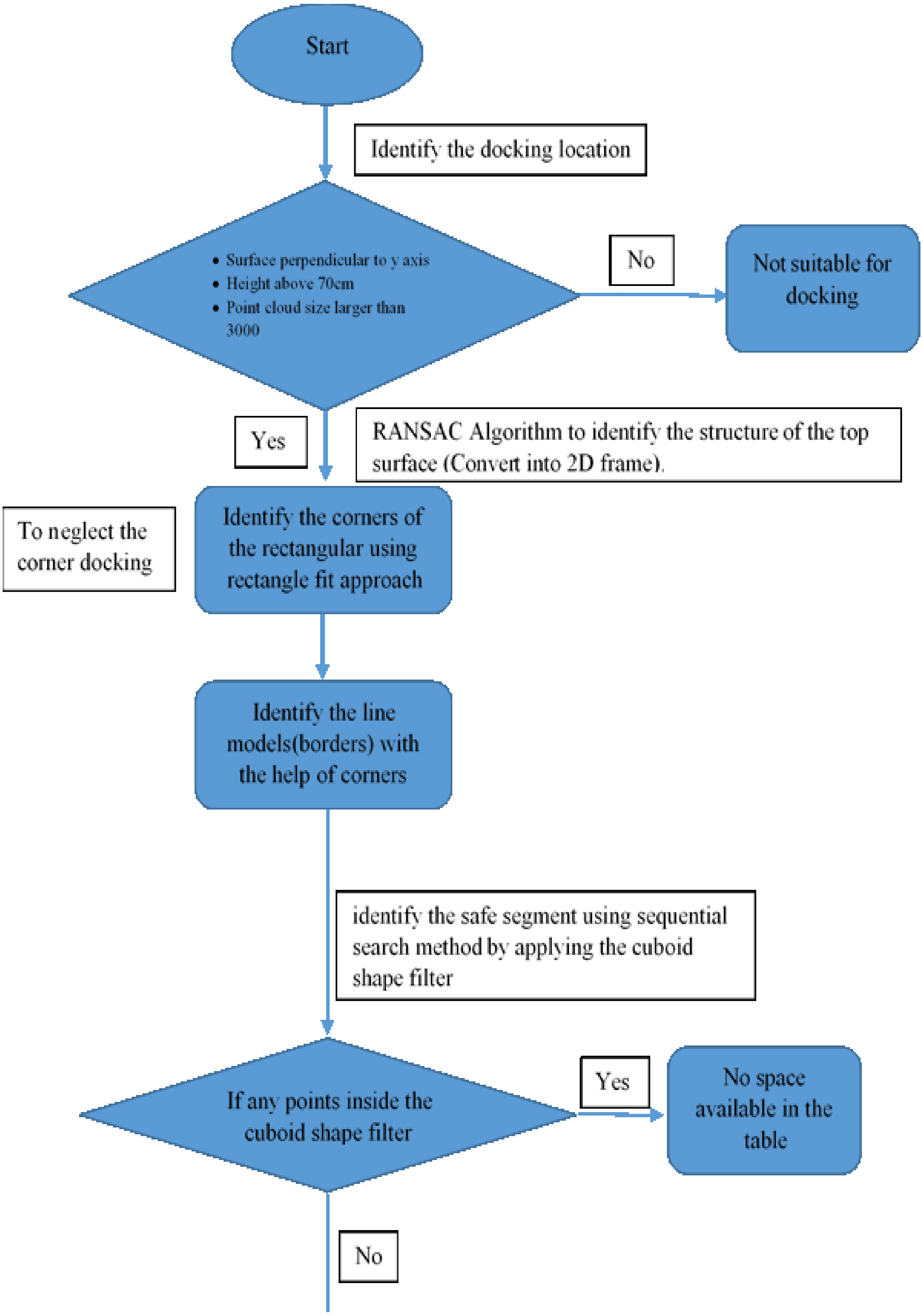
Figure 3.2 shows the design which was used in the human study process. In this figure, all parameters related to the study are marked and these were measured for each subject in the human study. Figure 3.3 shows the actual image of human study. In this study, it was only considered whether the wheelchair is near the table and without an object/obstacle affecting the scenario. This study only focuses on the docking behavior of the wheelchair. This was conducted in a space with a length of $3\ m$ and a width of $3.5\ m$.

Initially, the height and height above the hip for every subject were measured by using a measurement tool with the least amount of $0.1\ cm$. Then each human subject was allowed to perform the trial run for docking the wheelchair and then instructed to follow every scenario such as eating, writing, reading, and using a laptop on the particular table. During this, they have to adjust the wheelchair manually and dock with a table for each scenario. According to this procedure, the measurements were recorded as the data, and some questions regarding the comfort while performing the scenario were also asked from the subjects related to this study. The data collected for this research study was analyzed using the SPSS statistical software. The tests used for analysis were descriptive statistics, Analysis of Variance (ANOVA), correlation, and regression analyses. These analyses were performed at a 5% level of significance.

According to the human study performed, the distance between the wheelchair backend and table was varied for the different purposes to dock with the table such as reading, writing, eating, and using a laptop. But the wheelchair seat height is not vary based on the scenario for the particular table. These values were used in the algorithm to find a comfortable docking location. These analysis results were discussed in detail in chapter 4.

3.4 Algorithm Design

This section introduces an algorithm for docking based on object detection and user preference by detecting secure docking positions and docks with a table autonomously. The below flow charts in Figure 3.4 and Figure 3.5 show the flow diagram of the free space identification and object detection algorithms.


Start
Identify the docking location
Surface perpendicular to y axis
Height above 70cm
Point cloud size larger than 3000
No
Not suitable for docking
Yes
RANSAC Algorithm to identify the structure of the top surface (Convert into 2D frame).
To neglect the corner docking
Identify the corners of the rectangular using rectangle fit approach
Identify the line models(borders) with the help of corners
identify the safe segment using sequential search method by applying the cuboid shape filter
If any points inside the cuboid shape filter
Yes
No space available in the table
No

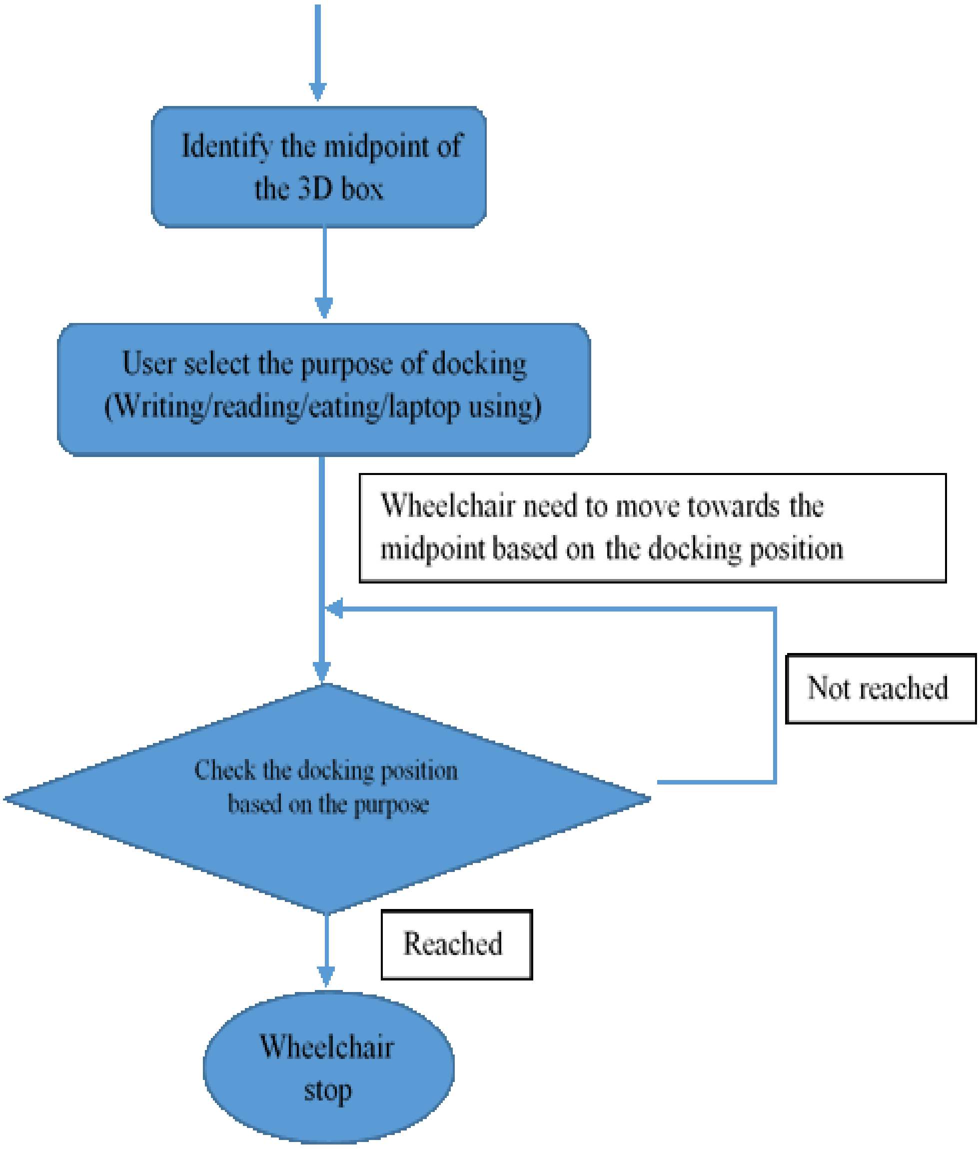


Figure 3.4 Flow diagram for docking based on the user preference with free space identification.

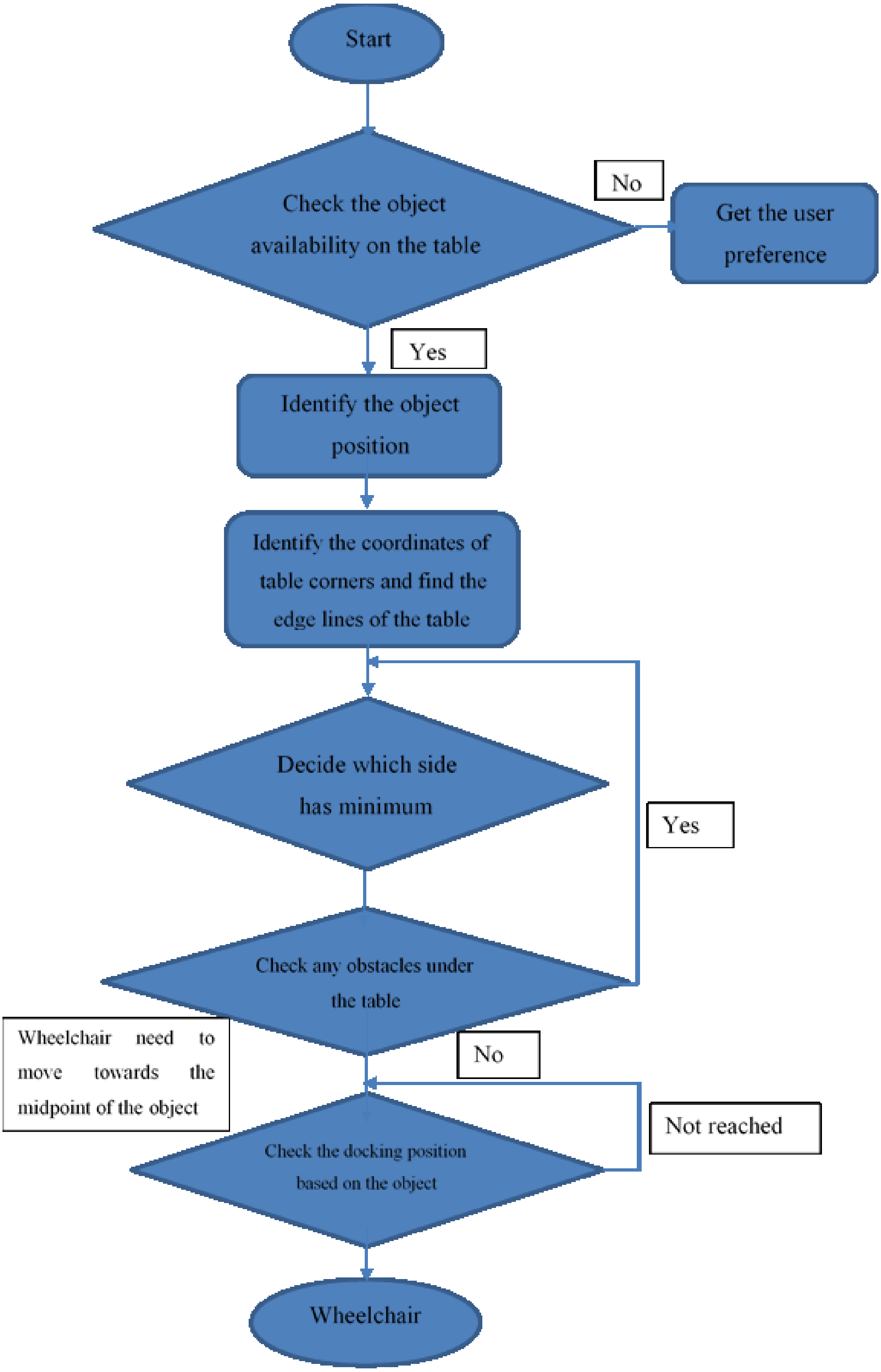


Figure 3.5 Flow diagram for docking based on the object on the table.

The algorithm first checks the docking area for whether it is suitable for docking or not.

Algorithm 1: Identification of Docking Area

Algorithm 1 Identification of Docking Area

h = table height
n =point cloud size

Require: : h , n
Ensure: : Activation

if $h > 0.7$ and $n > 3000$ **then**
    Activate RANSAC
else
    "Not suitable for docking"
end if

The RANSAC algorithm extracts the rectangular table's surface and corners. A point cloud of the scene was fed into the algorithm, with each point represented by a tuple containing its 3D location in camera coordinates $\langle x, y, z\rangle$. The Kinect sensor was attached to the wheelchair. The Y-axis was perpendicular to the ground in the camera coordinate shot, the Z-axis was back-to-front, and the X-axis was left-to-right.

Algorithm 2: Activate the RANSAC Algorithm

Algorithm 2 Activate the RANSAC Algorithm

$\{ai, aj\}, \{bi, bj\}, \{ci, cj\}, \{di, dj\}$ = table four corners

Require: : Activate
Ensure: : $\{a_i, a_j\}, \{b_i, b_j\}, \{c_i, c_j\}, \{d_i, d_j\}$

if activate = 1 **then**
    $RANSAC_{activate} = 1$
return $\{a_i, a_j\}, \{b_i, b_j\}, \{c_i, c_j\}, \{d_i, d_j\}$
end if

Algorithm 3: RANSAC Algorithm

Algorithm 3 RANSAC Algorithm

$3Dpoints$ = 3D points in the scene
n = number of iterations
t = threshold value
$inliers\_best$ = best inlier from 3D points
$inliers\_current$ = current inlier

Require: : n , t, $3Dpoints$, $inliers\_best$, $inliers\_current$
Ensure: : $inliers$

```
m = 1 inliers_best = , inliers_current =
for int i =0 ; i < n ; i++ do
    take four 3Dpoints to solve rotation matrix(R) and translation vector (t)
    for int j =0 ; j < 3Dpoints.size() ; j++ do
        errors(j) = 3Dpoints(j,1:3)' - (R*3Dpoints(j,4:6)' + t)
        if errors(j).norm < t then
            inliers_current (m++) = 3Dpoints(i,:)
        end if
    end for
    if inliers_best.size() < inliers_current.size() then
        inliers_best = inliers_current
    end if
end forreturn inliers_best
```

Using the corners, table edges were determined. Using two distinct points to identify the line method was used here to determine the edges. The next move was to search for the table's secure docking positions. Sequential search methods were proposed that allow the use of an aligned 3D box that was placed or slid underneath the table structure. The number of cloud points that come within this 3D box was countered and the knowledge to make safety decisions was used. The box's dimensions were determined by the Americans with Disabilities Act's table requirements. It was searched along the four edges. The 3D box was positioned under the edge positions to locate free segments that were safe for the wheelchair's diameter. The 3D box was tested iteratively in vertical strips as it slips around the four edges or the table structure's boundary. The box's midpoint served as a secure docking spot.

Algorithm 4: Docking Algorithm

Algorithm 4 Docking Algorithm

$\{a_i, a_j\}, \{b_i, b_j\}, \{c_i, cj\}, \{d_i, d_j\}$ = table four corners
$\{l_{ab}\}, \{l_{bc}\}, \{l_{cd}\}, \{l_{da}\}$ = table four edges
$d_{current}$ = current distance between wheelchair and table
$d_{reading}$ = docking distance for reading
$d_{writing}$ =docking distance for writing
d_{eating} = docking distance for eating
d_{laptop} = docking distance for using laptop
d_target = docking position based on the purpose
user_preference = reading/writing/using_laptop/eating

Require: : $\{a_i, a_j\}, \{b_i, b_j\}, \{c_i, cj\}, \{d_i, d_j\}$
Ensure: : Docked with the table

$l_{ab} = (b_j - a_j)x + (a_i - b_i)y = (b_j - a_j)a_i + (a_i - b_i)a_j$
$l_{bc} = (c_j - b_j)x + (b_i - c_i)y = (c_j - b_j)b_i + (b_i - c_i)b_j$
$l_{cd} = (d_j - c_j)x + (c_i - d_i)y = (d_j - c_j)c_i + (c_i - d_i)c_j$
$l_{da} = (a_j - d_j)x + (d_i - a_i)y = (a_j - d_j)d_i + (d_i - a_i)d_j$

function CHECKFREESPACE(line)
    freespace = Activate Cuboid Shape Filter
    return freespace
end function

function DOCK(table_edge)
    if user_prefernce = reading or OBJ_DETECTION(object) **then**
        $d_{target} = d_{reading}$
    else if user_preferne = writing or OBJ_DETECTION(object) **then**
        $d_{target} = d_{writing}$
    else if user_preferne = using_laptop or OBJ_DETECTION(object)
then
        $d_{target} = d_{usinglaptop}$
    else if user_prefernce = eating or OBJ_DETECTION(object) **then**
        $d_{target} = d_{eating}$
    end if
    return d_{target}
end function
function GET_DISTANCE(docking_edge)
    if docking_edge = l_{ab} **then**
        return $d_{current}$
    else if docking_edge = l_{bc} **then**
        return $d_{current}$
    else if docking_edge = l_{cd} **then**
        return $d_{current}$
    else if docking_edge = l_{da} **then**
        return $d_{current}$ 3

```
        end if
    end function
    while get_distance(docking_edge) > d_target do
        Move
    end while
```

The below algorithm was designed to identify the object on the table. Three objects were identified by this algorithm based on the height and height-width ratio of the bounding box of the object.

Algorithm 5: Object Detection Algorithm

Algorithm 5 Object Detection

h_{object} = Contour box height
$ratio$ = ratio between width and height of contour box

Require: : h , $ratio$
Ensure: : Object_type

```
function OBJ_DETECTION(object)
    if h < 50  then
        return Book
    else if ( thenh > 50 and ratio > 2.5)
        return Bowl
    else if ( thenh > 50 and ratio < 2.5)
        return Laptop
    end if
end function
```

The above algorithm was applied to the simulation design of the wheelchair in the Gazebo environment.

3.5 Simulation Design

Before going into the details of how ROS was implemented in our simulation system, this section going to review the robot kinematic model, the Unified Robot Description Format (URDF), the transformation (TF) System, navigation, and Simultaneous Localization and Mapping (SLAM).

