

Development of an Obstacle Detection and Navigation System for Autonomous Powered Wheelchairs

by

Heta Diwan

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

Faculty of Engineering and Applied Science

Mechanical Engineering

Ontario Tech University
Oshawa, Ontario, Canada

May 2019

©Heta Diwan 2019

THESIS EXAMINATION INFORMATION

Submitted by: **Heta Diwan**

Master of Applied Science in Mechanical Engineering

Thesis title: Development of an Obstacle Detection and Navigation System for Autonomous Powered Wheelchairs

An oral defense of this thesis took place on May 28th, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Carlos Rossa
Research Supervisor	Dr. Scott Nokleby
Examining Committee Member	Dr. Jaho Seo
Thesis Examiner	Dr. Jing Ren, FEAS

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

This thesis describes the development and prototyping of an after market system to convert an electric powered wheelchair into an autonomous wheelchair. The purpose of this research is to automatize powered wheelchairs for children who are impacted by conditions such as cerebral palsy. Maneuvering a powered wheelchair with a joystick is difficult and painful for users who have a high level of cerebral deficiency and other chronic conditions. The autonomous powered wheelchair is designed to maneuver in an indoor environment whilst avoiding static and dynamic obstacles. The add-on system comprised of stereo vision sensors (ZED Camera), and IMU is designed to use Robot Operating System (ROS) to communicate and control the movement of the wheelchair. With the addition of a 3D map of the environment generated using visual sensors through ROS packages, the system identifies and avoids obstacles. Simultaneous Localization and Mapping (SLAM) and autonomous navigation packages were tested and modified. Slopes and drops identified in the 3D map are converted such that they are compatible with the 2D navigation packages of ROS. Configuration settings were determined and tested to ensure that the system works as required. The results demonstrated that the powered wheelchair can be modified to become an autonomous wheelchair using ZED Camera and IMU, such that it can navigate indoors effectively avoiding static and dynamic obstacles.

Keywords: robotics; autonomous wheelchair; obstacle detection; slope and stair detection; autonomous navigation

To my family and friends.

Statement of Contributions

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Acknowledgements

I would like to thank my supervisor, Dr. Scott Nokleby. Without his support and advice completion of thesis was not possible.

I would like thank Sunrise Medical Inc. for their generous donation of the wheelchair Quickie Xperience2 and Grandview Children's Centre for entrusting us to develop a prototype for children with cerebral palsy.

Lastly, I would also like to thank my colleagues in the Mechatronic and Robotic Systems Laboratory for their input during the process of this thesis.

Table of Contents

Abstract	iii
Acknowledgements	vi
Table of Contents	ix
List of Tables	x
List of Figures	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Introduction to Cerebral Palsy	2
1.3 Scope	2
1.4 Problem Statement	3
1.5 Requirements	4
1.5.1 Functional Requirements	4
1.5.2 Physical Requirements	4
1.5.3 Performance Requirements	5
1.5.4 Safety Requirements	5
1.6 Contributions	5
1.7 Outline	5

2	Background	7
2.1	History of Wheelchairs	7
2.2	Knowledge Gap	14
2.2.1	Simultaneous Localization and Mapping (SLAM)	15
2.2.2	Obstacle Detection	16
2.2.2.1	Stereoscopic Camera	17
2.2.2.2	LiDAR Sensor	19
2.2.2.3	Safety Standards	19
2.2.2.4	Data Logger	21
2.3	Related State-of-the-Art Contributions	22
2.3.1	Autonomous Navigation for Unmanned Ground Vehicles	22
2.3.2	Autonomous Navigation for Electric Powered Wheelchairs	24
2.4	Summary	28
3	Wheelchair Modifications	30
3.1	Control System	33
3.1.1	Circuit Design	35
3.1.2	Kinematic Model	37
3.1.2.1	Differential Drive	37
3.1.2.2	Implementation of Kinematic Model	41
3.2	Add-on Device Design Prototype	42
3.2.1	Types of Sensors	42
3.2.2	Hardware Implementation of Add-on Device Accessories	46
4	Noise Filtration and Sensor Fusion	49
4.1	Sensor Noise	50
4.2	Sensor Noise Filters	53
4.3	ROS Implementation	55

4.3.1	SLAM Algorithms	56
4.3.2	ROS Implementation of Sensor Noise Filter for SLAM	57
4.4	Initial Test Results	60
5	3D Mapping, Slope Detection, and Autonomous Navigation	61
5.1	3D Mapping	62
5.1.1	Implementation Results	64
5.2	Slope and Stair Detection	66
5.3	Autonomous Navigation	69
5.4	Final Test Results	72
5.4.1	Test 1 - Hallway Autonomous Navigation	73
5.4.2	Test 2 - Stair Avoidance Autonomous Navigation	75
5.4.3	Test 3 - Ramp Acceptance Autonomous Navigation	77
6	Conclusions and Future Works	79
6.1	Lessons Learned	81
6.2	Future Work	81

List of Tables

3.1	List of Components for Conventional Control Systems	33
3.2	RoboteQ Motor Controller Technical Specifications	35
3.3	Types of Non-Contact Sensors and Technical Specifications	45
3.4	ZED Camera Technical Specifications [55]	46

List of Figures

2.1	Wheelchair Designed for King Phillip II of Spain [7]	8
2.2	Wheelchair Designed by Stephen Farfler in 1655 [7]	9
2.3	Wheelchair design by James Dawson [7]	10
2.4	First Patent of a Wheelchair in 1853 classified under Patent Number US9708A [9]	10
2.5	First Folding Wheelchair by Everest and Jennings US2095411A [10]	11
2.6	First Electric Powered Wheelchair [14]	13
2.7	BrainGate Device [15]	13
2.8	Occupancy Grid Map with Fused Data [34]	23
2.9	Traversability Map Built Using RPM Algorithm for Three Different LiDAR Applications [38]	24
2.10	Occupancy Grid Map with Fused Data [42]	26
3.1	QUICKIE Xperience 2 Wheelchair by Sunrise Medical Inc. [5]	31
3.2	R-Net Power Module 120AMP [5]	32
3.3	R-Net Encoder Module [5]	32
3.4	R-Net OBP and PC Programmer [5]	32
3.5	RoboteQ XDC2430 Brushed DC Motor Controller [50]	34
3.6	RoboteQ XDC2430 Discharge Circuit [50]	36
3.7	Differential Drive Model [51]	37
3.8	Roll, Pitch and Yaw Representation [53]	43

3.9	Zed Stereo Camera [55]	45
3.10	Backrest Sensor Mount	47
3.11	Modified Autonomous Wheelchair	48
4.1	General Schematic for Mobile Robot Localization [56]	49
4.2	Movement of Differential Drive Robot [56]	51
4.3	Range Error of the Mobile Robot [56]	52
4.4	Turn Error of the Mobile Robot [56]	52
4.5	General Schematic of SLAM Process in ROS	59
4.6	Environmental Map using EKF	60
4.7	Environmental Map using UKF	60
5.1	2D SLAM Algorithm for Multi-level Flooring	62
5.2	UML Diagram of the Most Common Octree and Node Classes [70]	63
5.3	3D Map of the Multi-level Environment using OctoMap in rviz	64
5.4	3D Map of the Multi-level Environment using RTAB-Map Visualization Tool	65
5.5	3D and 2D Map of the Multi-level Environment using RTAB-Map in rviz	66
5.6	An Example of a Local Costmap for Stair Detection in rviz Visualiza- tion Tool	68
5.7	A High-level Schematic of the Basic Autonomous Navigation System of the Wheelchair	70
5.8	A High-level Schematic of the move_base Node [77]	71
5.9	A High-level Schematic of the Autonomous Navigation System of the Wheelchair	72
5.10	An Example of Navigation Path in rviz for Hallway Environment	73
5.11	Test 1 - Validation of the Hallway Driving using the Add-on System	74
5.12	An Example of the Navigation Path in rviz for Stairs Avoidance	75

5.13	Test 2 - Validation of the Stair Avoidance using the Add-on System . .	76
5.14	An Example of the Navigation Path in rviz for Ramp Environment . .	77
5.15	Test 3 - Validation of the Ramp Driving using the Add-on System . . .	78

Chapter 1

Introduction

1.1 Motivation

Robotics is widely used in manufacturing and industrial industries; however, its use within the medical field is directly related to the public and is restricted to surgical, rehabilitation, and bio-robots. The purpose of this research is to extend the knowledge of robotics to automatize powered wheelchairs for children who are impacted by conditions, such as, cerebral palsy. Maneuvering a powered wheelchair with a joystick is difficult and painful for users who have a high level of cerebral deficiency and other chronic conditions. Depending on the severity of the condition, a child sometimes requires caregivers to maneuver them. This impact significantly affects the mental strength of the child and their self-esteem. Yet, the impact is not only significant for children who are affected by disabilities, but also their families and caregivers. Statistically, about 67.5% of families adjust their work schedule, reduce the amount of work, or sacrifice their career to adjust to the needs of their child [1]. Research to create an autonomous powered wheelchair will help reduce the impact on the mental strength of children and allow them to be independent. As a result, it increases flexibility to the parents and caregivers of children impacted by cerebral palsy.

An autonomous powered wheelchair is a motorized wheelchair that does not require human input for maneuvering. It is one of the upcoming research topics in the world of assisted mobility devices that is widely divided into two main sectors. Most of the research that is present today looks at the topics of obstacle detection and automatic navigation systems. These topics are vastly explored to improve on the mechanics of a powered wheelchair. Unfortunately, there has been little research that combines the two sectors to design a truly autonomous powered wheelchair. With little research available, the question remains whether the two sectors can be integrated to design an autonomous powered wheelchair?

1.2 Introduction to Cerebral Palsy

Cerebral palsy is a neurological disorder caused by a malformation or brain injury during the development of a child's brain. It primarily affects body movement and muscle coordination. In other words, the "muscles are constantly vibrating" [2]. The condition varies from person to person, where it can impact the body's movement in various forms, such as, muscle tone, reflexes, posture, and balance. Moreover, it is not limited to body movements, it can also impact fine motor skills, gross skills, and oral motor skills.

1.3 Scope

The scope of this research is to develop an add-on device that integrates obstacle detection and autonomous navigation for an existing powered wheelchair for indoor use. An add-on device is being developed to reduce the economic impact on families that require a powered wheelchair. In this research, various sensors will be examined for obstacle detection along with observing their interaction with the environment while using Simultaneous Localization and Mapping (SLAM) for navigation with real-time

results. Widely used in the Robot Operating System (ROS) (see www.ros.org), SLAM is the process of identifying a map of an unknown environment while simultaneously keeping track of the location of a robot within the environment [3]. The wheelchair will communicate with ROS using a laptop as a user interface for the purpose of this scope. However, as outlined in Section 6.2, the user interface will have to be designed depending on the needs of the specific child for when the system is deployed in the market.

1.4 Problem Statement

There are a number of different types of powered wheelchairs available to assist various forms of complex disabilities. Most of the wheelchairs cater to various input devices for maneuverability. Some of the known devices are Brain Computer Interface, eye movement controller, speech recognized controller, and head movement controller. These devices are ideal for users suffering through complex disabilities. However, the major downfall of these devices, especially for an individual suffering from cerebral palsy, is that it drains energy out of the user. As explained in Section 1.2, cerebral palsy restricts children from being active. To avoid such a scenario, this research presents a proof-of-concept prototype of an autonomous powered wheelchair.

To ensure that an existing powered wheelchair is converted to an autonomous powered wheelchair, there are key design concerns that need to be considered. The first design consideration is the control of the wheelchair and its modification. The second design consideration is the design of an obstacle detection system. Whereas, the last design consideration is the navigation system for the wheelchair. In order to ensure that the system is acceptable, the autonomous wheelchair should be able to meet the requirements listed in Section 1.5. For the accuracy of the navigation system, the wheelchair is validated by testing in a real-world environment.

1.5 Requirements

This section outlines four key requirements needed to ensure the scope of work is met efficiently. The first requirement discusses the wheelchair's functionality as an autonomous vehicle. The second requirement discusses physical limitations and modifications for the wheelchair, along with controllers design and its integration with the wheelchair. The third requirement discusses implementation of SLAM and slope detection for safe navigation. Lastly, the fourth requirement discusses safety standards and regulations to ensure safety for users and bystanders.

1.5.1 Functional Requirements

1. The wheelchair should reach the desired destination while avoiding all the obstacles (static and/or dynamic).
2. Ramps are acceptable for driving within the ranges of $\pm 5.7^\circ$. Otherwise, the ramp is considered as an obstacle.
3. Ramp edges, stairs, and walls should be avoided.

1.5.2 Physical Requirements

1. Modifications are subject to limitations based on wheelchair's physical structure.
2. The controller should be compatible with ROS.
3. The controller must have the correct wire connections to the wheelchair.
4. The wheelchair should be able to hold a user weighing 300 lbs with a maximum height of 6 feet (1.83 m).

1.5.3 Performance Requirements

1. Generate a 3D map of the environment and navigate the area in a safe manner.
2. Perform slope detection to fulfill functional requirements in Section 1.5.1.

1.5.4 Safety Requirements

1. The wheelchair should comply with ISO/TC 173/SC 1 - Wheelchairs for emergency safety, electrical designs and structural design [4].
2. Ensure that the speed is acceptable for crowded hallways, max speed allowed is 6.5 mph (10.5 km/h) [5].

1.6 Contributions

The main contributions of this thesis include:

1. Modification of the control system of the wheelchair
2. Customized noise filtration and data fusion ROS packages for the wheelchair application.
3. Tested and modified 3D mapping ROS packages to implement on the wheelchair for autonomous navigation.
 - (a) Customized the package to identify slopes and stairs in the 3D map.
 - (b) Feedback loop developed to relay the slope and stair data from the 3D map to 2D navigation.

1.7 Outline

This thesis is divided into six chapters, listed below:

- Chapter 1: Introduction - This chapter outlines the motivation and the scope of work within this thesis, followed by a discussion of the key requirements for the scope of the work.
- Chapter 2: Background - This chapter summarizes the history of the wheelchair and the research areas that were studied in this thesis, which is used to identify the knowledge gap between the state-of-the-art and the objectives of this thesis.
- Chapter 3: Wheelchair Modifications - This chapter discusses design modification for the controls of the wheelchair and the final prototype design.
- Chapter 4: Noise Filtration and Sensor Fusion - This chapter presents the method utilized to achieve sensor fusion and noise filtration. Followed by implementation of the filters to obtain SLAM system.
- Chapter 5: 3D Mapping, Slope Detection, and Autonomous SLAM Navigation - This chapter discusses the generation of 3D maps and slope detection using the results validated in Chapter 4, followed by the discussion of the autonomous navigation system.
- Chapter 6: Conclusions and Future Work - This chapter summarizes the work done within this thesis and discusses possible future improvements.

Chapter 2

Background

This chapter presents an overview of the history of wheelchairs, the knowledge gap that is surveyed within the scope of this review, and recent studies beneficial to the advancement of wheelchairs. The review of these works will serve as a guideline in the development of an autonomous powered wheelchair.

2.1 History of Wheelchairs

Powered wheelchairs are common in today's world to transport people with complex disabilities. Before discussing modern wheelchairs and their usage, it is important to know the history of wheelchairs. The wheelchair was initially introduced in the 6th century for King Phillip II (1595) of Spain, named "invalids chair", refer to Figure 2.1. This wheelchair was carved from stone, with armrests, a footrest, and a rolling mechanism. From the 6th century to the 18th-century, various designs for wheelchairs were available per users' preferences. Most of the wheelchair designs, at that time, were utilized by the wealthy as means of transportation. However, in 1655, Stephen Farfler designed and built the first three wheeled "self-propelled" wheelchair fit for individuals with disabilities, such as himself, to maneuver and transport themselves [6]. It was one of the first designs of a wheelchair that was considered to be "mechanical".

Yet, what is the definition of a “mechanical” wheelchair? It is a wheelchair that uses a form of chassis system rather than just a rolling mechanism. Farfler’s design is considered a “mechanical” design due to its three-wheeled chassis system with its use of a crank and cogwheel. However, the crank wheel was designed to move using a rotary handle on the front wheel (see Figure 2.2).



Figure 2.1: Wheelchair Designed for King Phillip II of Spain [7]



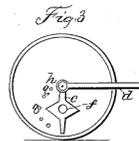
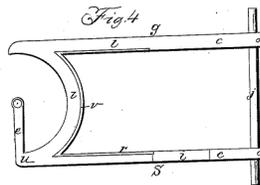
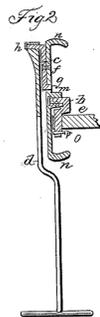
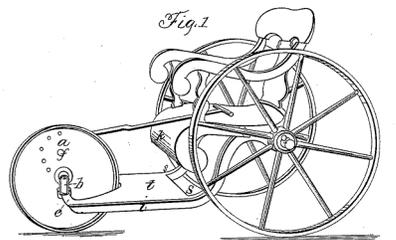
Figure 2.2: Wheelchair Designed by Stephen Farfler in 1655 [7]

The usage of wheelchairs increased during the late 17th century; however, these were primarily designed as means of transportation for the wealthy. The first wheelchair, that could truly be considered a mechanical design due to the similar design of modern wheelchairs, was built in the 18th century by John Dawson. In 1750, James Heath introduced a wheelchair design, which was commonly used as a rickshaw for the wealthy in Europe during the Victorian era [6]. However, in 1783 in Bath, England, Dawson designed a wheelchair similar to the design of James Heath. His design was considered the most widely used wheelchair until the beginning of the 19th century [8]. The wheelchair design consisted of a chair with two large wheels and one small one (see Figure 2.3). During the late 1800s, many improvements were implemented to wheelchairs. Some of the design modifications consisted of rear push wheels, small front casters, using rubber wheels on metal rims, and adding self-propulsion. One such known improvement was patented in 1853, under patent US9708A by Thomas Minniss known as the Invalid Carriage (see Figure 2.4).



Figure 2.3: Wheelchair design by James Dawson [7]

T. S. MINNISS.
 Invalid Carriage.
 No. 9,708. Patented May 10, 1853.



Inventor:
 Thomas S. Minnis

H. FETTER, PHOTO-LITHOGRAPHER, WASHINGTON, D. C.

Figure 2.4: First Patent of a Wheelchair in 1853 classified under Patent Number US9708A [9]

In 1916, the first mechanical wheelchair was made. This modification was a revolutionary change in wheelchairs since the design promoted a lighter and more maneuverable design. These wheelchairs were mainly self-powered, where they were maneuvered by the user by manually turning the wheels. However, if the individual was unable to maneuver themselves, an assistant or caretaker would be able to maneuver them by pushing the wheelchair from behind. Another modification that was revolutionary in the 20th century was the folding mechanism, which allowed for easier transport. The folding wheelchair was designed in 1932 by engineer Harry Jennings (see Figure 2.5). This wheelchair was designed by Jennings for his paraplegic friend Herbert Everest, which became the earliest wheelchair similar to what is used in today's modern society. This concept was such a success that they founded a company known as Everest & Jennings and patented the concept under patent number US2095411A, where the company monopolized the market of wheelchairs for years [7].

