Autonomous mobility scooters as assistive tools for the elderly

ANTHONY NTAKI

A thesis submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of

Master of Philosophy (MPhil)

April 2013

Abstract

The aim of this research is to investigate the development of an autonomous navigation system that could be used as an assistive tool for elderly and disabled people in their activities of daily living. The navigation environment is an urban environment and the platform is a Mobility Scooter (MoS).

To achieve this aim, a differentially steered MoS was modified to receive motion commands from a computer and outfitted with onboard sensors that included a Global Positioning System (GPS) receiver and two 2D planar laser range sensors. Perception methods were developed to detect the presence of an outdoor pedestrian walkway. These methods achieved this by processing the range data produced by the laser sensors to identify features that are typically found around walkways like curbs, low vegetation, walls and barriers. A method that utilises GPS localisation information to plan and navigate a route in an outdoor urban environment was also developed.

Extensive experimental work was conducted to test the accuracy, repeatability and usefulness of the sensory devices. The developed perception methodologies were evaluated in real world environments while the navigation algorithms were predominantly tested in virtual environments.

A navigation system that plans a route in an urban environment and follows it using behaviours arranged in a hierarchy is presented and shown to have the ability to safely navigate an MoS along an outdoor pedestrian path.

Publications

The following publication has been published as a direct result of this thesis:

Refereed Conference Paper

Anthony Ntaki, Ahmad Lotfi, Caroline Langensiepen, "Autonomous Mobility Scooter as an Assistive Outdoor Tool for the Elderly", in Smart Design, First International Conference Proceedings, Philip Breedon (Ed.), pp. 115-125, Springer-Verlag London 2012. [doi: 10.1007/978-1-4471-2975-2_14]

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

2D Two-Dimensional

3D Three-Dimensional

GPS Global Positioning System

Hz Hertz

MoS Mobility Scooter

NMEA National Marine Electronics Association

PCD Point Cloud Data

SLAM Simultaneous Localisation And Mapping

UAS User Assisted Mobility Scooter

V-REP Virtual Robot Experimentation Platform

Chapter 1

Introduction

This project aims to investigate and evaluate outdoor autonomous navigation as an assistive tool for the elderly and disabled people. The research will achieve this by developing an outdoor autonomous navigation system that utilises the current machine learning and sensor fusion techniques. The research will then move on to implement the developed system onto a mobility scooter for the purposes of investigating the feasibility of autonomous navigation in an outdoor urban environment.

1.1 Research Motivation

The elderly population is increasing all over the world. This has been attributed to the decline in fertility rates, the reduction in mortality rates and the increase in longevity [1]. Currently, the elderly population accounts for 11% of the total population. This figure is projected to rise to 22% by 2050. Additionally, this population is ageing. The proportion of the elderly population that is above 80 years is 14% but this figure is expected to rise to 20% by 2050. Meanwhile, the population growth of the young is projected to level off and by 2047 the population growth of the elderly will surpass that of the young. This has led to a decline in the support ratio especially in the developed and developing countries. The support ratio is defined as the number of individuals aged 15 - 65 for every person aged above 65 years. This has led to the elderly having to be more independent in performing their day to day activities.

With the increase in the ageing population, there has also been an increase in Mobility Scooter (MoS) usage. This electric powered mobility device is the preferred mode of transportation for the elderly in the urban environment. The reason for this is due to its ease of use and compact nature allowing it to be used both indoors and outdoors. The MoS also imbues the elderly with a sense of independence.

According to the UK Road Traffic Act (1988) [2], a MoS is defined as an invalid carriage. This means that it is a mechanically propelled vehicle whose weight should not exceed 254 kg. A MoS can be classified as either class 2 or 3 invalid carriage depending on its features and abilities. Those that belong to class 2 have a maximum speed of 4 mph and their weight is not allowed to exceed 113.4 kg. They are required to use pedestrian access areas and they are not allowed on the road unless it is to cross to the opposite sidewalk. For class 3, the MoS must have a maximum speed of 8 mph and allowed a maximum weight of 150 kg. These can use the road but not the motorway. In cases where the MoS uses a walkway or any other space that is reserved for pedestrians, it is required to keep the speed below 4 mph.

This increase in MoS usage led to a revisit of the UK legislation concerning its classification. The concern was enhanced by a few more factors:

- The majority of the users are the elderly who are prone to diminished cognitive abilities.
- There is no training requirement for using a MoS.
- There were reports of accidents that involved MoSs. These accidents were determined to be due to user error.

To aid with the UK legislation process two independent reports are already published [3, 4]. The first report was published in 2005 and it reviewed the classification of a MoS as an invalid carriage. Interviews were carried out of users, their family members and any other volunteers. Among the questions asked were if they thought that training was an appropriate requirement for the scooter users. They were also asked if they viewed the scooter as a potential hazard in pedestrian reserved spaces. Recommendations from this report included the introduction of a training scheme for scooter usage and the introduction of third party insurance for MoSs. The second report was published in 2010. From this report, it was decided that starting from 2013, a new category in the National Accidents Statistics Database be introduced. This category would record accidents that involved MoSs leading to a more complete analysis to inform future legislation efforts. Another recommendation was that regulators seek the views of the MoS users before introducing any new regulations that would affect MoS usage.

From these reports, it was determined that there is indeed a danger that the scooter poses to the other road users based on the fact that the scooter users are susceptible to cognitive deficiencies. The introduction of regulations could possibly lead to reduced scooter usage thereby depriving some people of the independence that MoS usage provides. So as the regulators endeavour to create regulations that would encourage MoS usage while mitigating the potential hazard that they cause, this project proposes a solution to try and bridge this gap.

To meet these challenges, the research question is that if a User Assisted mobility Scooter (UAS) would be a plausible solution, what are the challenges to develop such a system and how could it be implemented. Therefore the aim is to have a mobility scooter that is capable of autonomous navigation.

In literature, other terms such as Electric Wheelchair is also used for MoS. In this thesis we refer to the developed system as UAS or MoS. The UAS could provide the user with navigation assistance in terms of directions and it could provide an early warning detection system and end up making the usage of MoSs safer. This safety could extend to both the user and the pedestrians that share the space with the scooter. The navigation system could provide directions back to the user's home if they were unable to recall their way back. By adding a system that is capable of local and global navigation, this would provide confidence and safety to the user and the pedestrians.

An autonomous navigation system can be seen as any system that allows a vehicle to move in an environment with minimal human intervention. This has been achieved by installing sensors on or off the vehicle that monitor the internal and external environment of the vehicle. The data gathered by the sensors is used to provide navigation decisions to the vehicle. It has been considered vital for situations that call for exploration of environments that are hazardous to humans.

These could be deep sea, mining, active volcanoes and outer space. Autonomous navigation also proves to be useful in situations where processing large amounts of data in a limited time could improve safety like urban traffic environments.

The type of environment to be navigated affects the approach that is used to tackle the autonomous navigation problem. This is mainly due to the fact that environments provide features that tend to be specific to those kinds of environments. An example of this can be seen with indoor and outdoor environments. The indoor environments provide straight edges in form of walls, doors, chairs and tables while outdoor environments include wide open spaces with occasional straight edges.

1.2 Aims and Objectives

The aim of this research is to investigate how autonomous navigation can be used as an assistive tool for the elderly. The navigation environment is an urban environment and the platform is a MoS .

To accomplish this aim, the following objectives are identified:

1. Analyse the motion control system of the MoS. Through this analysis, several options will be identified that allow for the MoS to be controlled by a computer. Preference is given to the computer control system that has the least complexity. The computer motion control will endeavour to mimic the motion control signals that are produced by the user interface of the MoS.

2. Investigate the different types and variations of sidewalks and walkways that are available. Through this investigation, the characteristics of the

sidewalk and the walkway are determined which leads to a generalisation of these pedestrian reserved areas. This also provides an identification of the features that commonly surround sidewalks and walkways.

3. Investigate the methods used to detect the presence of environment features in an urban outdoor environment. The focus of this investigation will be on using onboard sensors like lasers sensors, ultrasound sensors and vision sensors to determine the features that are present in the environment surrounding the MoS. Through this investigation, the methods best suited for urban outdoor environment navigation will be identified. The criteria for the selected methods will include the ability to detect the presence of the features in their various iterations with a high enough confidence so as to allow for safe and accurate navigation.

4. Investigate the possible methods for localising the scooter on the walkway. From this investigation, a method that localises the MoS on a pedestrian walkway and sidewalk will be devised. This method will use the detected features to determine first the width of the path then move on to determine the location of the scooter on the path. This method will be used to enable the scooter to follow the shape of the path while in motion. This will be done such that when the path is straight the scooter will move straight on the path and when the path is curved the scooter will curve in the direction of the path. This method will fuse the information from the Global Positioning System (GPS) with the laser sensor data to determine the location of the scooter and the relative features that are around.

5. Investigate the methods that can be used to detect and classify obstacles.

This investigation will focus on methods that use range data to Devise an algorithm to detect and avoid obstacles that lie within the walkway or sidewalk. This algorithm uses the onboard sensors to detect the presence of obstacles, then determine if the obstacles lie within the borders of the walkway and then finally generate an appropriate plan to avoid the obstacle. The algorithm should have the ability to classify the detected obstacle as either static or dynamic.

6. Investigate ways of generating navigation routes for an urban outdoor environment. Then using the results from this investigation, a method to generate a route that is appropriate for a MoS will be devised. This method should be able to take as input the start and final destination in the format of GPS coordinates and then generate a route that is defined by GPS waypoints.

7. Devise an algorithm that allows the scooter to cross the road. This algorithm should be able to enable the scooter to identify a need for crossing. Then the method should search for areas that are most suitable for crossing. Then execute the cross of the road to get to the walkway that is located at the opposite side of the road.

1.3 Major Contributions of the Thesis

The main contributions of the thesis are listed below.

This thesis provides a method for modifying a MoS so that it can be controlled by a computer. The method used relies on first identifying how the user's commands are converted into drive signals to the motors. This information is then used to develop a system that can allow the computer to generate these signals and transmit them to the motor controller of the scooter.

This thesis presents a method for identifying an urban outdoor pedestrian walkway. By detecting the features that surround a typical walkway. The method identifies the different variations of walkways that may exist in an urban environment. This is done with the use of laser range readings that detect the nature of the surface and the horizontal plane around the scooter.

The next contribution is an algorithm for crossing the road to move to another walkway that is opposite the road. This uses a method to detect the side of the road that the walkway is located. This information is then combined with the curb location to determine the area of the walkway where the dropped curb should be searched for. The method then commands the scooter to cross at the point when it detects the presence of a dropped curb.

This presents also a method for navigating in an urban outdoor environment on the pedestrian accessible areas. The navigation system generates a pedestrian route to the destination from the starting point. This route is then followed by navigating on the pedestrian walkway and occasionally crossing the road when the need arises.

1.4 Thesis Layout

The rest of the thesis layout is as follows.

Chapter 2 presents the literature review. In this chapter autonomous navigation is defined by reviewing how other investigators have approached the problem.

It also presents the different methods used by other research groups to achieve autonomous navigation in the outdoors urban environment.

Chapter 3 presents the mobile platform that is used for the experiments. This chapter shows how the mobility scooter platform was modified for computer control. The components that are used to make the modification are introduced. The sensors that are used are also introduced and explained. These sensors include the laser sensors and the GPS receiver. The system architecture is also presented showing how all the components are interconnected to form the experiment platform.

Chapter 4 shows how the outdoor urban environment was characterised. The specific part of the environment that is characterised is the pedestrian walkway. The chapter then shows the features that are found around the walkway. This chapter also presents methods that are used to detect the presence of these features. The methods rely on processing laser range data.

Chapter 5 presents the method used for localisation in the urban environment. This shows how GPS is used to achieve the function of localisation in an outdoor urban environment. The chapter presents the reliability and the precision of this localisation method.