3.5.1 Overview of robot kinematic model

The steering mechanism influences the robot's mechanical design, therefore the kinematics model of the robot changes. In ROS, we may work with a variety of robot kinematic models to execute our algorithm. Variable motion, skid, and Ackerman(carlike) steering methods are the most well-known, but we may also create our kinematic model. There is a gazebo plugin that performs differential drive steering and includes the code in our robot description code (URDF) to create simulated autonomous systems in ROS and gazebo. This is also true for LIDAR and camera plugins. When the driving control receives a command velocity id, it calculates the front and back wheel joint stable transforms and generates the required energy to push and rotate the robot. Odometry calculations are also obtained, and the robot must return to its starting location to do so. A robot state publisher is in charge of the robot's static frame transformation. The next part gives a quick overview of how to make a robot model (robot description).

3.5.2 Description About Unified Robot Description Format (URDF)

To construct a wheelchair model to use in the simulation environment, the CAD design of the wheelchair (Figure 3.6) was designed using Solid Works software. The CAD design was then converted to URDF format (Figure 3.7). The URDF code of the autonomous wheelchair is attached in the Appendix section.

Figure 3.6 Solid work design of the wheelchair

Depending on the robot type, the Unified Robot Description Format (URDF) includes different information in the tag. In general, a sensor, such as a camera or a beam sensor, is depicted as a sensor/proposal. The Kino dynamic properties of a robot platform are included in the link. Transmissions serve as a link between the mechanical connections of joints and the actuators that control them. Joint describes the joint element's Kino dynamic nature. Model_state shows the state of the robot construction at a given point in time, while Gazebo shows simulation characteristics like damping and friction qualities. A robot structure's kinematic and dynamic characteristics are depicted in this model.

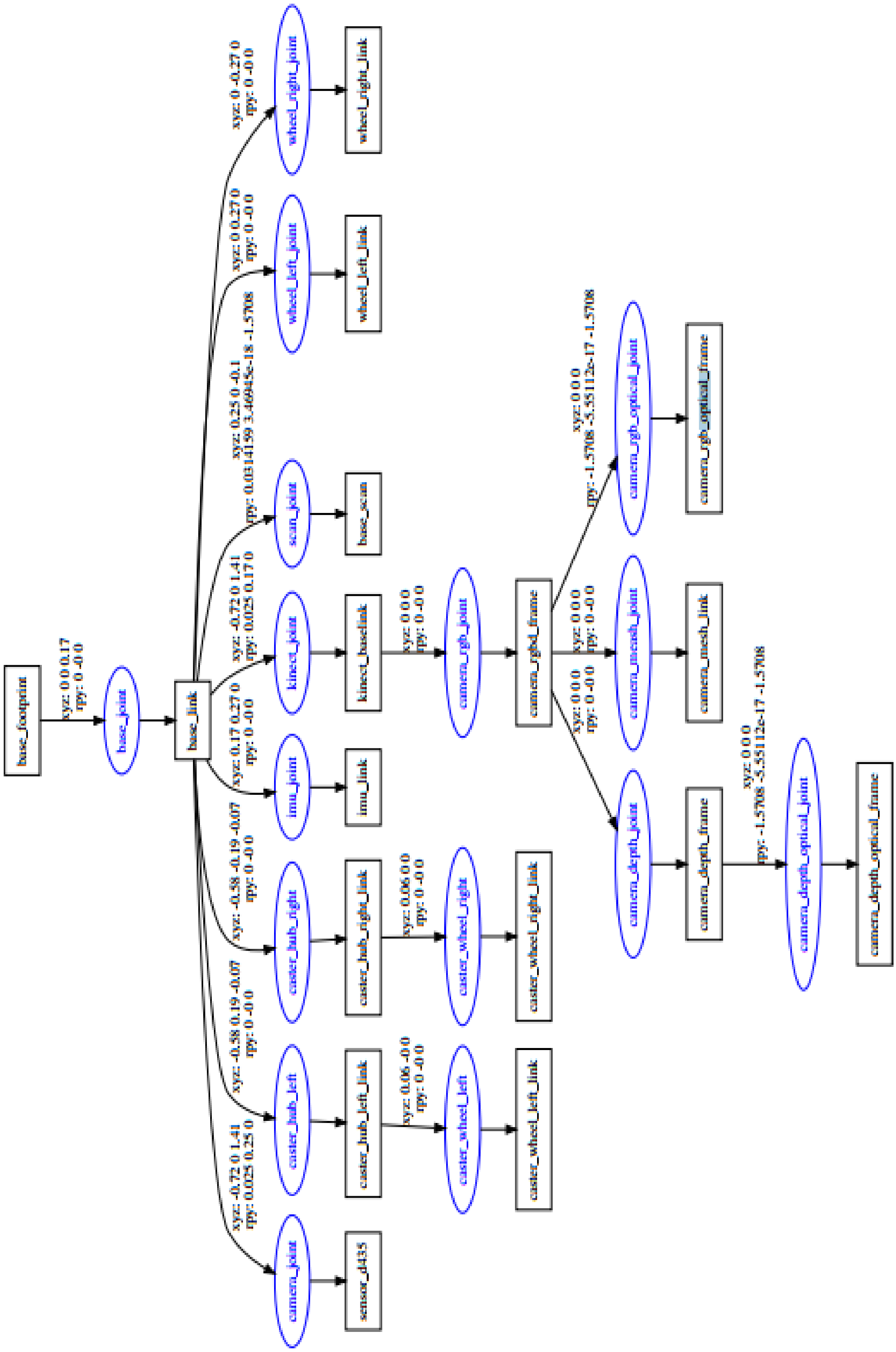


Figure 3.7 A diagram of URDF of intelligent wheelchair

3.5.2.1 Description of the link

A stiff body with inertia, visual, and collision tags is represented by the link element. In a URDF file, a link is defined by a particular tag. Various tags and their meanings as well as the nomenclature for the connection are going to be discussed here based on Figure 3.8. The link element's visual, collusion, inertia, geometry, and texture properties are represented by other tags. The start tag is only utilized by the Gazebo simulator for it to comprehend the inertia of the specified connection. A $3x3$ rotating matrix in inertial coordinates can be used to describe inertia. Based on the symmetrical characteristic of the momentum matrix, it may be described using just the six properties ixx, ixy, ixz, iyy, iyz, and izz. In a graphical interface, the visual tag can be used to display the part's connection geometry.

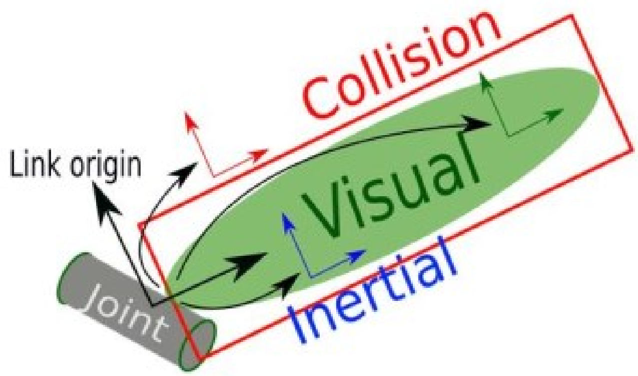


Figure 3.8 The connection between frames in the link components is depicted in this diagram.

The collusion tag, which is the same as the visual tag, denotes the area of the connection that is unaffected by other external or robot linkages. The origin tag, which contains characteristics XYZ specifying offset from x, y, z, and rpy, is the name of all the momentum, graphical, and collision tags. The matching object's structure is represented by the <geometry> tag. It can have the form of a box, cylinder, sphere, or mesh. Meshes may be created with CAD applications like SolidWorks, MeshLab, blander, or COLLADA, among others. SDF files are supported, however, dae files are preferred due to their color versatility.

3.5.2.2 Joints Description

The kinematics and dynamics of the joint, as well as the joint's safety restrictions, are represented by the joint element.

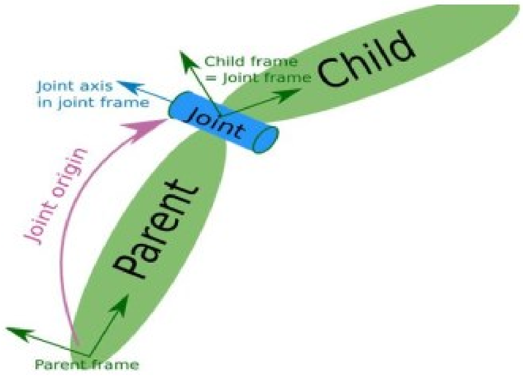


Figure 3.9 The connection between frames in a joint component is depicted in this diagram.

Dual characters <**name**> (needed) are used in the joint component to represent a new nomenclature of the joint. <**type**> is a required tag that specifies the type of joint, which can be any of the following: The first rotates at a fixed angle around an axis, the second rotates indefinitely around an axis, and the third slides along certain axes. Each of the coordinate frame's six axes may be moved using a floating joint. Finally, planar allows movement in the plane opposing the hub. The component of a joint is a crucial aspect of the URDF since it specifies the connection between two links. This tag includes a variety of components that are set at the child link's origin and specify the transition between the child and parent links. To comprehend the parent-child connection and a connection between frames, look at the diagram above (Figure 3.9). The XYZ and rpy attributes must be supplied to establish offsets, both linear and angular. The link property of the parent and child tags determines which is the parent and which is the corresponding, a child link for the robot tree-like structure (discretionary: defaults to (1,0,0)) axis. It describes the rotation link concerning another link on the robot when it is placed inside the joint tag. The dynamics, mimic, calibration, and safety_controller tags can also be used to make use of the many attributes they provide.

3.5.3 Overview of the Transformation System

Keep track of coordinate frames throughout time using this tool. Allow users to convert points, vectors, and other objects between coordinate frames by making it easier to connect coordinate frames in a time-based tree structure. Implemented as a publisher/subscriber model at the required time on the topics /tf and /tf static shown in Figure 3.10. To tune in to all broadcasted transformations, TF listeners utilize a cushion. Look for clear modifications in the change tree. Figure 3.11 represents the coordinate frame tree structure of the wheelchair according to the architecture of the transformation.

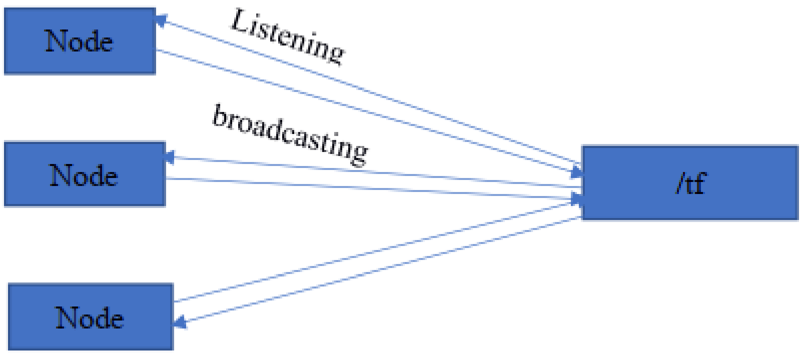


Figure 3.10 Represents TF node architecture

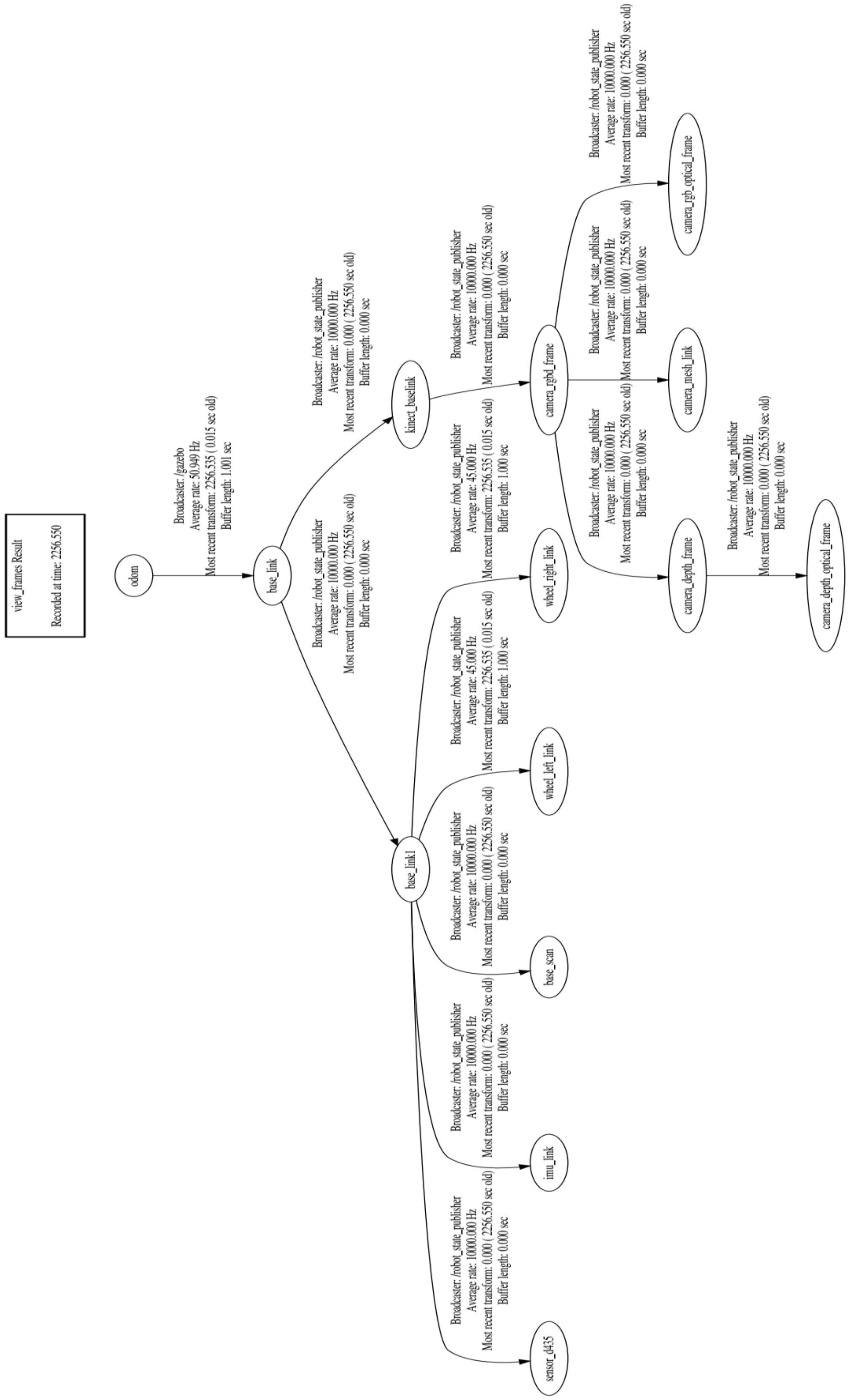


Figure 3.11 Represents the transformation system of the autonomous wheelchair

3.5.4 Description About Simultaneous Localization and Mapping (SLAM)

The robot is believed to be moving in an unfamiliar area across a trajectory defined by some random sequence variables in the probabilistic formulation of SLAM.

$$x_{1:t} = \{x_1, x_2, \dots \dots \dots \dots \dots \dots \dots, x_t\} \quad (1)$$

It acquires while moving sequences of odometry measurement

$$u_{1:t} = \{u_1, u_2, \dots \dots \dots \dots \dots \dots \dots, u_t\} \quad (2)$$

perceptions of environments

$$z_{1:t} = \{z_1, z_2, \dots \dots \dots \dots \dots \dots \dots, z_t\} \quad (3)$$

A map is a collection of geographically dense representations, well-placed markers such as grids of occupancy, surface maps, or raw sensor data, among other things. In order to solve the slam issue, high-dimensional state space must be computed. Taking into account a static environment and the Markov assumption. The Dynamic Bayesian Network (DBN) is a useful tool for explaining this topology [45]. The Bayesian network, which is a visual representation structure that displays the graph connectedness of a directed stochastic process, shows a recurring pattern in both the models of state change and observation.

In Figure 3.12, blue nodes denote variables that have been seen ($z1 : environment$ $T\ and\ u1 : T$), white color nodes denote needed variables ($x1 : T$), and m denotes the robot's route and the environment's grid map. The state transition model and the observation model both express a recurring pattern in the DBN's structure. The model of transition $P(x_t \mid x_{t-1}, u_t)$ attempts to determine the location of the robot x_t based on the previous position of the robot x_{t-1} and the odometry data u_t. The model of observation $P(z_t \mid x_t, m_t)$ attempts to derive observation z_t from an inside observation map m given the current robot location x_t. It may observe two equally essential solutions to the slam problem from a probabilistic point of view, which are online SLAM and full SLAM, which are discussed here.

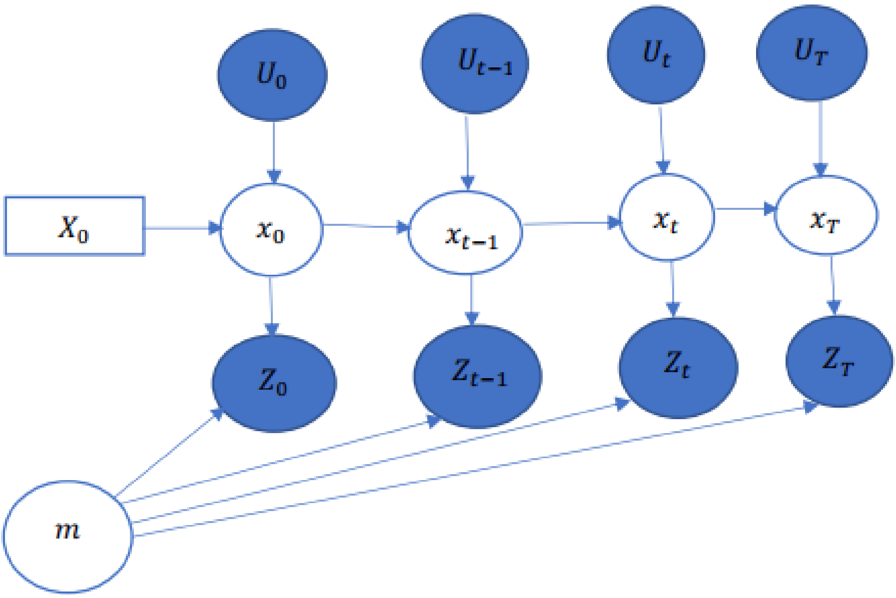


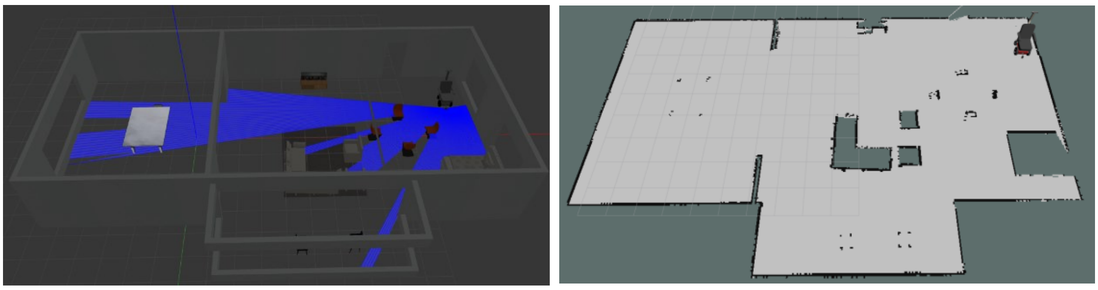
Figure 3.12 The SLAM method uses a DBN. Adapted from [40]

Online SLAM: This is the structure of the problem that manages the map and the estimation of the instantaneous pose, eliminating old measurements and managing current data as it becomes available equation 5 [46].

$$P(x_t, m \vee z_{1:t}, u_{1:t}) = \int \int \dots \int p(x_{1:t}, m \vee z_{1:t}, u_{1:t}) dx_1 dx_2 \dots dx_{t-1} \quad (4)$$

Full slam: Part of assessing the complete slam problem is computing the posterior probability of the robot's trajectory $x_{1:t}$ and the map of the environment. All the data plus a starting point is given by equation 6.

$$P(x_{1:\,t}, m | z_{1:\,t}, u_{1:\,t}, x_0) \quad (5)$$



(a) (b)

Figure 3.13 (a) Simulation environment design of the research (b) Generated map of the environment

The simulation environment (Figure 3.13(a)) was designed in the Gazebo simulator. The environment map (Figure 3.13(b)) was generated using the G-mapping technique.