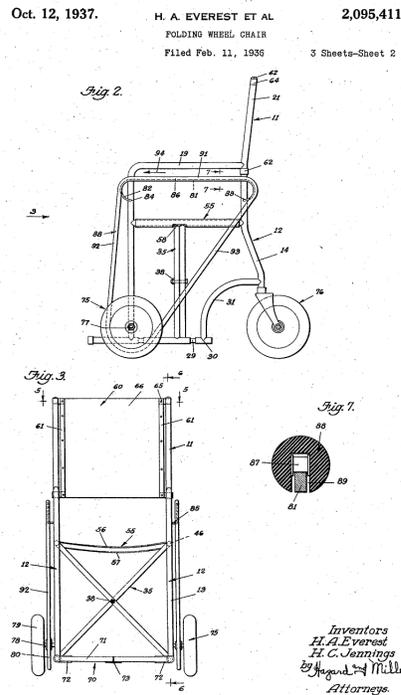


Figure 2.5: First Folding Wheelchair by Everest and Jennings US2095411A [10]

A motorized wheelchair, commonly known as an electric wheelchair, is driven by a user by using either a steering wheel mechanism or joystick. Electric wheelchairs reduce the user's energy required to travel compared to manual wheelchairs. In a manual wheelchair, the user has to use their upper body strength to maneuver themselves, which taxes the user and may induce muscular pain in their upper body. Not only that, there are various forms of disabilities that require mobility assistive devices. In these scenarios, each user has personalized requirements that limits them in using a manual wheelchair. The motorized wheelchair reduces the energy utilized by the user through various forms of power-based mechanisms [11]. One of the common mechanisms used in the construction of motorized wheelchairs are rim motors or a supplemental wheel. Rim motors generate assistive torque to the wheels when the user activates the motor. These are removable wheelchair motors that can turn any manual wheelchair into a motorized wheelchair [12]. The first design of an electric wheelchair was introduced during World War II by a Canadian inventor, George Klein and his team of engineers. This invention was funded by the National Research Council of Canada for a program to assist injured veterans in World War II [13] (see Figure 2.6). By early 1956, the invention developed by Klein and his team was mass produced by Everest & Jennings [7].



Figure 2.6: First Electric Powered Wheelchair [14]

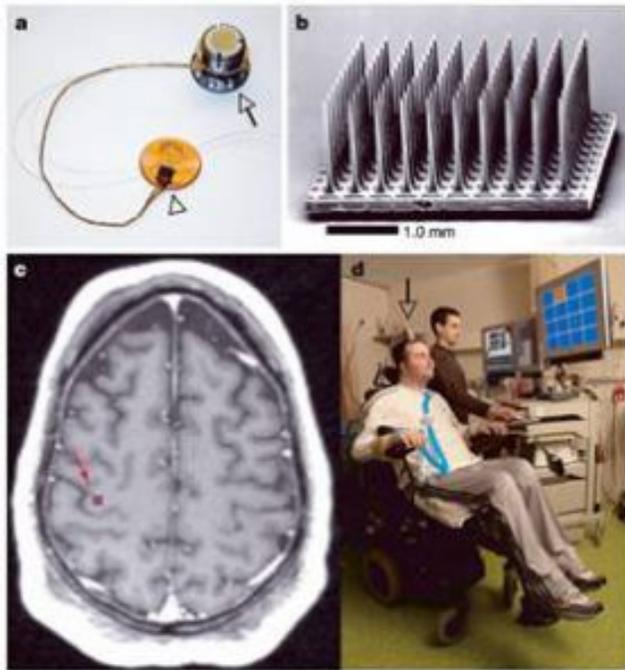


Figure 2.7: BrainGate Device [15]

There are various forms of research that have been conducted for power assisted wheelchairs. Some of the proposed designs consisted of a dynamic control model of rim motors using human input torque [16], a five-wheel wheelchair that uses an active-caster drive mechanism [17], and the most common implementation of using a joystick for maneuvering. The joystick model can be built in two ways, a four-wheel drive wheelchair or a three-wheel drive wheelchair. Unlike the four-wheel drive wheelchair, the three-wheel drive wheelchair is comparatively less stable since the grip forces between the ground and the wheels are smaller [11]. By the 21st century, a new phase of mobility came with the research of John Donoghue and Cyberkinetics Inc. under the research organization BrainGate. Similarly to this thesis, the duo invented a new wheelchair technology intended for a patient with limited mobility. The device invented by BrainGate is called the Brain Computer Interface (BCI). This is implanted within the brain of a patient and connected to a computer, which decodes the mental commands delivered by the patient such that any machine, including a wheelchair, can perform as per the user's choice [7], refer to Figure 2.7.

2.2 Knowledge Gap

Identified in Chapter 1, there are a number of challenges to overcome when designing a device that converts any powered wheelchair into an autonomous powered wheelchair. These challenges are the key areas of research that are discussed within this section. In this section, the state-of-the-art is reviewed to identify the gap between the current state-of-the-art and the objective for this research. An autonomous powered wheelchair is a motorized wheelchair with the removal of human input for maneuvering. This is not a highly researched topic, but one of the upcoming research topics in the world of assisted mobility devices. Autonomous wheelchairs are designed such that an individual inputs the command of a location and the wheelchair uses its

mechanism of localization and mapping, which aids in obstacle detection, to reach the destination inputted without collision or injury.

Various interfaces have been researched for users who cannot use commercially available wheelchairs. Some of the interfaces that have been proposed are a voice-based command, direction of the face, eye gaze, oral-motion, bio-signal, and electro-oculography (EOG) signal [18]. Similar research is presented by Murai [18] for a voice-activated wheelchair; however, it not only uses the command for maneuvering the wheelchair, but it also discusses the collision avoidance technique using sensors. In this research, the literature primarily reviewed analyzes Simultaneous Localization and Mapping (SLAM) based robotic wheelchair navigation and obstacle detection systems.

2.2.1 Simultaneous Localization and Mapping (SLAM)

SLAM is computationally generating a map of an unknown environment while simultaneously estimating the robot's position within the environment. This is applied in various fields, from self-driving cars, aerial vehicles, and domestic robots that are used for assistance to humans. One such robot is implemented as an autonomous powered wheelchair.

A study by Lankenau and Röfer discusses a self-localization navigation technique where the robot is tutored to adapt to various scenarios. After the robot has been tutored, it performs the navigation tasks in the environment. During its training process, the system builds the map of the environment that is then matched to the real world [19]. Another type of mapping they discussed is using a combination of topological and metrical maps. Topological frameworks only consider the distances between places, which then graphs the map of each object and arcs the path. A metrical framework is identified as 2-D object detection, these metrical maps are more precise for the coordinates of the object rather than distance while being noise sensitive. Using the combination of the two mapping frameworks allows the system

to get the precise result in coordinates and distance of the object.

Another type of SLAM based navigation system reviewed focuses on two main types of navigation: particle filter and Kalman filter. Particle filter-based SLAM is slow in response compared to Kalman filter [20]. Hence, Misono's use of Kalman filter-based algorithms for real world computation [20]. In this study, they utilized a laser rangefinder as the sensor for their SLAM algorithm. This was successfully implemented for the Intelligent Ground Vehicle Competition (IGVC) Navigation Challenge. Various studies have been conducted for SLAM-based wheelchair navigation. Some utilize only self-localization of the robot, while some combine SLAM with RGB-D sensors for obstacle detection and localization. Wu, et al. [21] combined SLAM with obstacle detection. It is highlighted that SLAM is as an essential tool for obstacle detection since the system detects the obstacle based on detection points. Yet, the accuracy of the obstacle is based on the accuracy of the robot's location and localizing is efficient if the mapping resolution and quality are achieved [21]. This is a critical study for autonomous wheelchairs since it combines the robot system and obstacle detection sensors. However, the difference in implementation would be that Wu's study consisted of tracking humans as obstacles while in this research the obstacles are varied objects.

2.2.2 Obstacle Detection

In any field, safety is the primary focus for the participating individuals and for those within the surroundings. Similarly, the powered wheelchair is efficient if it is safe for the users as well as the people that are in its operating surroundings. In order to ensure that the user and bystanders are safe, obstacle detection is utilized. Obstacle detection is used to detect ditches of various height, bystanders in the path of the user, pillars, and blockages along the path, as well as the height and clearance from stairways. Once obstacle detection has occurred, a message is relayed back to the

controller to avoid the obstacle. Thereafter, the device reroutes the path to avoid any obstacles. This is only effective if the detection is done in real-time. If the detection is not produced in real-time it will have a delayed response, which can cause accidents that are harmful to the user and/or bystanders.

Research has been done regarding the use of real-time embedded control systems for various devices and sensors to avoid obstacles. The most widely researched device is the stereoscopic camera, which allows the device to locate obstacles through vision detection. Another type of sensor that is used are Light Detection and Ranging (LiDAR) sensors. LiDAR sensors measure distance by illuminating a target with a laser light. In robotics, these sensors are widely used for military purposes rather than in the medicine field. However, these sensors are not restricted to robotics and are used for agriculture, archaeology, mining, and other fields. Thus, these types of sensors are analyzed for the purpose of obstacle detection for an optimized design of a powered wheelchair.

One of the preliminary studies for obstacle detection was presented by Borenstein and Koren in 1989 known as Virtual Force Field (VFF) method [22]. In this method, the robot and obstacle apply virtual forces in a counteracting manner. This method was modified in 1990 by the same authors, which is known as Vector Field Histogram (VFH) [23]. This method was implemented on robots that are passing through narrow corridors or clustered environments such that collision is avoided. To modernize this study, Fattouh and Nader presented a model in 2006, which integrated the powered wheelchair such that it would function in any desired environment with the desired algorithm in virtual reality [24].

2.2.2.1 Stereoscopic Camera

The stereoscopic camera sensor is a vision-based obstacle detection sensor. This sensor can be based on various systems. Some of the systems that were analyzed in the

literature were RGB-D sensors, Point Grey Research (PGR) program, Vector Field Histogram (VFH), and Digital Evaluation Map (DEM).

RGB-D sensors are utilized for colour and depth information for each pixel. The accuracy of the depth data provided by RGB-D sensors decreases as the distance from the sensor gets larger. This accuracy issue can be problematic for the mapping applications that will be required for this project. In a paper by Jafari, et al. [25], similar research was performed. RGB-D sensors were utilized in association with stereo applications for less complicated and faster detection processes. This robot contained a spatial operation radius of the tracking system of up to 15 m, which is suitable for mobile robots [25]. Another sensor type that was analyzed used PGR software for distance detection from the stereoscopic images acquired by a Bumblebee camera [26]. This software uses colder colours, such as blue, for the detection of offset values, and hotter colours, such as red, for objects that are extremely close. However, this resulted in a maximum error of ± 8 cm for distances closer than 3 m. It was effectively implemented by Nguyen, et al. [26] on powered wheelchairs for distance object detection.

For this research, various software programs were analyzed that focused and used stereoscopic camera sensors. The only variable was the type of algorithm used to create the mapping for the colour and depth analysis of the obstacles. Bernini, et al. discusses four types of algorithms: probabilistic occupancy map, digital elevation map, scene flow segmentation, and geometry based clusters [27]. The probabilistic occupancy map algorithm uses stereo sensors where depth data is measured in a 2D occupancy grid, where grey cells are unknown occupancy, white cells are free, and black cells are occupied [28]. To successfully analyze the obstacle, the authors presents three occupancy grids. Cartesian grid represents the world, disparity grid relates to the discretized values of the image coordinates, and the polar grid discretized values of image coordinates and the depth in the world coordinates [28]. Bernini, et al.

use Badino’s research topic to base their Digital Evaluation Map (DEM). Bernini’s approach consists of height based representation into a map similar to an occupancy grid [27]. This approach can be used with 2D sensors, such as, stereoscopic sensors, or 3D sensors, such as, LiDAR or radars. This approach has been analyzed for autonomous ground vehicles; therefore, it can also be implemented for autonomous wheelchairs.

2.2.2.2 LiDAR Sensor

LiDAR sensors were originally used as the technology to make high-resolution maps for forestry, archaeology, and military purposes. However, current research focuses on their use in automotive applications. The research performed by Shuqing Zeng [29] utilized LiDAR sensors to detect arbitrary obstacles and output lane change alerts (LCA) for a car. This test is applicable for powered wheelchair as it tested performance that uses depth analysis with system implementation of 360° field-of-view coverage, where LCA is used for the precise and robust performance of the wheelchair.

2.2.2.3 Safety Standards

Safety is one of the crucial requirements for a successful robot. For the wheelchair to function efficiently and safely, the validation of each field requires compliance with general safety standards. Some of the critical standards that ensure the safety of users are IEC 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety - Related Systems, WC19 - Wheelchair Transportation Safety Standard, and ISO 7176 - Assistive Products for Person with Disability (Wheelchairs). These safety standards are used as guidelines for the architecture of SLAM and real-time obstacle detection [4].

By reviewing software safety concerns, specifically the interference of sensors, identifies the priority sequence for the obstacle detector, the sensor’s ability to talk to the core

processor, and the strength of the obstacle detector. It is essential for the sensors to provide precise data of the surroundings, as well as for the system to identify the most critical obstacles to avoid. If this is not processed efficiently due to improper logic in the system, it could prove dangerous to the user; thus, a data logger is utilized. Another mechanism that is applied is priority sequence analysis for critical obstacle detection. This requires an implementation of identifying the closest obstacle as a critical obstacle while registering the next real-time obstacle in the sequence of proximity [4].

Mechanical safety concerns are based on a few basic components. For obstacle detection to be mechanically safe, it requires static stability, efficient braking of the wheelchair, an optimal turning radius for narrow corridors, and speed control. Static stability of the wheelchair determines the tipping angle of the wheelchair with and without locked brakes. This is important because the stability will determine the level of injury to the user if the sensors fail to detect obstacles efficiently. With higher static stability, the risk of injury reduces. The efficiency of braking is critical for the user in situations where the obstacle is detected shortly prior to the possible collision. To ensure braking is efficient, speed control is necessary. If the obstacle is detected in the path of the wheelchair, to avoid collision, a speed controller is used to reduce the speed for braking [4].

For the transportation of wheelchairs, the guideline that is utilized is WC19 - Wheelchair Transportation Safety Standards. This allows the user to be safe in a motor vehicle and use the wheelchair as a seat in the vehicle. For this, the wheelchair has to follow the requirement of WC19: “having at least four permanently labeled securement points that can withstand the forces of a 30 mph, 20 g impact, have specific securement point geometry that can receive a securement end fitting hook of a specified maximum dimension, be equipped with anchor points for a wheelchair-anchored pelvic belt and recommendations for purchasing a belt if not provided, such that the wheelchair and

pelvic belt will withstand a 30 mph, 20 g impact, and provide a standard interface on the pelvic belt to connect to a vehicle-anchored shoulder belt” [30].

2.2.2.4 Data Logger

For the wheelchair to relay correct output, it requires data to be analyzed and organized for optimal obstacle detection and maneuverability. The primary step to evaluate the usability and maneuverability is to collect data for the wheelchair from a real environment. A study by Komoto and Suzurikawa [31], generated a feasibility test to log the everyday usage of a wheelchair with a smartphone-based electronic recording equipment. This data logger combines a smartphone and a versatile A/D (analog to digital) converter to collect, transfer, and store various data, such as, acceleration and angular velocities of the wheelchair, GPS position, and the joystick inputs [31]. Another type of data logger that is reviewed is a motion-logger. This system collects motion information utilizing an Inertial Measurement Unit (IMU). This unit captures data in a secure digital (SD) memory card [32]. The study is conducted by Marquez, et al. [32] and successfully provides information about the motion and the attitude of the wheelchair for determining risky situations. The motion logger was also utilized to obtain and store battery conditions and temperature data for analysis. Lastly, the data logging platform utilized by Pineau, et al. [33] is an essential asset to the study of an autonomous powered wheelchair since they analyzed and recorded the 3D acceleration data of a wheelchair in real-time. However, this is implemented for electric powered wheelchairs (eg. joystick based) and was conducted using four different time-series features. The study by Pineau, et al. [33] was successful with a 98% accurate detection for unsafe events, and a 12% false positive rate. Even though none of the data loggers are utilized for an autonomous powered wheelchair, it can still be implemented [33].

2.3 Related State-of-the-Art Contributions

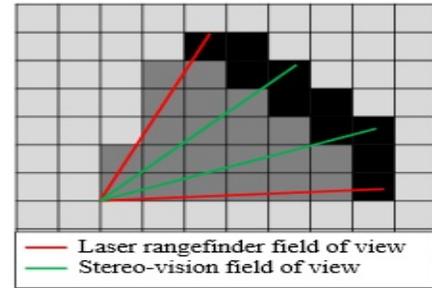
In recent years there have been major developments for autonomous mobile robots where various researchers have tried to bridge the knowledge gap identified in Section 2.2. However, the application and approach vary depending on the required final outcome. This section is divided into two topics: the first topic will discuss the state-of-the-art similar to the scope of this thesis for an Unmanned Ground Vehicles (UGV), while the second topic will discuss various navigation methods proposed by researchers to develop autonomous navigation for electric powered wheelchair.

2.3.1 Autonomous Navigation for Unmanned Ground Vehicles

Unmanned Ground Vehicles (UGV) are a widely researched topic in today's world. One such research is proposed by Hussein, et al., where their research targets the autonomous navigation of an off-road vehicle using stereo-vision and laser rangefinder for outdoor obstacle detection [34]. In this paper, Hussein, et al. are using an electric golf cart which is controlled in an ROS environment for an Intelligent Campus Automobile (iCab) project. The propose of the paper is to perform sensor fusion using a binocular camera and a laser rangerfinder for obstacle detection. The fused data is used to construct an occupancy grid map where each cell has a specific value that correlates to the level of obstacle detection (see Figure 2.8(a) and Figure 2.8(b)).

OBSTACLE CONFIDENCE VALUES			
Cell Category	Free Cell	Occupied Cell	Unknown Cell
Cell Value	100	0	150
Cell Color	Dark Gray	Black	Light Gray

(a) Obstacle Cell Occupancy Value



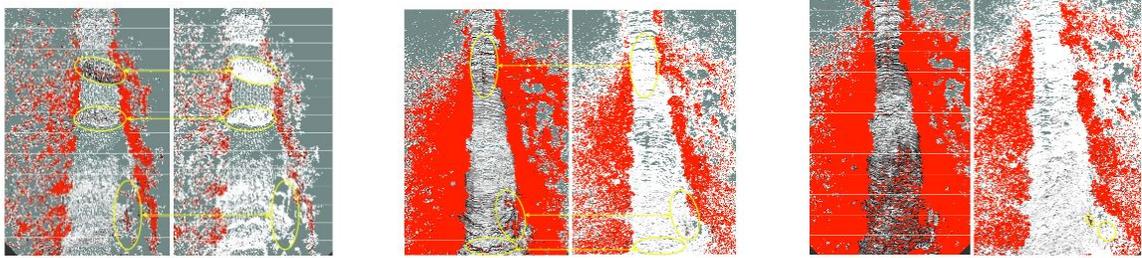
(b) Occupancy Grid Map Representation with Vehicle Field of View

Figure 2.8: Occupancy Grid Map with Fused Data [34]

Similar to Hussein, et al., Wang, et al. performed SLAM and obstacle detection using a laser rangefinder, however, the research extended to the detection and tracking of moving objects [35]. Wang, et al. used customized robot cars and trucks capable of autonomous driving or driver assistance. In this paper, they discuss a SLAM algorithm that combines maps for stationary objects and moving objects. Wang, et al. built an algorithm that acquires laser scan data from a rotating laser rangefinder. This approach is used in crowded environments and was successful in generating a map suitable for a robot to drive in outdoor conditions. The errors were corrected using the rotating rate of the scanning device and the velocity of the robot. The SLAM model presented by Wang, et al. can be modified and applied in ROS for indoor autonomous wheelchair driving.

Additional research that can be adapted for autonomous wheelchair driving are by Broggi, et al. [36], Jasper and Wuensche [37], and Chen, et al. [38]. All of these papers discuss autonomous navigation for off-road vehicles. Broggi, et al. and Jasper and Wuensche both discuss the use of stereo vision for B-Spline surface estimation. Both discuss the use of 3D point clouds to generate 2.5D occupancy grid map where a B-Spline fitting algorithm is applied to perform obstacle detection and slope estimation. A 2.5D occupancy grid stores in each cell of a discrete grid the height of objects above

the ground level at the corresponding point of the environment [39]. However, their approaches vary by their application of noise reduction. The algorithm presented by Broggi, et al. uses a Kalman filter to remove the noise acquired by the stereo vision [36], while the algorithm presented by Jasper and Wuensche utilizes the Gaussian model for noise reduction [37]. The paper presented by Chen, et al. also discusses mapping algorithm for off-road autonomous vehicles. Chen, et al., uses a mixture of Kalman filtering and Gaussian noise reduction utilizing LiDAR. They discuss different Relative Probabilistic Mapping (RPM) achieved by the use of 2D LiDAR, 3D LiDAR and multiple LiDARs [38]. As observed in Figure 2.9, the increase in degree-of-freedom for the sensors proves beneficial to identify various sized obstacles.