Chapter 6 presents the methods for navigation in an outdoor urban environment. The chapter presents the route generation method. The chapter shows how this generated route is used to navigate the scooter. The method for crossing the road is also presented. The chapter presents the autonomous navigation architecture.

Chapter 8 the conclusions are presented in this chapter. The conclusions discuss the evaluations of the navigation algorithm. The limitations are also presented. The recommended steps to improve the navigation algorithm are presented.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a review of the literature on autonomous navigation and how it can be implemented. It starts by providing a description of autonomous navigation and the elements that make it up. These elements are perception, localisation, environment representation and navigation architecture. The elements are presented in more detail showing the methods that have been used to achieve them.

2.2 Autonomous Navigation

Autonomous navigation can be defined as a methodology that allows to guide a mobile robot to accomplish its mission in an environment that contains obstacles in a good and safe way [5]. The navigation problem in the robotics field has been seen as providing the robot with the ability to answer a series of questions. In [6] autonomous navigation is broken down into three questions; "Where am I?", "Where am I going?" and "How do I get there?". Similarly in [7] the task of autonomous navigation is broken down into four questions; "Where am I?", "Where have I been?", "Where am I going?" and "What is the best way to get there?". Siegwart [8] presents navigation as made up of four building blocks. These blocks include; perception, localisation, cognition and motion control. While Reignier [9] presents navigation as an incremental process that includes four steps. These steps being environment perception, localisation, motion decision and planning and motion execution.

From the above definitions we can view that a big part of approaching the task of autonomous navigation is how you break it down into smaller sub tasks. Following this, we are going to define the problem of autonomous navigation as made up of perception, localisation, environment representation and navigation architecture. The perception of the robot involves sensors and sensor data processing. The sensors gather data from the surrounding environment. Gathered data is then processed to extract features that may have been detected.

Localisation is the process of determining the robot's position within the environment. This position can be an absolute location or relative to a specific environment landmark. This position is updated as the robot performs its tasks within the environment.

Environment representation refers to the way in which the surrounding environment is presented to the navigation system for planning. This process can also be seen as mapping. It is mentioned in [8] that the accuracy of the representation has to match the accuracy of the data returned by the sensors and the accuracy with which the robot achieves its goals. The complexity of this representation also has a direct impact on the computational complexity of the navigational system.

The navigation decisions and tasks are organised in a structure that is known as a navigation architecture. This shows how the data is organised and used to enable the vehicle to navigate in its environment. There are a great number of architectures that are used but they could be seen as falling into three major categories. These are the reactive, deliberative and the hybrid navigation architectures. The differences between them arising due to how the sensor data is used to navigate the vehicle in the environment.

2.3 Autonomous Navigation Applications

Autonomous navigation has been applied to different vehicles travelling in different mediums operating in different fields. The reasons for this include improving safety, productivity, precision and endurance.

In construction, industrial manufacturing and mining automated ground vehicles are used to transport materials between specified locations. In mining these automated vehicles are referred to as haulers and they transport materials from the mining site to locations were these materials are deposited. An example of these are the autonomous trucks from Komatsu that have the capability of hauling up to 290 tonnes in complex mining environments while being supervised by a central computing system [10]. In these cases, the vehicles allow for production to carry on despite the lack of sufficient man-power.

Autonomous navigation is also used in agriculture in the form of automated ground vehicles capable of ploughing large fields of crops without human intervention [11]. These vehicles use a combination of satellite positioning and closed feedback control strategies to follow a route in varying terrain. In agriculture, autonomous aerial vehicles are used to spray crops and in some instances to surveil farms for the purposes of monitoring crop yield and track livestock movement.

In exploration of harsh environments autonomous navigation has been used to endow vehicles with the capability of gathering information from the surrounding environments with minimal to no human supervision. An example of this is seen in deep sea exploration, outer space and other planet surfaces. In these instances, the risk to human life that these environment pose is eliminated.

Autonomous navigation has been used in domestic appliances like the vacuum cleaner and the lawn mower. The autonomous vacuum cleaner, the most popular of which is the Roomba made by iRobot [12], has the capability of cleaning a typical room while navigating around obstacles like table and chair legs. The area of operation in these instances could be demarcated by structures that are placed in the environment to assist the navigation system to identify the operation area. These structures could include infrared emitting devices or ground placed cables.

Autonomous motor vehicles are another application of autonomous navigation that has advantages that includes improvements in road safety and fuel efficiency to mention a few. This research has received support from organisations such as the Defence Advanced Research projects Agency (DARPA) in the United States that organised a series of competitions [13] that helped provide a renewed surge of research in the field. Other autonomous navigation competitions include the European Land-Robot (ELROB) trials [14] which come in two flavours, the military (M-ELROB) and civilian (C-ELROB) categories.

Autonomous navigation in urban traffic has produced vehicles able to detect the edges of the road and the lanes. This information is used to provide warnings to the driver of the vehicle and in some cases take control of the car and maintain movement within a particular lane. The research into autonomous navigation has led to the development of adaptive cruise control which is a system that allows a car to maintain a specified distance from the car infront of it in conditions of heavy traffic.

The move to introduce the autonomous vehicle as a part of the traffic is being pioneered by Google in the united states. They have successfully lobbied for legislation to legalise autonomous vehicles in Nevada, Florida and California. This legislation allows for the testing of autonomous vehicles in typical traffic conditions that are populated by other road users.

2.4 Sensors

Sensors are a critical component to navigation. They are transducers that measure a quantity. Although sensors vary in their functions and quantities that they measure, they are still subjected to performance evaluations. These performance factors include accuracy, precision, range, sensitivity, frequency and error. When it comes to autonomous navigation, some sensors have grown in popularity due to their ability to provide information about the internal and external environment of a mobile robot.

2.4.1 Laser Range Finder

The laser range finder which is also called laser sensor, laser radar, laser scanner or Light Detection And Ranging (lidar) has grown very popular in the mobile robotics community as a sensing device [15–17]. It is a non-contact optical device that measures the distance to an object using a pulsed laser beam [15]. It works by the principle of Time-Of-Flight (TOF) [18, 19]. It is made up of a source and a receiver. A pulsed laser beam is emitted by the source into the environment. When this beam strikes an object in the environment a portion of this beam energy is reflected back to the sensor and hits the receiver. The time it takes between emission and reception is considered as the time of flight of the laser beam. This time duration is recorded and using the speed of light through air, the distance to the object can then be calculated.

For a two-dimensional (2D) laser range finder such as those manufactured by SICK AG [20] and Hokuyo [21], the laser beam is reflected off a rotating mirror. This arrangement allows the sensor to acquire a fan-shaped 2D scan of the surrounding area. These usually have a scanning range between 180^o and 270^o.

For a three-dimensional (3D) lidar such as those manufactured by Velodyne [22], lasers are mounted in a spinning sensor block. The Velodyne HDL-64E, which the most popular of these sensors spins at up to $15Hz$, generates dense range Point Cloud Data (PCD) covering a 360^o horizontal field-of-view and a 30^o vertical field-of-view. It is commonly used for vehicle navigation [23–25] and is designed to withstand standard automotive G-forces [26].

2.4.2 Vision Sensor

The vision sensor is an attractive sensor because it emulates human vision. There are two competing technologies that are used in making vision sensors. These are Charged Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS) technology [8]. The difference between the two technologies is the way in which they collect data from the pixels. For CCD chip technology, pixel charges are frozen and moved to the part of the chip that reads the charges of the pixels. Care has to be taken in preserving these pixel charges long enough for them to be read. While for CMOS chip technology, each pixel has got circuitry that measures and amplifies the pixel charge. These charges are then transported to their destinations. The CMOS chip has the added benefit of not requiring the charges to be preserved but this pixel circuitry takes up valuable space on the chip and as a result it is unable to achieve the same pixel density resolution to a similarly sized CCD chip.

Vision sensors provide a rich representation of the environment but they are susceptible to noise from changing illumination conditions. This is made significant when the sensor is used for outdoor tasks. In the outdoors the position of the sun causes changes in illumination and also the structures that can be found in the outdoor environment can cause shades that are constantly changing positions. This is why this sensor is suitable for indoor use where the lighting conditions can be controlled.

A fundamental problem with visual images makes range finding relatively difficult. Any vision chip collapses the 3D world into a 2D image plane, thereby losing depth information. A solution to this is to recover depth by looking at several images of the scene to gain more information, hopefully enough to at least partially recover depth. Stereo vision is one of several techniques in which we recover depth information from two images that depict the scene from different perspectives. It works by having two cameras placed with their optical axes parallel, at a known separation [8, 27–29]. The cameras provide images of the environment at slightly different viewpoints. Points that are identified as appearing in both camera images are known as conjugate pairs. The difference in the relative positions of these points is calculated as the disparity. This disparity provides information on the depth of the points.

2.4.3 Structured Lighting Sensor

The structured lighting sensor contains a laser emitter and a camera. The laser emitter projects a known pattern, or structured light onto the environment. The reflected light is collected by a lens and projected onto a camera [8, 30]. The reflection of the known pattern is captured by a receiver and, together with known geometric values, the system can use triangulation to establish range measurements.

2.4.4 Ultrasound Sensor

Ultrasound sensors use time of flight of sound waves to measure range. The sensor sends out a packet of sound and then measures the amount of time it takes for the echo to travel back to the sensor [8]. Due to nature of sound waves to spread out as they travel through a medium, the range that is measured is for an area and not a point [31, 32]. This means that the precision of an ultra sound sensor is not as high as that of a laser range finder.

2.4.5 Global Positioning System

The Navstar Global Positioning System usually referred to as GPS is a space based radio navigation system that provides localisation information using the transmission of satellite signals [33]. It was made operational in 1993 by the United States Department of Defence (DoD). It was initially developed for military use but is now freely available for civilian navigation.

It is made up of at least 24 satellites that orbit every 12 hours. Each satellite continuously transmits data that indicates its location and the current time to GPS receivers. The GPS satellites synchronize their transmissions so that their signals are sent at the same time. When a GPS receiver reads the transmission of multiple satellites, the arrival time differences inform the receiver as to its relative distance to each satellite. By combining information regarding the arrival time and location of at least four satellites, the receiver can infer its own position.

The fact that the GPS receiver must read the transmission of four satellites simultaneously is a significant limitation. GPS satellite transmissions require direct line-of-sight communication with the satellite. Thus, in outdoor environments such as city blocks that have tall buildings or in dense forests where the view of the sky is limited, one is unlikely to receive signals from four satellites.

A few strategies have been developed to improve the resolution of GPS [34,35]. The most popular of these strategies is differential GPS (DGPS), which makes use of a second receiver that is static and at a known exact position [36–38]. A number of errors can be corrected using this reference, and so resolution improves to the order of $1m$ or less.

2.4.6 Inertial Measurement Unit

Inertial Measurement Unit (IMU) is a unit that contains gyroscopes and accelerometers [39,40]. A typical IMU contains 3 gyroscopes arranged orthogonally and three accelerometers also arranged orthogonally. With this unit, the motion of a robot can be tracked and used in determining its absolute position within the environment. This unit is sometimes referred to as Inertial Navigation System (INS) when it is used for mobile robot navigation.

Gyroscopes are sensors that are able to maintain their orientation relative to a fixed reference frame regardless of the movement of the mobile system. They are used to measure the angular velocity of the robot. They can be either mechanical or optical. Optical gyroscopes use two monochromatic light beams, or lasers, emitted from the same source, instead of moving, mechanical parts. These are built using microfabrication technology, thereby providing heading information with resolution and bandwidth far beyond the needs of mobile robotic applications [8].

An accelerometer is a device that measures the acceleration of a body along a single or along multiple axes. This acceleration is measured as a vector quantity with both magnitude and direction. Accelerometers are usually used in mobile robots to sense the orientation of the robot. They have been improved with the advancement of the Micro-Electrical-Mechanical System (MEMS) technology and they have been rapidly becoming smaller, more lightweight and more inexpensive [41].