3.5.5 GMapping Package

The gmapping ROS module implements gmapping, a particular SLAM algorithm. This indicates that someone has implemented the gmapping algorithm within ROS so that we don't have to do it ourselves. As a result, if we utilize the ROS Navigation stack, all we need to know (and worry about) is how to configure gmapping for our wheelchair robot. The slam gmapping ROS Node in the gmapping package allows us to construct a 2D map utilizing the laser and posture data provided by our robot while traveling around an environment. This node takes data from the laser and the robot's transformations and converts it into an occupancy grid map (OGM).

The map server package is yet another option in the ROS Navigation Stack. This package contains the map saver node, which enables us to obtain map data from a ROS service and save it to a file. When you ask the map saver to save the current map, the data is stored in two files: the YAML file, which includes the map metadata and the image name, and the image itself, which contains the encoded data of the occupancy grid map.

3.5.6 Overview of the Navigation

The navigation stack's objective is to move a wheelchair robot from one location to another while keeping it safe. It employs odometry and sensors, as well as an objective posture, to provide the wheelchair robot with safe speed limit paths. The navigation stack presumes that the robot is programmed to follow a certain path. A review of this concept may be seen in the diagram below (Figure 3.14). The white segments are necessary portions that have already been completed, the gray segments are optional elements that have already been completed, and the blue segments are essential for each wheelchair robot level. The sections below contain the prerequisites of the route stack, as well as for instructions on how to meet each one. Table 3.1 contains the list of ROS packages used in this navigation.

Table 3.1 This study made use of ROS packages.

Packages	**Application**
Transformation (TF)	The connection between various coordinate frames must be maintained.
AMCL	For a robot moving in two dimensions, a probabilistic localization method is used.
MOVE_BASE	Act as moving to the final destination.
RVIZ	2D simulator
GAZEBO	Simulator for many robots in 3D

3.5.5.1 Use of navigation stack

Ascertain if the wheelchair robot has been using ROS and that the platform is broadcasting information more concerning the transformation broadcasting node frame of reference using TF to utilize the ROS navigation stack (ROS transition frame). The stack relies on sensor data to avoid obstacles in the environment, and it expects data to be delivered in response to ROS signals. As a range sensor, a laser sensor was used. Furthermore, ROS movement necessitates the transmission of sensor data via TF and ROS messages to assess the wheelchair robot's movement.

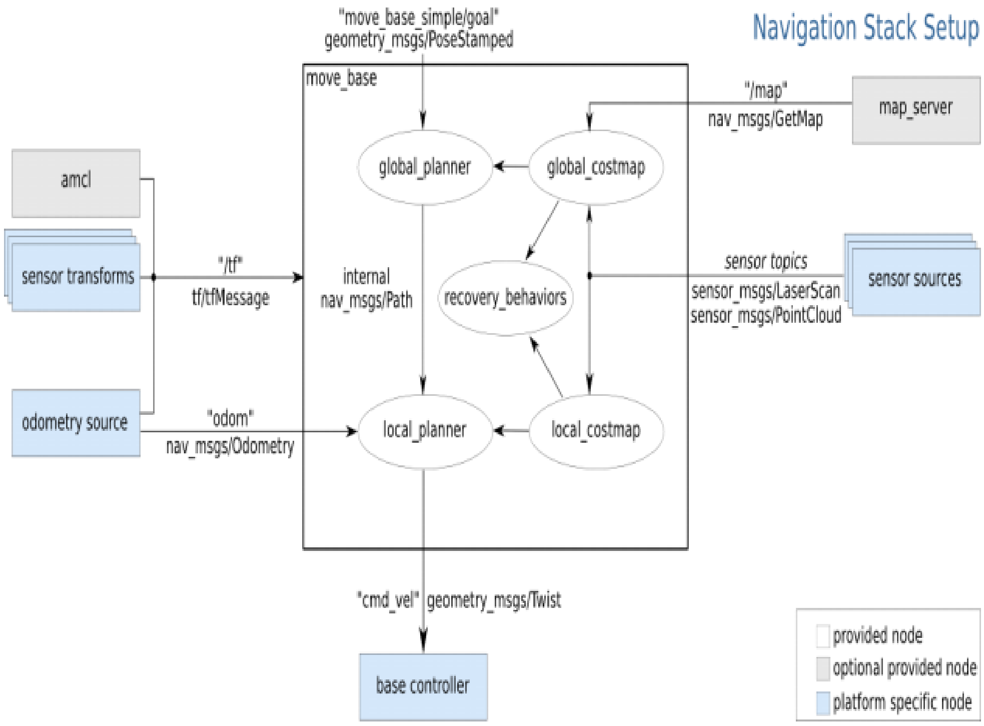


Figure 3.14 The ROS navigation stack's general architecture. Adapted from [40]

In ROS navigation, velocity commands are sent via geometric and twist signals that are converted from the platform's base frame and sent via the "cmd_vel" topic. This implies that there must be an odometry node and other nodes subscribing to it that accept messages of angular and linear velocity and convert them to motor commands for transmission to the mobile base.

3.5.5.2 Cost map

Inside a grid map, are a collection of values (matrix) that depict a region that is beneficial to the wheelchair user. For successful navigation, it analyzes and uses the surrounding occupied and empty cells, as well as the specified wheelchair robot inflating radius. The cost map is used to avoid barriers based on the parameter estimation provided to it. The cost map 2D package in ROS manages two different types of cost maps. The global cost map collects occupied and empty cells to create the robot's global plane, whereas the local cost map is used in support of the local planner. The global planner simply designs away from signaling A to B using the current guide and the robot's footprint. It is unconcerned with how the robot will move to achieve this aim. The local planner uses a robot's kinematic model to find the most stable set of directions for creating a path worldwide. Below Table 3.2 shows the list of packages required for the navigation stack.

Table 3.2 Packages required for the navigation stack

Packages	**Application**
Map server	Map database of ROS
AMCL	Provides a robot platform localization system.
MOVE_BASE	Act as moving to the final destination.
Global_planner	Implement a navigational quick global planner.
Local_planner	For local robot navigation, use trajectory rollout and local planner.

3.5.7 Path Planning

Moving from one location to another is a simple activity for humans. In a single second, one selects how to move. Such a simple and fundamental activity poses a significant difficulty to a robot. Path planning is a crucial topic in autonomous robotics. The common task is to determine a safe path for a robot, whether it's a vacuum cleaning robot, a robotic arm, or a magically flying item, from a starting point to a destination point. The challenge is to find a path from a starting point to a destination point. Depending on the environment model, the type of robots, the nature of the application, and so on, this topic has been handled in a variety of ways in the literature. Because the quality of the produced path has a large impact on the robotic application, safe and successful mobile robot navigation necessitates an efficient path planning algorithm. Typically, the primary goal of the navigation process is to minimize the traveled distance because it affects other metrics such as processing time and energy usage. There are two types of path planning: global path planning and local path planning shown in Figure 3.15.

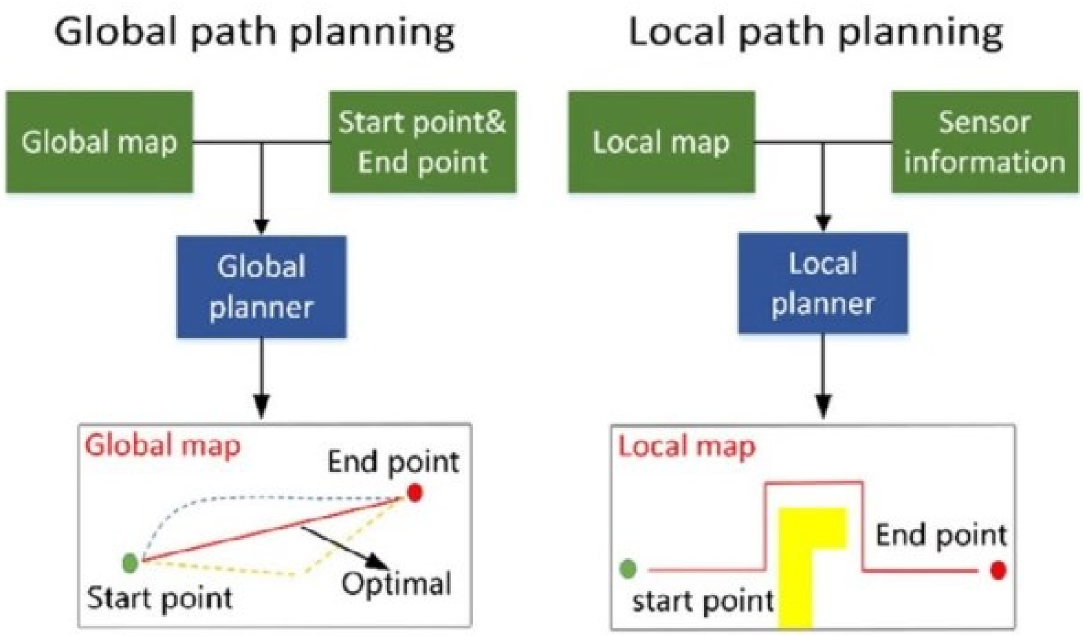


Figure 3.15 Global path planning and local path planning

3.5.8 Adaptive Monte Carlo Localization (ACML) Algorithm

This is a probabilistic localization system for a navigation robot traveling in a two-dimensional plane. This technique employs a particle filter to compare the posture of a robot platform to a known map. The AMCL node is primarily concerned with laser scanners and laser scanner-based maps. The "rviz" interface may be used to set the initial position. Then, in accordance with the plan, ROS begins sending velocity orders to two motors. There are two types of paths: "Global Path" and "Local Path." In this stage, a global path is created to establish a path from the starting point to the specified target.

3.5.9 Docking

The above section explained how wheelchair navigation was implemented. After reaching the navigation goal, the wheelchair should identify the object in the table before docking. It will dock with the table based on the object identification and free space availability according to the algorithm discussed in section 3.4. If no object is on the table, it will dock based on the user-selected preference. Figure 3.16 shows the wheelchair position after the navigation goal. The results of the object identification and docking and docking based on the user preference are discussed in detail in chapter 4.

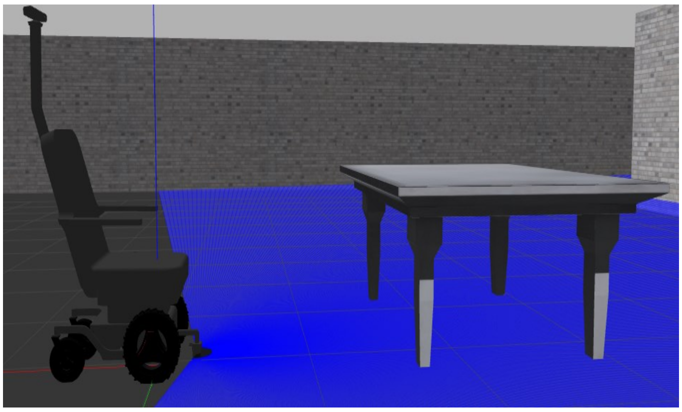


Figure 3.16 Free space identification by wheelchair after navigating to the table.

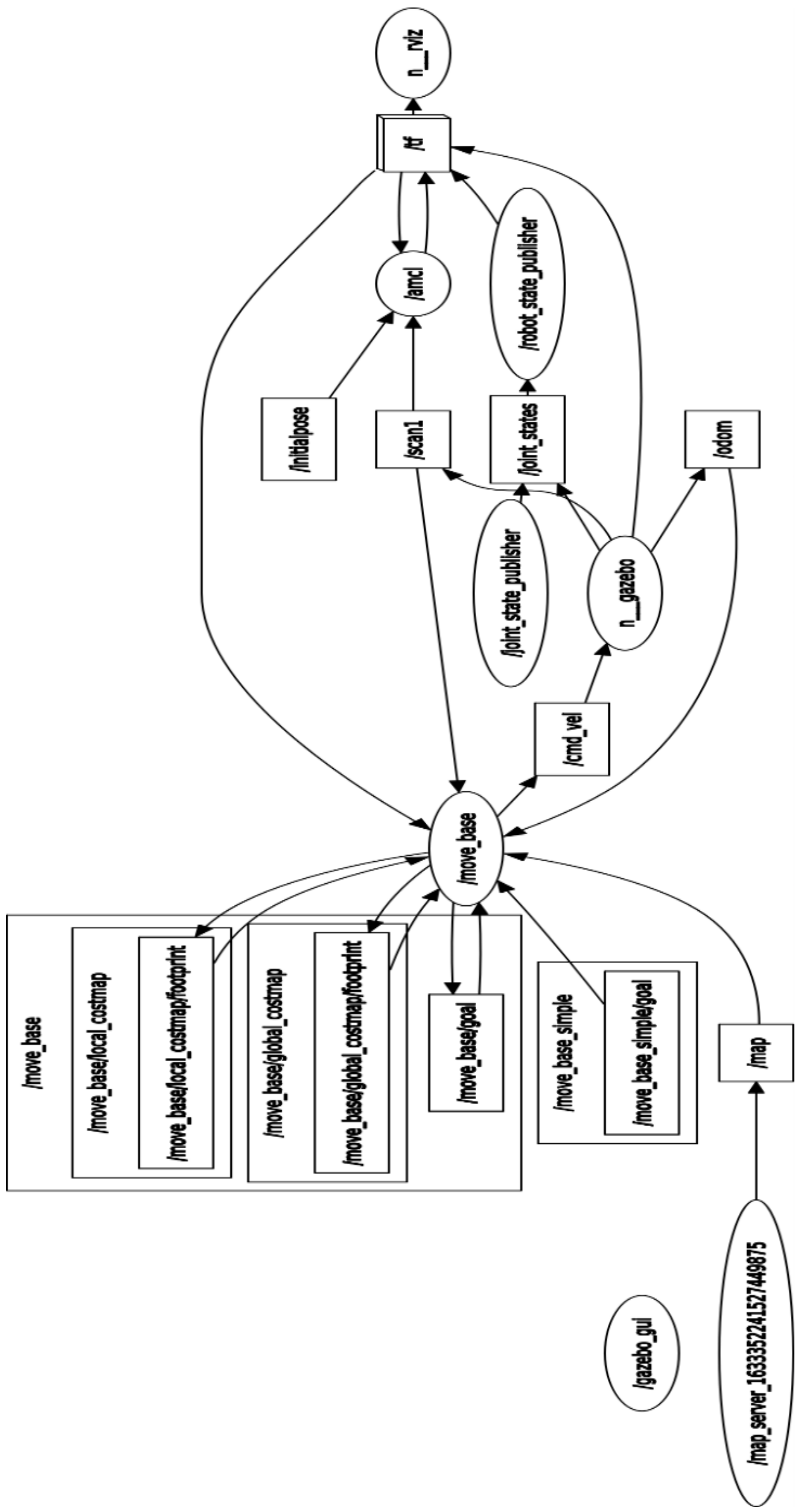


Figure 3.17 The node of the ROS control system is represented by an RQT graph.

Figure 3.17 depicts all of the nodes in the system, as well as the subjects that have been registered for communication with the Gazebo simulator.

3.6 Hardware Design

The results of the research can be implemented in a jazzy air wheelchair2. The simulation was designed based on the available jazzy air wheelchair2 with Microsoft Kinect SDK 2.0 sensor and laser sensor. Intel Core i5 quad-core processor, 8GB of RAM, and Ubuntu 18.04 installed system can be used like in the simulation. Figure 3.18, Figure 3.19, and Figure 3.20 show the input connection, output connection, wiring diagram of the above wheelchair respectively. The Kinect sensor can be connected to the wheelchair like in the below figure to replicate the simulation to the hardware implementation.