(a) 2D LiDAR Measurements, left: non-RPM Map; right: RPM Map
 (b) 3D LiDAR Measurements, left: non-RPM Map; right: RPM Map
 (c) Multiple LiDAR Measurements, left: non-RPM Map; right: RPM Map

Figure 2.9: Traversability Map Built Using RPM Algorithm for Three Different LiDAR Applications [38]

2.3.2 Autonomous Navigation for Electric Powered Wheelchairs

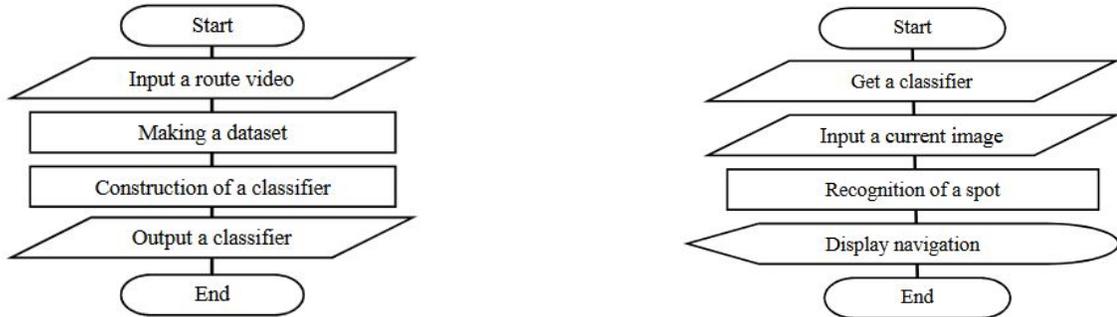
Lately, the increase in electric wheelchair designs have reflected on the increase in research towards the impact of a disability to the individual and their caregivers lives. The need for autonomous wheelchairs has increased and to fill the gap, many researchers have proposed various methods to automate an electric powered wheelchair. Research proposed by Maatoug, et al. uses fuzzy logic controller to autonomously

control a powered wheelchair. For autonomous wheelchair navigation Maatoug, et al. developed the Zero-order model using Sugeno fuzzy logic controller. This model performs navigation in intermediate stages, which considers that going from one room to another involves various types of obstacles [40]. In order to perform validation of the developed model, Maatoug, et al. used a unicycle kinematic model to simulate the wheelchair in MATLAB. They simulated three ultrasonic sensors in the kinematic model to ensure that the simulation mimics real environment situation. The ultrasonic sensor will detect the obstacle to activate a fuzzy inference algorithm to avoid the obstacle. However, this method is over simplified and not ideal for real life implementation [40].

Research proposed by Zhang, et al. discusses automating an electric wheelchair by combining Brain-Computer Interface (BCI) with automated navigation [41]. In this paper, Zhang, et al. primarily proposes a concept of controlling a wheelchair using brain signals; however, due to electroencephalogram (EEG) signals being noisy and unstable, it was proposed to combine the brain signals with automated navigation techniques. In order to achieve the proposed theory, Zhang, et al. modified a commercially available mid-wheel drive wheelchair by adding laser range-finder and an array of three ultrasonic sensors. These sensors are used to generate a map for obstacles which is utilized by BCI that combines motor imagery (MI). In this system the user selects a destination via BCI and the wheelchair navigates towards the destination without any control input by the user. For this system to navigate autonomously, the system uses two webcams to localize obstacles [41]. The research proposed by Zhang, et al. is ideal for structured environments, yet it is not suited for an environment with moving obstacles.

Few researchers have used the technique of image processing to develop an autonomous navigation system. Nakayama, et al. developed a navigation method using image processing by applying deep learning techniques [42]. Nakayama, et al. divides the

process into two sections: learning process and navigation process. In learning process, the model is trained using video input of a route (see Figure 2.10(a)). The trained model classifies the input image into a divided and labeled dataset which is used by navigation process (see Figure 2.10(b)). In order to construct the classifiers assigned in the learning process, Nakayama, et al. use Convolutional Neural Network (CNN). For this learning section, route teaching was carried out by using a webcam under different weather conditions. Similar to Nakayama, et al., Lee, et al. also used image processing to develop an obstacle avoidance and navigation system for autonomous wheelchair applications. Lee, et al. uses Canny Edge detection and Erosion Noise Filtering for obstacle detection using a webcam. As for the navigation system, Lee, et al. use two compass modules as an input to the system. The combined system is ideal for structured environments and where ramps and stairs are not considered as obstacles [43].



(a) Learning Process

(b) Navigation Process

Figure 2.10: Occupancy Grid Map with Fused Data [42]

ROS has been used in various mobile robot platforms, however, there has been limited use of ROS for an electric powered wheelchair. There have been many research projects proposed for navigation systems or obstacle detection. One such research project developed an intelligent wheelchair called ATEKS by Akar, et al. [44]. ATEKS is designed using two controllers, intelligent controlling unit high (AKBH) and intel-

lignant controlling unit low (AKBL), 10 ultrasonic sensors, a Microsoft XBOX 360 Kinect sensor, encoders, a joystick, and an indoor positioning system (İÇKON) to be compatible with ROS [45]. The system detects obstacles using ultrasonic sensors, and localizes itself using İÇKON. The İÇKON system was developed by Yeniçeri, et al. It uses ultrasonic signals to calculate the position to an accuracy of 0.01 meters. Yeniçeri, et al. developed an automatic transmitter position calculating system using the trilateration method. It is a surveying method which measures length of the side of the triangle electronically to compute the angles [46], based on time of flight technique [45]. However, research proposed by Yayan, et al. suggests indoor navigation software ATEKS developed by Akar, et al. [44]. The proposed software called User Interface Unit (KAB) is a mobile application designed for Android and iOS platforms to drive ATEKS [47]. KAB uses position information from AKBH and AKBL using WiFi and converts the XML data saved on an SD card into a global coordinate frame. Once communication is established between KAB and ATEKS, the predefined map is updated for the location of the wheelchair. Once the location is identified, the A* algorithm is used to detect the shortest path to navigate indoors to the desired location [47]. Similar to Yayan, et al., Li, et al. uses the A* algorithm to detect the shortest path for navigation. However, it combines the A* algorithm to Gmapping and the Adaptive Monte Carlo Localization (AMCL) algorithm. Li, et al. uses an Arduino microcontroller to control a wheelchair's motors, along with an RGB-D camera to produce a depth point cloud. The depth point cloud is converted to 2D laser scan data to generate 2D map of an environment using Gmapping algorithm. This system gets combined with AMCL and the A* algorithm to localize the wheelchair and perform navigation, which is activated using the Android App as a remote control for the wheelchair. Both the proposed methods are ideal for level floors with static and dynamic obstacles, however, neither of the two proposed methods discusses about identifying edges and ramps as obstacles.

Another ROS-based indoor navigation system for wheelchairs was developed by Grewal, et al. [48] [49]. They do not present a user interface unit, but present the initial stages of converting powered wheelchairs to autonomous wheelchairs. The first paper proposed by Grewal, et al, uses a 2D LiDAR to generate an obstacle map. Similar to Li, et al., Grewal, et al. uses Gmapping and the AMCL algorithm for SLAM navigation; however, the LiDAR sensor is used instead of an RGB-D camera [48]. In the second paper proposed by Grewal, et al., the powered wheelchair carries three sensors, two LiDAR sensors and one camera instead of just one LiDAR sensor. The proposed system combines machine learning and computer vision to perform navigation for unmapped indoor environments [49]. Grewal, et al. uses similar techniques as Nakayama, et al. where a trained data set is used to detect store fronts for the trials performed. Therefore, the system proposed by Grewal, et al. is not ideal for cases where dynamic traffic is involved since it is suited for static obstacles only in ideal conditions.

2.4 Summary

There are many different types of wheelchairs that have been introduced over the years. However, little research has been performed to develop fully autonomous electric powered wheelchairs. For a wheelchair to be fully autonomous, it needs to navigate the environment within a defined location while performing SLAM without the user's support for navigation. Maatoug, et al. [40] and Zhang, et al. [41] investigated for autonomous electric powered wheelchairs. The study by Maatoug, et al. is a theoretical model that has not been validated with a physical prototype, while Zhange, et al. utilizes Brain-Computer Interface for navigation assistance. While the research proposed by Grewal, et al. [49] is similar to the proposed topic in this research, the system is not acceptable as it requires a machine learning process prior to autonomous navigation.

The system proposed by Grewal, et al. is not ideal for randomized encounters. Due to this, the literature on UGVs was reviewed to identify possible SLAM and obstacle detection algorithms that could be used to develop an add-on system that is multi-level floor traversable. Through analysis, it was deemed that to perform SLAM in a multi-level environment, obstacle detection is also required, such that, the system can perform 2D and 3D mapping of an environment and autonomously navigate. Most of the autonomous navigation research for electric powered wheelchairs and UGVs were limited to terrain environments or single-level floors. Therefore, the gap in knowledge was largely identified by the lack of research found in the autonomous navigation for electric powered wheelchairs in multi-level environments, especially where ramps and stairs are considered as obstacles.

Chapter 3

Wheelchair Modifications

This chapter presents the design requirements identified in Section 1.5 and discusses design considerations to modify the powered wheelchair to an autonomous wheelchair. The wheelchair used for the prototype is a QUICKIE Xperience 2, donated by Sunrise Medical Canada Inc. (see Figure 3.1). As mentioned in Section 1.3, the scope of this research is to develop an add-on device to turn a conventional powered wheelchair into an autonomous wheelchair. An add-on device has to be designed such that it is compatible with any wheelchair as an after market product with minor modifications. One may ask what constitutes minor modifications? Minor modifications means minimal amount of physical changes made without losing the integrity of the object. The developed device consists of an interface machine and autonomous navigation system. For the purpose of this thesis, the interface system is replaced with a laptop which will harbour the processing of the control system and temporary user interface. To develop the prototype, the first step is to identify which components are compatible with the objective of this thesis.



Figure 3.1: QUICKIE Xperience 2 Wheelchair by Sunrise Medical Inc. [5]

For the wheelchair to be modified to an autonomous wheelchair, controller compatibility needs to be identified. As mentioned in Section 1.5, the key criteria for an add-on device is for the control system to be ROS compatible. The control system equipped with the donated wheelchair consisted of a R-Net Power Module 120AMP and a R-Net Encoder Module (refer to Figure 3.2 and Figure 3.3). Using R-Net OBP and PC Programmer (see Figure 3.4) to have encoder module work with ROS, it is concluded that the controller is not programmable. An alternative option to reprogramming the controller through the provided programmer is to directly tap into the Input Capture Unit (ICU) of the module using a microcontroller. This technique is used to identify the IP address where the controller commands are being communicated on, such that the controller can be reprogrammed. While doing so, it was identified that the control system uses various IP addresses to communicate. As such, it is difficult to differentiate between the joystick and encoder data input. Through an elimination process, it was determined that the control system provided with the wheelchair is not adaptable with ROS.

Once the controller compatibility was identified, the next step was to ensure joystick

connectivity with ROS. While doing so, it was determined that the joystick provided with the wheelchair is only compatible with R-Net Power Module, since the communication to the joystick is limited and controlled by the bus on the ICU. However, the wheelchair batteries can only be charged using the provided joystick connection. Other options are not acceptable, due to this, it was necessary that all the connections equipped with the wheelchair stay as is.



Figure 3.2: R-Net Power Module 120AMP [5]



Figure 3.3: R-Net Encoder Module [5]



Figure 3.4: R-Net OBP and PC Programmer [5]

Through preliminary testing, it was concluded that the control system had to be modified to implement an autonomous wheelchair. This chapter outlines the process of the modifications to the control systems and the modifications done to the wheelchair.

3.1 Control System

Before designing a control system that is compatible with ROS, it is vital to identify the components of the provided wheelchair that require modification. Table 3.1 identifies all the components of the wheelchair that are related to the control systems. There are four components from the original control systems that need to be modified.

Table 3.1: List of Components for Conventional Control Systems

#	Components	Technical Specifications	Modification (Yes/No)
1	Motors (Left and Right)	<ul style="list-style-type: none"> • 24V Brushed Motor - 4 Pole • In-built encoders 	No
2	R-Net Power Module 120AMP	<ul style="list-style-type: none"> • Max Current - 120 Amp • Advanced and precise drive control • Mirrors all system programming • Two universal inhibit inputs 	Yes
3	R-Net Encoder Module	<ul style="list-style-type: none"> • Motor Encoder Interface • Two motor encoder inputs • Reverts to conventional control if encoder faults are detected 	Yes
4	R-Net Bus Cable, MALE-MALE	<ul style="list-style-type: none"> • Length - 0.5m • Connects R-Net Encoder Module to R-Net Power Module 	Yes
5	R-Net Colour Joystick with 700mm Cable	<ul style="list-style-type: none"> • Two jack sockets as standard • On-Board Programming (OBP) option • Charging socket 	Yes
6	Cable for External Switch on Encoder Module	<ul style="list-style-type: none"> • Connects R-Net Encoder Module to the Motor Encoders 	No
7	GP24 Battery Harness and Cover Kit for Compact Frame	<ul style="list-style-type: none"> • Battery Connector in Parallel and Safety Kit 	No
8	GP24 Battery (2 Batteries)	<ul style="list-style-type: none"> • Max Voltage - 24V • Max Current - 140 AMP 	No

As seen in Table 3.1, components 2 - 5 need to be modified as they are not compatible with ROS. To ensure that these items are modified efficiently, the new controller has to comply with basic requirements and connectors of the motor and battery equipped with the wheelchair. Given these conditions, a key factor being compatibility with ROS, there are few motor controllers that can be applied for the modified control system.

One of the options explored to modify the controller was to use a Arduino micro-controller. The Arduino is capable of communicating with ROS; however, given the technical requirements it will not suffice. The Arduino is not capable of carrying such heavy load of current efficiently. Other than the Arduino, the RoboteQ Motor Controller is able to communicate with ROS through various means (CAN, RS232, and USB). RoboteQ makes a variety of motor controllers. Given the technical requirements, their industrial motor controller XDC2430 was most ideal for this application (see Figure 3.5). The maximum voltage for the motor controller is 30 V, maximum allowable current is 150 A per motor, and it also has the ability to connect to motor encoders through auxiliary connection. Refer to Table 3.2 for detailed technical specifications.



Figure 3.5: RoboteQ XDC2430 Brushed DC Motor Controller [50]

Table 3.2: RoboteQ Motor Controller Technical Specifications

Components	Technical Specifications
RoboteQ XDC2430 (See Figure 3.5)	<ul style="list-style-type: none"> • Max Current - 150 Amp • Max Voltage - 30V • 0-5V Analog Input for Encoder • Closed loop position control with analog or pulse/frequency feedback • PID control loop with separate gains for each channel • ROS Compatible

3.1.1 Circuit Design

The motor controller is an essential component in the autonomous navigation system. It is used to control the movement of the motors utilizing Pulse Width Modulation (PWM). The motor controller requires an external power source, which is provided by the battery itself. However, since the wheelchair is powered by a DC power source, connecting the motor controller straight to the battery is not ideal. The inner workings of the motor controller consists of non-linear electric devices, such as, inductors, capacitors and transistors. The utilization of these non-linear devices causes transients upon the connection of the battery to the motor controller. As a result of these transients, the current and voltage exhibit a higher magnitude of both metrics than the rated values. This creates a very high current where the motor controller can be damaged. To prevent damage by surcharge, a discharge circuit was required (see Figure 3.6).

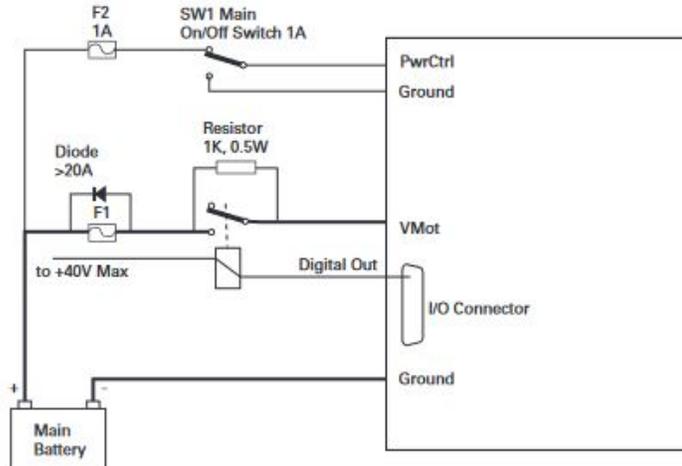


Figure 3.6: RoboteQ XDC2430 Discharge Circuit [50]

A discharge circuit was implemented in order to counter the effects of the current which resulted from the transients. The positive and negative terminals of the motor controller essentially act as a big capacitance, which results in the transients upon battery connection. Once the contactor for the battery is closed, there are hundreds of amps that flow through the motor controller; thereby, permanently damaging the controller. The discharge circuit consists of a pre-charge resistor and two switches. Initially, the pre-charge resistor is connected to the battery to absorb the transient, and once the current has stabilized, the second switch is opened to connect the motor controller. The diagram seen in Figure 3.6 identifies the pre-charge resistor as a current limiting device, it is capable of dissipating lower power and acts as the charge controller. Industry standards require charging up to 90% of the battery operating voltage, this is the shortest amount of time required to hold the pre-charge closed before closing the motor controller contact. The transients will eventually stabilize to steady state and, at this point, the motor controller contact can be closed.

3.1.2 Kinematic Model

After the circuit design for the motor controller, the next step in designing the control system is to program the controller to drive the wheelchair. Prior to programming the control system for the wheelchair, it was important to understand the kinematic model of the wheelchair. The QUICKIE Xperience 2 is a mid-wheel drive wheelchair, which translates to a differential drive kinematic model. The motor controller (RoboteQ XDC2430) is designed to control left and right motors individually. This is ideal for designing a differential drive control system for the wheelchair.

3.1.2.1 Differential Drive

Differential drive is a popular kinematic model for various mobile robots. Differential drive robots consist of two wheels controlled by separate motors whose axes are collinear (see Figure 3.7). As mentioned before, the powered wheelchair is a mid-wheel drive wheelchair. Due to this, the wheelchair can be modelled with a differential drive kinematic model [51].

where ω is the angular velocity of the wheels, R is the distance from the ICC to the midpoint between the wheels, (x,y) is the centre point of the axis of the wheels, and V_r and V_l are right and left linear velocity, respectively,

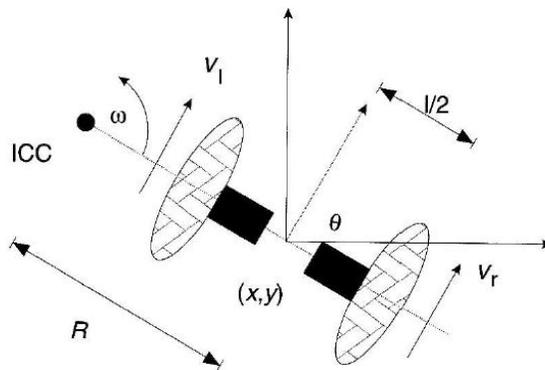


Figure 3.7: Differential Drive Model [51]

The wheelchair contains four castor wheels, two in the front and two at the back for the support. Defining velocity of each wheel is crucial, as a small difference in wheel velocities can change the trajectory of the wheelchair. The wheelchair has to perform rolling motion, the wheels rotate about the common axis of the left and right wheels. The point they rotate about is known as the Instantaneous Center of Curvature (ICC) (see Figure 3.7). The trajectory of the wheelchair is dependant on the velocity of the wheels. Since both the wheels rotate at the ICC, the angular velocity of the wheelchair can be written as [51]:

$$\omega(R + l/2) = v_r \quad (3.1)$$

$$\omega(R + l/2) = v_l \quad (3.2)$$

ω = angular velocity of the robot

R = distance from the ICC to the midpoint between the wheels

v_r = velocity of the right wheel

v_l is the velocity of the left wheel, and l is the distance between the wheels.