2.5 Perception

Perception is the processing of sensor data to extract meaningful information about the surrounding environment. It involves filtering out the sensor noise from the sensor data and using mathematical methods to detect the presence of features in the sensor data. For the mobile robot navigating in an urban environment, the perception is tasked with extracting environment features from the laser sensor range data. The features that are extracted include the pedestrian walkway. This information is used to generate plans for navigation of the UAS within the urban environment.

The navigation system needs to navigate the scooter within an urban outdoor environment. Due to the UK legislation [2], the mobility scooter is allowed to move in the pedestrian accessible areas. This means that the scooter will have to identify these pedestrian accessible areas and restrict movement to those areas. The pedestrian areas include walkways and sidewalks. The task of the navigation system is then to identify these walkways and move along them to get to the prescribed destination.

The identification of the walkway is split into sections. The sections represent the various features that surround the walkway. These include the curb, vegetation (grass and hedges), walls and barriers. The following sections show how these features are detected by others in literature.

2.5.1 Curb Detection

Curb detection has mainly been investigated for the purpose of navigating autonomously on urban roads, using a wide range of sensing modalities and algorithms. In a typical urban road setting an experiment could involve having a car drive on a street with curbs located on both sides of the vehicle. These approaches are not entirely suited for navigation in pedestrian reserved areas found in urban environments. For pedestrian reserved areas curbs appear differently due to the fact that the mobile robot is moving along the walkway enabling it to have a different viewpoint. A review of the contributions in the field is presented together with their relation to our application.

In [42] authors present a method for detecting a curb or ramp using a Canesta TOF imaging range camera. The curb is detected by detecting individual planar patches from the returned sensor data using the Connected Component RANdom SAmple Consensus (CC-RANSAC) method which is a modified version of the classic RANSAC method. They argue that the CC-RANSAC method provides more accurate detection results with less iterations compared to the RANSAC method.

In [43] a method that uses cameras and laser sensors to detect and track the presence of a curb is presented. This method uses the laser to detect the presence of a curb. The position of this curb is compared to the image from the camera. The position of the curb is determined in the image and this information is used to estimate the shape of the road boundary. This method is shown to estimate the road boundary regardless of the direction of travel of the vehicle onto which the sensors reside.

A method to Detect a curb using a laser line stripper is presented in [44]. The previously detected curb positions are used to anticipate the future position of the curb. This method is susceptible to noise giving rise to false detections. This noise is due to the nature of the laser stripper sensor that uses a camera to detect the laser points.

Detecting curbs from 3D Point Cloud Data (PCD) is presented in [45]. This method extracts a 3D model of the curb from the cloud point data. Based on this model, the points are divided into separate regions representing the road and sidewalk. The parameters of the curb are then estimated using the points from these regions. This method can be used on any sensor that produces 3D PCD.

In [46] authors present a method for detecting the curb at the road side for the purpose of navigating a car in an urban environment. The method presented uses a laser sensor together with a CCD camera. The camera is used to provide a wider view of the road. This assists the navigation to view the trend of the curb. This information is used to anticipate the position of future curb detections which proves to be vital in situations like road junctions, intersections and corners. The method is implemented and tested in simulation and a prototype autonomous vehicle. This combination of laser and camera for curb detection displays improved curb following compared to laser only methods.

Superpixel segmentation and condition random field is used in [47] to detect the position of a curb in an urban environment. The method does not make any assumptions about the position of the curb next to the robot. The method is able to detect curb positions in complex areas such as T-junctions. However, due to the computation requirements of the algorithm, the curb detection method is unable to provide data for real time navigation.

Authors in [48] present a method that uses stereo cameras that produce dense stereo data. This data is then used to construct an elevation map of the surrounding environment. Edge detection is used on the elevation data to detect the presence of curbs in the environment. It uses multi frame temporal filtering on the curb points to reduce the number of false positives. The method is able to detect both straight and curved curbs in real time. The method finds challenges in detecting dropped curbs or curbs with significantly low heights.

Closer to our approach, Youjin et al [49] and Wende [50] present methods for curb detection that use a 2D plane laser. The laser is tilted down allowing it to return a profile of the environment surface. The curb is detected as having a distinct / significant height above the road surface. We actually take inspiration from their methods and develop a curb detection method for navigation on the sidewalks. Our method differs from theirs in that we use it for navigation along sidewalks while their methods are used for navigation along roads.

2.5.2 Vegetation Detection

In outdoor navigation scenarios, the classification of the terrain is important due to the fact that the majority of mobile robots have not been built to navigate over surfaces covered by low vegetation like grass. Navigation over these surfaces is undesirable because it causes the wheels to slip which causes errors in odometry. These surfaces could also lead to the mobile robot getting stuck. Additionally, moving over surfaces covered by vegetation is uncomfortable to the user. This provides a motivation for developing a method to detect the presence of vegetation such as grass in the environment.

There exist several approaches for detecting vegetation. They however fall into two broad categories. The first category uses the distinct reflective characteristics of chlorophyll to detect the presence of vegetation. The second category uses the distribution of laser range readings to determine the presence of vegetation.

2.5.2.1 Vegetation Indices

The detection of vegetation can be done by using a method that is used in the satellite imaging field. This method uses the unique reflective properties of chlorophyll to distinguish plants from the rest of the environment.

Plants contain water and chlorophyll. The chlorophyll strongly absorbs the red and blue wavelengths of the visible light spectrum. As a result, those wavelengths that are not absorbed by either the water or chlorophyll are reflected. This property has led to the development of a reflectance index for vegetation. One of the most popular indices used for remote sensing applications is the Normalized Difference Vegetation Index (NDVI) [51,52]. This index varies from -1 (blue sky) to 1 (chlorophyll-rich vegetation) [53].

Laser sensor measurements are used in [54] to detect the presence of low vegetation such as grass. They use the laser remission values of the returned laser measurements. These remission values are modelled as functions of distance, angle of incidence and material. Then using a vibration based terrain classification technique they use self supervised learning to train the vegetation classifier. However, the fact that this remission value is a function of the angle of incidence provides a problem. Vegetation is characterised as generally being made up of a large number of surfaces that are not aligned. This misalignment leads to non uniform angles of incidence for the laser beams. This makes the detection of vegetation using the remission values unreliable.

A comparison of the red and Near Infra Red (NIR) reflectance of a material to detect the presence of vegetation is reported in [55]. They use a camera and laser sensor to provide inputs to the perception system. The PCD collected by the laser sensor are superimposed onto the images from the camera. To this, the local image properties like colour and NDVI value are added. This information is then used to create a discretised space around the robot that is used for path planning in the environment. This technique is computationally costly compared to our application of navigation in an urban outdoor environment using a UAS.

2.5.2.2 Lidar Points Distribution

Lidar data is used to construct a PCD of the surrounding environment. The local distributions of the points in this cloud are used to classify the different features of the environment in [56]. These features include vegetation (tree trunks, bushes and grass) and bare solid ground. From their work, they show that points that produce a local flat distribution are characteristic of flat surfaces. Also that points that produce one-dimensional distributions indicate a tree trunk while points that produce an isotropic distribution that varies equally in all directions indicate the presence of a bush. For classification of the environment features, a Bayesian classifier that is trained with offline labelled data is used. This technique was further developed in [57] to include semantic interpretation of the features and high level geometric modelling of the environment.

An alternative approach is presented in [58] where they use a laser range sensor to distinguish between grass and rocks. They use the range distribution of the readings to determine the presence of vegetation. In their work the rocks or bare ground return regular distributions of range readings while grass returns spatially scattered readings.

Laser range measurements are used in [59] to detect sections of the environment that are either suitable or not suitable to navigation. The non suitable (non-navigable) areas considered in their definitions include low vegetation such as grass while the suitable (navigable) areas include pavements. The technique they use includes checking for flat surfaces and rough surfaces. They use a laser sensor tilted downwards pointing towards the ground infront of the robot. Flat surfaces are classified as those surfaces that return range readings that are well aligned and have minimal variance in height. Rough surfaces are those that return range readings that are not well aligned and have some variance in height. Hidden Markov Models are used to classify the returned range readings into either navigable or non-navigable areas.

2.5.3 Walls and Barrier Detection

When advanced autonomous robots navigate within environments, they have to be endowed with the ability to follow walls and barriers. Walls and Barriers are easily detected by range data and in some instances only 2D images. The range data provides a simplified method for their detection and it is simple to interpret by both the human and the robot navigation system. It is because of this reason that there are several methods developed to detect walls from range data as opposed to image data.

Walls show up as line segments in range data. Line segments are the most suitable geometric primitives that can be used to represent structures in urban environments. The challenge of wall and barrier detection then transforms into a line extraction problem of which several algorithms and methods exist. One of these algorithms is the Successive Edge Following (SEF) [60] algorithm which determines a line starts when two consecutive range points are apart by a distance that is greater than a defined threshold.

Another line extraction algorithm is the Line Tracking (LT) [61] algorithm, also known as Incremental. This builds a line model that passes through the first and second range points, it then goes on to successively add a new range point when the line criterion is validated. For instances when the line criterion is not validated, the line is terminated and a new one is started, this is repeated until the end of the dataset is reached.

The Iterative End Point Fit algorithm (IEPF) [62] is another recursive method for line extraction. It constructs a line using the first and last range points. Then, it finds the range point furthest from the line, and if it is far enough, two subsets are created taking this point as the splitting point. This process is performed recursively for all the subsets to the point when the validation criterion is unable to hold.

The RANdom SAmple Consensus (RANSAC) algorithm [63] uses a probabilistic approach that allows it to fit models despite the presence of data outliers. This method works by first constructing a line using two randomly chosen range points from an initial set. Then a consensus set of line inliers is created, and if this set is large enough, the line is re-adjusted to the points included in this consensus set. If the consensus set is not large enough, the process is repeated until a consensus set is attained or the number of loop iterations reaches a defined maximum value.

The main advantage of SEF, LT and IEPF is that they are computationally fast when fitting lines to range points. This speed however comes with a price, the methods tend to be non-robust. RANSAC on the other hand, is a robust method. However, due to the fact that RANSAC is a non-deterministic method, its results and processing time are not always the same. Because of this reason we use the SEF line extraction algorithm. It is fast and this enables it to keep up with the real time demands of the navigation. To improve the robustness, this perception method will be combined with information from other perception methods when making navigation decisions for the UAS.

However lines are not the only features that are extracted from the range data. Other features that are extracted include corners and circular features that could belong to pillars or lamp posts.

2.6 Localisation

Determining the position of a robot within its environment is a vital component to its navigation. It is through this information that the navigation instructions for the next movement are based. This is also known as localisation. It is determined differently based on whether the navigation is indoors or outdoors.

For outdoors, the preferred method is to employ GPS. This is the method that provides the simplest solution and it is readily available. It however has an approximate accuracy of $3m$ [33] making it unsuitable for precision navigation. The other limitation is that GPS updates are not more than $5Hz$. This does not allow for high speed navigation of the mobile robot. GPS requires that the receiver have an unobstructed view of the sky. For urban outdoor environments, this view is periodically obstructed by tall buildings or other infrastructure such as tunnels. GPS accuracy also deteriorates due to multipath effects and clock bias errors.

GPS is commonly combined with an Inertial Navigation System (INS), which is a system that incorporates accelerometers and gyroscopes. These are responsible for monitoring the vehicles linear accelerations and rotation rates then use this information to determine its position, velocity and attitude. However, the accuracy of INS deteriorates over time due to the integration of inherent sensor errors that include white noise, correlated random noise and bias instability. As long as the system receives periodic position updates from GPS, these errors are prevented from growing to disastrous levels. Research has been conducted into aiming to provide an optimal GPS/INS integrated module with Kalman Filter being the most popular [64, 65].