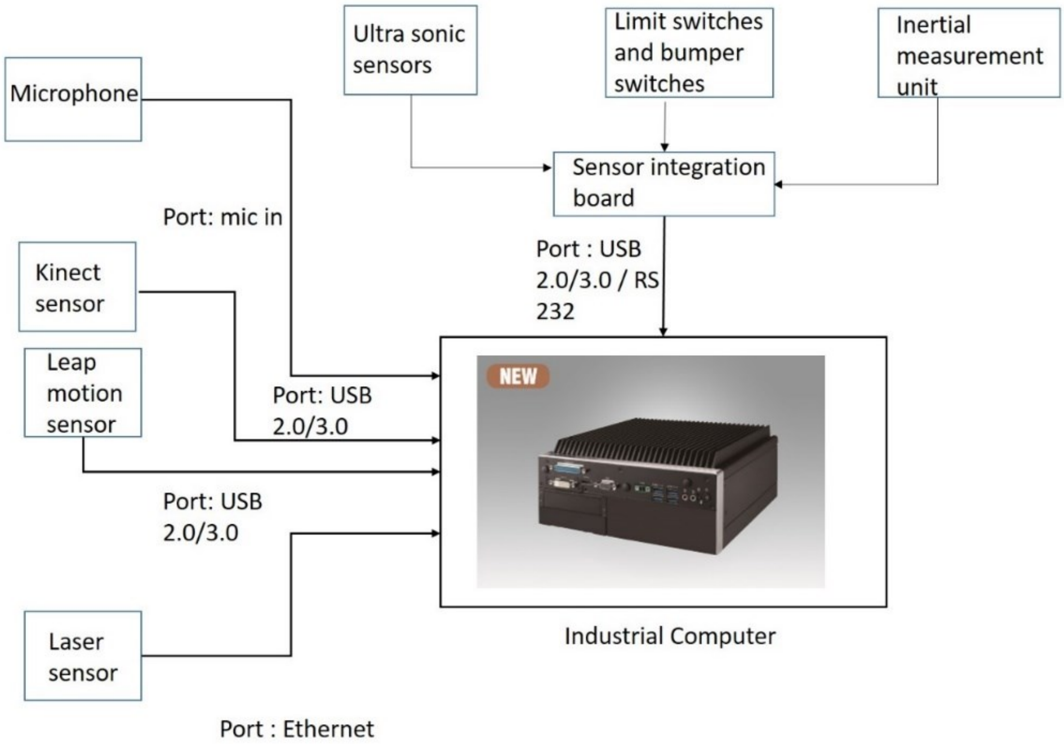


Figure 3.18 Input connections of the wheelchair

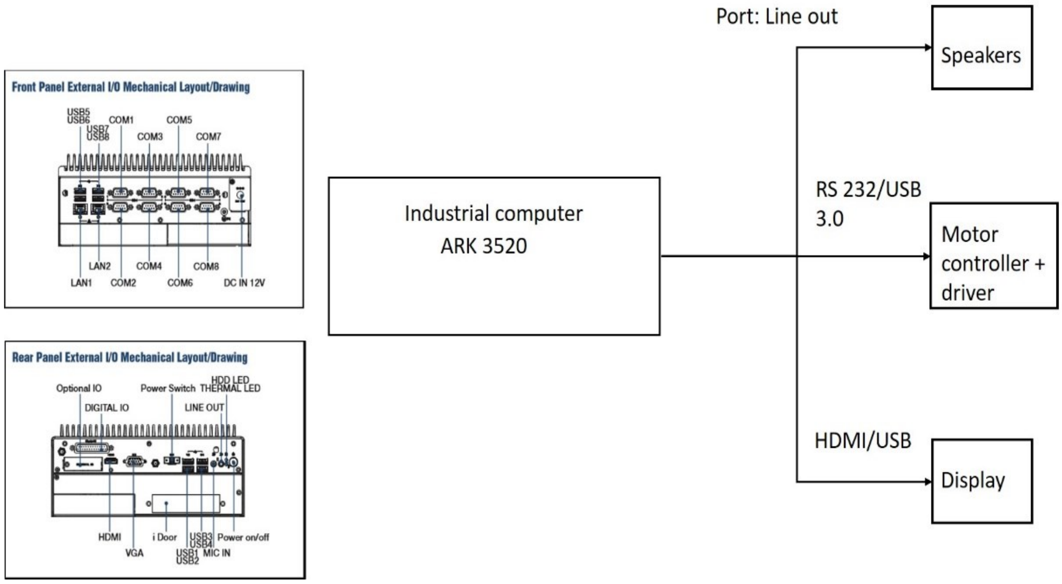


Figure 3.19 Output connections of the wheelchair

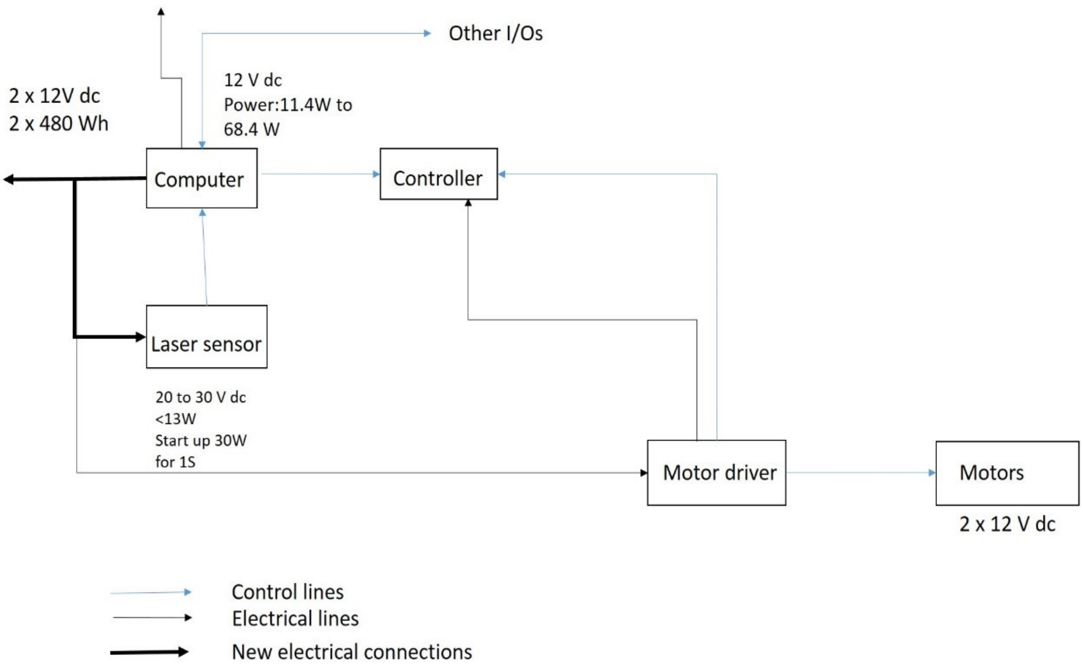


Figure 3.20 Wiring diagram of the wheelchair

Figure 3.21 was gathered from the Jazzy air wheelchair manual. It should be understood thoroughly before connecting the wiring diagram implemented above and also understood the specification mentioned in Table 3.3.

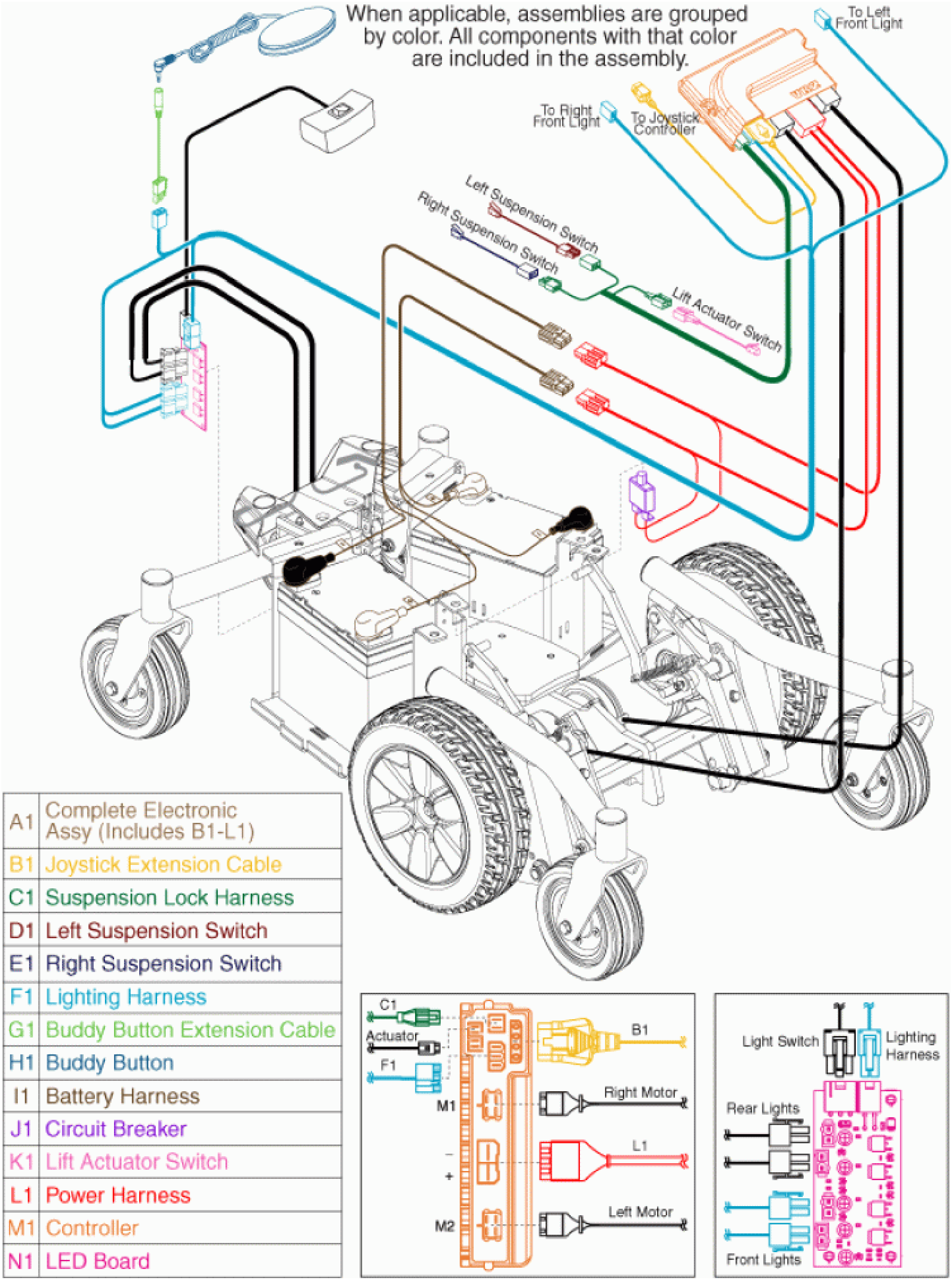


Figure 3.21 Actual wiring diagram of jazzy air wheelchair

Table 3.3 Specification of the jazzy air wheelchair

Weight	300 lbs.
Maximum Velocity	Up to 4 mph (6.44 km/h) Up to 3.5 mph (5.6 km/h) with Jazzy Air feature elevated
Clearance of ground	2.42 in. (61.5 mm) in the lowest point of the footplate bracket 1 in. (25.4 mm) at a motor mount
Radius of rotation	22.5 in. (571.5 mm)
Full Size	Length: 43.2 in. (1100 mm) with rigging in the front 38.6 in. (980 mm) without rigging in the front Width: 25.5 in. (650 mm)
Wheels on the Road	10 in. (254 mm), flat-free
Wheels with casters	6 in. (152.4 mm), flat-free
Suspension	Active-Trac® with Mid-Wheel 6 Technology
Range	Up to 18.4 miles (29.64 km) with 40 Ah batteries Up to 15.3 miles (24.6 km) with U-1 batteries
Weights of Components	Complete Assembly (without batteries): 181 lbs. (82 kg) Batteries: U-1: 24.5 lbs. (11.1 kg) each; 40 Ah: 32.2 lbs. (14.6 kg) each
Drivetrain	Two motors, mid-wheel-6 drive
Brakes	Electronic regenerative disc park brake with "Intelligent Braking"
Batteries	Two 12 volt, deep-cycle U-1; 40 Ah
Charger description	3.5 amp, off-board
Operational Force on the Move	< 60 N
Operational Force (Drive Control)	< 5 N

CHAPTER 4 RESULTS AND DISCUSSION

This chapter is going to discuss the results of human study, simulation results, and user survey results. The first section contains the analyzed results of the human study. The next section is going to discuss the results gained from the simulation such as navigation to the table, object detection, and free space identification. The last section contains the validation part of the research and the results obtained from the user survey.

4.1 Results of Human study

In the human study, the following analyses were done to identify the variations and relationships associated with the variables. The analyses are

- Analyzing the suitable distance between the wheelchair back end and the table and finding out the suitable height difference between the table surface and the wheelchair seat.
- Analyzing the relationship between human height and height above hip.
- Analyzing the relationship between the distance between the wheelchair back end and table and the height difference between the table surface and wheelchair's seat at different scenarios.
- Analyzing the distance between the wheelchair back end and table and the height difference between the table surface and wheelchair seat for different human heights and heights above hip.

4.1.1 Analyzing the Distance and Height Difference

According to the human study, data were entered and analyzed through SPSS statistical software, checking the relationship between the parameters and deciding the parameters that were affecting the docking process of a wheelchair. Based on the study for wheelchair docking to the table, the distance between the wheelchair's back end and the table decides the stopping position of a wheelchair from the table. The height difference between the table surface and the wheelchair's seat decides the height or seat adjustment of the wheelchair. The boxplots as indicated in Figure 4.1 and Figure 4.2 below were drawn to identify the suitable distance between the wheelchair back end and table and suitable height difference between the table surface and wheelchair seat at different scenarios performed such as eating, writing, reading, and using a laptop.

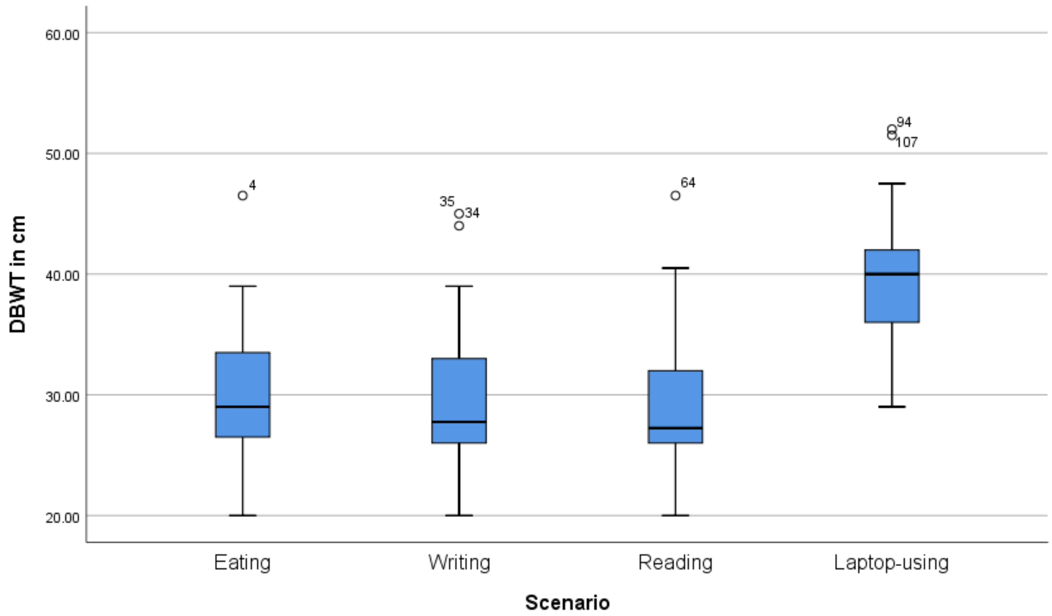


Figure 4.1 shows the boxplots of the distance between the wheelchair's back end and the table.

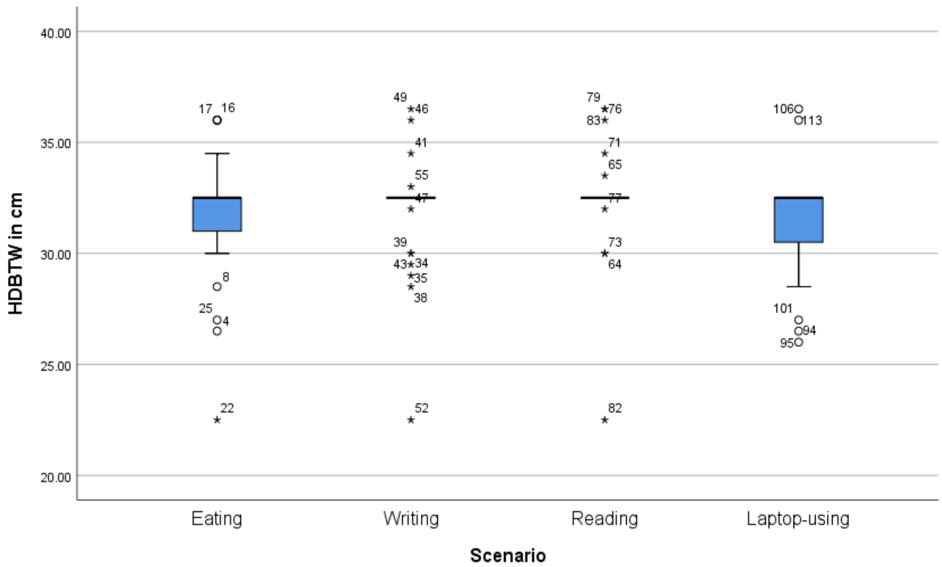


Figure 4.2 shows the boxplots of the height difference between the table surface and wheelchair seat. The box plots have the usual notation; box: inter quartiles, horizontal line, median, whiskers: minimum and maximum, and star sign: outliers.

According to the boxplot, the optimized distance between wheelchair back end and table were identified for eating, writing, reading and using a laptop as 29 cm, 27.75 cm, 27.25 cm and 40 cm respectively. The optimized height difference between table surface and wheelchair seat for all scenarios was obtained as the same value as 32.25 cm for a particular table (height = 81 cm). Seat height was not dependent on the scenario.

4.1.2 Relationship between Human Height and Height above the hip

The relationship between human height and height above the hip was analyzed using correlation and regression analyses. The results of these analyses implied that there is a significant linear relationship between these two measurements (P-value = 0.000) and this relationship is a strong positive linear relationship (Correlation coefficient = 0.886**).

From the regression analysis, this relationship can be modeled by the equation given below:

$$human\_height = height\_above\_hip + 66.109 \quad (6)$$

The β values for modeling this equation were significant (P-value = 0.000) and the regression analysis also implies that 78.5 % of the effect on "Human Height" was described by the independent variable "Height above hip "(R square value = 0.785).

4.1.3 Analysis of Distances and Height Differences at Different Scenarios

The distances between the wheelchair, table, and the height difference between the table surface, wheelchair seat were analyzed to find out whether there is a significant difference between those distances at different scenarios performed while sitting in the wheelchair.

The ANOVA implies that there is no significant difference in the height differences between the table surface and the wheelchair seat at different scenarios performed at the table (P-value = 0.378). But it implies that there is a significant difference in the distances between the wheelchair back end and table at a 5% level of significance (P-value = 0.000).

Therefore, the change of the scenario performed at the table while sitting in a wheelchair should be taken into consideration as there is a significant difference in the height differences between the table surface and wheelchair seat in different scenarios.

4.1.4 Analysis of Distances, Heights for Different Human Heights and Heights above Hip

In this study, it was also analyzed to find out whether there is a significant difference between the distances between the wheelchair back end, table, and the height differences between the table surface and wheelchair's seat at different scenarios for different human heights and heights above hip. This was also done for the height differences between the table surface and wheelchair seat.

The result of ANOVA for this purpose implies that there is no significant difference between the distances for different human heights. But there is a significant difference among height differences between the table surface and wheelchair seat for different heights above hip when eating (P-value = 0.020), writing (P-value =0.002), reading

(P-value =0.001), and this isn't significantly different for distances between wheelchair back end and table for different heights above hip.

4.2 Results of Simulation Design

This section is going to discuss the results obtained from the simulation design. This section is divided into three sub-sections such as results gained in the navigation of the wheelchair to the goal, results of wheelchair docking based on the user preference, and results obtained from the wheelchair docking based on the object detection on the table.

4.2.1 Navigation

4.2.1.1. Mapping

The ROS "SLAM GMapping" module is a wrapper for an open-source SLAM implementation. The ROS package includes an executable node named "SLAM GMapping" that implements the SLAM method and can generate a 2D occupancy grid map from laser scan data and mobile robot attitude. The "SLAM GMapping" node receives laser data (sensor_msgs/scan) and transmitted data and provides occupancy grid map data as output. The map may be stored by launching the "map server" node. The maps are saved in two files: a YAML file and an image file. The YAML file specifies the meta-data for the map and names the picture file, which encapsulates the occupancy data. Figure 4.3 depicts the environment map utilized for navigation.

The simulation environment was designed as a domestic environment like a bedroom with a table, bed, chairs, TV with stand, etc. The environment map was created by the wheelchair movement using teleoperation. The laser scanner determines the distance and direction of an item, whilst the encoders provide data on the robot platform's movement. Using this multisensory data, the mapping and localization operation in ROS was carried out using a SLAM algorithm with a particle filter method. The "scan" node in ROS provides its laser data. The transform tree was then built to connect the "base link" and "base scan" coordinate frames. In this study, an occupancy grid map with size $(x, y) = (19, 11)$ (i.e., 19 elements in the longitudinal direction and 11 elements in the lateral direction) was considered. In occupancy grid maps like in gmapping do not consider (x, y) coordinates, it was considered as a cell shown below Figure 4.3. 20*20 grids were generated and origin was also marked in the center point of the grid.

Figure 4.3 Origin representation in the grid map

4.2.1.2. Path Planning

For environmental modeling, a grid-based map was employed, which divides the environment into uniform cells. Each cell's value shows the likelihood that the cell is filled by the barrier. The AMCL method employed to tackle the problem was based on particle filters. The technique was used to track the position of a dynamic robot platform against a known map. Route planning was handled via the "move base" package, which is separated into two modules: global path planning and local path planning. This is the most often used approach for dealing with complicated path planning problems. The global path planner was used to find the quickest route to the destination, and the local path planner (also known as the driver) provides the actual orders to follow the global path while avoiding obstacles.