At any given time, the distance from the ICC to the midpoint of the wheels and the angular velocity of the wheels can be identified using:

$$R = \frac{l (v_l + v_r)}{2 v_r + v_l}; \quad (3.3)$$

$$\omega = \frac{v_r - v_l}{l} \quad (3.4)$$

Differential drive can be presented using three cases:

1. If $v_r = v_l$, then $R = \infty$, the wheelchair will move either forward or backward in straight line.
2. If $v_l = -v_r$, then $R = 0$, the wheelchair rotates in place (rotation at the midpoint

of wheel axes) left or right.

3. If $v_l \neq v_r$, the wheelchair will follow a curved trajectory about a point a distance R from the centre (it changes not only position, but also the orientation).

The structure of the kinematics allows the wheelchair to turn on the spot, either left or right, which is ideal system for narrow environments. However, it restricts the wheelchair from moving in the direction of the common axis of the wheels.

There are two types of kinematic models for differential drive robots: forward kinematics and inverse kinematics. In a forward kinematic model, the control parameters of the robot are known, while the final destination (also known as the goal pose) is undefined. The given control parameters will define the trajectory and the final pose of the wheelchair. If the robot has a pose (x, y, θ) at any given time t , and the velocities v_r and v_l of the robot are known, during the period $t \rightarrow t + \delta t$, then the ICC is given by [51]:

$$ICC = (x - R\sin(\theta), y + R\cos(\theta)), \quad (3.5)$$

where θ is the heading angle and at time $t + \delta t$ the pose of the robot is given by [51]:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix} \quad (3.6)$$

Integrating Equation (3.6) with initial conditions (x_0, y_0, θ_0) , at any time t , the pose of the robot can be identified as [51]:

$$\begin{aligned}
x(t) &= \int_0^t V(t) \cos(\theta(t)) dt \\
y(t) &= \int_0^t V(t) \sin(\theta(t)) dt \\
\theta(t) &= \int_0^t \omega(t) dt
\end{aligned} \tag{3.7}$$

where $V(t)$ is the velocity of the robot.

The forward kinematic model is ideal to utilize for simple trajectory cases, where $v_r = v_l$ or $v_r = -v_l$, such that $V(t)$ is the linear velocity of the wheelchair system. However, for the third case, where $v_r \neq v_l$, the linear velocity of the wheelchair is $V = v_r + v_l$. This is a special case, where the velocity constraints cannot be integrated to positional constraints [51]. This constraint is identified as a non-holonomic constraint. This is when v_r and v_l are in the same direction, but varied such that R is greater than $l/2$. This special case is defined by [51]:

$$\begin{aligned}
x(t) &= \frac{1}{2} \int_0^t (v_r(t) + v_l(t)) \cos(\theta(t)) dt \\
y(t) &= \frac{1}{2} \int_0^t (v_r(t) + v_l(t)) \sin(\theta(t)) dt \\
\theta(t) &= \frac{1}{l} \int_0^t (v_r(t) + v_l(t)) dt
\end{aligned} \tag{3.8}$$

Using a forward kinematic differential drive model for the mobile robot for the special case produces an infinite number of solutions for the combination of v_r and v_l , such that $R = l/2$. Using the forward kinematic for this situation can be complicated.

However, if a global pose or trajectory is specified, the robot inverse kinematic model can be used.

3.1.2.2 Implementation of Kinematic Model

Adopting the differential drive model for the control system of the wheelchair means controlling the speed of each wheel individually in ROS. The RoboteQ XDC2430 controller is programmed for the ROS environment utilizing the `roboteq` package, presented by Clearpath Robotics Inc. It is developed to control various RoboteQ controllers, due to this, the package has to be modified for different controllers to be efficient. The package uses basic controller implementation that allows the motors and motor controller to communicate in the ROS environment. However, this package is not designed to implement a differential drive kinematic model. Therefore, the package was modified, where Equations (3.1) and (3.2) are utilized to set the individual wheel speeds. The modified controller package is programmed for two conditions, open-loop system and closed-loop system. If the system is open-loop, it uses motor power to be equated for each wheel, as no encoder data is utilized for the feedback of the wheel count. While if the system is closed-loop, the motor speeds are equated for each wheel, using the encoder data. The programmed wheelchair was tested and it was observed to adopt the third case where $v_r \neq v_l$. Even though the code is adopted to follow the case where $v_r = v_l$ for differential drive, the wheelchair was outputting different motor speeds for the left and right wheels. This is due to an error in the feedback for the closed-loop system. To ensure that the motor speed for the left and right wheels are the same, the PID controller for each wheel was tuned.

Once the PID was implemented, the wheelchair followed a straight line while moving forward, instead of mimicking the special case scenario of the motion. The control system of the wheelchair was not only programmed to drive in a straight line, it was also programmed to drive using keyboard teleoperation commands. This allows the

wheelchair to be connected to ROS with a laptop as the user interface.

3.2 Add-on Device Design Prototype

Modifying the control system of the wheelchair was crucial to ensure that the wheelchair was compatible with ROS and can be controlled via a laptop. As defined in Section 1.3, the wheelchair has to navigate autonomously avoiding static and dynamic objects. Visual sensors, like cameras, aid to recognize static and dynamic objects whilst preventing the wheelchair colliding with obstacles. Apart from that, the wheelchair also has to detect slopes and stairs, as mentioned in Section 1.5. For that purpose, the orientation of the wheelchair with respect to the normal plane (ground) has to be identified. Hence, a sensor that publishes information regarding angles and rate of rotation of the wheelchair with respect to the ground was required.

3.2.1 Types of Sensors

Sensors are classified into two main categories: internal-state sensors and external-state sensors. Internal-state sensors are ideal for feedback of a robotic system, such as wheel positions, battery level, or rotation angles (yaw, pitch, and roll) [52]. External-state sensors are utilized to get feedback from the environment the system is operating in. They can be contact sensors, such as bumpers, and non-contact sensors, such as cameras [52].

One of the key requirements of the navigation system is to detect slopes and stairs during navigation; this can be obtained using internal-state sensors. The motors are equipped with built-in encoders that measure the wheel speed; however, encoders will not send data about wheelchair angle relative to the normal plane. Therefore, a feedback system was required. An Inertial Measurement Unit (IMU) is an ideal sensor to provide angle and position feedback of the wheelchair. There are various

types of IMUs, the most common are the 6-DoF and 9-DoF systems. 6-DoF is a combination of gyroscope and accelerometer. It is ideal to identify the acceleration, yaw, pitch and roll of the system. While 9-DoF is a combination of a gyroscope, an accelerometer, and a magnetometer. For the purpose of this thesis, a 6-DoF IMU is sufficient. It will give feedback of the wheelchairs acceleration, as well as yaw, pitch and roll angles. Yaw is a rotation of α about the z -axis. A pitch is a rotation of β about the y -axis. A roll is a rotation of γ about the x -axis (refer to Figure 3.8). For this purpose, a MPU6050 IMU was chosen as an internal-state sensor for feedback about the wheelchair’s orientation.

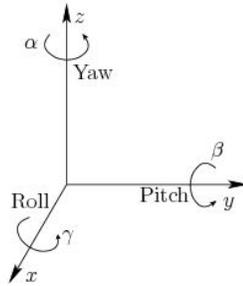


Figure 3.8: Roll, Pitch and Yaw Representation [53]

Apart from internal feedback of the wheelchair, the autonomous navigation system requires external environment feedback to avoid obstacles. The contact sensors are not ideal for this application, as the wheelchair navigation system is designed to avoid obstacles without bumping into them. Therefore non-contact sensors are analyzed. There are two types of non-contact sensors that are available: visual sensors and non-visual sensors. For the purpose of this thesis, both types of non-contact sensors were looked into (see Table 3.3). The process of selecting an ideal sensor for obstacle detection included two main criteria. First, the sensor is able to produce 3D imaging of the environment; whereas, the second criteria is that the sensor is affordable. As mentioned in Section 1.1, the economic strain on patients and their families is to be avoided. Therefore, the sensors have to be economical and efficient.

There are four types of sensors identified for the task of obstacle detection.

1. Structured Light 3D Sensor - ideal for measuring 3D shape image as it uses projected light patterns and a camera system [54].
2. Stereo Sensor - uses two or more lenses with separate image sensors for stereo photography, which gives the ability to produce 3D images. They are also ideal for range imaging [54].
3. LiDAR - light detecting and ranging sensors. Generally used to determine the depth between the sensor and the object by use of a laser scanner. They are typically found in 2D, however, by adding a nodding mechanism it can be utilized to produce 3D point clouds [54].
4. Time-of-Flight Camera - similar to LiDAR, however, it does not carry a laser scanner. It utilizes a light pulse to capture the environment [54].

Through economic process of elimination, three sensors were deemed acceptable for the purpose of this thesis. The Kinect (structured light camera), the ZED (Stereo Camera) and Scanse (LiDAR) are the three sensors that can be utilized for obstacle detection system (see Table 3.3). However, the Kinect has a smaller range of vision compared to the ZED and the Scanse. Due to this, the Kinect was eliminated from the selection process. It was difficult to decide between the ZED and the Scanse, as to which sensor to implement for the application of obstacle detection. Using a LiDAR sensor, there is an increase risk for an inaccurate construction of a 3D point cloud, as it is designed to generate a 2D point cloud. To achieve a 3D point cloud from a LiDAR sensor, a nodding mechanism has to be added. As for the stereo camera, the ZED sensor, it is beneficial for indoor and outdoor application. It is capable of long range data procession, as well as able to produce a depth cloud. Due to this, it was determined that a stereo sensor will be utilized in development of the obstacle

detection system. Refer to Table 3.4 for technical specification of ZED Stereo Camera (see Figure 3.9).

Table 3.3: Types of Non-Contact Sensors and Technical Specifications

Sensor	Technical Specifications	Cost
Kinect (Structured Light 3D Sensor)	<ul style="list-style-type: none"> • Range - 1.2m to 3.5 • Weight - 1.4 kg • FoV - $57^\circ(\text{H}) \times 43^\circ(\text{V})$ 	\$ 300 USD
ZED (Stereo Sensor)	<ul style="list-style-type: none"> • Range - 20m • Weight - 159 g • FoV - $90^\circ(\text{H}) \times 60^\circ(\text{V}) \times 110^\circ(\text{D})$ 	\$ 449 USD
Bumblebee2 (Stereo Sensor)	<ul style="list-style-type: none"> • Range - 7m • Weight - 342 g • FoV - $97^\circ(\text{H}) \times 67^\circ(\text{V})$ 	\$ 3,500 USD
Scanse (LiDAR)	<ul style="list-style-type: none"> • Range - 40m • Weight - 120 g • FoV - $360^\circ(\text{H}) \times 0.5^\circ(\text{V})$ 	\$ 400 USD
Velodyne Ultra Puck (LiDAR)	<ul style="list-style-type: none"> • Range - 200m • Weight - 925 g • FoV - $360^\circ(\text{H}) \times 40^\circ(\text{V})$ 	\$ 4,000 USD
Basler (Time-of-Flight Camera)	<ul style="list-style-type: none"> • Range - 13m • Weight - 400 g • FoV - $57^\circ(\text{H}) \times 43^\circ(\text{V})$ 	\$ 2,000 USD



Figure 3.9: Zed Stereo Camera [55]

Table 3.4: ZED Camera Technical Specifications [55]

Components	Technical Specifications
Dimensions and Weight	<ul style="list-style-type: none"> • 6.89" (W) × 1.18" (H) × 1.3" (L) • Weight - 159 g
Features	<ul style="list-style-type: none"> • High-Resolution and High Frame-rate 3D Video Capture • Depth Perception Indoor and Outdoor - max 20m • 6-DoF Positional Tracking • Spatial Mapping
Video	<ul style="list-style-type: none"> • Video Mode - 720p • Frames per Second - 60 • Output Resolution - 2560 × 720
Lens	<ul style="list-style-type: none"> • Wide-angle all-glass dual lens • Field of View - 90°(H) × 60°(V) × 110°(D) max • $f/2.0$ aperture
Compatible OS	<ul style="list-style-type: none"> • Windows 7, 8, 10 and Linux
Operating Temperature	<ul style="list-style-type: none"> • 0°C to + 45°C

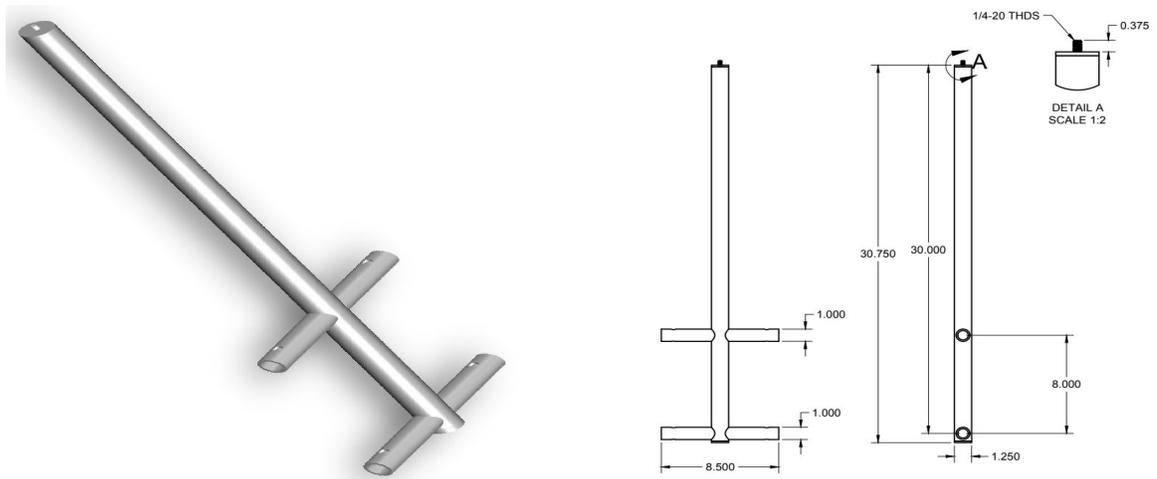
3.2.2 Hardware Implementation of Add-on Device Accessories

The final prototype of the wheelchair contains a modified controller system and attachments for the sensors. The modified controller is placed in-between the space of the seating system and the driving system. This allows the controller to be secured without modifying the original wiring of the wheelchair. The safety circuit is placed in position of the original controller. This gives easy access to power the wheelchair. The placement of the sensors is a bit more complex, than the placement for the modified control system. Since they are an add-on devices, it is essential that the sensors can be placed on any wheelchair. Therefore, it was decided that the sensors will be placed on attachments equipped with the wheelchair. It was essential to

identify the placement requirements for the sensors to determine which attachments can be utilized or need to be modified.

As discussed in Section 3.2.1, the IMU is utilized to detect the acceleration, yaw, pitch and roll of the wheelchair with respect to the flat ground (normal plane). Due to this, it is crucial that the IMU is placed such that it is flat against the surface of the driving system. Since the dimensions of the IMU is 1" × 2", it can be placed in a tight space. For this reason, the placement of IMU is parallel to the controller, between the seating assembly and the driving system.

The ZED sensor used for identifying the static and dynamic obstacles in the navigation system, was required to be mounted at a height where there are no obstructions in the field of view of the sensor. The camera has to be placed at such a position that it is not obstructing the users view and have a clear view of the front of the wheelchair. The only attachment for the seating assembly is the backrest. Due to this, the backrest was deemed acceptable for the placement of the sensors; however, it is not suited to install the sensors without modification. Thus, the backrest was modified by adding a post and an extra horizontal support to ensure rigidity (see Figure 3.10).



(a) Rendered Image of the Sensor Mount

(b) Sensor Mount Dimensions (Dimensions in inches)

Figure 3.10: Backrest Sensor Mount

Once all the components were mounted, the wheelchair was ready to communicate with ROS to be automatized. Figure 3.11 shows the final prototype for the autonomous wheelchair.



(a) Front View of the Wheelchair



(b) Back View of the Wheelchair



(c) Side View of the Wheelchair



(d) Isometric View of the Wheelchair

Figure 3.11: Modified Autonomous Wheelchair

Chapter 4

Noise Filtration and Sensor Fusion

Navigation is one of the most challenging aspect of a mobile robot. A successful navigation system is dependant on four things: perception, localization, cognition, and motion control (see Figure 4.1) [56]. For a wheelchair to be considered autonomous, it must be able to answer the following questions:

1. What is in my surrounding environment? (Perception)
2. Where am I? (Localization)
3. How do I get to the final position? (Cognition)
4. What do I need to do to get there? (Motion Control)

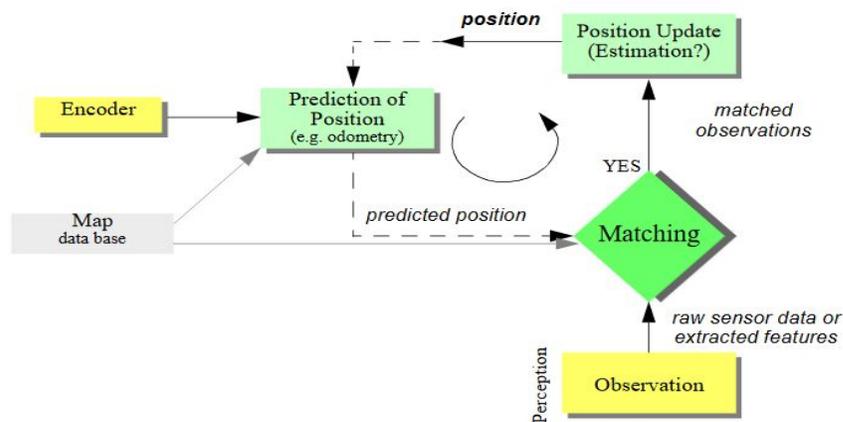


Figure 4.1: General Schematic for Mobile Robot Localization [56]

In this chapter, the methods used for perception and localization are explored, where it will be divided into three topics. First, Section 4.1 discusses how a sensor's uncertainty can be responsible for localization failure. Second, Section 4.2 describes various methods used to eliminate sensor noise. Lastly, Section 4.3 presents the implementation methods of perception and localization of the wheelchair in a ROS environment.

4.1 Sensor Noise

One of the biggest challenges for an indoor robot is to answer the question, "Where am I?" Sensors like Global Position System (GPS) are able to answer this question with quite ease; however, they are not suited for indoor application. A GPS is utilized to find an absolute position of an object relative to the Earth's reference frame [56]. For an efficient navigation system, the wheelchair has to know its relative position with respect to static or dynamic objects in its surroundings. Sensors are the most important robot input for the process of perception. They allow the robot to identify its world state and an error in doing so will limit the consistency of the robot in the same environmental state [56]. As mentioned in the previous chapter, two sensors were selected in addition to the in-built motor encoders of the wheelchair to develop a navigation system. It is imperative that the sensors produce minimal error to ensure that the navigation system runs flawlessly. Inaccuracy of the sensors can pose a challenge to localize the wheelchair.

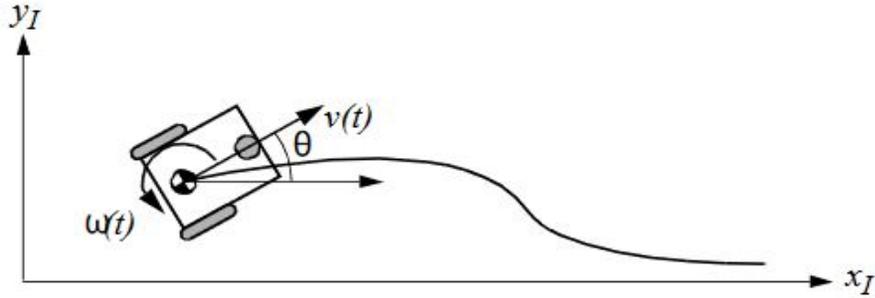


Figure 4.2: Movement of Differential Drive Robot [56]

The wheelchair primarily receives data from the wheel encoders to identify its location. As the wheelchair moves, the encoders are integrated to compute position. Due to this, the position error accumulates over time [56]. There are various reasons for the wheelchair to acquire odometric error. These odometric errors are categorized as deterministic or random. In the case of random errors, it becomes difficult to determine the source of the error.