Authors in [66] provide a technique to deal with position error accumulation during GPS signal outages. They do this by designing an alternative INS/GPS integration scheme that incorporates artificial neural networks. This method proves superior to Kalman Filter based methods when the length of GPS outages exceeds 90 seconds.

A multi-sensor data fusion methodology for navigation systems is presented in [67]. Synthetic Aperture Radar (SAR) which is a sensing technique that uses active microwave imaging radar is fused with INS and GPS. They combine local decentralized fusion with global optimal fusion to enhance the accuracy and reliability of the integrated navigation system. They demonstrate that the developed INS/GPS/SAR integrated navigation system out performs an INS/GPS integrated system.

Input-Delayed Neural Networks (IDNN) is used in [68] to model the INS position and velocity errors based on current and some past samples of INS position and velocity. In their work, they demonstrated improvements in positioning accuracy for cases of tactical grade INS and long GPS outages.

A localisation method that fuses data from GPS and a wide angle stereo camera is presented in [69]. They use the visual information from the stereo camera to obtain vehicle dead-reckoning. By using a wide angle stereo camera instead of an IMU they are able to generate a map of the environment while being able to detect loop closings using visual appearance information.

2.7 Environment Representation

Environment representation refers to the process of the navigation system taking the sensor data to build a model of the environment. This is one of the main requirements for autonomous mobile robots. This model is closely related to the nature of the tasks that the robot is required to perform in the environment and the fidelity of the sensors. Sometimes the type of environment such as indoors or outdoors plays a factor in the type of representation that is used by the navigation system. This model is sometimes referred to as a map of the environment. Maps can be classified by the way in which they represent the collected information about the environmental obstacles and free space: maps can be topological or metric.

2.7.1 Topological Maps

Topological maps show the interconnections between features in the environment [70–73]. These kinds of maps are based on graph data structures and are usually not to scale. Environment features in this kind of map are usually nodes. The interconnections between the nodes are edges. The definition of a node is specified by the programmer. In some cases these nodes could be clearly defined locations while in other cases these nodes could be loosely defined locations within the environment.

2.7.2 Metric Maps

Metric maps are more spatial representations of the environments. They are usually in the form of points [74], line segments [75] or grids [76] and they are usually to scale. They are more readily constructed with range data. There are different types of these maps that are used in mobile robots. Point and line segment maps provide a more compact representation of the environment compared to grid maps. This makes them easier to manage.

2.7.3 Simultaneous Localisation and Mapping

It is often time consuming to manually construct a map of the environment. Map building by the robot has proven to be much faster than building the map manually by hand [77]. For this reason, the motivation becomes to provide the mobile robot with the ability to create its own map. This has led to the development of the field of Simultaneous Localisation And Mapping (SLAM) . It is a situation where the mobile robot builds a map of the surrounding as it navigates within the environment. To build this map, the robot needs to determine its position within the environment and how this position relates to the perceived features detected by the sensors. In [78] SLAM is defined as an online map creation process that involves building, extending and improving a map of the environment while the robot is moving and simultaneously localising the robot with respect to the map.

The challenge with this methodology of map making stems from the fact that the location data contains errors due sensor noise. This presents uncertainties in the robot position which, in turn, presents uncertainty in the robot location estimate. From this it can be seen that the landmark and robot pose estimates are dependent.

A Gaussian Probability Density Function (PDF) is used to represent the uncertainties and correlations associated with SLAM. This PDF is propagated using an Extended Kalman Filter (EKF). This form of SLAM is known as EKF-SLAM [79]. In [80] authors claim that EKF-SLAM is limited to environments that contain geometric-shaped landmark models. To overcome this limitation they present Scan-SLAM, a new generalization of SLAM that presents a new way to define arbitrary shaped landmark models.

A Relative Simultaneous Localisation and Mapping System (RSLAM) that uses a binocular stereo camera system is presented in [81]. They use a topo-metric environment representation that allows them to achieve real-time performance without placing severe limits on the size of the map that can be built.

To address the issue of SLAM being computationally costly, a set of optimal algorithms that lower the computational requirements without imposing any penalties on the accuracy of the results are presented in [79].

2.8 Navigation Architectures

The method in which the information is arranged and used to generate navigation decisions for the robot is referred to as the architecture. This architecture is what allows the robot to achieve its navigation goals in the environment that it is placed into. There are currently many architectures that are used by different users but most of them fall into 3 broad categories. These categories are the reactive, deliberative and the hybrid architectures.

2.8.1 Reactive Architecture

This architecture is inspired from entomology. It was suggested that organisms such as insects do not have a reasoning and planning structure but instead opting to react to the stimuli from the environment. They however exhibited intelligent behaviours that allowed them to interact with their environment. This architecture attempts to replicate this idea to impart intelligence to the mobile robot navigation. The architecture discards the modelling and planning phase from the decision making process and instead lets the navigation decisions come from the environment stimuli. Brooks [82] suggests that the best model of the world is the world itself. This architecture leads to a shorter decision making process. It is sometimes referred to as a behavioural based architecture.

Subsumption architecture is a sub category of the reactive architecture. It was developed by brooks [82] and it is the best known representative of the reactive architecture. In this architecture, planning is decomposed into a collection of concurrent layered behaviours. Each of these behaviours is connected to its own sensory input. These behaviours are responsible for controlling the robot. The behaviours are organised into a hierarchy and at any given time, only one behaviour is responsible for controlling the robot. The rest of the outputs from the other behaviours are suppressed. The advantages of using this type of navigation architecture include the following;

- The architecture allows for short response times. This enables the robot to cope with dynamic environments.

- Due to the independent nature of the behaviours, the controller is able to function even when one of the behaviours is unable to function.
- The incremental nature of the architecture allows it to be improved by adding behaviours.
- The architecture allows for simplified behaviour construction which reduces production costs and also allows for miniturisation.

Other navigational systems using reactive planning have been developed. These include Payton's reflexive behaviours [83], Kadonoff's arbitration strategies [84] and Arkin's motor schemas [85]. Although these methods have different ways of integrating, controlling and selecting primitive behaviours, they never the less share a common decomposition of motor action into a collection of primitives which can be closely tied to incoming sensory information.

The architecture's inability to reason or plan about the navigation however proves to be a limitation that is hard to overlook. This usually leads to unintelligent actions on the part of the robot. These could include navigating back and forth a section of the environment without making any overall progress to the goal of the robot. High level reasoning and planning is quickly becoming the rule and not the exception.

2.8.2 Deliberative Architecture

Deliberative architecture takes the sensor data and then places it in a deliberative component of the navigation. This component analyses the data and then generates navigation commands that are used to move the actuators of the robot. It allows for high level specifications during the planning phase enabling the robot to handle complex missions that could include various goals and constraints. The disadvantages of a purely deliberative architecture include the following;

- Navigation decisions are delayed. This deliberation process can take time to generate navigation decisions. During this time between the reception of the sensor data and the generation of the navigation command, the robot is unable to act.
- This system is unable to cope with unpredicted events. This is due to the long process that it takes to sense and react to a new situation.
- The modelling of the environment is not straight forward. Relating sensor data to objects in the real environment is a process that is prone to errors. These errors are from the inaccuracies of the sensor, the noise in the sensor and the spread nature of the data.
- The difference between the sensor data and the real world produces uncertainties in the robot positioning. It becomes necessary to account for these uncertainties in the modelling process to produce a more robust navigation. This however introduces more computational requirements that in turn increase the time it takes to produce navigational decisions.

One of the earliest mobile robots to employ this navigational architecture was Shakey [86]. The Stanford Cart [87] also used a deliberative architecture.

2.8.3 Hybrid Architecture

This architecture combines the features of reactive and deliberative architectures. The aim is to enable the robot to react to its sensor data in a way that advances its movement towards its ultimate destination. In [88] it is suggested that the union of these architectures does not necessarily guarantee a superior architecture. The union needs to ensure that the developed architecture is capable of robust navigational plan execution while taking into account a high level understanding of the nature of the world.

In [89] a hybrid architecture for robot control that combines a servo-control, subsumption and symbolic layer is presented. These layers are combined in a way that allows each of their advantages to be fully exploited. They achieve this by building situation recognizers that link the servo and subsumption layers, and event detectors that link the subsumption and symbolic layers.

2.9 Conclusion

This chapter presented the methods that have been used to achieve the different components that make up autonomous navigation. It shows that the outdoor contains features that can be detected by sensors and that these sensors can produce data that is processed to enable a mobile robot to perform autonomous movement in an outdoor environment in a safe and intelligent way.

To navigate in an urban outdoor environment, this research will follow a walkway by identifying and following a curb. This curb will be detected by a down facing laser sensor that will return laser range readings of the ground in front of the UAS. The difference in height that is exhibited by the curb will be used to identify its presence in the laser data.

Our method for pavement and vegetation discrimination draws inspiration from the lidar points distribution methods reviewed in Section 2.5.2.2 and uses the classification of smooth range readings to determine the presence of a walkway and uneven distribution of range readings to infer the presence of vegetation such as grass bordering the walkway.

GPS will be used to provide absolute localisation. However, due to the limitations presented in Section 2.6, the localisation technique of using only GPS will be used to serve in situations that will not require precise positional accuracies. The GPS will provide general positional information and for more precise localisation the navigation system will use the range data from the laser sensor to provide relative localisation. The use of more precise laser range data for localisation ensures that the UAS is able to determine its precise position in relation to the detected features in the environment like the walkway borders while maintaining a general sense of its global position.

Although SLAM is a robust mapping technique for mobile robot navigation it tends to be computationally costly. SLAM is also proven to be more suitable for indoor environments. For this research, the UAS will use a map that is provided before hand. Currently, there are several maps of urban outdoor environments that are easily accessible.

From the review presented in Section 2.8, we can extract that purely reactive systems though quick to provide navigation decisions in dynamic environments, require a model of the environment to facilitate intelligent navigation towards specified goals. And that an ideal navigation architecture needs to incorporate some form of high level reasoning about the environment. This combination of the two architectures needs to be performed in a manner that allows the system to draw from the advantages of both systems. And that we should be aware of the disadvantages that could arise from improper combination of reactive and deliberative architectures.

The chosen navigation architecture for the UAS will be a hybrid architecture that performs periodic high level planning. High level functions like route planning will invoke the deliberative component of the architecture. These functions will have lower temporal requirements. Lower level functions such as following walkway paths will use the reactive component of the navigation architecture. These functions with their higher temporal requirements will be able to allow the UAS to react in time to any dynamic aspects in the environment.

All these navigation strategies will be developed and tested on a mobile platform that is a modified MoS. The sensors for proximity detection and pavement detection are laser sensors that are chosen for their accuracy and precision while GPS localisation will be performed by a standard inexpensive GPS receiver.

Chapter 3

Mobility Scooter Platform and Sensors

3.1 Introduction

This chapter introduces the test platform and the sensors that are used to carry out the implementation of the autonomous navigation system. First a description of a Mobility Scooter (MoS) that is of the same type as the experimental platform is presented. Then the chapter moves on to describe how the control system of this MoS can be modified to receive commands from a laptop computer. The chapter then moves on to introduce the onboard sensors and how they were mounted onto the mobile platform. Finally, a brief outline of the system architecture is provided showing the connectivity of the components that make up the mobility scooter test platform.

(a) (b)

Figure 3.1: Typical examples of mobility scooters (a) Class 2, (b) Class 3.

3.2 Mobility Scooter

A MoS is an electrically powered mobility aid like those shown in Figure 3.1. It is used by people with fatigue based conditions and people with mobility impairments. A MoS can be classified as either belonging to class 2 or 3 invalid carriage depending on its features and abilities. For class 2 mobility scooters they are limited to a top speed of $4mph$, and they are only allowed to travel along pedestrian reserved areas like pavements. While class 3 mobility scooters that are slightly bigger than class 2 mobility scooters and they have a top speed of $8mph$. Class 3 mobility scooters can travel along the road except on motorways.