The "move base" package keeps two cost maps that describe the region in which the robot can move. The "global cost map" and "local cost map" variables are utilized by global and local planners, respectively. The cost map is used to record and preserve information regarding environmental barriers and navigation directions in an occupancy grid. The global path planner is based on the "global cost map," which is often derived from a previously constructed static map and is updated at the user-specified frequency. Before the robot begins moving, the global route planner generates a long-term plan from the present stance to the goal pose. The "move base" package creates and updates the obstacle map in real-time. The local path planner is based on the local cost map, which employs real-time sensor data and is fed by the global planner. The local route planner attempts to follow the global path as closely as feasible while taking into account the robot's kinematics and dynamics, as well as any additional obstacle information in the local cost map.

The "Navfn" module in ROS was used to find the best path given a goal by "rviz." The "Navfn" function offers quick interpolated navigation that may be used to construct plans for a move-base. This path planner uses a cost map to identify the shortest way from a start point to an endpoint in a grid. The Dijkstra algorithm was used to calculate the shortest path with the lowest cost. Following the transmission of the global plan to the controller, velocity directives were generated and transmitted to the wheelchair. The controller then attempts to drive the robot along with a piece of the global path (defined by the size of the local cost map), and this sectional navigation on the global path is repeated until the objective is reached. If we introduce additional barriers or objects into the environment, as a result, updating the cost map is required to prevent such shifting impediments. As a result, we require a Local Planner that determines the course in the surrounding vicinity while moving the platform in real-time navigation.

The wheelchair's obstacle avoidance capacity was restricted due to its speed and acceleration limits when utilized for obstacle avoidance. The Dynamic Window Approach (DWA) is a velocity-controlling local planner that determines the robot's ideal velocity for collision-free navigation to the given target. DWA planner converts a cartesian goal (x, y) into velocity vector space (v, ω) composed of linear speed v and rotational speed ω orders for the wheelchair. The ideal velocity was chosen to optimize the robot clearance, velocity, and heading closest to the target. Figure 4.4 depicts how the path was generated once a goal was assigned to the mobile base using "rviz," which automatically produces the global path within ROS while avoiding static obstructions. The minimum and maximum linear velocity of the wheelchair were set to $0.2\ m/s$ and $0.5\ m/s$. Wheelchair speed can vary within this range based on the above algorithm.

Figure 4.4 Wheelchair initial pose array indicates in green color and goal posed indicates in red arrow.

The static map's obstacles or the wall are shown in Figure 4.5 by the black limits and dots. The inflation radius for the cost map (the blue-colored layer around barriers) had been configured. The inflation radius should be set to the greatest distance from barriers that may incur a cost. Setting the inflation radius to 0.50 meters, for example, indicates that the robot will regard any pathways that are 0.50 meters or more away from barriers as having the same obstacle cost.

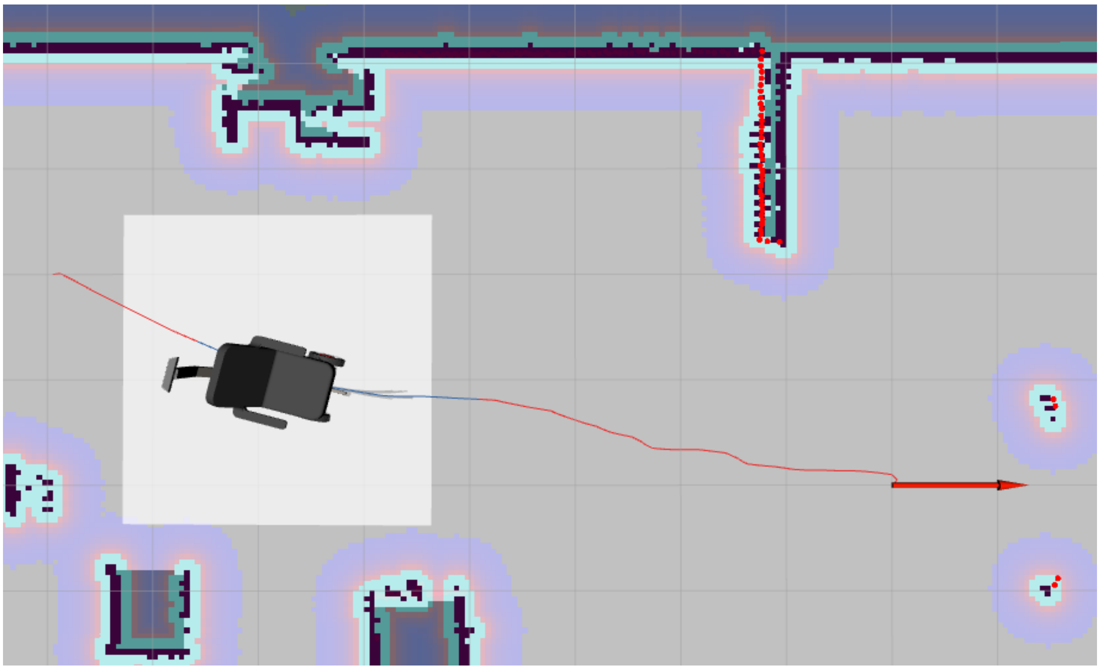


Figure 4.5 The red dot line indicates laser rays, the red arrow indicates the goal pose and the red line represents the global plan, while the blue line represents the local plan.

Wheelchair's initial location was selected randomly in the environment such as Figure 4.6(a) - $(x = 12.06, y = 0.89)$, Figure 4.6(b) - $(x = 9.58, y = 3.56)$, Figure 4.6(c) - $(x = 7.65, y = 5.02)$, Figure 4.6(d) - $(x = 3.30, y = 6.20)$ in the room. The wheelchair was moved to the goal by avoiding obstacles and using the shortest pathfinding algorithm. Wheelchair navigation was tested by changing the initial location and recording their navigation path below in Figure 4.6.

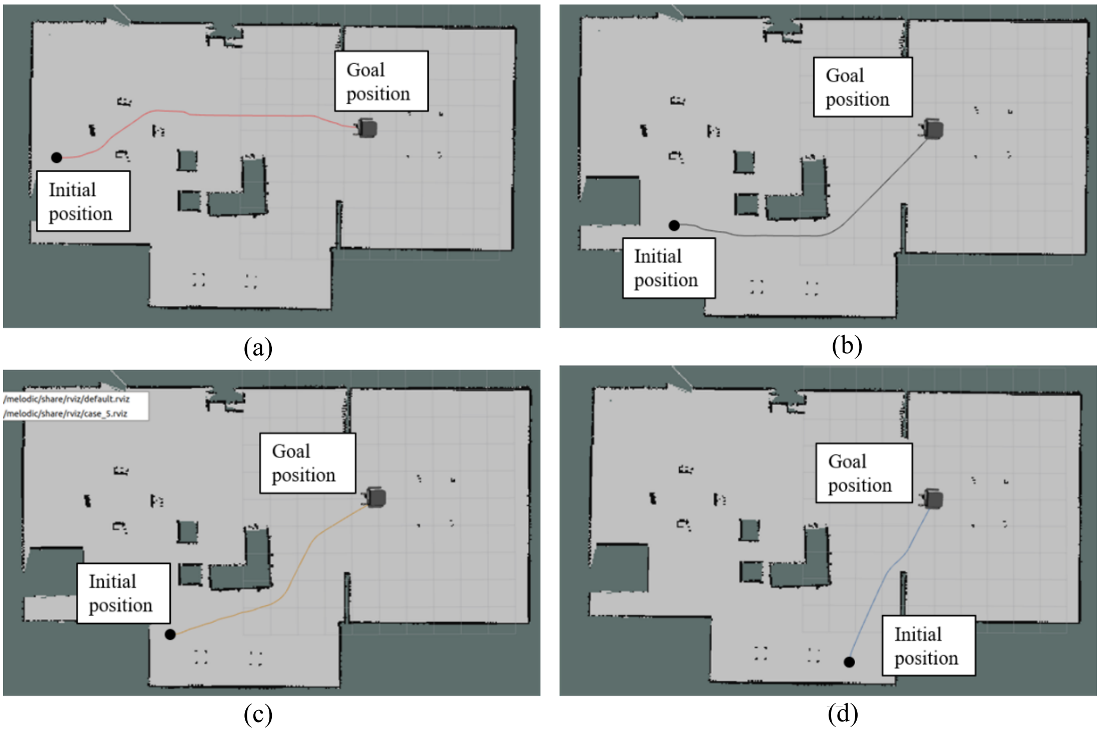


Figure 4.6 Represents the navigation path of the wheelchair in different initial locations to reach the goal.

4.2.2 Path planning for docking

After the wheelchair has reached the navigation goal position, next it will be ready to dock with the table based on the object on the table or based on the user preference. If an object is available on the table, it will identify the object position then it will find the docking position. If there is no object on the table then it will search the free space which is closer to the wheelchair. How the path planning for the wheelchair docking when there is no space in the wheelchair facing side and right side of the table or if an object is on the left side of the table, will be constructed will be explained in below Figure 4.7. Based on the free space identification algorithm/object identification algorithm, it will first decide which direction to dock. Based on the above scenario, the wheelchair first turns the left side $(theta = -1.5)$ shown in Figure 4.7(b). Then wheelchair will move to the $(x = -0.3, y = -1.5, theta = -1.8)$ position according to the navigation algorithm mentioned in Section 4.2.1.2 shown in Figure 4.7(c). The next wheelchair will move to the $(x = -2.1, y = -1.8, theta = math\,\pi/2)$ position as same as the previous navigation shown in Figure 4.7(e).

Finally, the linear velocity of $0.2\ m/s$ it will move to dock with the table based on the analyzed ergonomics data for the user preference or object identification which is shown in Figure 4.7(f). Below test cases are also designed as in the above section explained.

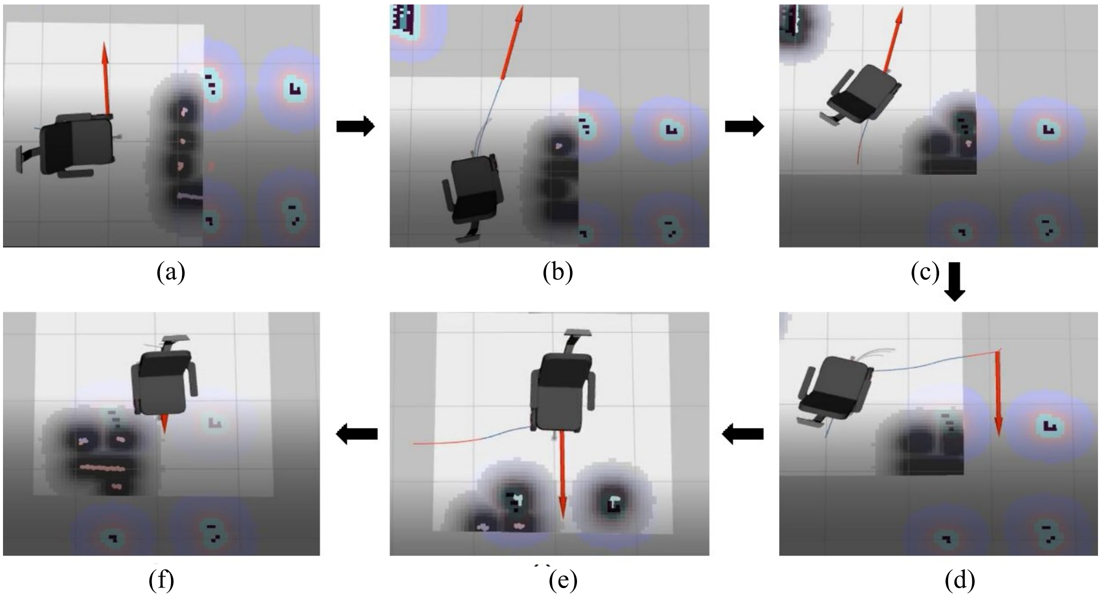
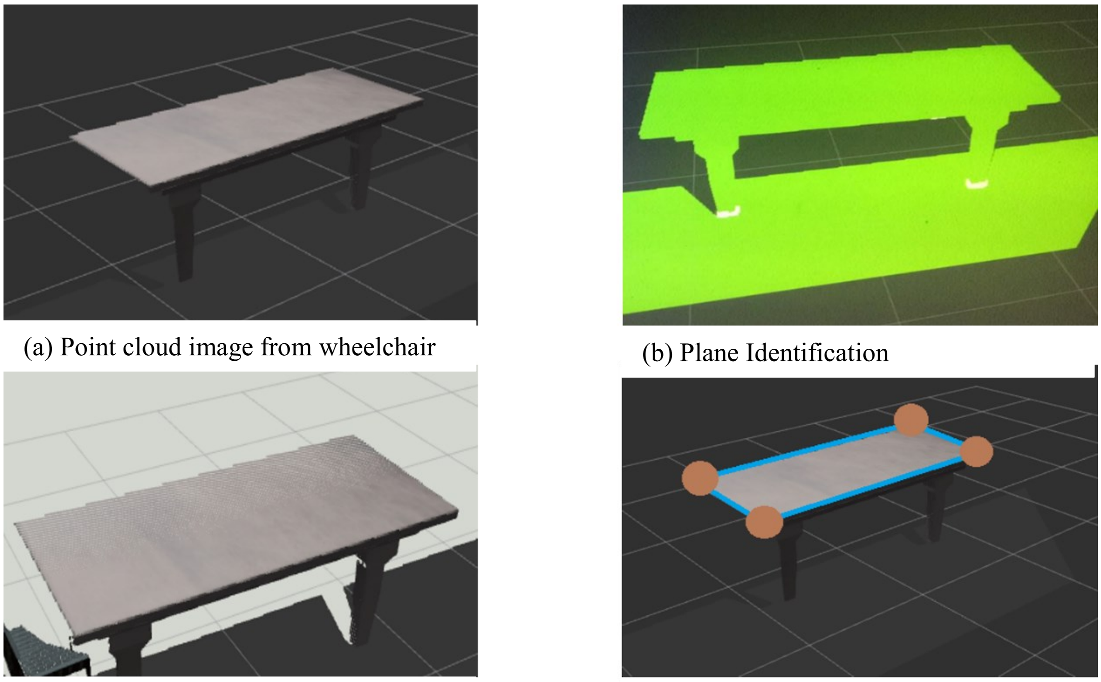


Figure 4.7 Path planning for docking to the particular test case.

4.2.3 According to user preference wheelchair docks with the table

After the wheelchair reached the table using navigation, it was stopped nearly 1.5 m away from the table before docking. The navigation goal was set as an origin point of the grid map and the simulation environment was designed according to that. The table was checked whether it had an object or not. If no object was on the table, user preference (eating, reading, writing, and using a laptop) was asked to dock with the table.



(a) Point cloud image from wheelchair
(b) Plane Identification
(c) Surface Identification
(d) Identification of corners and edges

Figure 4.8 Table surface and edge identification

Then the top surfaces were discovered using a RANSAC planar model (Figure 4.8) that is perpendicular to the Y-axis i:e the vertical axis of the Kinect sensor frame, fits with table size (height greater than 70 cm above the ground plane) and orientation constraints (point cloud size > 3000). The top surface was considered as a minimum-area bounding rectangle to use to compute the four corners using basic trigonometry. Corner points were identified in x,y coordinates. Using the two corner points, edges were derived from the equation of a line from two points. The free space under the table was detected by a cuboid shape filter that gets rotated under the table edges to find free space. If there are no points inside the cuboid shape filter it's considered free space. After that according to the user preference, the stopping position was decided based on the analyzed ergonomics data. Figure 4.9 shows the docking process of the wheelchair if there is no object on the table.

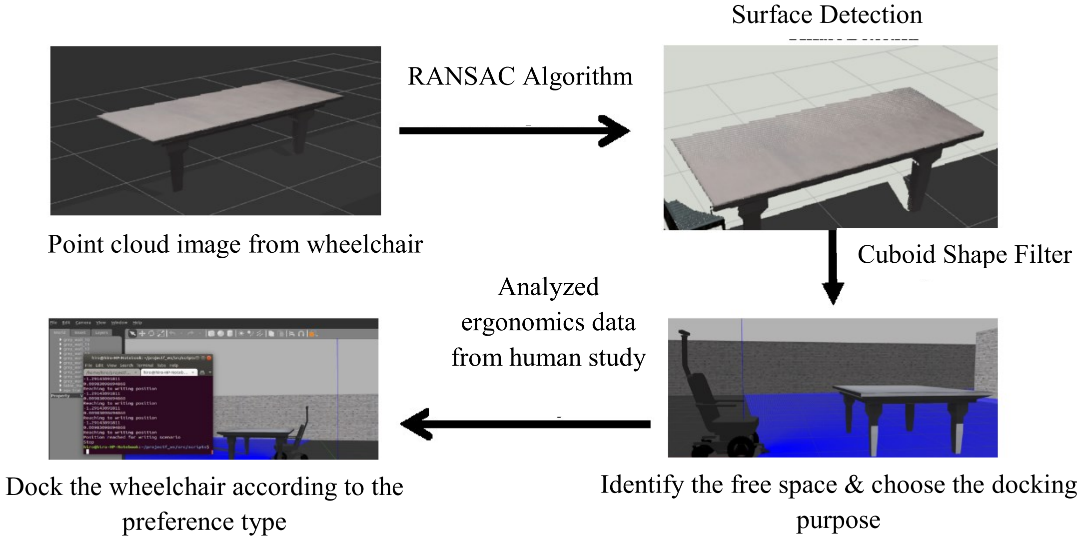
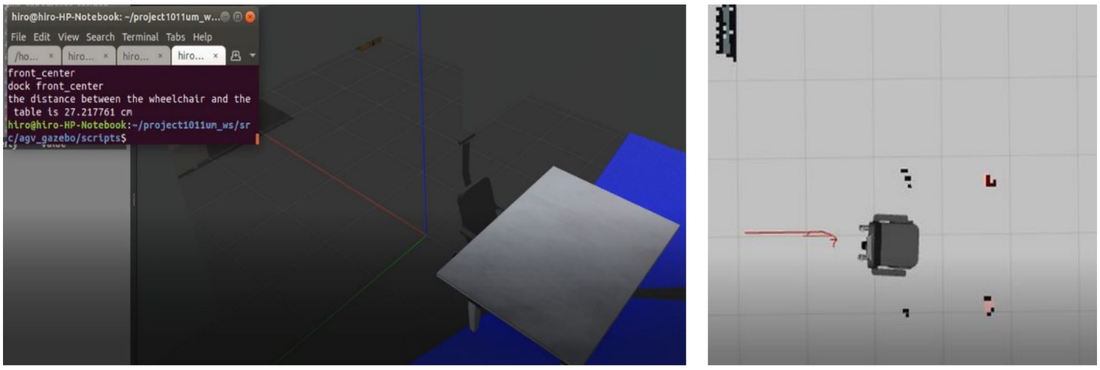


Figure 4.9 Wheelchair docking flow based on the user preference

The wheelchair docking process was checked under the below conditions without an object on the table scenario.