Types of odometric errors [56]:

1. Range error - integrated path length of the robots movement; sum of the wheel movements.
2. Turn error - similar to range errors but are related to turns; differences pertain to the wheel's motion.
3. Drift error- difference in the wheel encoder, resulting in an error for the robot's angular orientation.

Over a period of time, turn and drift errors have more impact to the overall position and orientation of the wheelchair than range errors. The error model for odometric position estimation presented by Siegwart et, al. [56] considers the random errors such as wheel deformation, slippage, unequal floor, and errors in the encoders. For the model presented, two assumptions are made: the two errors of the individually driven wheels are independent and that the errors are proportional to the absolute value of the traveled distances (see Figures 4.3 and 4.4).

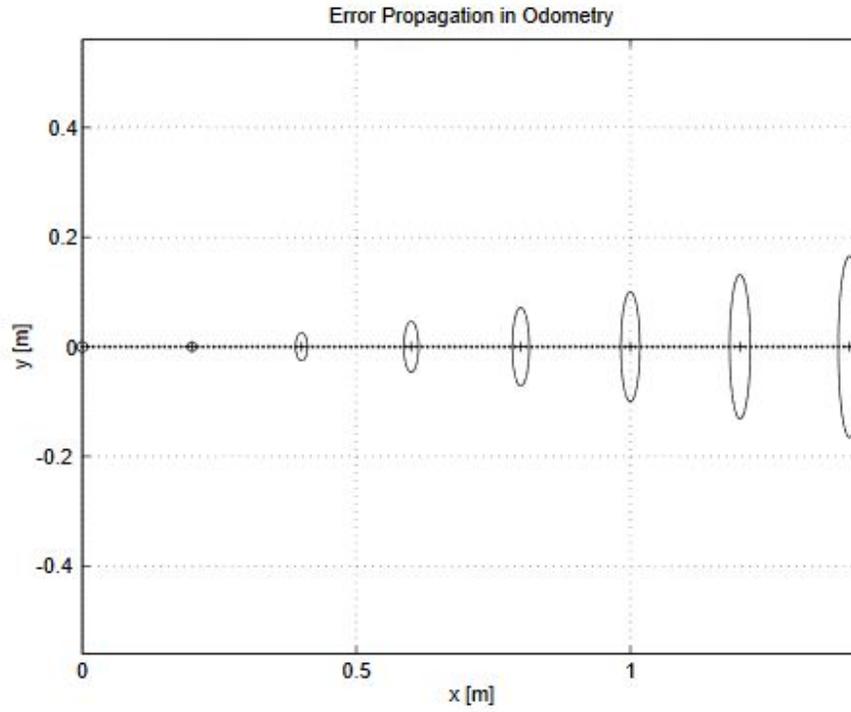


Figure 4.3: Range Error of the Mobile Robot [56]

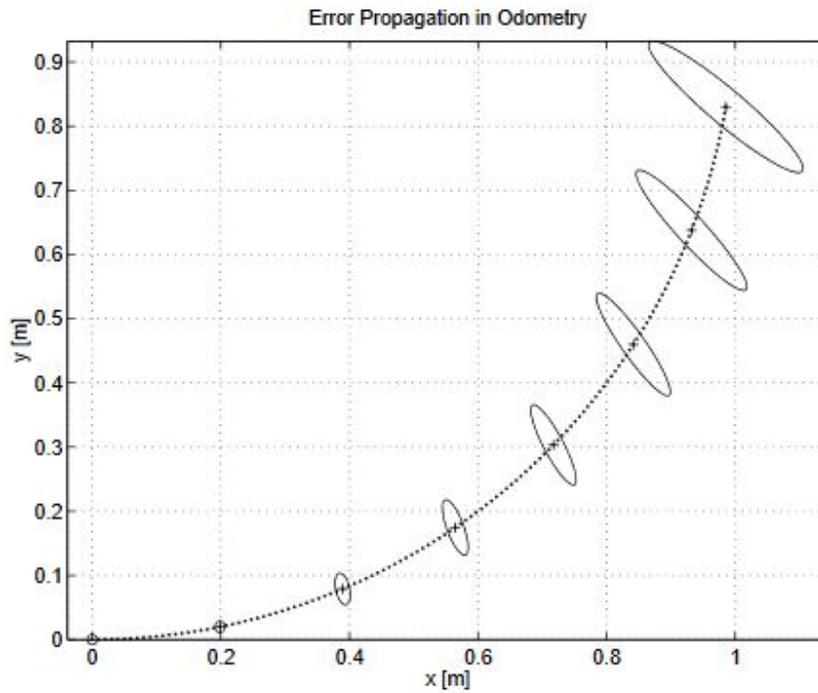


Figure 4.4: Turn Error of the Mobile Robot [56]

Other than wheel encoders, the wheelchair can receive odometric information from the IMU. The IMU is designed to provide the pose (x , y , z , yaw , $pitch$, and $roll$) of an object it is placed on. The MPU6050, the IMU used in this research, is not calibrated, which means it has sensor noise that increases the inaccuracy of the sensor over a period of time. Even after calibration, there is a possibility that the sensor may still produce noise. The IMU noise can be categorized using two types of errors: deterministic and stochastic [57]. Deterministic errors are caused by slowly varying sensor bias, while stochastic error is an additive noise that fluctuates very rapidly (white noise) [58, 59].

Apart from odometry from wheel encoders and IMU, the wheelchair can also localize itself using a stereo sensor. The ZED camera does not provide cartesian odometry; however, it provides information for localization through visual odometry. Visual odometry is estimating the motion of the camera in real-time using sequential images. The error of estimation using visual odometry can be the result of illumination changes in the environment, poor texture in close range, and dynamic objects and their size domination in the image view [60].

It is evident that sensor noise reduces the useful information from the sensor reading, which may result in difficulty localizing the wheelchair in its environment. Various researchers have proposed different ways to reduce or remove sensor noise. Most of the researchers have proposed different forms of sensor fusion or methods to fuse multiple sensor data to increase the overall information content of the robot's inputs.

4.2 Sensor Noise Filters

In order to remove process and measurement noise for localization of the mobile robot, a number of filtering options are available. Most of these options fall under two main categories: probabilistic methods and particle filter methods. The models

presented for noise filtration typically exhibit complex non-linear and non-Gaussian distributions. The two most popular probabilistic methods are the Markov method and Kalman filter. Markov localization method uses an explicitly specified probability distribution across all possible robot positions [56]. It localizes starting from an unknown position and recovers the pose by tracking multiple and completely disparate possible positions of the robot. However, the computational process of the Markov method is lengthy as it requires discrete representation of the space to update the position as the robot moves. Due to this, the precision and map size is limited [56]. As for the Kalman filter, it is precise and efficient since it does not independently consider the possible pose in the robot's configuration. The Kalman filter localizes and tracks the robot from an initial known position, making it an ideal application for continuous map construction of the environment of the robot. The two types of Kalman filters presented are the Extended Kalman Filter and Unscented Kalman Filter [56].

In 1979, Anderson and Moore [61], presented an algorithm to solve the problem of filtering, known as the Extended Kalman Filter (EKF). It is based upon the principle of linearizing the measurements and evolution model using Taylor series expansion. However, this filter is not suited for non-linear systems and probability distributions of interest [62]. In 1997, Julier and Uhlmann [63] introduced a filter that uses a Gaussian distribution to approximate arbitrary non-linear functions. This filter is known as the Unscented Kalman Filter (UKF). The UKF method is ideal for non-linear systems, instead of the EKF, as it generates better estimation of the covariance of the state. However, there is one limitation for UKF, it is not applicable to general non-Gaussian distributions [62].

As for particle filters, one of the most popular of them is the Sequential Monte Carlo (SMC) method. This is an old solution proposed in 1993 by Gordon et, al. [64]; however, due to the computer processing power required for it to function, it was not

popular until the early 2000's. This method provides a complete representation of the states by performing statistical estimation. The easy computation of the statistical estimation allows the system to identify and deal with the non-linearities or distributions. Apart from SMC, Adaptive Monte Carlo Localization (AMCL) method is widely used for sensor noise filter for localization of the robot. AMCL is similar to SMC, however, it adapts the size of the sample particles as the algorithm is processing. The algorithm uses KLD-Sampling, as it chooses a small number of samples if the density is focused on a small part of the state space, and it chooses a large number of samples if the state uncertainty is high [65]. It is computationally more efficient than SMC. For localization of the robot, the AMCL doesn't require initial position of the robot to be assigned. As the robot moves the environment data received from the sensors, re-weights the samples to localize the robot [66]. For a particle filter to work efficiently, it requires a large sample size. Due to this, typically particle filters are accompanied by probabilistic models [62].

Therefore, for the purpose of this research, Kalman filters are considered the most ideal noise filter method for localization and mapping of the wheelchair for obstacle detection.

4.3 ROS Implementation

The application of sensor noise filters for the localization of the wheelchair is a foundation to ensure that the autonomous navigation system works efficiently. However, this implementation has to happen in a ROS environment. As discussed in Section 4.1, the wheelchair can identify its location using odometry from a known location. However, uncertainty over a period of time can be problematic for the wheelchair to localize itself. To ensure that the wheelchair can localize itself, in relation to its environment map, it relies on its on-board sensors. Due to this, the localization process

is generally a two-step process that is done simultaneously. The localization updates primarily use an encoder sensor and has support from external sensors, along with mapping itself while in motion.

Since the localization process uses encoders and external sensors, this section will present the types of SLAM algorithms available in a ROS environment and the process to apply a Kalman filter, such that, it stabilize the wheelchair odometry using wheel encoders, IMU data, and visual odometry of the ZED camera. As mentioned in Section 4.2, there are two types of Kalman filter: EKF and UKF. To identify which filter is ideal for this work, both of the filters are compared and the most suited filter for wheelchair localization and mapping is identified.

4.3.1 SLAM Algorithms

As reviewed in Section 2.2.1, there are various SLAM algorithms applied for the navigation of a wheelchair. However, as identified, most of the implementations were not applied through the ROS framework. SLAM implementation in ROS is available through various packages. Each package utilizes different SLAM algorithms. The most popular SLAM packages available in the ROS framework are `hector_mapping` and `gmapping`.

The `hector_mapping` package is a SLAM approach that can be used without odometry for platforms that exhibit roll/pitch motion. It utilizes 2D pose estimates from LiDAR sensors. The system does not provide explicit loop closing ability, but it can still be sufficiently applied to real world scenarios. The system has successfully been used on various projects related to unmanned ground robots [67].

The package `slam_gmapping` contains a ROS wrapper for OpenSlam's Gmapping. To use `slam_gmapping`, a mobile robot is required that provides odometry data and is equipped with a horizontally-mounted, fixed, laser range-finder. The `slam_gmapping` node creates a 2D occupancy grid map (presented similar to a building floor plan)

from laser and pose data collected by a mobile robot and attempts to transform each incoming scan into the odometry (odom) `tf` frame [68].

Even though both of the packages use LiDAR as a perception sensor, each package can be modified, such that, it can use the data received from a ZED camera. Considering that the `slam_gmapping` package uses odometry from the mobile robot and is able to close the loop for the environmental map, it is considered ideal for implementation of SLAM in a ROS environment.

4.3.2 ROS Implementation of Sensor Noise Filter for SLAM

Prior to implementing a filter in a ROS environment, it is crucial to ensure that the sensors are connected to the laptop and communicate efficiently in the ROS environment. The MPU6050 is ideally connected using an Arduino for outputting data; however, the Arduino cannot communicate with ROS independently. Therefore, the ROS package `rosserial_arduino` is utilized as a liaison to communicate with the Arduino in the ROS environment. Once the Arduino is able to communicate with ROS, the package `mpu6050_serial_to_imu` is used to stabilize and publish the IMU data in the ROS environment. This package uses the Arduino script MPU6050 Digital Motion Process (DMP) to receive filtered orientation values. The ROS node reads the IMU data from the Arduino serial port and publishes the linear accelerations, rotational velocities, and the orientation as a ROS `sensor_msgs/Imu` message. This node is also designed to broadcast a `tf` transform of the sensor. The `tf` transform is used to publish the relative pose and coordinate to the system and to setup the relationship between two coordinate frames. This is useful for localization as it identifies the IMU's frame with respect to other sensor frames in the world frame of the robot. Implementing the package for the wheelchair, the measured offset values of the MPU6050 needed to be modified with respect to the mounted position on the wheelchair.

Similarly, connecting the ZED camera in a ROS environment requires the ROS package `zed_wrapper`. This package is developed and maintained by Stereolabs. It allows one to use the ZED stereo camera with ROS. The ROS node is developed to output the information regarding the camera's left and right images, depth map, point cloud, and pose information [55].

There are two packages available to implement Kalman filters in a ROS environment: `robot_pose_ekf` and `robot_localization`. The package `robot_pose_ekf` is limited to implementing an EKF model to the system. The `robot_localization` package, developed by Charles River Analytics, Inc., is designed to implement both types of Kalman filters. Therefore, the ROS package `robot_localization` is utilized to combine the odometry of the sensors and reduce the noise error. The `robot_localization` package provides four types of sensor estimation nodes: `navsat_transform`, `dual_ekf_navsat`, `ekf`, and `ukf`. The `navsat_transform` node uses three variables in order to function: a world-referenced heading (yaw), the robot's current pose odometry data in its environment, and a latitude/longitude/altitude [69]. The output of this node is an odometry message that contains the GPS data transformed into the robot's world coordinate frame. The `dual_ekf_navsat` node is a compilation of three nodes: first node, an EKF instance that fuses odometry and IMU data and outputs an odom-frame state estimate; second node, a second EKF instance that fuses the same data, but also fuses the transformed GPS data from the third node; and lastly, an instance of the `navsat_transform` node, which takes in GPS data and produces pose data that has been transformed into the robot's world frame (map of the environment) [69]. The `ekf` and `ukf` nodes produce a map-frame state estimate using a generalized Kalman filter. As `navsat_transform`, and `dual_ekf_navsat` are developed to fuse GPS signals with the robot odometry using a Kalman filter, for this research, `ekf` and `ukf` are the two nodes that were tested and compared.

As seen in Figure 4.5, for the localization of the wheelchair in its environment, the

system has two stages. First, the odometry data from the wheel encoders, IMU, and visual odometry from the ZED camera is processed using the `robot_localization` packages to produce filtered odometry readings. The second step is to localize the wheelchair with respect to its environment. The filtered odometry does not provide information regarding the layout of the surroundings or the position of the wheelchair with respect to its environment. To generate the environmental map of the wheelchair, the ROS package `slam_gmapping` is used along with the visualization tool `rviz` in a ROS environment.

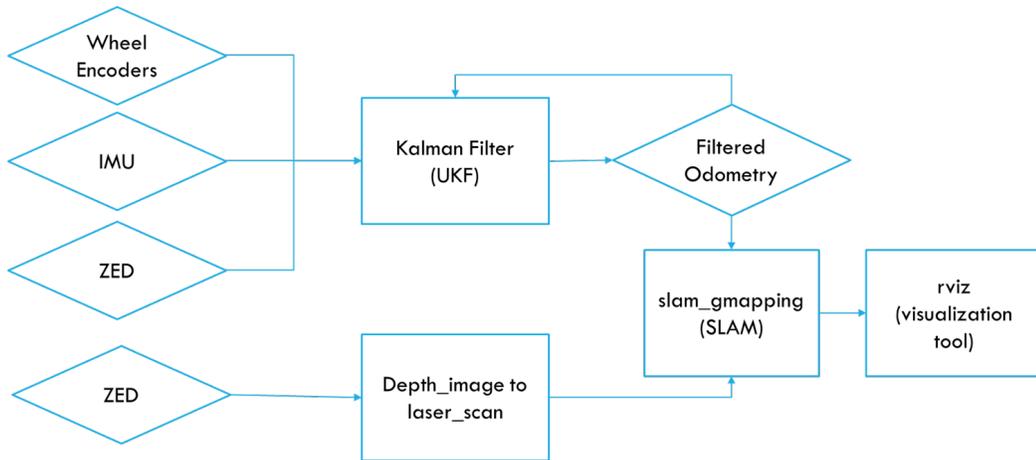


Figure 4.5: General Schematic of SLAM Process in ROS

As mentioned in Section 4.3.1, the `slam_gmapping` package uses a laser sensor as the perception sensor. Since neither LiDAR or any other laser sensors are utilized in this research, the 3D point cloud data acquired using the ZED camera is converted using `depthimage_to_laserscan` to send 2D scan data to the `slam_gmapping` package. The `depthimage_to_laserscan` package takes a depth image and generates a pseudo 2D laser scan based on the provided parameters. However, it is not designed to subscribe image or camera info until there is a subscriber for the scan data. Once the `slam_gmapping` node receives odometry data from the `robot_localization` node and the 2D laser scans data from the `depthimage_to_laserscan` node, the wheelchair is able to generate the map of the environment in motion, which is visualized in `rviz`.

4.4 Initial Test Results

The system was tested in an indoor environment with long hallways, slopes, and stairs. There were two tests performed to construct the environmental map: the first test was performed using the EKF for odometry fusion and the second test was performed using the UKF for odometry fusion. As seen from Figures 4.6 and 4.7, the map generated using the UKF presents less noise and produces better results of the environment. While using the EKF it was observed that the map is fuzzy and does not have clear obstacle identification. Therefore, for the application of the wheelchair, the UKF is better suited than the EKF.

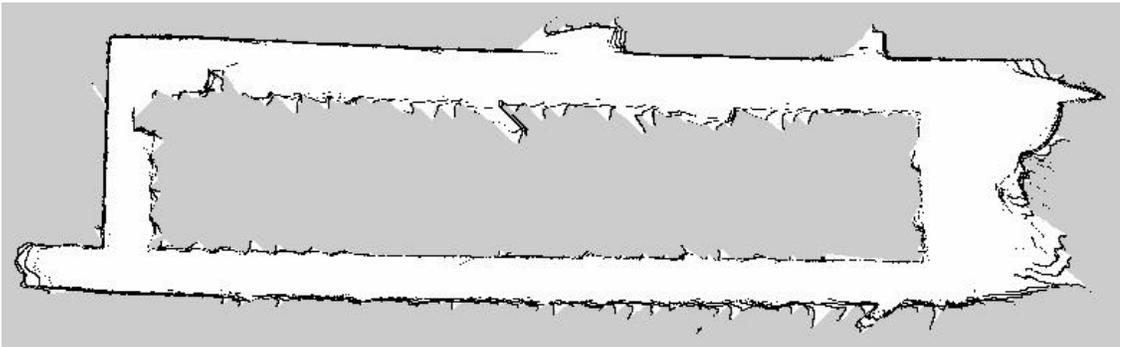


Figure 4.6: Environmental Map using EKF

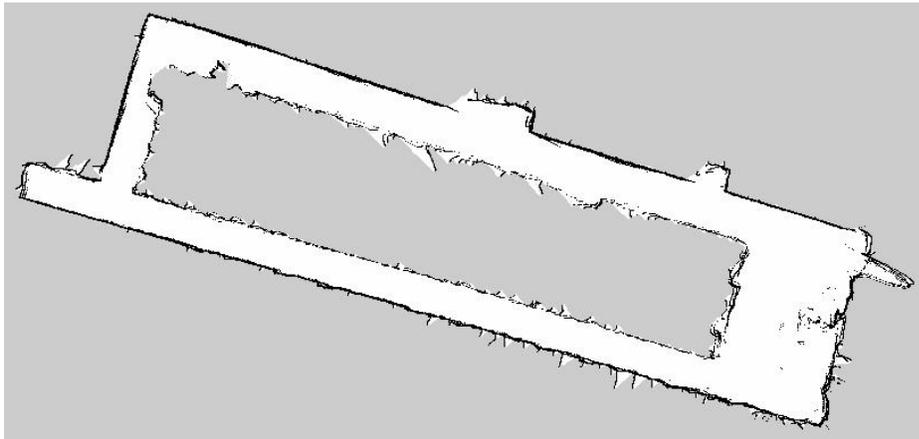
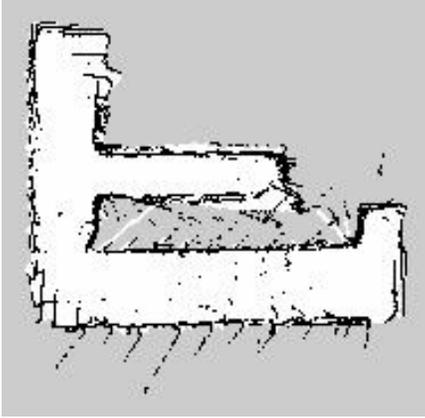


Figure 4.7: Environmental Map using UKF

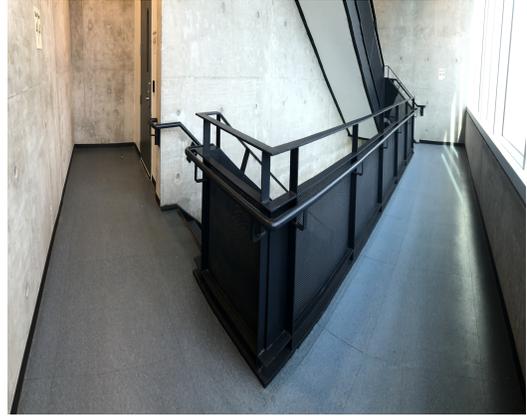
Chapter 5

3D Mapping, Slope Detection, and Autonomous Navigation

A conventional wheelchair user drives on ramps and avoids stairs when travelling from one floor to another. An autonomous wheelchair must mimic similar behaviour. The proposed algorithm for SLAM in Chapter 4 is tested in a multi-level environment to identify stairs as obstacles. Through Figure 5.1(a) and 5.1(b), it is observed that the SLAM algorithm presented in Chapter 4 does not identify the stairs in the environmental map as an obstacle, instead it identifies it as an opening that is traversable for the wheelchair. Due to this, the SLAM algorithm is not suited for the application of an autonomous wheelchair in a multi-floor environment.