This project is going to focus on the navigation of Class 2 mobility scooters. The scope of the developed navigation system will be limited to navigation outdoors on pavements and other pedestrian reserved areas. Out of scope will include the navigation of Class 3 mobility scooters that travel along roads among other road traffic.

The experimental platform used to design, develop and test the navigation system will be carried out on a Pride Jazzy powerchair which belongs to the Class 2 invalid carriage category [90]. This type of MoS was chosen for its excellent maneuverability and its differential steering ability. A few modifications are carried out on this powerchair to enable it to perform this function. But first we present a description of a typical powerchair. Then we move on to show how it is modified for assistive navigation testing.

3.2.1 Drive

Powerchairs generally have four or six wheels. Some powerchairs have the ability to fold while others can be partially dismantled to allow for easier transit. For a four wheeled powerchair, two drive systems exist, the front and back wheel drive. The powered drive wheels propel the powerchair while the other two wheels are caster wheels. The drive wheels are bigger than the caster wheels. For a six wheel layout, the powered wheels are positioned in the middle of the powerchair while the caster wheels are located at the front and rear.

3.2.2 Power

The electric motors of powerchairs are powered by two 12 volt rechargeable deep cycle batteries with a rating of 12 to 80 ampere-hours. These batteries are designed to provide enough power to last throughout a typical day of usage. Many powerchairs come with a charging unit that allows the powerchair to be charged from a standard wall outlet.

Figure 3.2: Mobility scooter experimental platform.

3.2.3 Controller

The powerchair controller is usually an arm rested joystick. With this controller, the user is able to control the movement of the powerchair including turning the powerchair within its own length. The controller may also have additional controls to allow the user to configure the powerchair to their needs like adjusting the sensitivty. Alternative controllers such as sip-and-puff controllers, that work by blowing into a sensor are available for users unable to use a hand controller.

3.2.4 Environment

The typical dimensions of a powerchair coupled with its differential steering ability allow it to manuever in tight environments. They are designed for indoor and outdoor use. Powerchairs are intended for pavement use while outdoors.

3.3 Mobile Platform Modification

The experimental platform shown in Figure 3.2 is modified from a Pride Jazzy powerchair. This modification is divided into two sections. The first section shows how the motion controller of the MoS is modified to allow for computer controlled motion instead of a joystick controller. The second section presents the mechanical modification that involves removing the seat of the MoS and replacing it with an aluminium box that houses the sensors and other electronic components.

3.3.1 Controller Modification

In a normal setting, the MoS is controlled by a user who inputs motion commands through a joystick. These commands are converted into signals that enable the motors to move. To control the movement of the MoS using a computer, a strategy is developed to enable the computer to generate these signals. These generated signals are then passed on to the MoS to move the motors.

To achieve this task, a check is made to find an ideal place in which to interface the computer generated signals. The control of this particular MoS which is a pride jazzy powerchair is made up of a joystick, a Pilot-Plus module and a motor controller also known as a power module as illustrated in Figure 3.3. The joystick connects to the Pilot-Plus module which is a control module that acts as an interface to the user. It displays the status of the MoS in form of messages. These messages could include the battery level, the selected mode and the current drive state. This Pilot-Plus module is connected to the motor controller which connects directly to the motors of the MoS. The motor controller is a plastic black

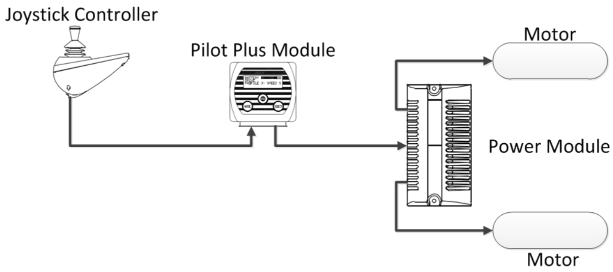
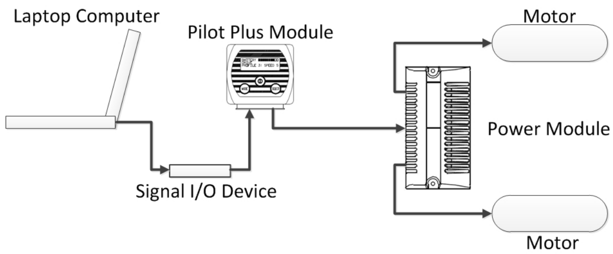
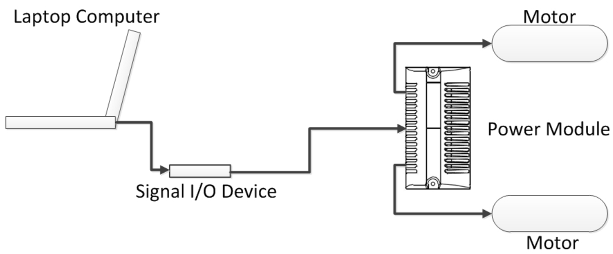


Figure 3.3: Configuration of scooter controller.



(a)



(b)

Figure 3.4: Configuration of scooter control by computer (a) option 1, (b) option 2.

box that generates the PWM signals that are used to run the motors.

From the presented configuration shown in Figure 3.3, it can be seen that there are two places where the computer can be placed to control the MoS. A connection could be made to the Pilot-Plus module as illustrated in Figure 3.4a or to the motor controller also illustrated in Figure 3.4b. The connection to the Pilot-Plus module is however the less complicated of the two options. Because of

Table 3.1: Table from pilot plus manual showing the pin assignments of the 9-way D-Type connector port.

pin	Analog Function	Digital Function
1	Joystick Speed	Forward
2	Joystick Direction	Reverse
3	Joystick Reference	Left
4	-	Right
5	Detect	Detect
6	-	Fifth Switch
7	12V, 100mA	12V, 100mA
8	Joystick Ground	0V
9	Connected to pin 7	Connected to pin 7

the need to enable third party manufacturers to produce alternative controllers, the Pilot-Plus interface is made simple and straight forward. This means that there is a clear outline of the expected signal levels and how these signal levels can be used to control the motion of the MoS. The information about the Pilot-Plus interface is found in its operational manual.

Pilot-Plus module expects controllers that are connected to it to produce analog or digital signals. This information is displayed in Table 3.1 that is acquired from the manual. The table shows a clear outline of the signals that are required to move the MoS forward, reverse, turn left and turn right. For this project preference was given to digital signals. This is due to their relative simplicity to be generated by a computer as opposed to the analog signals that would require a digital to analog converter. The digital signals that correspond to the various motion commands are shown in Table 3.2.

From Table 3.2, it can be seen that only four pins are required to control the full range of the MoS motion. This table also shows that for each movement only one pin will change state while the rest remain at the same logic level. In the

Table 3.2: Table showing the signal levels that correspond to the various motion commands

Pins	Stop	Forward	Reverse	Turn Left	Turn Right
pin 1	1	0	1	1	1
pin 2	1	1	0	1	1
pin 3	1	1	1	0	1
pin 4	1	1	1	1	0

Figure 3.5: Mirrorbow USB-25IO allows the computer to generate digital signals.

case of a normally closed configuration, all pins have to be at logic 1 for the MoS to be at rest. To move the MoS forward, pin 1 needs to be set to logic 0 while the rest of the other pins are at logic 1. Similarly to move the MoS in reverse, pin 2 this time needs to be set to logic 0 while the rest of the other pins are set to logic 1.

To generate these signals, a Mirrorbow USB-25IO device shown in Figure 3.5 is used. This is a component that contains digital output pins and is connected to the computer through the USB port. It allows the computer to control the digital levels of the pins that are on the component. These pins are connected to the 9-way D-type pins on the Pilot-Plus module.

3.3.2 Mechanical Modification

The hardware modification includes removing the seat of the MoS as illustrated in Figure 3.6. The seat is replaced with an aluminium box also illustrated in

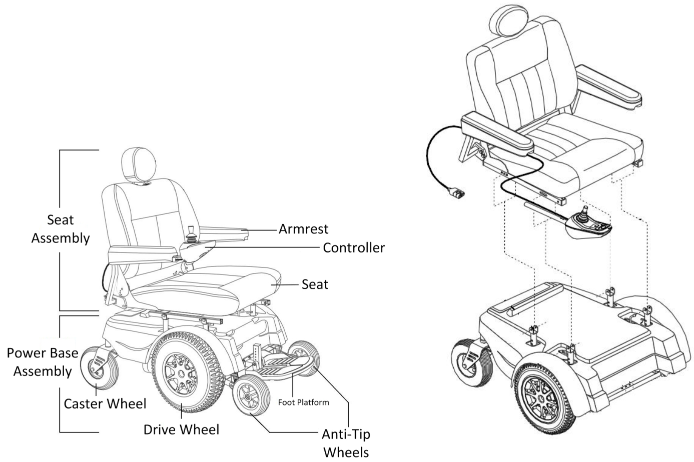


Figure 3.6: Image showing the removal of the seat from the MoS.

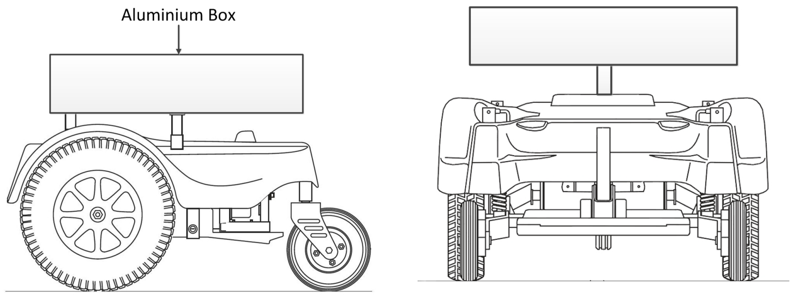


Figure 3.7: Image showing the aluminium box attached to the MoS.

Figure 3.7. The box is meant to house the components of the MoS. Among these components are the sensors, computer and any other electronic components vital to the navigation of the MoS. This box also provides placement for the MoS components that in normal circumstances would be attached to the seat of the MoS like the Pilot-Plus module and the motor controller. The dimensions of the box are made to allow for a large enough space to store the components that are anticipated to assist in navigation and any other components that may be added in the future. Size of the box does not protrude the volume of the base from any of the sides. This configuration helps prevent damage to the inner navigation

Figure 3.8: Hokuyo URG-04LX Laser Range Sensor.

components of the MoS like the sensors and the computer in case of collisions.

3.4 Sensors

A combination of sensors is used to provide the MoS with the ability to detect its surroundings allowing it to safely navigate its environment. Laser sensors are used for obstacle and surface detection while localisation is performed by a GPS receiver. The sensors are placed on the MoS in positions that allow them maximum coverage of the environment quantity they measure.

3.4.1 Hokuyo URG-04LX Laser Range Sensor

The laser sensor type used is the Hokuyo urg04lx shown in Figure 3.8. This sensor operates at a frequency of $10Hz$ with a range of $5m$ and an angular resolution of 0.05 radians [91]. It has a scanning range of 240^o ($\frac{4\pi}{3} radians$) and produces 725 range readings every 100 milli seconds. It does not work well in areas that have low light like dark corners and also has a problem getting range readings

Figure 3.9: GlobalSat BU-353 GPS Receiver.

from highly polished surfaces. The laser passes through transparent materials like glass making features like glass windows and cabinets undetectable to this type of sensor. The Hokuyo urg04lx is unable to detect objects that form an angle of incidence of 45^o or less with the sensing laser beam [15].

This laser sensor connects to a computer through a USB cable. The USB cable is used to provide power to the sensor and also facilitate data transfer between the sensor and the computer. The sensor uses serial communication when connected to a computer. The communication protocol used to interface with the sensor to configure data output is SCIP which is a standard that was developed to provide flexible and efficient sensor interfacing for robotic applications [21].