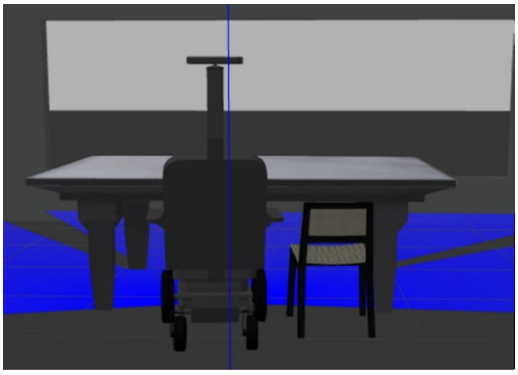
- Table with no chair
- Free space available on the facing side with chair/s.
- No free space on the facing side
- No free space on the facing side and right side
- No free space on the facing side, right side, and opposite to the facing side
- No free space in the table



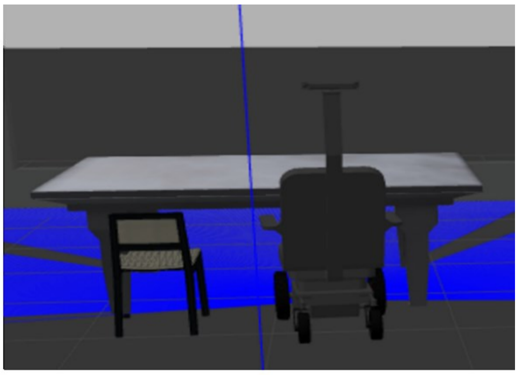
(a) (b)

Figure 4.10 Docking to the front side

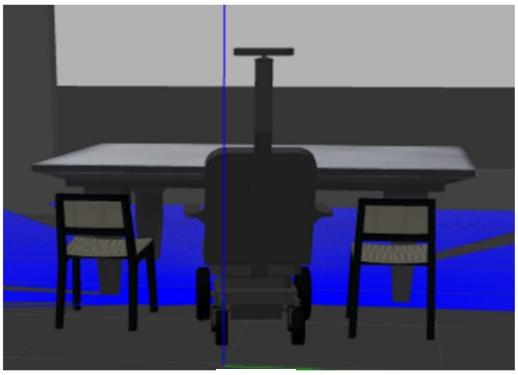
Figure 4.10 shows the wheelchair docking when there is no chair and object on the table for reading purposes. If no object is on the table, then it will dock based on the user preference such as reading, writing, eating, and using a laptop.



(a)



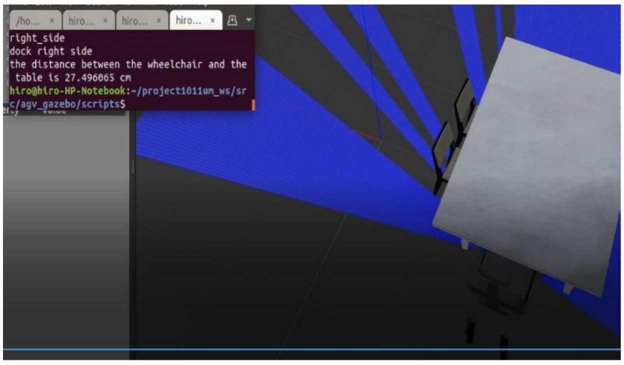
(b)



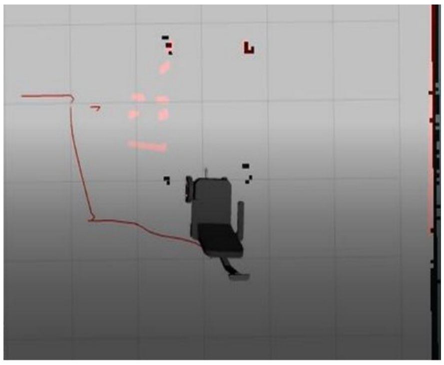
(c)

Figure 4.11 Docking to the front side with various conditions

Figure 4.11 shows the wheelchair docking when there is a chair/s and free space in right or left or center for docking. Also no object in the table for reading purposes. Chair/s can be placed anywhere on the wheelchair-facing side.



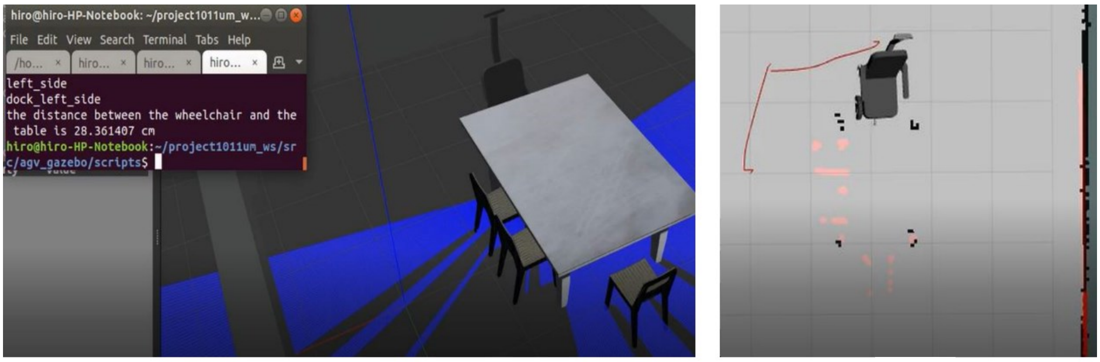
(a)



(b)

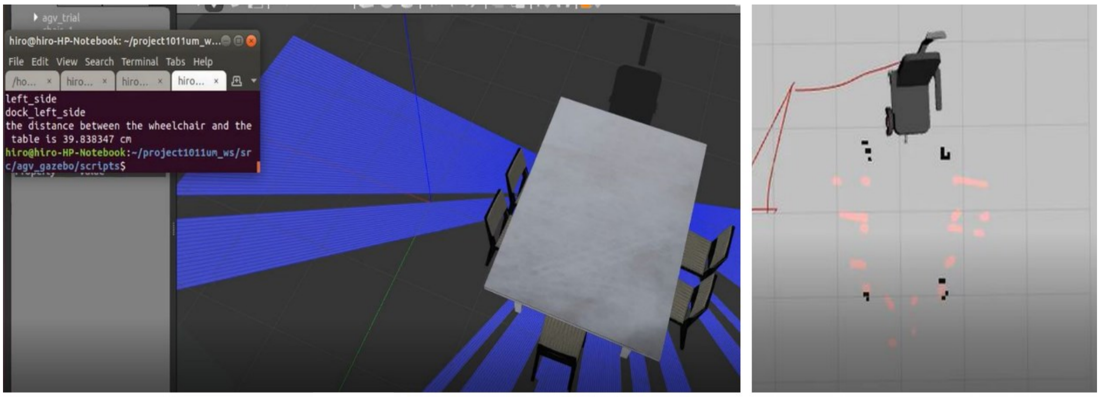
Figure 4.12 Docking to the right side of the table

Figure 4.12 shows there is no free space on the wheelchair facing side, then it will dock to either the left or right side of the table if there is no object on the table.



(a) (b)

Figure 4.13 Docking to shortest distance side



(a) (b)

Figure 4.14 Docking with three chairs on the table.

Figure 4.13 shows there are chairs on the wheelchair facing side and right side of the table, it will dock to the shortest path side to dock to the table. Figure 4.14 indicates there are chairs on any three sides of the table, then it will dock to the other side which has no chairs.

Table 4.1 Summary Results of docking based on the user preference

Test Cases	**Preference was chosen by the user**	**Analyzed Ergonomics Data (cm)**	**Docking Distance based on the simulation (cm)**
Table with no chair	Writing	27.75	27.218
Free space available on the facing side with a chair placed on the left side	Reading	27.25	26.856
Free space available on the facing side with a chair placed on the right side	Using laptop	40	39.827
Free space available on the facing side with two chairs	Eating	29	28.854
No free space on the facing side	Reading	27.25	27.496
No free space on the facing side and right side	Eating	29	28.361
No free space on the facing side and left side	Reading	27.25	27.352
No free space on the facing side, right side, and left side	Using laptop	40	39.599
No free space on the facing side, right side, and opposite to the facing side	Using laptop	40	39.838
No free space on the facing side, right side, and left side and chair placed on the left side of the opposite to the facing side	Reading	27.25	27.399
No free space on the facing side, right side, and left side and chair placed on the right side of the opposite to the facing side	Eating	29	28.965

All the possible scenarios were tested using this simulation, and the expected wheelchair behavior was successfully observed shown in Table 4.1. Distance between the table and wheelchair after the docking was measured in all the scenarios in the simulation platform.

4.2.4 According to the object on the table wheelchair docks with the table

After the wheelchair reached the table using navigation, it was stopped 1.5 m away from the table before docking. The table was checked whether it has an object or not. The assumption was made as only one object on the table. If any object was on the table, free space was checked to dock with that object. If any obstacle under the table, the wheelchair was docked to the other side which was the nearest to the object. To find the nearest position to the object, it will check all four edges to find the minimum distance edge. Only three objects were considered for object detection such as a bowl, book, and laptop. Objects were identified using the OpenCV library as shown in Figure 4.16. The detected image was then converted into HSV format. The bounding box was then drawn using the OpenCV function. Figure 4.15 shows the docking process of the wheelchair if there is an object on the table.

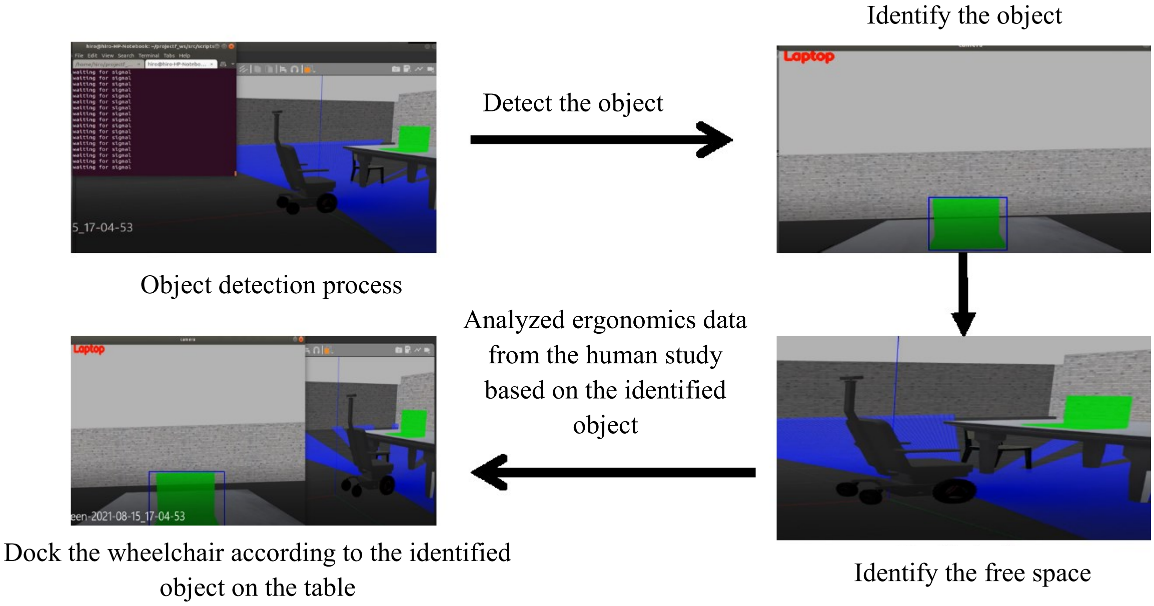
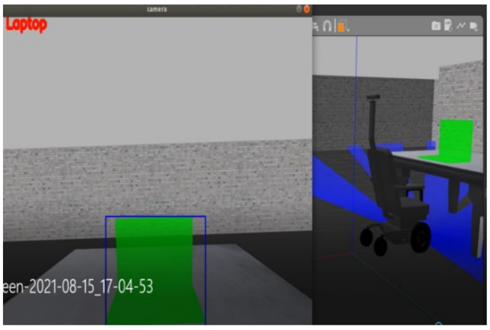
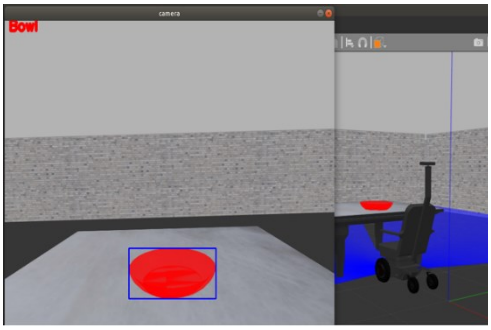


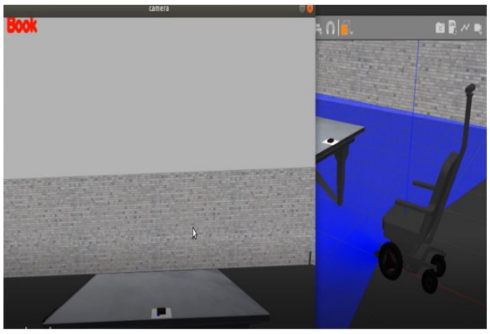
Figure 4.15 Wheelchair docking flow based on the object in the table.



(a)



(b)

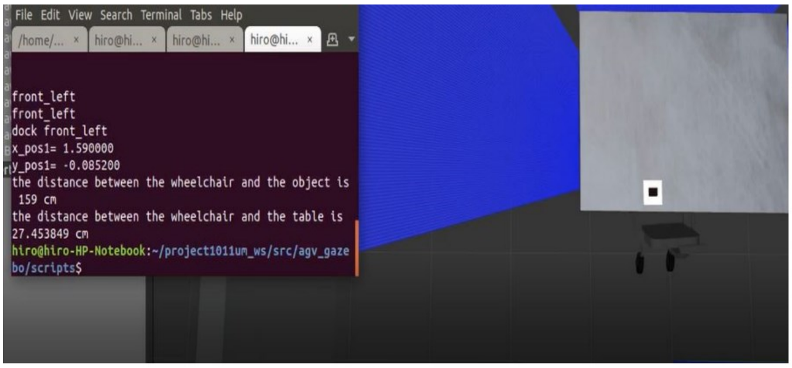


(c)

Figure 4.16 Object detection for laptop, bowl, and book

The wheelchair docking process was checked under the below conditions with an object on the table scenario.

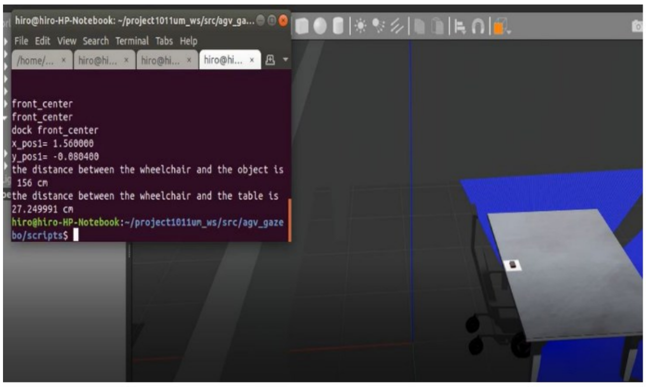
- An object placed in any location of the table (front, left, right, back)
- Object detection integrated with above section (4.2.3) test cases



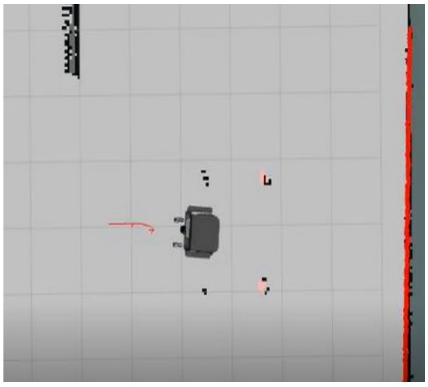
(a)

(b)

Figure 4.17 Object placed in the front left side

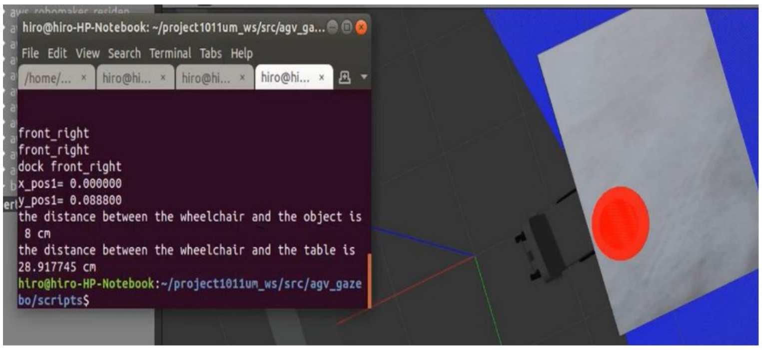


(a)

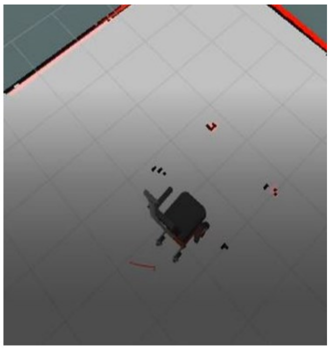


(b)

Figure 4.18 Object placed in the front center side

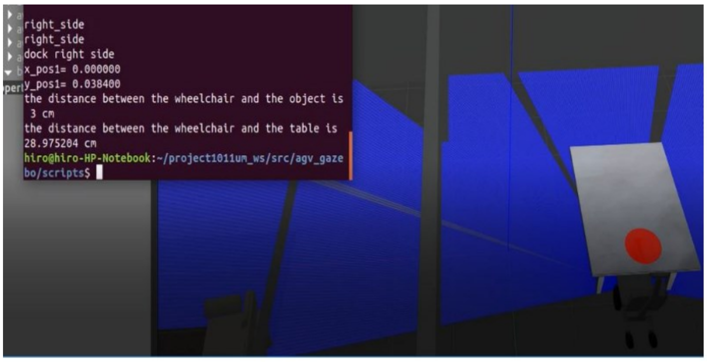


(a)

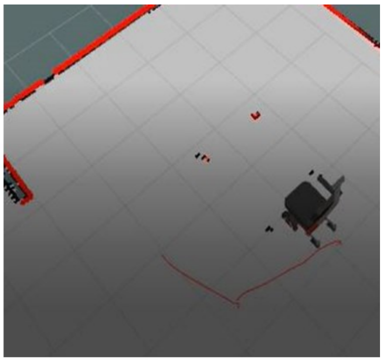


(b)

Figure 4.19 Object placed in the front right side

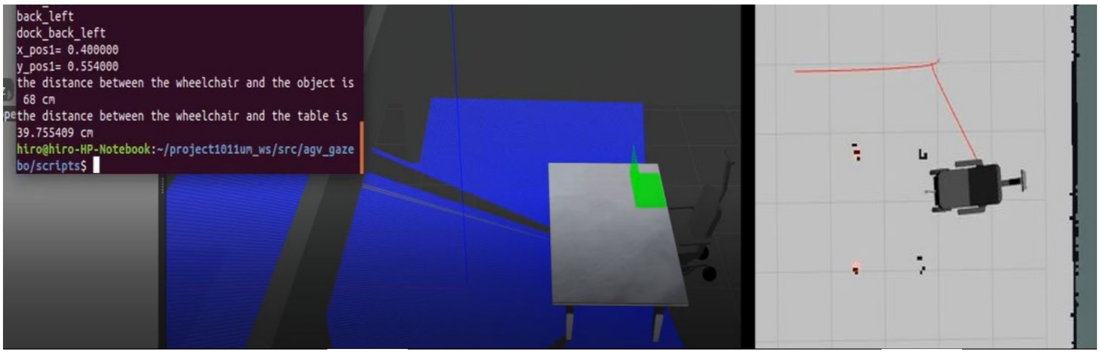


(a)