(a) Environment 2D Map



(b) Environment Image

Figure 5.1: 2D SLAM Algorithm for Multi-level Flooring

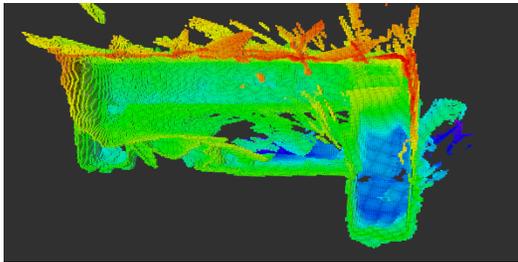
This chapter presents various algorithms that can be utilized to generate a 3D map of the environment, such that traversable areas are correctly identified and stairs are identified as obstacles, thus allowing autonomous navigation of the wheelchair in ROS for a multi-level environment.

5.1 3D Mapping

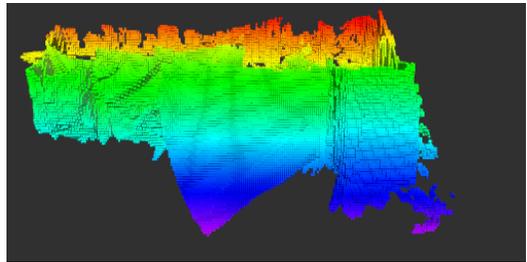
3D mapping is a useful tool to generate a detailed map of an environment. In a ROS environment, the 3D mapping of an environment can be primarily achieved using two packages: `octomap` and `rtabmap_ros`. The `octomap` package is a ROS package implementation of the OctoMap algorithm. The OctoMap algorithm is based on octrees and uses probabilistic occupancy estimation. It represents occupied space along with free and unknown areas. It proposes an octree map compression method that keeps the 3D models compact [70]. This can be implemented with `rviz`, a visualization tool in ROS, using the `octomap_server` node. The node uses the 3D point cloud generated by the sensor. Using probabilistic sensor fusion allows the node to compress the data and achieve computable data for 3D mapping [70]. The octree and node classes in the OctoMap framework can be represented using UML in Figure

5.1.1 Implementation Results

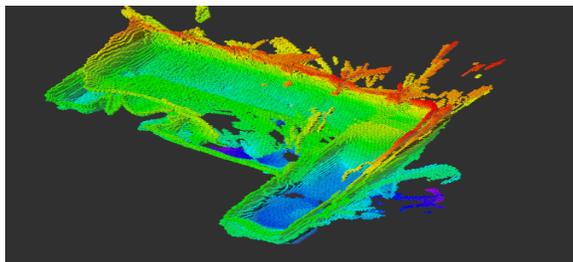
The 3D mapping algorithms, OctoMap and RTAB-Map, are widely used in ROS. To identify which 3D mapping algorithm is most appropriate for the wheelchair, both algorithms were tested. The algorithms were tested in a multi-floor environment (see Figure 5.1(b)). The `ocotmap` package was utilized to generate a 3D map occupancy grid of the environment in `rviz`. As observed in Figure 5.3, the 3D map of the environment identifies depth of the slope efficiently. It also defines the occupancy of the grid using colour schemes, cool colours (i.e., purple) are obstacles below the ground plane, while the warm colours (i.e., red) are obstacles above the wheelchair. The green areas represent the obstacles on the ground plane of the wheelchair. OctoMap is an efficient 3D mapping algorithm to identify the stairs as an obstacle. Since, the ramp going downwards will be below the ground plane, the system will identify it as an obstacle.



(a) Top View of the 3D Map of the Multi-level Environment



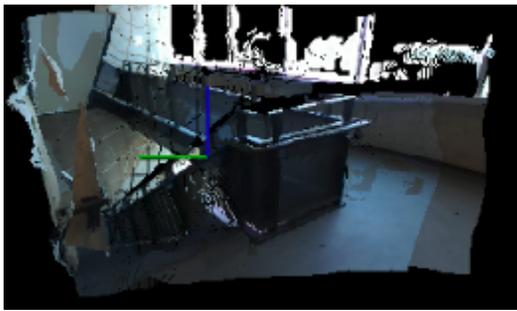
(b) Side View of the 3D Map of the Multi-level Environment



(c) Isometric View of the 3D Map of the Multi-level Environment

Figure 5.3: 3D Map of the Multi-level Environment using OctoMap in `rviz`

Next, the `rtabmap_ros` package was tested with both the RTAB-Map visualization tool and the `rviz` visual tool. Figure 5.4 presents the results obtained using RTAB-Map visualization tool. As observed, the RTAB-Map identifies windows texture as an obstacle and detects stairs efficiently. However, it does not consider stairs as an obstacle. This implementation of the algorithm is ideal for standalone systems that do not contain an autonomous navigation system. In order to develop an autonomous navigation system, it will require more processing power as the navigation system is visualized using the `rviz` visualization tool. Running both the `rviz` and RTAB-Map visualization tool will reduce the reaction time of the wheelchair with obstacles.



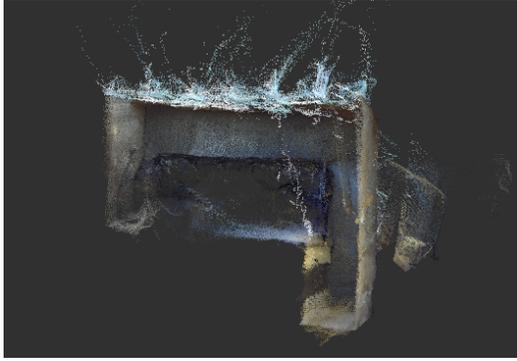
(a) 3D Map of the Environment



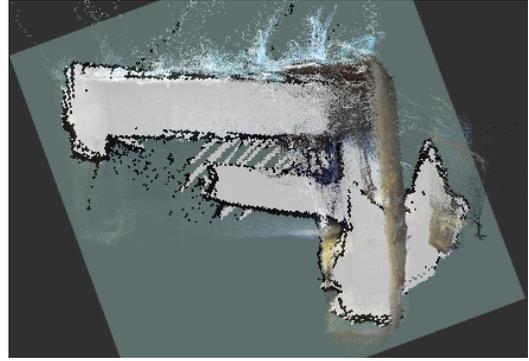
(b) Camera Image from ZED

Figure 5.4: 3D Map of the Multi-level Environment using RTAB-Map Visualization Tool

Lastly, the `rtabmap_ros` package was tested utilizing the `rviz` visualization tool. Figure 5.5 presents the 3D mapping in the `rviz` visualization tool using the `rtabmap_ros` package. This is the most ideal system as it publishes a 2D map and overlays a 3D map cloud of the environment. The 3D map cloud allows the system to identify the distance to an object relative to the wheelchair in the environment using depth from the point cloud. This is the most useful combination for navigation and slope and stair detection for the wheelchair.



(a) 3D Map of the Environment



(b) 2D and 3D Map Overlay of the Environment

Figure 5.5: 3D and 2D Map of the Multi-level Environment using RTAB-Map in rviz

5.2 Slope and Stair Detection

The 3D mapping algorithms does not provide the wheelchair system with information on whether the stairs are considered obstacles and/or the slopes are considered traversable. Therefore, a slope and stair detection algorithm is necessary in combination with the 3D mapping algorithm discussed in Section 5.1.1 for the obstacle detection system of the wheelchair. Slope and stair detection is not a widely researched topic in robotics using ROS. Drwiega and Jakubiak developed a package called `depth_nav_tools` to detect stairs as an obstacle using depth sensor. The package is designed to work efficiently with any depth sensors; however, all the testing performed for this package were limited to Kinect [74]. Since the visual sensor utilized in this thesis is the ZED camera, a stereo vision sensor, the package was not directly adaptable to the system.

The `depth_nav_tool` package consists of four standalone ROS packages:

- `lascerscan_kinect` - Converts depth image published by Kinect to 2D Laser-Scan data.
- `cliff_detector` - Detects negative objects like cliffs or downstairs.

- `depth_sensor_pose` - Detects the ground plane on depth image and estimates height and tilt angles of the depth sensor.
- `nav_layer_from_points` - Creates navigation costmap layer based on received points from `cliff_detector`.

Each package was modified to incorporate the size of the depth image, the data type published by the ZED camera, and the published topic with their respective messages to communicate the ZED camera with the `depth_nav_tool` package. The `depth_nav_tool` package was implemented in the system coinciding with the package `costmap2D`. This was used to identify stairs as an obstacle since the `depth_nav_tool` package produces a costmap layer for stair detection. The `costmap2D` contains various layers, each layer dedicated towards an obstacle detection system of the robot. Even after implementing the modifications to the package to be compatible with the ZED camera, in correlation with the `costmap2D` package, the stair detection algorithm still failed to produce the cliff detection. Instead, the stairs were not identified as an obstacle in the costmap layer.

The stair detection failure was caused due to the system avoiding the ground points, in lieu of avoiding the ground as an obstacle. By doing so, the system could not identify the negative space to the ground plane. For the modified package to work, the node for `obstacle_detection` in the `rtabmap_ros` package was utilized. Where a 3D point cloud from the ZED camera is utilized to identify topics `/ground_obstacles` and `/ground_cloud`. The obtained ground point data from the point cloud was then published in the `cliff_detector` node. Figure 5.6 shows the output of the costmap, overlaying the 2D map of the environment, where stairs are identified as an obstacle. The pink and purple markers on the map are the obstacles in the environment of the wheelchair, where the system identifies walls and stairs as obstacles.

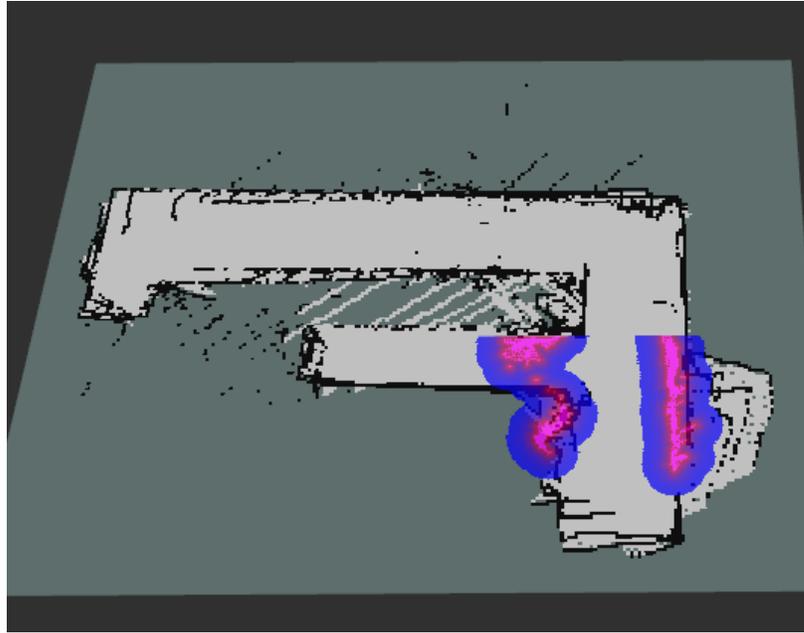


Figure 5.6: An Example of a Local Costmap for Stair Detection in rviz Visualization Tool

The package developed by Drwiega and Jakubiak only detects stairs as an obstacle; however, as mentioned in Section 1.5, the slope of the ramp cannot be more than defined by the Ontario and Canada Building Codes. According to the Ontario Building Code, the ramp should not be greater than 1:10 ratio (5.7°) [75], while according to the Canada Building Code the ramp should not be greater than 1:12 ratio (4.8°) [76]. To ensure that only ramps that meet the building codes are traversable, the ratio of 1:10 was utilized to identify greater ramp slopes as a obstacle. This was carried out by developing a custom node `slope_detection` that utilizes the Point Cloud Library (PCL) in a ROS environment. From the PCL, the normal estimation node was adopted to communicate with the 3D point cloud published by the ZED camera. The custom package consists of two nodes: publisher and subscriber. The publisher node publishes the topics from the PCL to the ROS environment while receiving the data from the ZED camera point cloud. For the publisher node to avoid extensive computational processing, the RANSAC (Random Sample Consensus) algorithm is used. The

subscriber node subscribes to the publisher node to output the distance of the ground from the sensor optical frame. Furthermore, it publishes the angles in X and Y axes to calculate the slope of the ramp.

The node was tested in an indoor environment where the measured slope of the ramp was 4° while the calculated slope through the node published the angle for the X axis to be 3.93° . The observed error for the angle is 1.75% ($\pm 0.07^\circ$ accuracy of the angle). This error could be due to the system averaging the depth points rather than taking the farthest point to calculate the angle. Since the stairs are designed between 1:1.6 ratio and 1:2 ratio [75], the observed error of the slope detection system, $\pm 0.07^\circ$ accuracy of the angle, does not possess risk for the wheelchair to identify stairs as ramps; therefore, the functionality of the wheelchair will not be impeded nor risk the safety of the occupant. Using the farthest point can help reduce the error percentage, but it can cause issues for the system as there can be outliers for depth points (Z points) in the point cloud. The outliers can exist due to various factors, such as the texture of the floor and the lighting of the environment.

5.3 Autonomous Navigation

Once the obstacle detection for the wheelchair was developed, the wheelchair needs a navigation system to make it autonomous. The design of an autonomous navigation system is divided into two test systems. The first autonomous navigation system consisted of the basic navigation system for the wheelchair, where only the SLAM process (refer to Section 4.3.2) is utilized for obstacle detection. This is to ensure that the wheelchair is capable of driving autonomously on a level surface. The second autonomous navigation system develops on the basic navigation system, where the slope and stair detection system are added as obstacles.

The basic autonomous navigation system in the ROS environment is performed using

the `move_base` package. As seen in Figure 5.7, the SLAM process data is inputted into the `move_base` package along with the AMCL information and the map of an area. As mentioned in Section 4.2, AMCL does not require an initial position of the robot for localization, so to refine the path planning and localizing for the autonomous navigation system, AMCL is combined with UKF. The `move_base` package provides an implementation of an action for the robot, where a goal pose is defined and travel is attempted. The `move_base` package contains a node that provides a ROS interface for configuring, running, and interacting with the navigation stack on a robot. The `move_base` node utilizes two planners to perform the navigation task: global and local. To ensure that obstacles are identified, the node maintains two costmaps: one for the global planner and one for the local planner. Figure 5.8 presents a high-level view of the `move_base` node. The processed data in the `move_base` package which publishes `cmd_vel`, the velocity, to the wheelchair’s controller and the visualization of the navigation system in `rviz`.

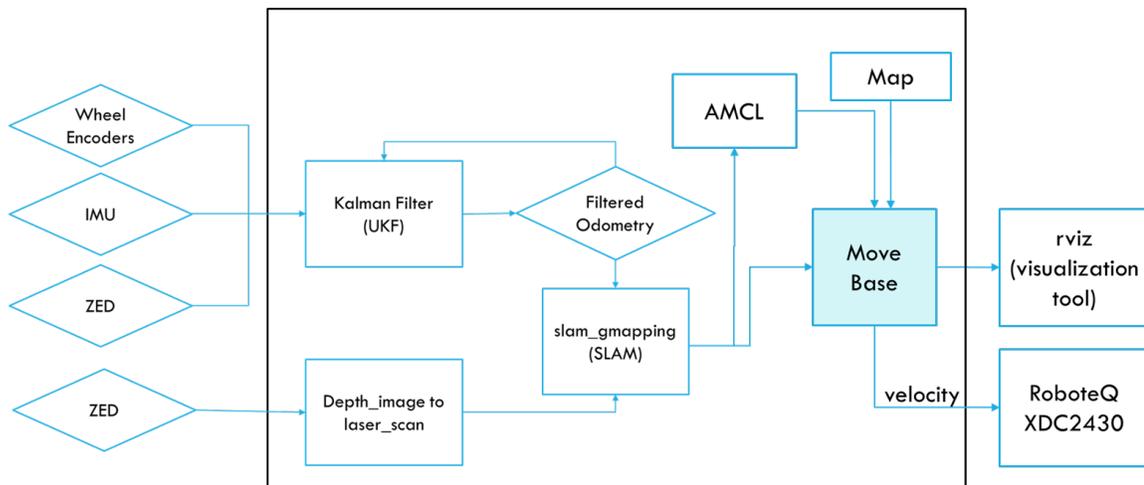


Figure 5.7: A High-level Schematic of the Basic Autonomous Navigation System of the Wheelchair

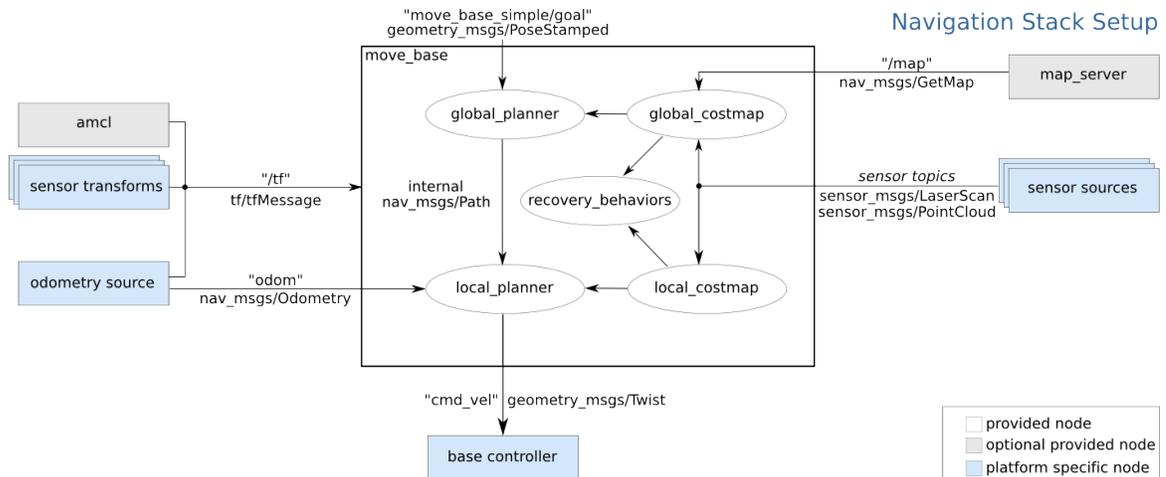


Figure 5.8: A High-level Schematic of the `move_base` Node [77]

The `move_base` node for the wheelchair utilized three plugins to ensure obstacle detection during basic autonomous navigation. The static map layer provides unchanging data information to the system in the global planner. It identifies the obstacles in the known map that are inputted in the `move_base` node, while the obstacle layer and inflation layer are utilized by the local planner. The obstacle layer utilizes the information from the point cloud generated by the ZED camera. It is used to identify both static and dynamic obstacles. The inflation layer ensures that the wheelchair can travel through the environment without getting stuck. Since the wheelchair is 0.8 meters wide, the inflation radius for the costmap is selected as 0.4 meters. This ensures that the wheelchair has enough clearance to travel around obstacles. The local planner generates a local map of the environment of the wheelchair, where if a dynamic object, such as a person, is identified it will inflate the obstacle to the size of the inflation radius, and redirect the path of the wheelchair while avoiding the obstacle. Thus, with the combination of the three plugins for the global and local planners, the wheelchair system is designed to successfully autonomously navigate indoors on leveled floors (refer to Section 5.4.1).