3.4.2 GlobalSat BU-353 GPS Receiver

For outdoor localisation, a GlobalSat BU-353 GPS receiver shown in Figure 3.9 is used. The sensor outputs location, bearing and time data at a rate of 1 Hertz. This particular type of receiver has 12 channels with a warm start of less than 30 seconds and a cold start of less than 2 minutes. Warm start is the time it takes to recover GPS data after the satellite signal has been reacquired. Cold start is the time it takes to get a GPS fix when the sensor has been turned on for the first time.

This GPS receiver connects to the computer through the USB cable and it uses this cable to transfer data and get power. Communication with the computer is through serial communication and the sensor outputs NMEA (National Marine Electronics Association) sentences. Unlike the laser sensor this sensor engages in one way communication with the receiver constantly outputting the NMEA sentences. This sensor is supposed to be placed in a position that allows it the best view of the open sky.

3.4.3 Sensor placement

The laser sensors and the GPS receiver are placed on the MoS as illustrated in Figure 3.10. Ideally, the sensors should be mounted to positions where they have the best performance while avoiding unnecessary disturbances. The laser sensor used in our experiments is meant for indoor use. However this sensor can still be used for outdoor navigation provided it is shielded from direct sunlight. This means that both laser sensors have to be placed in positions that allow them an ample view of the environment while shielding them from the sun's rays.

The first sensor is used to detect the presence of obstacles that might lie within the surrounding environment. This sensor is placed in a horizontal position as illustrated in Figure 3.11. The second sensor is used to detect the nature of the surface infront of the MoS. This sensor is placed tilted downwards such that its detection plane intersects the fore ground. The placement of the down facing laser is also illustrated in Figure 3.11.

In both positions, the aluminium box provides adequate shielding to the sen-

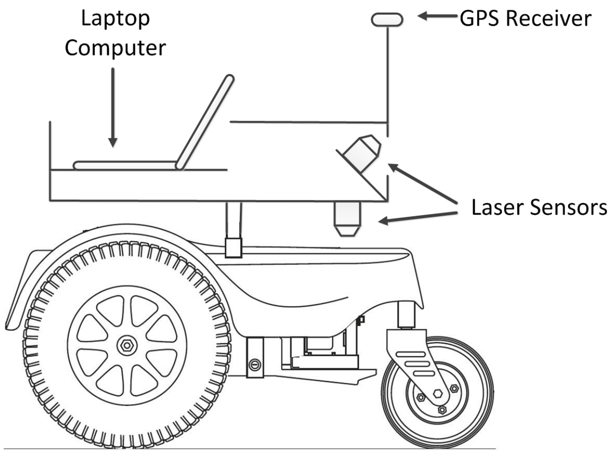


Figure 3.10: Placement of the sensors and computer on the experimental MoS.

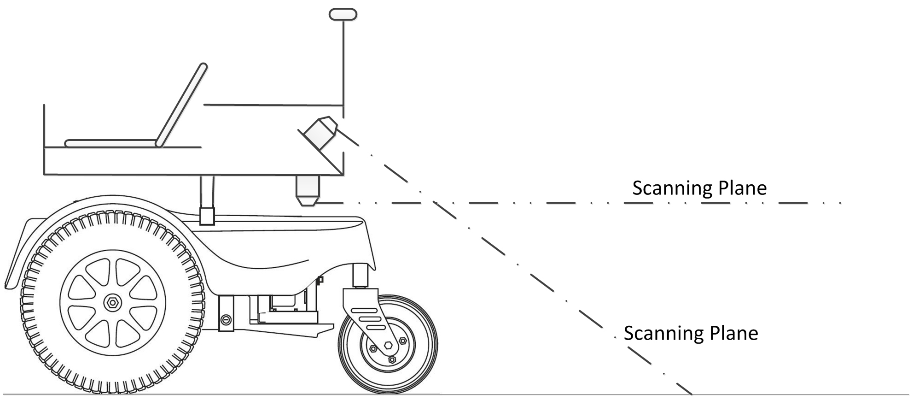


Figure 3.11: Detection planes of the onboard laser sensors.

sors without obstructing their access to the environment.

3.5 Software Development

The navigation system software of the UAS was written in C++ programming language using Microsoft Visual Studio. This was done on a laptop computer running windows operating system. All the data from the sensors is processed by the laptop computer. This computer communicates with the sensors by serial communication through the USB ports.

The algorithm development cycle involves collecting extended datasets using the experiment platform then analysing this data to develop navigation algorithms. To facilitate this function a remote control function for the mobile platform is developed. Data logging programs for the sensors are also developed. Then the control mechanism is coupled with the sensor data logging functions allowing the MoS to be controlled in an outdoor environment while the sensors gather data. To analyse this data, Matlab is used to both visualise the data offline and develop feature perception methods and navigation algorithms. The developed navigation algorithms are then tested on the experiment platform in a controlled outdoor urban environment.

While field testing is paramount, it is time consuming and not always possible. This is because it involves transportation of the MoS to the experiment site and then performing experiments. Other situations cannot safely be tested like the navigation in an area that has pedestrians and other dynamic obstacles. Thus a simulator is used to perform extensive testing of the navigation algorithms prior to testing them on site or in cases where the scenario would present a hazard to other path users or the MoS. The simulator chosen for this function is Virtual Robot Experimentation Platform (V-REP) [92] . It provides a simplified interface and it contains models of the sensors.

A model of the MoS is developed in V-REP together with a virtual outdoor environment as illustrated in 3.12. Programming of the virtual models is done using a scripting language. The default script language that is used in V-REP is Lua although there is an option to write code in C++. This would simplify transfer of code between the virtual models to the MoS. However the process

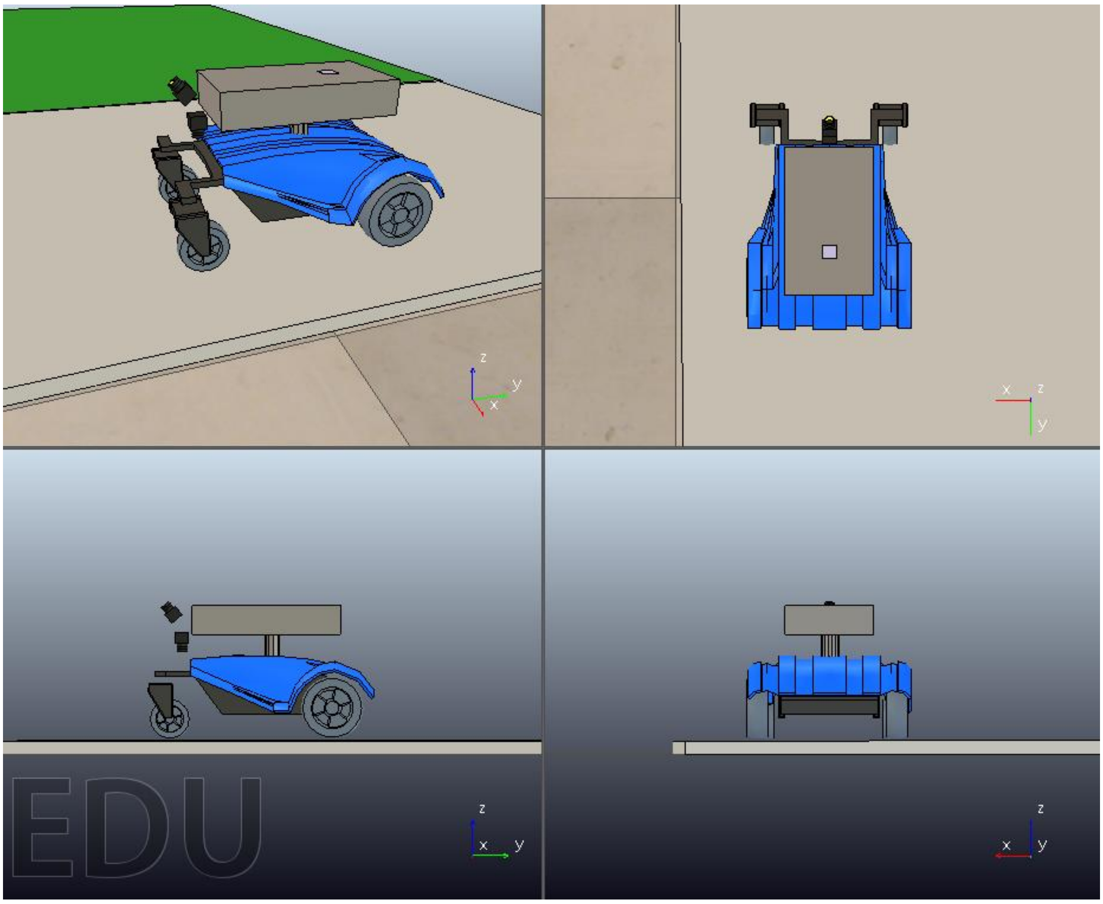


Figure 3.12: Model of the MoS in a simulated environment constructed in V-REP software.

of changing from the default scripting language to C++ is complicated and not worth the extra work involved since the algorithms can be easily transferred between the simulator and the MoS.

3.6 System Architecture

The components of the navigation system are all connected to the computer as shown in Figure 3.13. The computer provides the central processing functions of the system. The sensors that gather data from the environment and the digital

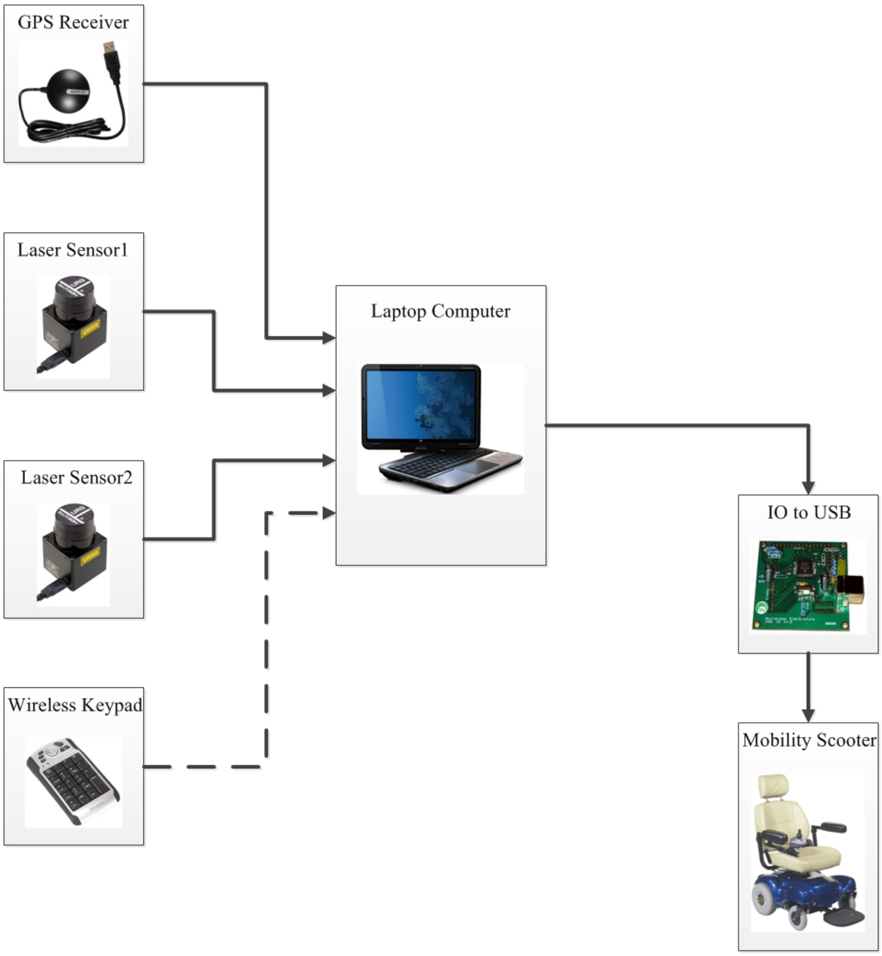


Figure 3.13: Components connections of the autonomous navigation system.

signal generator are connected to the computer through the USB ports. These draw power from the computer through the USB. The computer runs on a laptop battery that has a higher capacity than a typical laptop battery. This is in an effort to support longer durations of outdoor field tests.