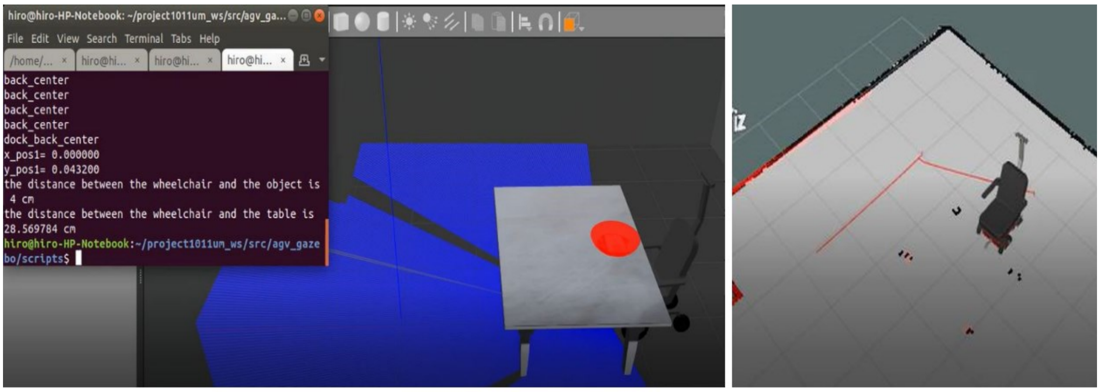
(b)

Figure 4.20 Object placed on the right side of the table



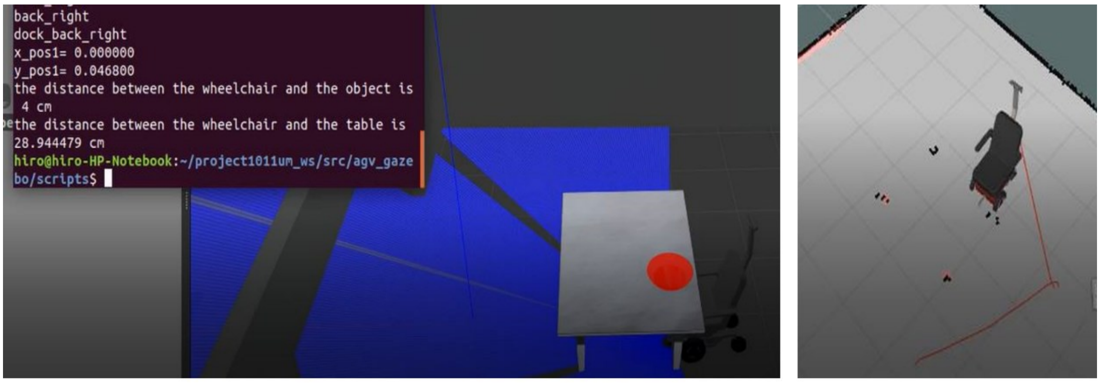
(a) (b)

Figure 4.21 Object placed on the back left side



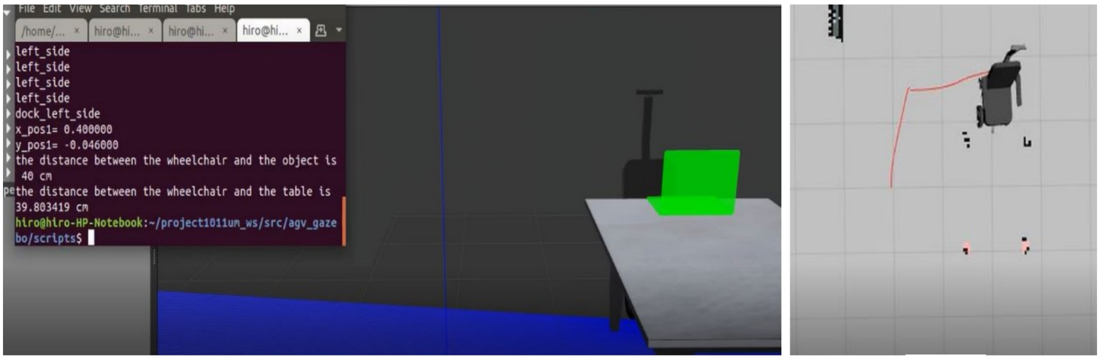
(a) (b)

Figure 4.22 Object placed on the back center side



(a) (b)

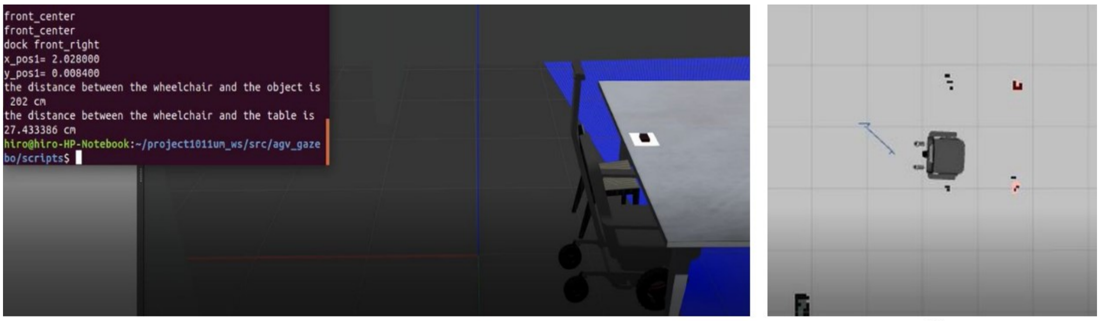
Figure 4.23 Object placed on the opposite right of wheelchair facing side



(a) (b)

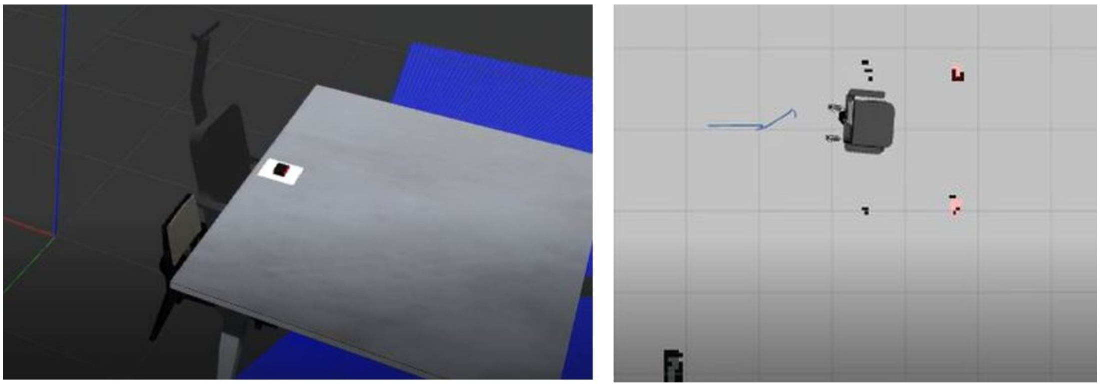
Figure 4.24 Object placed on the left side of the table

The above figures show the wheelchair docking when there is an object on the table. Objects such as a book, a bowl, and a laptop were placed on the front left side (Figure 4.17), front right (Figure 4.19), front middle side (Figure 4.18), left side (Figure 4.24), right side (Figure 4.20), and the opposite side (Figure 4.21, Figure 4.22, Figure 4.23) of the table and observed the docking. At a time only one object was placed on the table. The distance between the wheelchair and the table and the distance between the wheelchair and the objects were measured.



(a) (b)

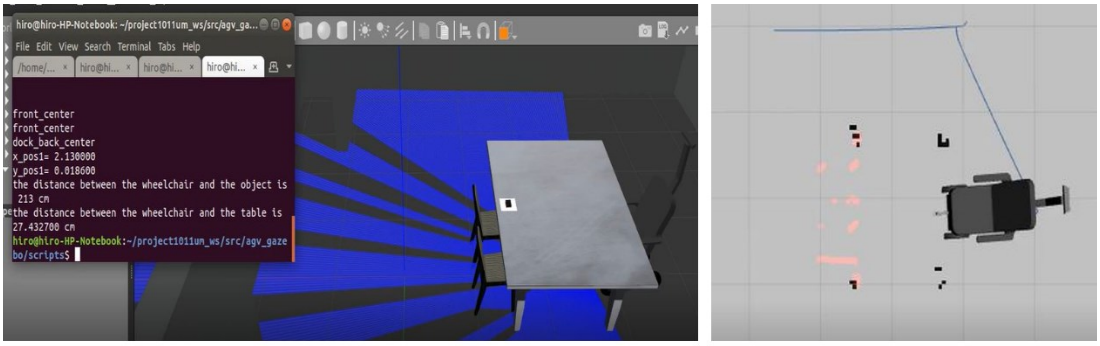
Figure 4.25 Object placed on wheelchair facing side



(a) (b)

Figure 4.26 Object placed on wheelchair facing side with a chair.

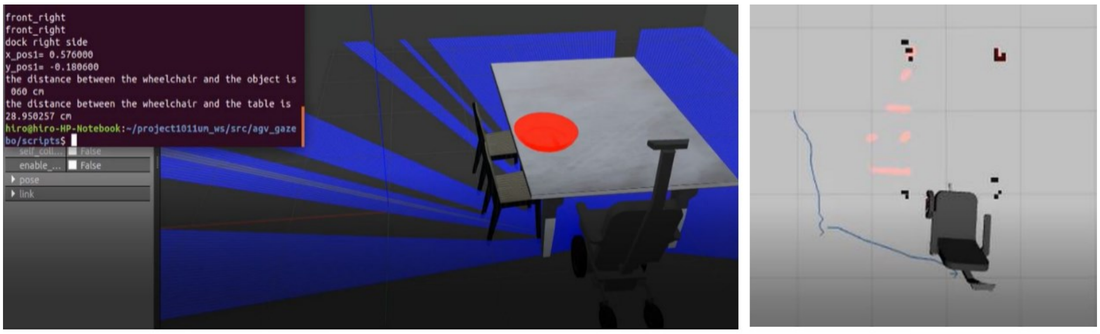
Figure 4.25 and Figure 4.26 indicate there is an object in the front center position of the table, and there is a chair near to it which is disturbing the wheelchair to dock with that object, then it will dock to the object possible nearest position.



(a) (b)

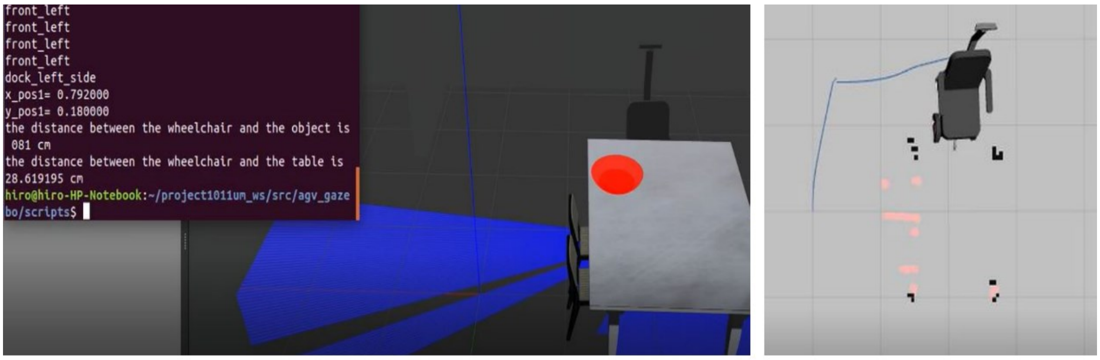
Figure 4.27 Object placed on the wheelchair facing side with no free space

Figure 4.27 shows there is an object on the front center side of the table and there is no free space on the front side, then it will dock to the opposite side which is the nearest position to reach the object.



(a) (b)

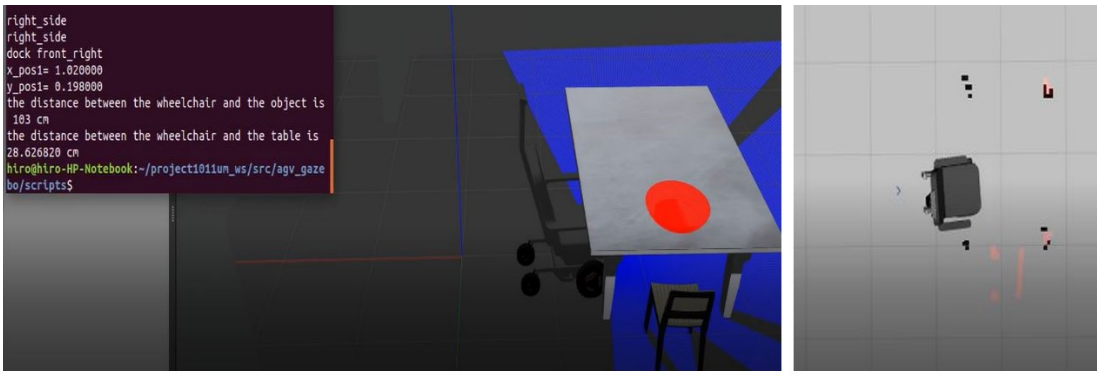
Figure 4.28 Object placed in the wheelchair facing side with no free space on that side.



(a) (b)

Figure 4.29 Object placed on the wheelchair facing side with no free space and near to the left side

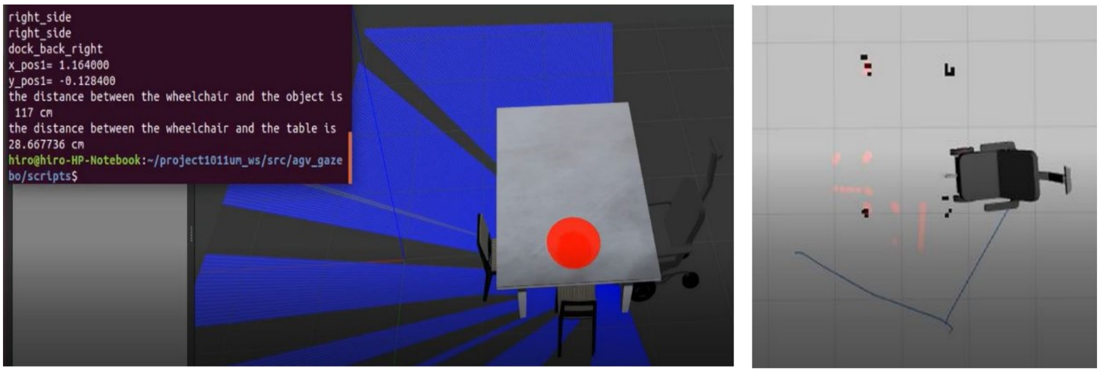
If there is an object on the front right side of the table, and there is no space to dock with that side, then it will dock to the right side of the table as it is the possible nearest position to reach the object. Also if the object is placed on the front left side of the table and there is no space to dock, then it will dock to the left side of the table. Above both scenarios were tested and Figure 4.29 shows the results of left side docking and Figure 4.28 shows the right side docking.



(a) (b)

Figure 4.30 Object placed on the right side of the table with a chair on the same side.

Figure 4.30 shows if an object is on the right side of the table and there is no space on the right side, then it will dock to the wheelchair-facing side of the table as it is the nearest to the object.



(a) (b)

Figure 4.31 Object placed on the right side of the table with chairs on the same side and facing side

Figure 4.31 shows if an object is placed on the right side of the table, and there is no space on the right side and wheelchair facing side, then it will dock to the opposite of wheelchair facing side.

Table 4.2 Summary Results of docking based on the object detection

Test Cases	The object on the table	Analyzed Ergonomics Data (cm)	Docking Distance base on the simulation (cm)
The object placed on the front left side	Book	27.25	27.454
The object placed in the front center side	Book	27.25	27.249
The object placed on the front right side	Bowl	29	28.918
The object placed on the right side of the table	Bowl	29	29.975
The object placed on the opposite left of wheelchair facing side	Laptop	40	39.755
The object was placed on the opposite center of wheelchair facing side	Bowl	29	28.569
The object placed on the opposite right of wheelchair facing side	Bowl	29	28.944
The object placed on the left side of the table	Laptop	40	39.803
The object was placed on wheelchair facing side with a chair on the left side	Book	27.25	27.433
The object was placed on wheelchair facing side with a chair on the right side	Book	27.25	27.318
The object was placed on the wheelchair facing side with two chairs	Bowl	29	28.912
The object placed in the wheelchair facing side with no free space on that side	Book	27.25	27.432
The object was placed on the right side of wheelchair facing side with no free space on that side	Bowl	29	28.950
The object was placed on the left side of wheelchair facing side with no free space on that side	Bowl	29	28.619

The object was placed on the right side of the table with a chair on the same side	Bowl	29	28.627
The object was placed on the right side of the table with chairs on the same side and facing side	Bowl	29	28.668
The object was placed on the left side of the table with a chair on the same side	Bowl	29	28.577
The object was placed on the left side of the table with chairs on the same side and facing side	Bowl	29	28.613
The object was placed on the left side of the table with chairs on the same side, facing side and right side	Bowl	29	28.676
The object was placed on the left side of the table with chairs on the same side, facing the right side and right side	Bowl	29	28.666
The object was placed on the left side of the table with chairs on the same side, facing the right side, right side of the opposite side	Bowl	29	28.699
The object placed on the left side of the table with chairs on the same side, facing side, right side, and back right side	Book	27.25	27.110

The object was placed in several places on the table with different table setups, as discussed above shown in Table 4.2. Expected wheelchair behavior was observed successfully.

4.3 Feedback result for simulation survey

The research was planned to validate using the test trial in the actual environment. Then based on their feedback gathered in the simulation survey, the comfort of the docking behavior of the autonomous wheelchair was planned to be analyzed. Due to the COVID pandemic, it was impossible to do the test trial in a physical environment user. Therefore, the results were analyzed based on the gathered information from the survey performed by the participants of the survey, by watching the simulation videos. Videos were recorded based on the test cases mentioned in previous sections. Analysis was performed to assess the wheelchair's docking behavior based on the user preference and object detection.

The simulation survey along with the recorded videos was shared among the university students via google form and seventy-five responses were received. After watching the videos, participants were asked to complete the survey. Participants were asked to rate the following claims on a 5-point Likert scale for each question. Videos details were mentioned below:

Video 1 - Docking to the table without any chairs
Video 2 - Docking to the table with one chair
Video 3 - Docking to the table with chairs on two sides
Video 4 - Docking to the table with chairs on three sides
Video 5 - Docking with table with chairs on all sides
Video 6 - Docking to the table with laptop

- **Q1:** Rate your satisfaction with the wheelchair behavior in video 1.
- **Q2:** Rate your satisfaction with the wheelchair behavior in video 2.
- **Q3:** Rate your satisfaction with the wheelchair behavior in video 3.
- **Q4:** Rate your satisfaction with the wheelchair behavior in video 4.
- **Q5:** Rate your satisfaction with the wheelchair behavior in video 5.
- **Q6:** Rate your satisfaction with the wheelchair behavior in video 6.
- **Q7:** Rate your satisfaction with the wheelchair's final alignment in videos 1 to 5.
- **Q8:** What do you think about the movements of the wheelchair in the above videos?
- **Q9:** Rate your satisfaction level for obstacle identification and movement of the wheelchair in video 1-5
- **Q10:** Rate your satisfaction level for object detection and docking of a wheelchair in video 6.

5-Strongly Agree, 4-Agree, 3-Neutral, 2-Disagree, and 1-Strongly Disagree were the potential input scores for Q1-Q10 except for Q8.
The survey respondents' responses to the question statements were evaluated to assess the success of the built docking process. The scores received from participants for the above question statements are depicted in Figure 4.32.