Once the basic autonomous navigation was successfully developed, the final autonomous

navigation system was developed, where the data retrieved using the slope and stair detection system was subscribed by the `move_base` package (see Figure 5.9). To develop the obstacle detection consisted of slope and stair detection and the `move_base` node for the wheelchair utilized four plugins to ensure obstacle detection. The plugins consists of static map layer, obstacle layer, and inflation layer from the basic autonomous navigation system, along with stair detect layer. The stair detection layer developed using the `nav_layer_from_points` node will identify stairs as an obstacle. Thus, with the combination of the four plugins and the global and local planners, the wheelchair system is designed to successfully autonomously navigate indoors while driving traversing ramps and avoiding stairs as an obstacle (refer to Section 5.4.2 and 5.4.3).

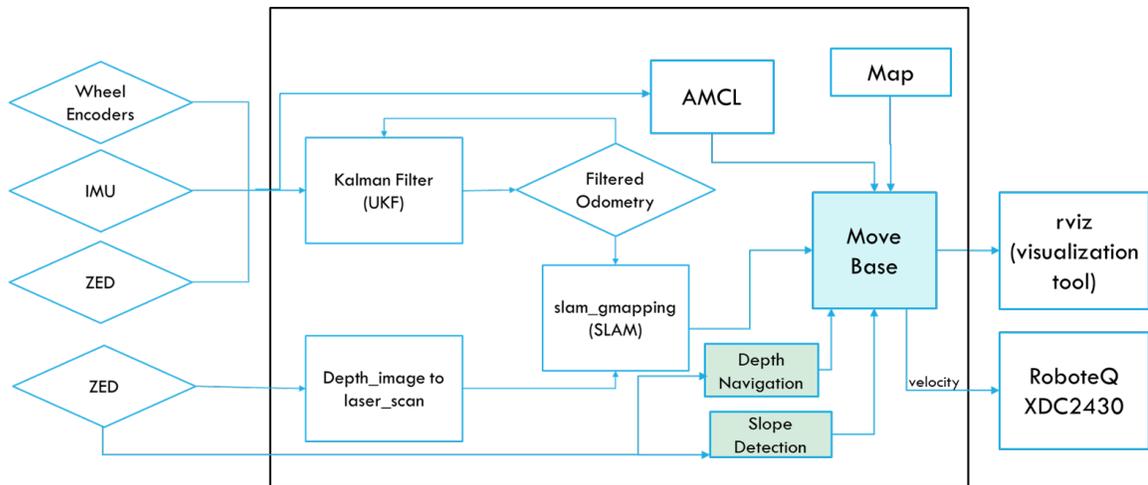


Figure 5.9: A High-level Schematic of the Autonomous Navigation System of the Wheelchair

5.4 Final Test Results

To validate the add-on system three tests were conducted in different environments:

1. Test 1 - Hallway, no stairs or ramps, to validate base autonomous navigation.
2. Test 2 - Stairs, to validate stairs as an obstacle.

3. Test 3 - Ramp, to validate an acceptable ramp for driving the wheelchair.

5.4.1 Test 1 - Hallway Autonomous Navigation

The first test was conducted in a hallway, to validate that the developed add-on system for autonomous navigation works. For this test, the final destination was defined in rviz. As observed in Figure 5.10, the costmap overlay, seen in pink and purple, indicates the walls as an obstacle that need to be avoided by the wheelchair and the path to navigate to the final destination, seen as a green line, is achieved using the `move_base` package. The inflation of the obstacles ensures that the wheelchair is traversable through the hallway without scratching the sides to the wall. Figure 5.11 shows the physical representation of the wheelchair in the environment. In Figure 5.11(a), the red circle defines the final position assigned for the wheelchair.

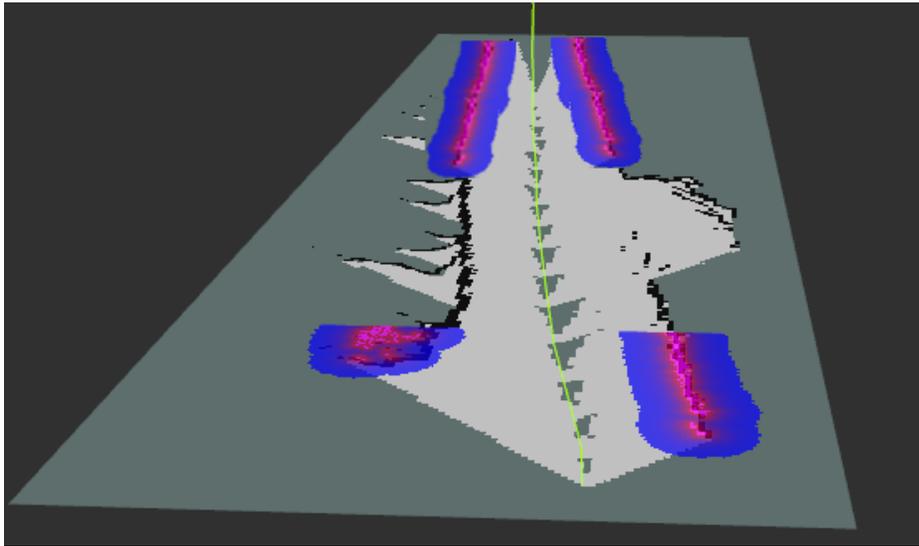
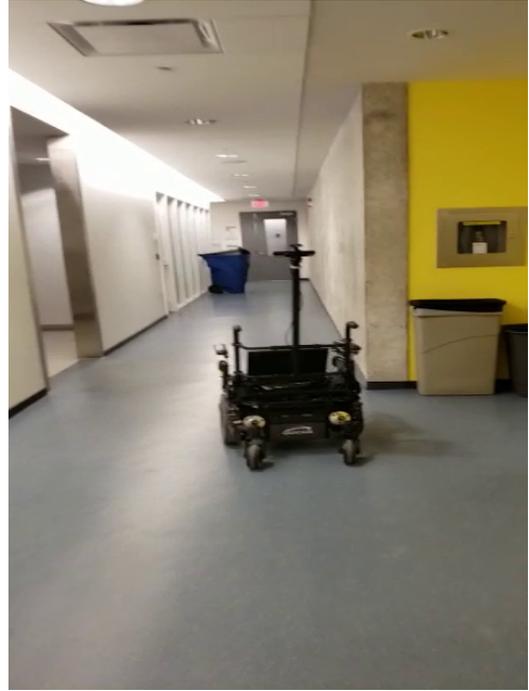


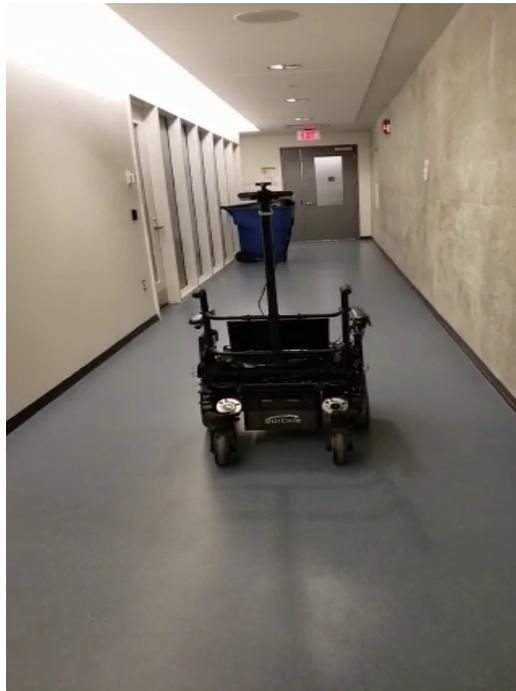
Figure 5.10: An Example of Navigation Path in rviz for Hallway Environment



(a) Starting Point of the Wheelchair with Defined Final Destination



(b) Mid Point of the Wheelchair During Autonomous Navigation



(c) Final Destination of the Wheelchair

Figure 5.11: Test 1 - Validation of the Hallway Driving using the Add-on System

5.4.2 Test 2 - Stair Avoidance Autonomous Navigation

The second test was conducted in a hallway leading to a stair case. This test was conducted to validate that the wheelchair detects the stairs and stops even if the goal is defined beyond the stairs. Similar to the first test, the final destination was defined in rviz. Figure 5.12 shows the path utilized to achieve the final destination (red dot) in rviz and the stairs identified as an obstacle using the costmap are marked in pink and purple, the size of the markings defines the inflation to ensure that the wheelchair is able to navigate around the obstacles. The black outline identifies walls as an obstacle. Figure 5.13 shows the physical representation of the wheelchair in the environment. In Figure 5.13(a), the red arrow defines the final position assigned for the wheelchair, which is at the midpoint on the stairs. As can be seen in the figures, the wheelchair successfully stops before the stairs.

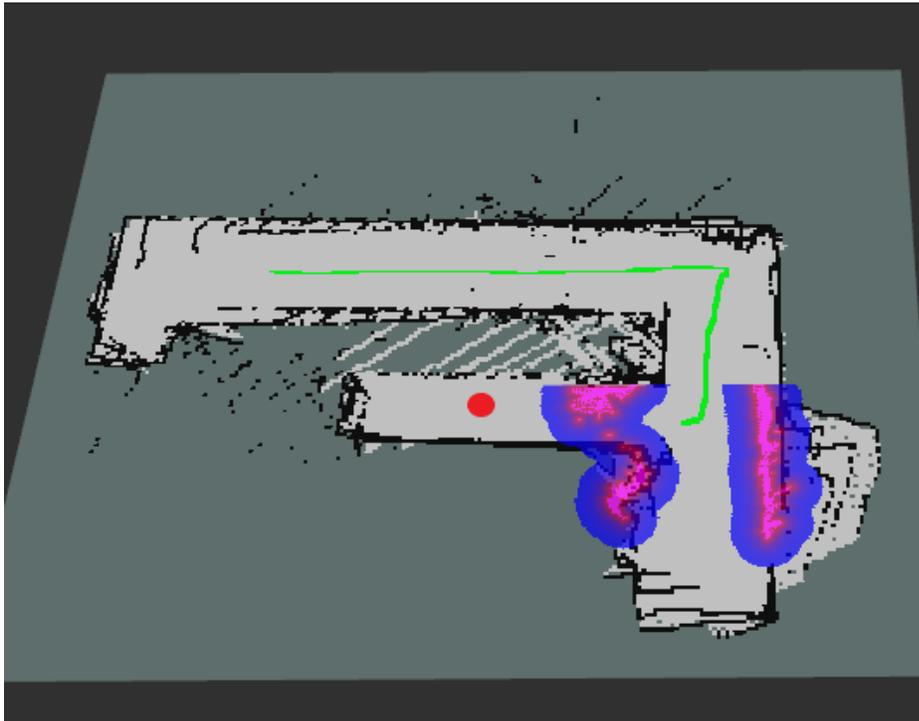
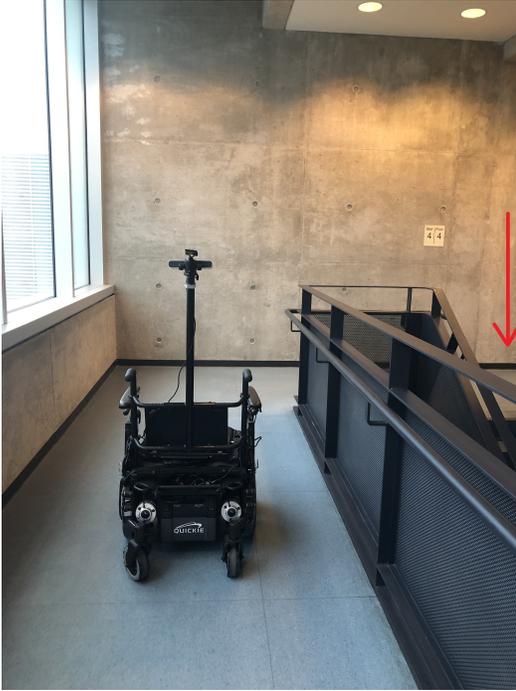
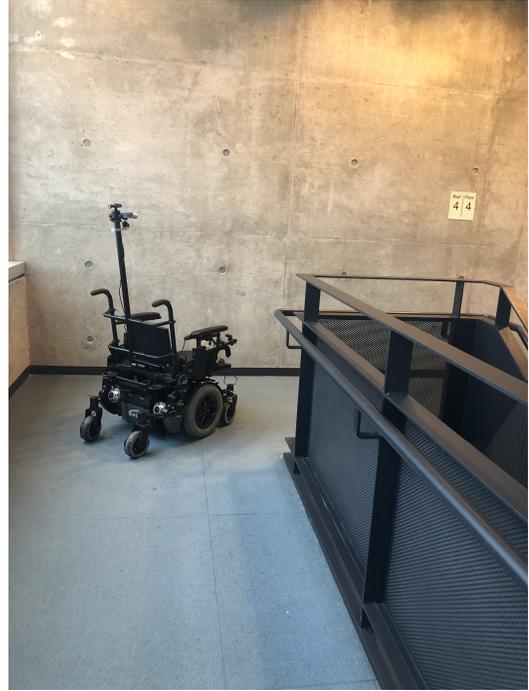


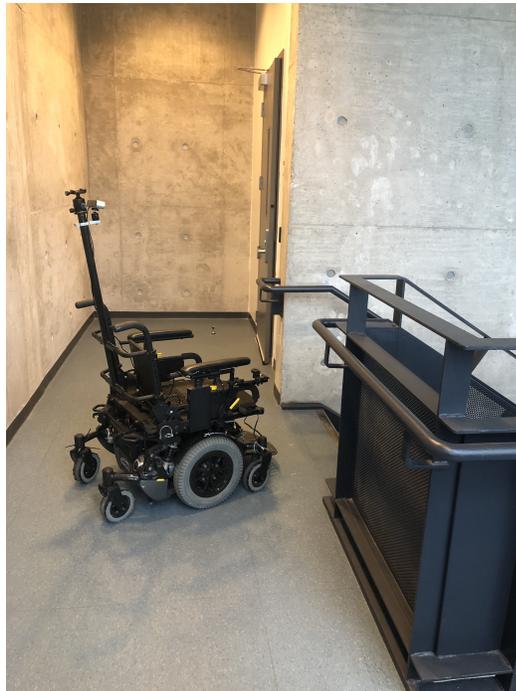
Figure 5.12: An Example of the Navigation Path in rviz for Stairs Avoidance



(a) Starting Point of the Wheelchair with Defined Final Destination



(b) Mid Point of the Wheelchair During Autonomous Navigation



(c) Final Position of the Wheelchair

Figure 5.13: Test 2 - Validation of the Stair Avoidance using the Add-on System

5.4.3 Test 3 - Ramp Acceptance Autonomous Navigation

The last test was conducted in an area with a ramp that was acceptable for a wheelchair. The defined final goal required the wheelchair to drive down the ramp to validate that the add-on system correctly identifies acceptable ramps for the wheelchair. Similar to the previous tests, the final destination was defined in rviz. Figure 5.14 shows the obstacles identified by the wheelchair and the path achieved to arrive at the defined final destination in rviz. Since the area was wide and the parameter for the range was defined at 2 meters, the wheelchair does not build a costmap while travelling other than in the beginning, where it detects the movements of students going to a classroom. The reason for the cluster to be only mapped in the beginning is due to the costmap constructed using the local planner. There were no other obstacles identified in the local costmap as the walls in the environment are static obstacles as observed by the black line in the map for the hallway. Figure 5.15 shows the physical representation of the wheelchair in the environment. In Figure 5.15(a), the red circle is the defined final position assigned for the wheelchair.

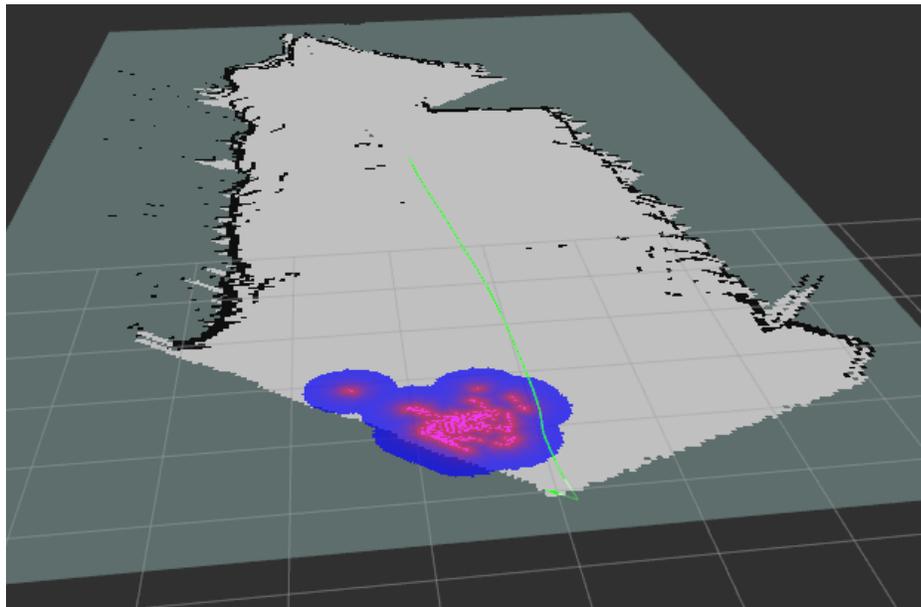
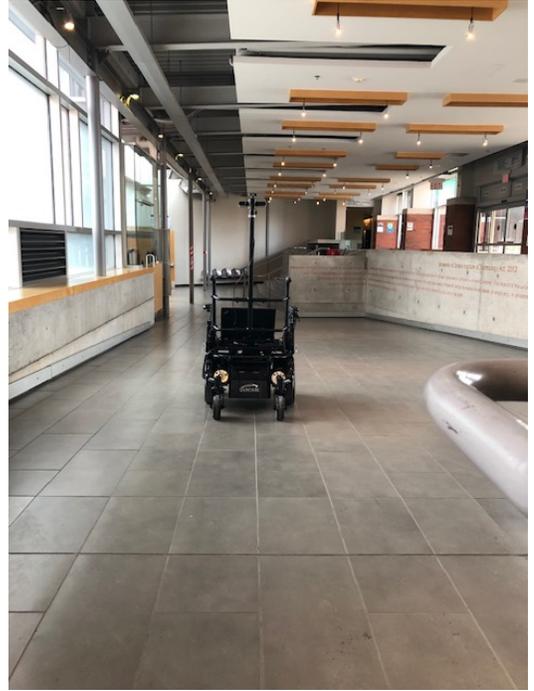


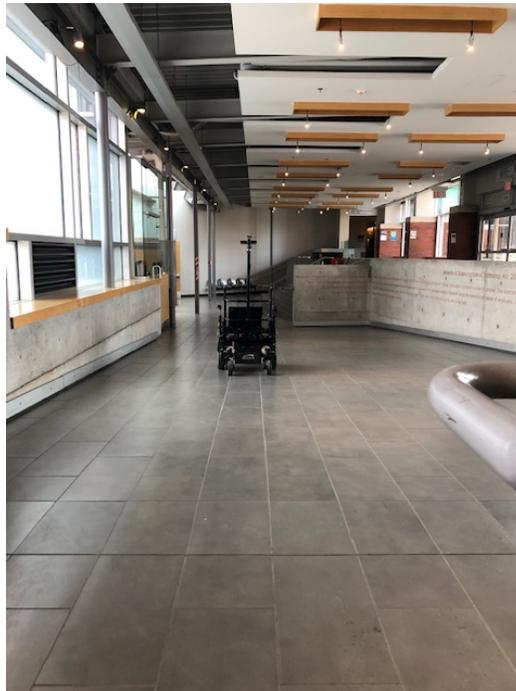
Figure 5.14: An Example of the Navigation Path in rviz for Ramp Environment



(a) Starting Point of the Wheelchair with Defined Final Destination



(b) Mid Point of the Wheelchair During Autonomous Navigation



(c) Final Destination of the Wheelchair

Figure 5.15: Test 3 - Validation of the Ramp Driving using the Add-on System

Chapter 6

Conclusions and Future Works

This thesis described the development of a fully functioning proof-of-concept prototype to convert a powered wheelchair into an autonomous wheelchair. The main focus of this thesis was to develop an obstacle detection system and a navigation system for an indoor environment.