Having a common laptop computer perform the central processing functions of the UAS provided flexibility to the navigation system. Incase of a breakdown, this laptop can be replaced by another generic laptop computer. This applies also to the use of components that interface through USB which is a common feature

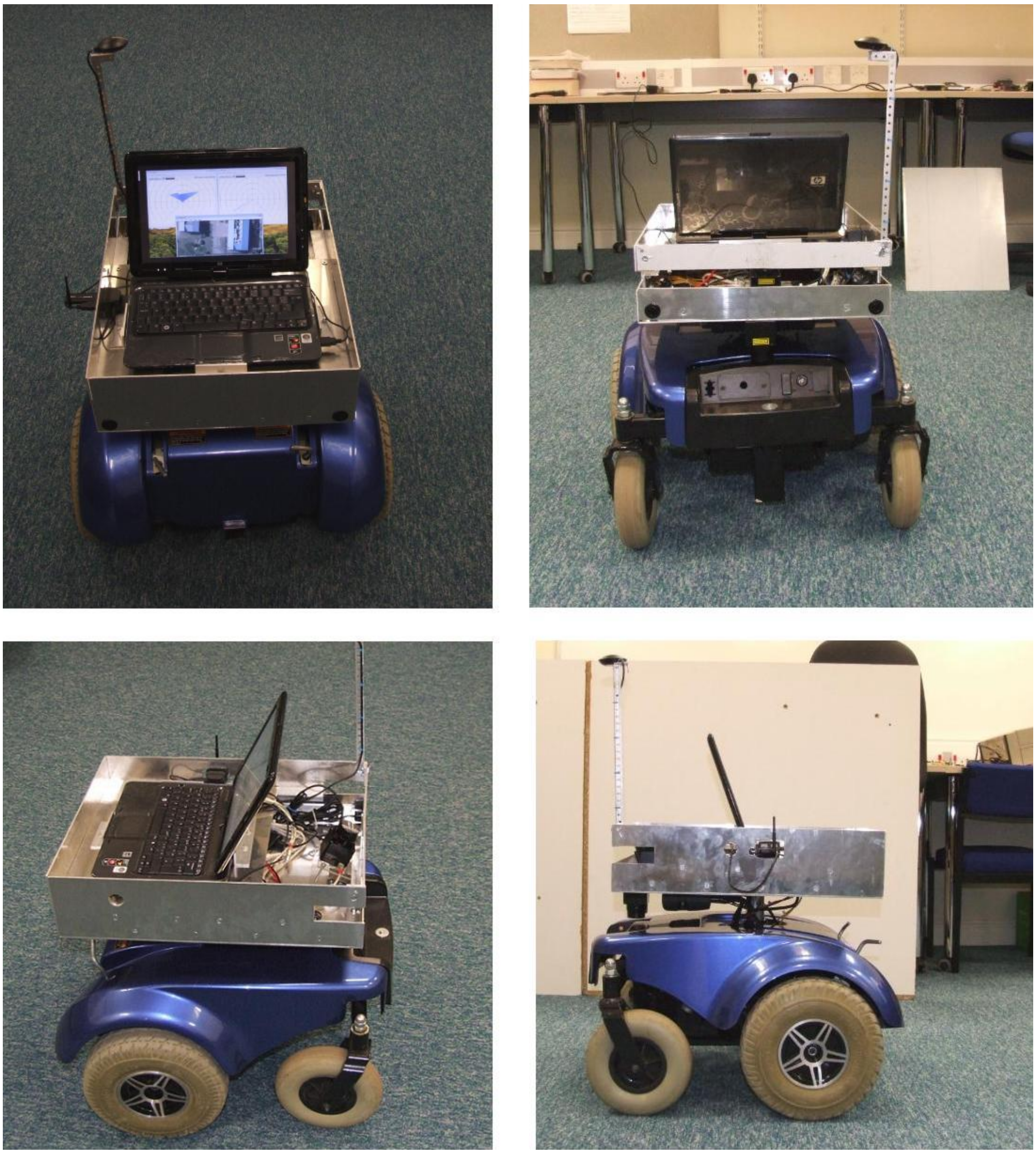


Figure 3.14: Experiment platform in an indoor environment.

that is found on most laptop computers.

The MoS runs on the default 12 Volt batteries that have a capacity of 32 Ampere Hours. These are charged using the MoS wall charger with a typical charge providing about 12 hours of constant run time. Having the computer and

Figure 3.15: Experiment platform in an outdoor environment.

the MoS draw from different power sources enables the separation of the computer from the MoS without cutting the power to the computer. This separation also simplifies the electrical power system of the UAS.

3.7 Design Prototype of User Assisted Mobility Scooter

A prototype of the UAS is designed in software to depict its feasibility. The prototype is modelled in 3DSmax which is a Computer Aided Design (CAD) software package by Autodesk. This model is then exported to V-REP where it is simulated in a virtual outdoor environment.

The design of the prototype UAS starts by building on the original design of a typical power chair. From this original power chair design, the modifications that are specific to the UAS are added. These modifications are driven by the need to find a suitable placement position for the sensors, the central processor

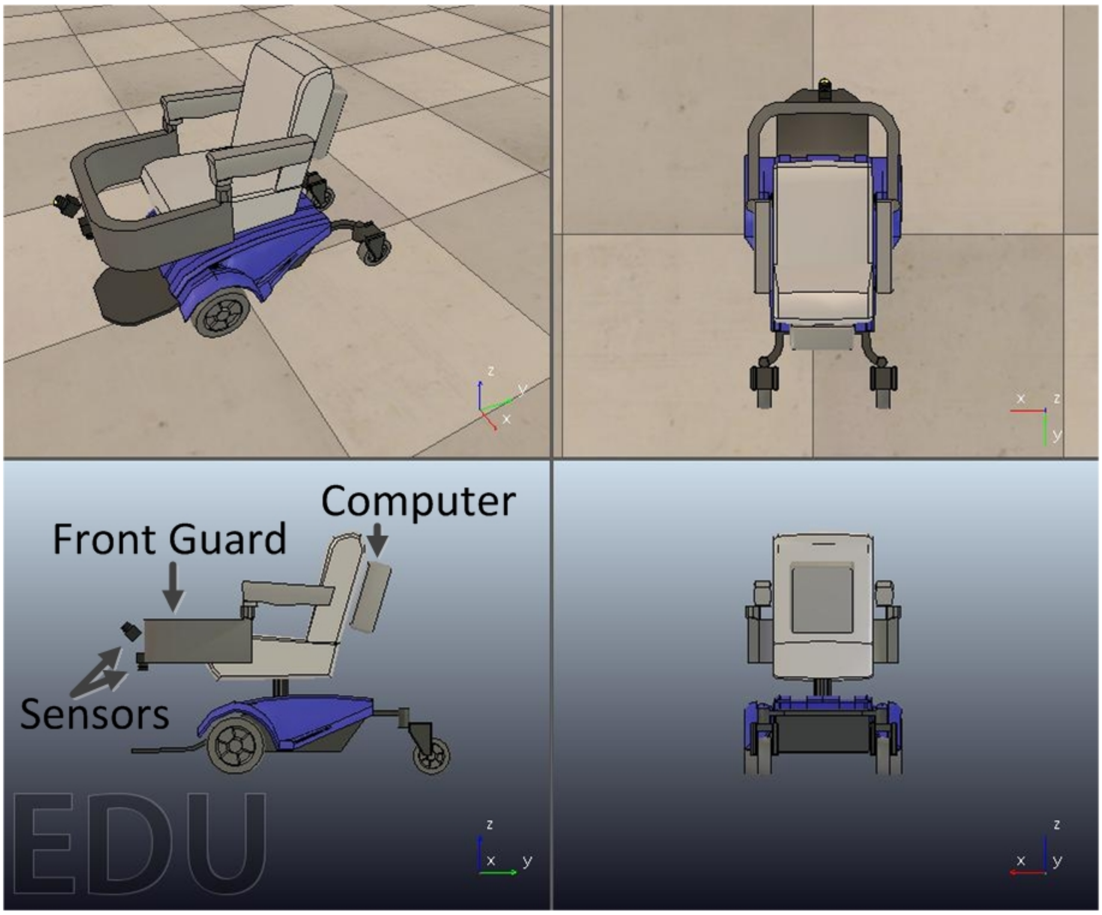


Figure 3.16: Design prototype of a UAS as modelled and simulated in a virtual environment constructed in V-REP software.

of the UAS which is a computer and any other components that are vital to the functionality of the UAS.

A box is designed and placed at the back of the seat of the UAS. This box is meant to hold the central processor and any other components like the signal generator that are required by the navigation system. Any hard material like aluminium could be used to construct this box, the requirement is that it needs to be able to support the components that are placed into it and protect them from damage.

For placement of the sensors, a structure that surrounds the front of the UAS is designed. This structure shown in Figure 3.16, attaches to both armrests of the UAS. One of the arm rests is hinged to the structure while the other arm rest has a removable attachment. This configuration allows the structure to pivot or swing at one of the arm rests forming a gate-like mechanism that enables the user to have easy access to and from the UAS. For this project, this structure is referred to as the front guard. The design of the front guard also strives to ensure that the leg-room of the user is not strongly diminished. This front guard provides a suitable position to place the laser sensors. Both the down-facing and horizontal laser sensors can be placed in positions that enable them to have an unobstructed view of their environment while also not restricting the user's motions.

The designed model shown in Figure 3.16 meets the requirements of the prototype UAS. These requirements include having a place to accommodate the extra components that are as a result of the onboard assistive navigation system. The model is exported from 3DSmax into V-REP where the model is fitted with virtual entities such as motors and sensors that allow it to interact with the virtual environment. This model can then be programmed to navigate the virtual environment for the sole purpose of testing and evaluating the developed outdoor navigation algorithms.

3.8 Conclusion

The experimental platform that is used to test the navigation of the UAS is presented as occurring in two stages. The first stage is the enabling of computer motion control and the second is the replacement of the seat with an enclosure that houses the navigation hardware. The software development makes use of data recording and offline processing to develop perception methods and navigation algorithms. Simulation is also used to develop the navigation of the UAS because it provides a safe and fast method to test navigation algorithms. The computer acts as the central processor of the UAS running on a power supply that is separate from that of the UAS.

The presented modification and architecture allows for system development and testing in a manner that is expandable, efficient and safe to both the UAS and the environment around.

Chapter 4

Environment Feature Detection

4.1 Introduction

In urban outdoor environments, the Mobility Scooter is required to move along pedestrian reserved walkways and sidewalks. These walkways and sidewalks contain a wide variety of static and dynamic obstacles all of which have to be avoided in a safe and reasonable manner. To fulfil this requirement, perception methods are developed to enable the navigation system to identify an outdoor walkway by using laser range sensor data. This chapter presents the methods developed to detect the features that can be found on and around a typical pedestrian walkway or sidewalk. The features that are detected include curbs, vegetation such as grass and hedges, walls and barriers. All these methods and techniques make up the perception system for a UAS that is responsible for providing information about the locations of obstacles, traversable ground and the borders of the walkway.

Figure 4.1: Examples of walkways.

4.2 Walkway

Pedestrian areas can be seen as belonging into two categories namely the sidewalk and the walkway. The sidewalk is found next to the road and allows the pedestrians to travel alongside the road. It is usually found along both sides of the road although there are some cases where it is found along only one side of the road. It has sections that allow access to and from the road. These are to facilitate the crossing of the road by the pedestrians so that they can get to the opposite side.