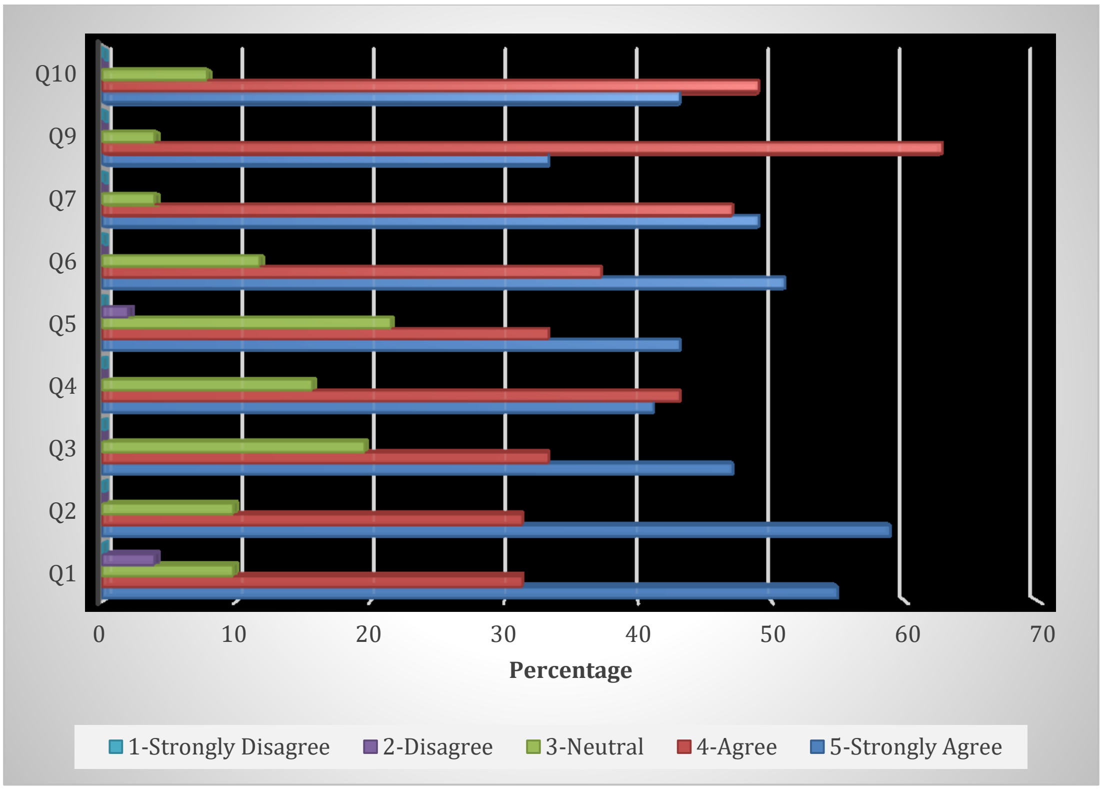


Figure 4.32 The results of the 5-point Likert scales used to receive participant scores for each question are plotted here.

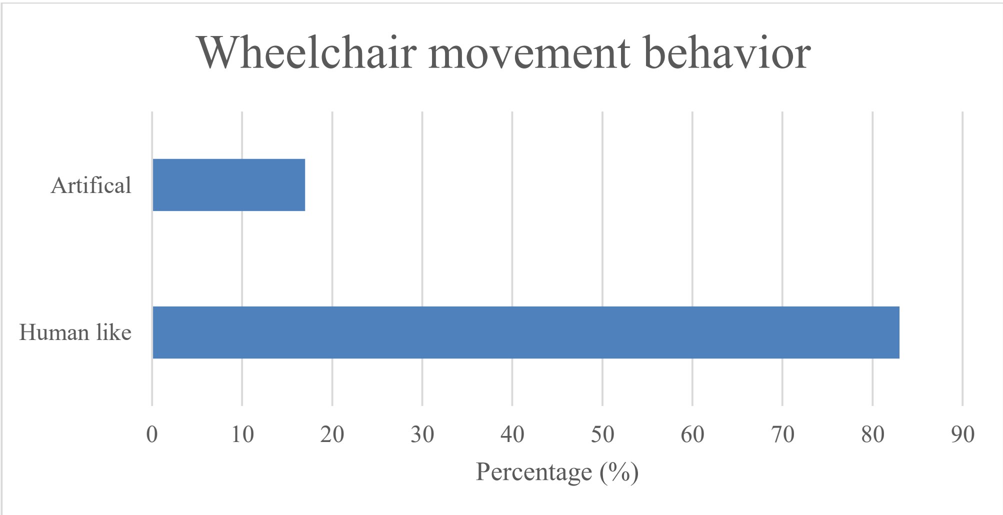


Figure 4.33 Wheelchair movement behavior results from the simulation survey

The above Figure 4.33 was shown the wheelchair movement and docking behavior analyzed based on the below movement type, such as Humanlike behavior i.e wheelchair movement like a manual wheelchair for navigation and docking, Artificial behavior i.e wheelchair docking behavior is like not a human behavior in docking mentioned in [47].

CHAPTER 5 CONCLUSION AND FUTURE WORKS

In the case of intelligent wheelchairs, as well as autonomous mobile robots, finding safe docking areas is critical. This research focused to fulfill three sub-objectives to achieve the main goal as to improve the docking behavior of the autonomous wheelchair. A novel method is presented here for determining comfortable docking locations based on the analyzed ergonomics data which was gathered through human subjects on actual wheelchair usage, to be applied within a single algorithm to get a safe location using 3D point cloud data, which requires no visual fiducial or environment customization. Two scenarios were considered while docking with the table. One is a table with an object on the table, the other one is without an object on the table. If the object is on the table, it will dock based on the object's position on the table and free space availability. If there is no object, then it will dock based on the user preference by considering the free space availability. Navigation and obstacle avoidance are also integrated into this wheelchair to enhance its autonomous behavior to navigate in the domestic environment.

Also, this study proves that there is a significant difference in the distances between wheelchair back end and table among the four different scenarios and it also proves that the height differences between the table surface and wheelchair seat at different scenarios vary significantly according to the heights of individuals above the hip. Hence, the type of scenario performed should be considered when deciding the suitable height difference between the table surface and wheelchair seat for an individual. For each scenario, the height above the hip of an individual should be considered when deciding the suitable height differences between the table surface and wheelchair seat. As there is a strong positive relationship between height above hip and height of an individual, human height can also be considered when designing a wheelchair.

5.1. Conclusions about the Research Objectives

Sub-objective 1: What are the parameters that affect the comfort of the docking

Analysis of the human study proves that there is a significant difference in the distances between wheelchair back end and table among the four different scenarios (eating, writing, reading, and using a laptop) and it also proves that the height differences between the table surface and wheelchair seat vary significantly according to the heights of the table.

Sub-objective 2: According to user preference wheelchair docks with the table

If there is no object on the table, the wheelchair was a dock to the table based on the user preference and considering the free space availability. It was docked to the other side of the table if any obstacle under the table. It also considered the results obtained from the human study to decide the docking distance based on the user preference.

Sub-objective 3: *According to the object on the table wheelchair docks with the table*

The object was identified by the wheelchair and docked to the table based on that. Docking distance was determined from the results obtained from the human study.

The overall objective was achieved either by automatically identifying the object in the table, and dock with the table based on that purpose or based on the preference of the user dock with the table by applying the docking parameters.

5.2. Further works

Future work includes the distance between the wheelchair back end and table, and the height difference between the table surface and wheelchair seat can be analyzed with height above the hip of an individual to further improve the research. Also, this research is limited to rectangular tables only. As a future development safe docking can be implemented for various structures of the table. Furthermore, deep learning techniques can be integrated with object detection to improve accuracy.

PUBLICATION

[1] H. Vadivel, K. S. Priyanayana, and A. G. B. P. Jayasekara, "Enhancing the capabilities of approaching to service scenarios and settling for intelligent wheelchair robots," in *2021 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2021, pp. 711–716.

REFERENCES

[1] H. W. Nam, "Design of AI-based gamification platform for effective educational service using child behavior prediction/change," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 5, pp. 286–292, 2021, doi: 10.17762/turcomat.v12i5.899.

[2] M. J. Johnson *et al.*, "Task and Design Requirements for an Affordable Mobile Service Robot for Elder Care in an All-Inclusive Care for Elders Assisted-Living Setting," *Int. J. Soc. Robot.*, vol. 12, no. 5, pp. 989–1008, 2020, doi: 10.1007/s12369-017-0436-5.

[3] J. Holland *et al.*, "Service robots in the healthcare sector," *Robotics*, vol. 10, no. 1, pp. 1–47, 2021, doi: 10.3390/robotics10010047.

[4] L. Boccanfuso *et al.*, "Emotional robot to examine different play patterns and affective responses of children with and without ASD," *ACM/IEEE Int. Conf. Human-Robot Interact.*, vol. 2016-April, pp. 19–26, 2016, doi: 10.1109/HRI.2016.7451729.

[5] M. Ohkita, Y. Bando, E. Nakamura, K. Itoyama, and K. Yoshii, "Audio-visual beat tracking based on a state-space model for a robot dancer performing with a human dancer," *J. Robot. Mechatronics*, vol. 29, no. 1, pp. 125–136, 2017, doi: 10.20965/jrm.2017.p0125.

[6] H. S. Ahn *et al.*, "Entertainment services of a healthcare robot system for older people in private and public spaces," *ICARA 2015 - Proc. 2015 6th Int. Conf. Autom. Robot. Appl.*, pp. 217–222, 2015, doi: 10.1109/ICARA.2015.7081150.

[7] F. Hussain, S. Mazhar, A. Shah, A. Ali, I. Ali, and H. Kumar, "Voice and Hand Controlled Wheelchair System," vol. 1, pp. 34–42, 2021.

[8] H. Yoshitake, K. Futawatari, and M. Shino, "A VR-Based Simulator Using Motion Feedback of a Real Powered Wheelchair for Evaluation of Autonomous Navigation Systems," *Adjun. Proc. - 13th Int. ACM Conf. Automot. User Interfaces Interact. Veh. Appl. AutomotiveUI 2021*, pp. 26–29, 2021, doi: 10.1145/3473682.3480257.

[9] C. T. Rusek, M. Kleven, C. Walker, K. Walker, R. Heeb, and K. A. Morgan, "Perspectives of inpatient rehabilitation clinicians on the state of manual wheelchair training: a qualitative analysis," *Disabil. Rehabil. Assist. Technol.*, vol. 0, no. 0, pp. 1–9, 2021, doi: 10.1080/17483107.2021.1993359.

[10] S. Desai, S. S. Mantha, and V. M. Phalle, "Advances in smart wheelchair technology," *2017 Int. Conf. Nascent Technol. Eng. ICNTE 2017 - Proc.*, 2017, doi: 10.1109/ICNTE.2017.7947914.

[11] A. Hartman and V. K. Nandikolla, "Human-Machine Interface for a Smart Wheelchair," *J. Robot.*, vol. 2019, 2019, doi: 10.1155/2019/4837058.

[12] Á. Rocha, A. M. Correia, F. B. Tan, and K. A. Stroetmann, "A Survey on

Intelligent Wheelchair Prototypes and Simulators," *Adv. Intell. Syst. Comput.*, vol. 275, no. VOLUME 1, 2014, doi: 10.1007/978-3-319-05951-8.

[13] S. V. and N. L. Luís Paulo Reis, Brígida Mónica Faria, *Multimodal Interface for an Intelligent Wheelchair*, vol. 325. 2015.

[14] M. Letaief, N. Rezzoug, and P. Gorce, "Comparison between joystick- and gaze-controlled electric wheelchair during narrow doorway crossing: Feasibility study and movement analysis," *Assist. Technol.*, vol. 33, no. 1, pp. 26–37, 2021, doi: 10.1080/10400435.2019.1586011.

[15] G. Beraldo, E. Termine, and E. Menegatti, "Shared-Autonomy Navigation for Mobile Robots Driven by a Door Detection Module," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11946 LNAI, no. October, pp. 511–527, 2019, doi: 10.1007/978-3-030-35166-3_36.

[16] M. Derry and B. Argall, "Automated doorway detection for assistive shared-control wheelchairs," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1254–1259, 2013, doi: 10.1109/ICRA.2013.6630732.

[17] S. Jain and B. Argall, "Automated perception of safe docking locations with alignment information for assistive wheelchairs," *IEEE Int. Conf. Intell. Robot. Syst.*, no. Iros, pp. 4997–5002, 2014, doi: 10.1109/IROS.2014.6943272.

[18] A. L. Barriuso, J. Pérez-Marcos, D. M. Jiménez-Bravo, G. Villarrubia González, and J. F. De Paz, "Agent-based intelligent interface for wheelchair movement control," *Sensors (Switzerland)*, vol. 18, no. 5, 2018, doi: 10.3390/s18051511.

[19] T. Carlson and J. Del R. Millan, "Brain-controlled wheelchairs: A robotic architecture," *IEEE Robot. Autom. Mag.*, vol. 20, no. 1, pp. 65–73, 2013, doi: 10.1109/MRA.2012.2229936.

[20] G. B. Frédéric Leishman, Odile Horn, *Multimodal Laser-Vision Approach for the Deictic Control of a Smart Wheelchair*, vol. 5597. 2009.

[21] Y. Ren, W. Zou, H. Fan, A. Ye, K. Yuan, and Y. Ma, "A docking control method in narrow space for intelligent wheelchair," *2012 IEEE Int. Conf. Mechatronics Autom. ICMA 2012*, pp. 1615–1620, 2012, doi: 10.1109/ICMA.2012.6284378.

[22] H. Sermeno-Villalta and J. Spletzer, "Vision-based control of a smart wheelchair for the Automated Transport and Retrieval System (ATRS)," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, no. July, pp. 3423–3428, 2006, doi: 10.1109/ROBOT.2006.1642225.

[23] M. Nuttin, D. Vanhooydonck, E. Demeester, and H. Van Brussel, "Selection of suitable human-robot interaction techniques for intelligent wheelchairs," *Proc. - IEEE Int. Work. Robot Hum. Interact. Commun.*, pp. 146–151, 2002, doi: 10.1109/ROMAN.2002.1045613.

[24] R. C. Simpson, D. Poirot, and F. Baxter, “The Hephaestus Smart Wheelchair System,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 2, pp. 118–122, 2002, doi: 10.1109/TNSRE.2002.1031980.

[25] Z. Wei, W. Chen, and J. Wang, “3D semantic map-based shared control for smart wheelchair,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7507 LNAI, no. PART 2, pp. 41–51, 2012, doi: 10.1007/978-3-642-33515-0_5.

[26] Y. Chen, D. P. Member, and B. L. Member, “Virtual Landmark-Based Control of Docking Support for Assistive Mobility Devices,” *IEEE/ASME Trans. Mechatronics*, 2021, doi: 10.1109/TMECH.2021.3081426.

[27] A. Ravankar, A. A. Ravankar, M. Watanabe, Y. Hoshino, and A. Rawankar, “Multi - robot path planning for smart access of distributed charging points in map,” *Artif. Life Robot.*, no. 0123456789, 2020, doi: 10.1007/s10015-020-00612-8.

[28] B. Sebastian, “Autonomous Docking of Hybrid- Wheeled Modular Robots With an Integrated Active Genderless Docking Mechanism,” *J. Mech. Robot.*, vol. 14, no. February, pp. 1--36, 2022, doi: 10.1115/1.4051519.

[29] Y. Zhu and Q. Meng, “Wheelchair Automatic Docking Method for Body-Separated Nursing Bed Based on Grid Map,” vol. 9, pp. 79549–79561, 2021, doi: 10.1109/ACCESS.2021.3084620.

[30] Q. Zhang, Juzhong and Cai, Liming and Chu, Yuyi and Zhou, “A Sectional Auto-docking Charging Control Method for the Mobile Robot,” in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2019, pp. 330–335.

[31] T. Bailey, E. M. Nebot, and J. K. Rosenblatt, “Behaviour-Based Docking using the DAMN Arbiter,” *Engineering*, 2006.

[32] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, and A. G. B. P. Jayasekara, “Replicating Natural Approaching Behavior of Humans for Improving Robot’s Approach toward Two Persons during a Conversation,” *RO-MAN 2018 - 27th IEEE Int. Symp. Robot Hum. Interact. Commun.*, no. August, pp. 552–558, 2018, doi: 10.1109/ROMAN.2018.8525834.

[33] C. Weber, S. Wermter, and A. Zochios, “Robot docking with neural vision and reinforcement,” *Knowledge-Based Syst.*, vol. 17, no. 2–4, pp. 165–172, 2004, doi: 10.1016/j.knosys.2004.03.012.

[34] T. Martínez-Marín and T. Duckett, “Fast reinforcement learning for vision-guided mobile robots,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2005, pp. 4170–4175, 2005, doi: 10.1109/ROBOT.2005.1570760.

[35] D. Muse, C. Weber, and S. Wermter, “Robot docking based on omnidirectional vision and reinforcement learning,” *Res. Dev. Intell. Syst. XXII - Proc. AI 2005, 25th SGAI Int. Conf. Innov. Tech. Appl. Artif. Intell.*, pp. 23–36, 2006, doi: 10.1007/978-1-84628-226-3_3.

[36] M. Williamson, R. Murray-Smith, and V. Hansen, "Robot docking using mixtures of Gaussians," *Adv. Neural Inf. Process. Syst.*, no. December 1998, pp. 945–951, 1999.

[37] X. Liang, X. Li, S. Jia, and Y. Sun, "Wheelchair/bed docking control based on the combination of vision and ultrasound," *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, pp. 4441–4446, 2017, doi: 10.1109/CAC.2017.8243562.

[38] X. Li, X. Liang, J. Fan, and S. Jia, "Position-based visual servo control of intelligent wheelchair/bed docking," *2018 IEEE Int. Conf. Inf. Autom. ICIA 2018*, no. August, pp. 1497–1502, 2018, doi: 10.1109/ICInfA.2018.8812315.

[39] P. Fearn, Tomos and Labrosse, Frederic and Shaw, "Wheelchair Navigation Wheelchair Navigation : Automatically Adapting to Evolving Environments," in *Annual Conference Towards Autonomous Robotic Systems*, 2019, pp. 496--500.

[40] L. Joseph, *Robot Operating System for Absolute Beginners*. 2018.

[41] B. Bingham, C. Ag, T. Lum, and M. Rawson, "Toward Maritime Robotic Simulation in Gazebo," no. October, 2019, doi: 10.23919/OCEANS40490.2019.8962724.

[42] A. A. Kusumo, B. S. Marta, B. Sena, B. Dewantara, and D. Pramadihanto, "2D Mapping and Localization using Laser Range Finder for Omnidirectional Mobile Robot," *2019 Int. Electron. Symp.*, pp. 126–131, 2019.

[43] J. Jung, J. Park, T. Kang, and J. Kang, "applied sciences Mobile Robot Path Planning Using a Laser Range Finder for Environments with Transparent Obstacles," *Appl. Sci.*, vol. 10, no. 8, p. 2799, 2020.

[44] J. Pu, Y. Jiang, X. Xie, X. Chen, M. Liu, and S. Xu, "Low Cost Sensor Network for Obstacle Avoidance in Share-Controlled Smart Wheelchairs under Daily Scenarios," *Microelectron. Reliab.*, vol. 83, pp. 180--186, 2018.

[45] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM Giorgio," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 38, no. 4, pp. 30–48, 2017, [Online]. Available: https://scholarsarchive.byu.edu/facpubhttps://scholarsarchive.byu.edu/facpub/1963%0Ahttps://scholarsarchive.byu.edu/etd/6609%0Ahttp://www.roboticsproceedings.org/rss09/p37.pdf%0Ahttps://youtu.be/CsJkci5lfco.

[46] P. Kudriashov, Andrii and Buratowski, Tomasz and Giergiel, Mariusz and Malka, *SLAM Techniques Application for Mobile Robot in Rough Terrain*. Springer, 2020.

[47] M. Destephe, M. Brandao, T. Kishi, M. Zecca, K. Hashimoto, and A. Takanishi, "Walking in the uncanny valley: Importance of the attractiveness on the acceptance of a robot as a working partner," *Front. Psychol.*, vol. 6, no. FEB, pp. 1–11, 2015, doi: 10.3389/fpsyg.2015.00204.