Converting a powered wheelchair to an autonomous wheelchair required modifications to hardware and software. The provided controller was removed and replaced with a RoboteQ XDC2430 motor controller. The RoboteQ XDC2430 motor controller was programmed for the ROS environment utilizing the `roboteq` package. The package was modified to implement a differential drive kinematic model along with a PID controller to ensure straight driving of the wheelchair.

The odometry data from the wheel encoders, IMU (MPU6050), and visual odometry from the ZED camera, were fused using the `robot_localization` package to produce filtered odometry data. This ensured that the wheelchair avoids slippage error and provides more accurate odometry data of the wheelchair for localization. Localization is essential to develop an obstacle detection system for the environment. To be considered an efficient system, the obstacle detection system has to avoid stairs and other static and dynamic obstacles. The obstacle detection system testing was

divided into two sections: the preliminary testing and secondary testing. The preliminary testing was done on the leveled floor to ensure that the sensors correlated efficiently. Using the ZED camera, 3D point cloud data was acquired and converted using the `depthimage_to_laserscan` package to send 2D scan data to the `slam_gmapping` package. The `slam_gmapping` package was used to produce a 2D map of the environment while localizing the wheelchair in motion on level floors.

Secondary testing was conducted by testing various 3D mapping algorithms. The `rtabmap_ros` package was utilized to produce the `voxel_grid` of an environment, which generates a 3D map of the environment. The `voxel_grid` output produces the point cloud of an environment along with the ground plane as an obstacle. The ground point clouds were utilized by the `cliff_detector` package to identify obstacles which are below ground level, such as stairs. The identified obstacles were implemented using the `costmap2D` package as a layer in the costmap generated while performing autonomous navigation.

Lastly, for the system to be considered autonomous, a navigation system was developed. This navigation system consisted of the costmap produced using the `costmap2D` package with `move_base` package to define the goal pose. AMCL is a localization package within `move_base` which was used for path planning along with the UKF package from the `robot_localization` package.

In addition, the hardware and software developed here could potentially be used to convert any powered wheelchair into an autonomous wheelchair. However, some retrofitting would be required in order to add the sensors and motor controller equipment. Three tests were conducted to validate the add-on system. The first test was conducted in a hallway with no stairs or ramps. Second in an open area with stairs, where the final destination was defined past the stairs. Lastly, the wheelchair autonomously driving through an acceptable ramp. All the tests successfully validated the add-on system developed here.

6.1 Lessons Learned

While developing the add-on device to convert the powered wheelchair to an autonomous wheelchair, a few key lessons were learned. These key lessons include:

- There is significant wheelchair slippage error if only the wheels' odometry is utilized for localization. This is due to the castor wheels of the wheelchair.
- For stair detection, it is crucial for the ground point clouds to be identified, otherwise, the system will fail to perform stair detection.
- For efficient autonomous navigation, the system has to apply AMCL along with the UKF for ideal localization and path planning for the wheelchair.

6.2 Future Work

The developed proof-of-concept system effectively converts a powered wheelchair into an autonomous wheelchair. This thesis presents hardware and software development of an add-on device and indoor testing of the system. However, for the device to be considered ready for the market, it requires the system to be tested outdoors and develop an easy to use user interface depending on the need of the child. The charging system of the wheelchair has to be optimized such that the wheelchair can be charged using other methods instead of the current state, where the wheelchair has to be connected to the original controller since the battery charger connection is through the provided joystick connection.

Lastly, the wheelchair currently uses the ZED camera for sensing its environment. Objects behind the wheelchair are not detected, which could pose a risk for reversing the wheelchair in crowded and narrow environments. There are two possible solutions: add secondary sensor systems to detect objects behind the wheelchair such as additional ZED camera and/or implement a nodding or rotating LiDAR sensor in order to generate 3D scans of the environment.

References

- [1] J. C. Brehaut, D. E. Kohen, P. Raina, S. D. Walter, D. J. Russell, M. Swinton, M. O'Donnell, and P. Rosenbaum, "The health of primary caregivers of children with cerebral palsy: how does it compare with that of other canadian caregivers?" *Pediatrics*, vol. 114, no. 2, pp. e182–e191, 2004.
- [2] K. A. Stern, "Cerebral palsy," <https://www.cerebralpalsy.org/about-cerebral-palsy/definition>, 2018, [Online; accessed 26-May-2018].
- [3] S. Riisgaard and M. R. Blas, "Slam for dummies," *A Tutorial Approach to Simultaneous Localization and Mapping*, vol. 22, no. 1-127, p. 126, 2003.
- [4] ISO, "Wheelchairs-part 1: 10," International Organization for Standardization, Geneva, Switzerland, ISO, 2014.
- [5] "Quickie xperience 2 powered wheelchair," <http://www.sunrisemedical.ca/power-wheelchairs/quickie/mid-wheel-drive/quickie-xperience-2>, 2017, [Online; accessed 26-May-2018].
- [6] B. W. Nick Watson, "History of the wheelchair," <https://www.britannica.com/topic/history-of-the-wheelchair-1971423>, 2015, [Online; accessed 26-May-2018].
- [7] M. S. Otago, "History of the wheelchairs," <http://www.mobilityscooters.co.nz/history/wheelchairs>, 2015, [Online; accessed 11-June-2016].

- [8] B. Museum, “Bath chair,” http://www.bbc.co.uk/ahistoryoftheworld/objects/qcI7cMgiR0qmLnD_QPyIGQ, 2010, [Online; accessed 26-May-2018].
- [9] T. S. MINNIELSS., “Invalid carriage,” patentus 9708 A.
- [10] H. A. Everest and H. C. Jennings, “Folding wheel chair,” patentus 2 095 411A.
- [11] H. T. Kim, C. W. Ko, G. Y. Kim, J. H. Lee, and T. S. Bae, “A motorized and easy-docking wheelchair drive for persons with lower-limb disabilities,” in *Industrial Technology (ICIT), 2015 IEEE International Conference on*. IEEE, 2015, pp. 52–56.
- [12] S.-H. Chen and J.-J. Chou, “Motion control of the electric wheelchair powered by rim motors based on event-based cross-coupling control strategy,” in *System Integration (SII), 2011 IEEE/SICE International Symposium on*. IEEE, 2011, pp. 1340–1345.
- [13] BrainGate, “About braingate,” <https://www.braingate.com/about-braingate>, 2009, [Online; accessed 26-May-2018].
- [14] N. R. C. Canada, “25 - useable, motorized wheelchairs for vets,” https://www.nrc-cnrc.gc.ca/eng/about/centennial/100_years/1946_1964.html, 2017, [Online; accessed 26-May-2018].
- [15] D. J. Phillipson, “George klein,” <http://www.thecanadianencyclopedia.ca/en/article/george-klein>, 2015, [Online; accessed 26-May-2018].
- [16] S. Nomura and T. Murakami, “Power assist control of electric wheelchair using equivalent jerk disturbance under slope environment,” in *2010 11th IEEE International Workshop on Advanced Motion Control (AMC)*. IEEE, 2010, pp. 572–576.

- [17] Y. Munakata, A. Tanaka, and M. Wada, “A five-wheel wheelchair with an active-caster drive system,” in *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.
- [18] A. Murai, M. Mizuguchi, M. Nishimori, T. Saitoh, T. Osaki, and R. Konishi, “Voice activated wheelchair with collision avoidance using sensor information,” in *ICCAS-SICE, 2009*. IEEE, 2009, pp. 4232–4237.
- [19] A. Lankenau and T. Röfer, “Smart wheelchairs - state of the art in an emerging market,” *KI*, vol. 14, no. 4, pp. 37–39, 2000.
- [20] Y. Misono, Y. Goto, Y. Tarutoko, K. Kobayashi, and K. Watanabe, “Development of laser rangefinder-based slam algorithm for mobile robot navigation,” in *SICE, 2007 Annual Conference*. IEEE, 2007, pp. 392–396.
- [21] B.-F. Wu, C.-L. Jen, W.-F. Li, T.-Y. Tsou, P.-Y. Tseng, and K.-T. Hsiao, “Rgb-d sensor based slam and human tracking with bayesian framework for wheelchair robots,” in *Advanced Robotics and Intelligent Systems (ARIS), 2013 International Conference on*. IEEE, 2013, pp. 110–115.
- [22] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.
- [23] —, “Real-time obstacle avoidance for fast mobile robots in cluttered environments,” in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE, 1990, pp. 572–577.
- [24] A. Fattouh and S. Nader, “Preliminary results on dynamic obstacle avoidance for powered wheelchair,” in *2006 2nd International Conference on Information & Communication Technologies*, vol. 1. IEEE, 2006, pp. 849–853.

- [25] O. H. Jafari, D. Mitzel, and B. Leibe, “Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5636–5643.
- [26] T. H. Nguyen, J. S. Nguyen, D. M. Pham, and H. T. Nguyen, “Real-time obstacle detection for an autonomous wheelchair using stereoscopic cameras,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 4775–4778.
- [27] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, “Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 873–878.
- [28] H. Badino, U. Franke, and R. Mester, “Free space computation using stochastic occupancy grids and dynamic programming,” in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, vol. 20, 2007.
- [29] S. Zeng, “A tracking system of multiple lidar sensors using scan point matching,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2413–2420, 2013.
- [30] W. T. S. Standard, “Wc19: Wheelchairs,” <http://wc-transportation-safety.umtri.umich.edu/wts-standards/wc19-wheelchairs>, 2012, [Online; accessed 28-May-2016].
- [31] K. Komoto and J. Suzurikawa, “Feasibility test and characterization of wheelchair everyday life log with smartphone-based electronic recording equipment,” in *System Integration (SII), 2012 IEEE/SICE International Symposium on*. IEEE, 2012, pp. 176–181.
- [32] A. F. Marquez, M. Castillo-Effen, S. Fitzgerald, and W. A. Moreno, “Motion-logger: An attitude and motion sensing system,” in *2011 50th IEEE Conference*

- on Decision and Control and European Control Conference.* IEEE, 2011, pp. 5311–5316.
- [33] J. Pineau, A. K. Moghaddam, H. K. Yuen, P. S. Archambault, F. Routhier, F. Michaud, and P. Boissy, “Automatic detection and classification of unsafe events during power wheelchair use,” *IEEE journal of translational engineering in health and medicine*, vol. 2, pp. 1–9, 2014.
- [34] A. Hussein, P. Marín-Plaza, D. Martín, A. de la Escalera, and J. M. Armingol, “Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection,” in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 104–109.
- [35] C.-C. Wang and C. Thorpe, “Simultaneous localization and mapping with detection and tracking of moving objects,” in *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 2918–2924.
- [36] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, “Terrain mapping for off-road autonomous ground vehicles using rational b-spline surfaces and stereo vision,” in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 648–653.
- [37] H. Jaspers and H.-J. Wuensche, “Fast and robust b-spline terrain estimation for off-road navigation with stereo vision,” in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 140–145.
- [38] C. Chen, Y. He, F. Gu, C. Bu, and J. Han, “A real-time relative probabilistic mapping algorithm for high-speed off-road autonomous driving,” in *Intelligent Robots*

- and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 2015, pp. 6252–6258.
- [39] M. Herbert, C. Caillas, E. Krotkov, I.-S. Kweon, and T. Kanade, “Terrain mapping for a roving planetary explorer,” in *Proceedings, 1989 International Conference on Robotics and Automation.* IEEE, 1989, pp. 997–1002.
- [40] K. Maatoug, M. Njah, and M. Jallouli, “Autonomous wheelchair navigation in indoor environment based on fuzzy logic controller and intermediate targets,” in *Advanced Systems and Electric Technologies (IC_ASET), 2017 International Conference on.* IEEE, 2017, pp. 55–59.
- [41] R. Zhang, Y. Li, Y. Yan, H. Zhang, S. Wu, T. Yu, and Z. Gu, “Control of a wheelchair in an indoor environment based on a brain–computer interface and automated navigation,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 24, no. 1, pp. 128–139, 2016.
- [42] Y. Nakayama, H. Lu, J. K. Tan, and H. Kim, “Environment recognition for navigation of autonomous wheelchair from a video image,” in *Control, Automation and Systems (ICCAS), 2017 17th International Conference on.* IEEE, 2017, pp. 1439–1443.
- [43] Y. K. Lee, J. M. Lim, K. S. Eu, Y. H. Goh, and Y. Tew, “Real time image processing based obstacle avoidance and navigation system for autonomous wheelchair application,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017.* IEEE, 2017, pp. 380–385.
- [44] B. Akar, U. Yayan, H. Yücel, V. Bayar, and A. Yazici, “Akıllı tekerlekli sandalye kontrolü için sonlu durum makinesi tasarımı,” *Otomatik Kontrol Ulusal Toplantısı TOK 2013*, 2013.

- [45] C. Yeniçeri, T. Tuna, A. Yazıcı, H. Yücel, U. Yayan, and V. Bayar, “A smart solution for transmitter localization,” in *2013 IEEE INISTA*. IEEE, 2013, pp. 1–5.
- [46] i. Encyclopaedia Britannica, “Trilateration,” <https://www.britannica.com/science/trilateration>, 2016, [Online; accessed 26-May-2018].
- [47] U. Yayan, B. Akar, F. Inan, and A. Yazici, “Development of indoor navigation software for intelligent wheelchair,” in *Innovations in intelligent systems and applications (INISTA) proceedings, 2014 IEEE international symposium on*. IEEE, 2014, pp. 325–329.
- [48] H. Grewal, A. Matthews, R. Tea, and K. George, “Lidar-based autonomous wheelchair,” in *Sensors Applications Symposium (SAS), 2017 IEEE*. IEEE, 2017, pp. 1–6.
- [49] H. S. Grewal, N. T. Jayaprakash, A. Matthews, C. Shrivastav, and K. George, “Autonomous wheelchair navigation in unmapped indoor environments,” in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2018, pp. 1–6.
- [50] RoboteQ, “Xdc24xx brushed dc motor controller datasheet,” <https://www.roboteq.com/index.php/docman/motor-controllers-documents-and-files/documentation/datasheets/xdc2xxx-datasheet/199-xdc2230-datasheet/file>, 2018, [Online; accessed 26-May-2018].
- [51] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press, 2010. [Online]. Available: <https://books.google.ca/books?id=bZx8tAEACAAJ>
- [52] A. Kelly, *Mobile Robotics: Mathematics, Models, and Methods*. Cambridge

- University Press, 2013. [Online]. Available: <https://books.google.ca/books?id=laxZAQAAQBAJ>
- [53] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006. [Online]. Available: <https://books.google.ca/books?id=-PwLBAAAQBAJ>
- [54] B. Furht, *Encyclopedia of Multimedia*, ser. Encyclopedia of Multimedia. Springer US, 2008. [Online]. Available: https://books.google.ca/books?id=Ipk5x-c_xNIC
- [55] S. Lab, “Zed stereo camera,” <https://www.stereolabs.com/zed/>, 2018, [Online; accessed 26-July-2016].
- [56] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, ser. Intelligent Robotics and Autonomous Agents series. MIT Press, 2011. [Online]. Available: <https://books.google.ca/books?id=4of6AQAAQBAJ>
- [57] K. Nirmal, A. Sreejith, J. Mathew, M. Sarpotdar, A. Suresh, A. Prakash, M. Saffonova, and J. Murthy, “Noise modeling and analysis of an imu-based attitude sensor: improvement of performance by filtering and sensor fusion,” in *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II*, vol. 9912. International Society for Optics and Photonics, 2016, p. 99126W.
- [58] P. Furgale, T. D. Barfoot, and G. Sibley, “Continuous-time batch estimation using temporal basis functions,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2088–2095.
- [59] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

- [60] C. Golban, P. Cobarzan, and S. Nedeveschi, “Direct formulas for stereo-based visual odometry error modeling,” in *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2015, pp. 197–202.
- [61] B. Anderson and J. Moore, *Optimal Filtering*, ser. Dover books on engineering. Dover Publications, 2004. [Online]. Available: <https://books.google.ca/books?id=v7KcAQAACAAJ>
- [62] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, “The unscented particle filter,” in *Advances in neural information processing systems*, 2001, pp. 584–590.
- [63] S. J. Julier and J. K. Uhlmann, “New extension of the kalman filter to nonlinear systems,” in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–194.
- [64] N. J. Gordon, D. J. Salmond, and A. F. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” in *IEE proceedings F (radar and signal processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113.
- [65] D. Fox, “Kld-sampling: Adaptive particle filters,” in *Advances in neural information processing systems*, 2002, pp. 713–720.
- [66] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” *AAAI/IAAI*, vol. 1999, no. 343-349, pp. 2–2, 1999.
- [67] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, “A flexible and scalable slam system with full 3d motion estimation,” in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

- [68] G. Grisetti, C. Stachniss, W. Burgard *et al.*, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE transactions on Robotics*, vol. 23, no. 1, p. 34, 2007.
- [69] T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 335–348.
- [70] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, 2013, software available at <http://octomap.github.com>. [Online]. Available: <http://octomap.github.com>
- [71] M. Labbé, “Rtab-map: Real-time appearance-based mapping,” <https://introlab.3it.usherbrooke.ca/mediawiki-introlab/index.php/RTAB-Map>, 2019, [Online; accessed 27-Sept-2018].
- [72] R. Toldo, U. Castellani, and A. Fusiello, “A bag of words approach for 3d object categorization,” in *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*. Springer, 2009, pp. 116–127.
- [73] M. Labbe and F. Michaud, “Appearance-based loop closure detection for online large-scale and long-term operation,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013.
- [74] M. Drwiega and J. Jakubiak, “A set of depth sensor processing ROS tools for wheeled mobile robot navigation,” *Journal of Automation, Mobile Robotics & Intelligent Systems (JAMRIS)*, 2017, software available at http://github.com/mdrwiega/depth_nav_tools.

- [75] Ontario, *Ontario Building Code 2012: containing the Building Code Act, 1992, and O. Reg. 332/12 in effect January 1, 2014*. [Toronto], Ont: Queen’s Printer for Ontario, 2015.
- [76] N. R. C. of Canada, C. C. on Building, and F. Codes, *National building code of Canada, 2015*, 14th ed. Ottawa, Ont: National Research Council Canada, 2015.
- [77] E. Marder-Eppstein, “move_base - ros wiki,” http://wiki.ros.org/move_base, 2018, [Online; accessed 27-Sept-2018].