A walkway can be described as a paved path that usually allows passage through areas that may not be seen as strictly belonging to the road network. An example of this could be a paved path that crosses the park. For both walkways and sidewalks the recommended minimum width is 2m [93]. This minimum width allows two pedestrians to use the path comfortably whether they are walking side-by-side or passing by each other. In this research the word walkway and sidewalk is used interchangeably to mean a pedestrian reserved paved path. Figure 4.1 shows some examples of typical walkways.

Walkways are used by pedestrians for many reasons. These reasons vary from moving for miles to moving from the home to the local grocery. These walkways differ in visual and physical structure. This difference is due to the fact that there are various constraints taken into consideration beyond the basic safety requirements when designing them. Some of these constraints include the nature of the environment that the walkway resides. Town centers are characterised by high foot traffic and the walkway design has to take this into account. As a result, these are often wide walkways with clear demarcations from other traffic like cars. While in the residential areas, due to the low volume of pedestrians, the walkways are narrow.

Apart from being next to a road, the walkway can also be bordered by a building. In this case, the walkway border is formed by a wall or a barrier. Other times the walkway is bordered by vegetation such as grass. This project detects the presence of the walkway by associating a flat surface in close proximity with a curb, wall or vegetation.

4.3 Laser Sensor Placement

Laser sensors are used to detect some of the environment features. There are two laser sensors and they are placed in different configurations. They are used to detect different features in the environment. Both laser range sensors produce range readings in the form of polar coordinates of the laser strikes relative to the position of the sensor. The sensors produce range readings in the form (r_i, θ_i), where i is the index of the reading and it varies from 1 to n, r_i is the distance to the detected obstacle and θ_i is the orientation of the laser beam. $\theta_i = \theta_{res} * i$ where θ_{res} is the angular resolution of the laser sensor. These range readings

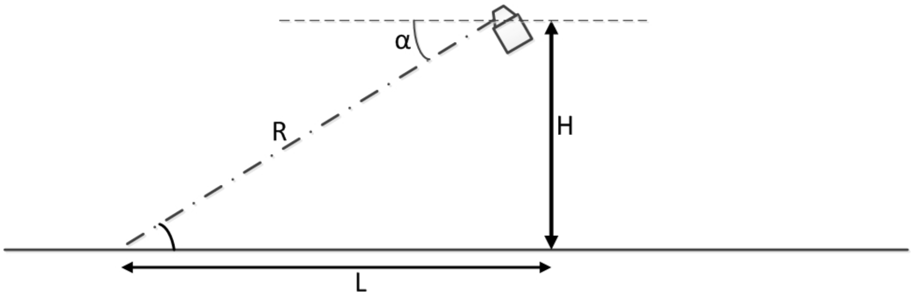


Figure 4.2: Down facing laser.

are processed to identify any features that may be detected in the environment. However, due to the different placement configurations of the sensors, the range data is interpreted in slightly different ways. The following sections introduce the sensor parameters and how these parameters affect the laser sensor range readings.

4.3.1 Downfacing Laser Sensor

The downfacing laser sensor is placed at a height of H above the ground plane with an angle of depression α as shown in Figure 4.2. This placement configuration allows the detection plane of the laser sensor to intersect with the ground plane L metres infront of the MoS. With this configuration, the sensor is used to determine the surface characteristics of the ground infront of the MoS. This placement of the sensor aims to provide height measurements of the ground surface that is infront of the MoS. There is a correlation between the distance that the sensor can look ahead and the angle of depression. The distance infront of the MoS that the sensor can look ahead is determined by the angle of depression. From Figure 4.2 it can be seen that an increase in the angle of depression leads to a decrease in the look ahead distance. The angle of depression also affects the angle of incidence of the laser beams as they strike the ground infront of the MoS. This angle of incidence is directly proportional to the angle of depression and this can be seen in Figure 4.2. However there is also a correlation between the angle of incidence and the the laser range reading errors. This means that when the angle of incidence gets lower, the errors due to incoherent reflections increase. The reflections get to the point that most of the beam is reflected away and the receiver is unable to detect any reflection from the emitted beam. From the study carried out in [15] it is reported that the Hokuyo laser does not return any readings when the angle of incidence of the emitted laser beam is 45^o or less. So although there is a need to place the detection plane as far away infront of the MoS as possible, there is a constraint that is introduced by the characteristics of the Hokuyo laser sensor. The placement therefore balances the look ahead distance with the angle of incidence to provide both reliable range data and environment information as early as possible.

From Figure 4.2 it can be seen that;

$$L = \sqrt{R^2 - H^2} \tag{4.1}$$

The cartesian coordinates (x_i^d, y_i^d) of the laser strikes relative to the downfacing (d) laser sensor can be stated as follows;

$$x_i^d = L\tan(\theta_i) \tag{4.2}$$

$$y_i^d = L \tag{4.3}$$

$$\theta_i = (m - i) * \theta_{res} \tag{4.4}$$

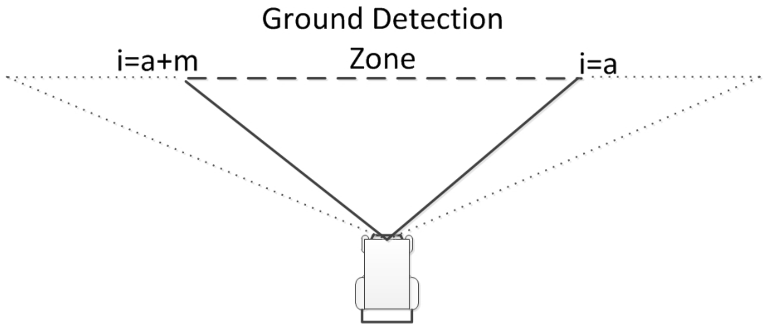


Figure 4.3: Detection zone of the down facing laser due to the angle of incidence limitation.

where m is the mid Index value, θ_i is the orientation of the laser beam and θ_{res} is the angular resolution of the laser sensor.

For the downfacing laser sensor, not all the span of readings are used. This is due to the fact that as the readings move further away from the central range reading, the angle of incidence with which the beam strikes the ground decreases. This means that at a certain point from the central laser reading any strike from the sensor to the ground infront of the MoS will fail to return a coherent reflection. This is assuming the MoS is situated in an environment with a relatively flat surface. Taking the selected detection range as m, the span range can be written as starting from $i = a$ to $i = a + m$ as illustrated in Figure 4.3. This span range is centered around the laser central reading $i = n/2$.

So it can be seen that placing the laser sensor in a down facing position allows the surface characteristics of the ground infront of the MoS to be observed. This placement however has to balance between errors caused by incoherent reflection and the length by which the sensor can look ahead. This is due to the fact that the laser strikes of the downfacing laser depend on the look ahead distance which is in turn determined by the angle of depression and the height of the sensor

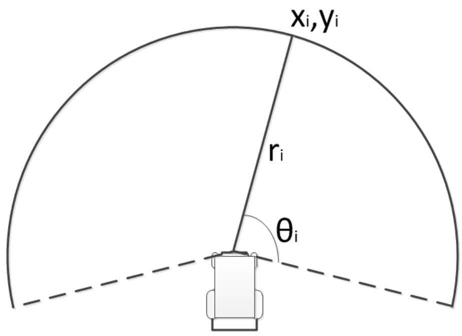


Figure 4.4: Position of laser strike relative to the sensor location.

above the ground.

4.3.2 Horizontal Laser Sensor

The horizontal laser sensor is placed parallel to the ground plane. This configuration allows for the detection of any obstacles that may lie in an area around the MoS. The maximum span of detection is θ_{max} and the maximum range is r_{max}. Then from Figure 4.4 the cartesian coordinates (x_i^h, y_i^h) of the laser strikes relative to the horizontal (h) laser sensor can be stated as follows;

$$x_i^h = r_i cos\theta_i \tag{4.5}$$

$$y_i^h = r_i sin\theta_i. \tag{4.6}$$

For the horizontal laser, the full span of the laser readings are utilised unlike the downfacing laser. Provided the laser is placed in a horizontal position, the location of the laser strikes do not depend on the height above the ground like the downfacing laser.

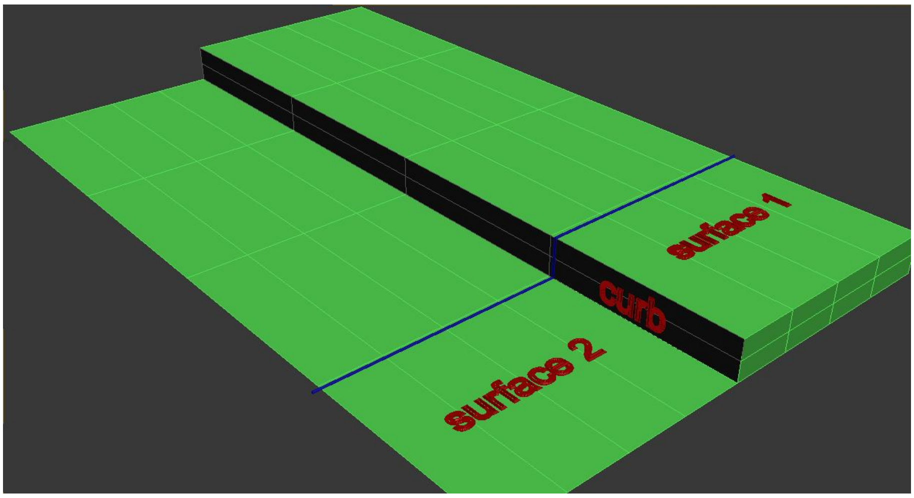


Figure 4.5: Curb Structure.

4.4 Curb Detection

A curb or kerb, is the edge where a raised sidewalk meets an unraised street or road. It separates the road from the roadside, discouraging drivers from parking or driving on the pedestrian sidewalk. This separation also allows the pedestrians to be aware of the boundary of the sidewalk. For the MoS the curb provides a tripping hazard that cannot be safely traversed. This is due to the low ground clearance of the MoS that depends on the wheel radius and size. Curbs also provide structural support to the pavement edge and they are predominant in urban areas.

Curbs often have a vertical or nearly-vertical face as illustrated in Figure 4.5 making them difficult to mount or unmount. However, in some areas the curb is sloped towards the road. This feature is sometimes referred to as a lowered curb or dropped curb and it allows easy wheel access on or off the sidewalk. Lowered curbs are used to mark points on the sidewalk where the pedestrians can cross the road to the opposite sidewalk.

To perform the detection of the curb, the laser sensor is placed with an angle of depression. This placement of the sensor aims to provide height measurements of the ground surface that is infront of the MoS. Due to the angle of incidence limitation of the laser sensor beam, only a portion of the span range is used for surface height detection. The laser range data is in the form of polar coordinates, that are converted into cartesian coordinates as shown in Section 4.3.1 with the horizontal ,x , component representing the width and the vertical, y, representing the height. The y component displays the height of the surface infront of the MoS. With the angle of depression fixed, any variations in this y component is due to the nature of the surface. This means that if the MoS were to move along a path with a flat even surface, the value of this component would be approximately constant. This property is used to identify erroneous readings in the returned sensor data.

4.4.1 Pre-Processing

Considering the maximum height of a curb and the placement of the downfacing laser, the range of the height detected by the sensor can be determined. Let us consider that the height returned by the sensor when the MoS is on a flat surface is y. If the MoS is then placed on a walkway that is bordered by a road separated from the walkway by a curb of height c, there will be some laser strikes that will fall on the walkway surface and some that will fall on the surface of the road. The height returned by the laser strikes that fall on the walkway will be y while those that fall on the road surface will be $y + c$. This shows that at any given time, the height values of the laser sensor will range between the height of the surface and the height of the surface plus the height of the curb. When there is an obstacle infront of the MoS, this value will be lower than the value of y. This means that the maximum value of the y component of the downfacing laser will be $y + c_{max}$, where c_{max} is the maximum curb height. This information is used in the preprocessing stage to eliminate erroneous readings. This is done by specifying a threshold, $y + c_{max} + y_{env}$ where y_{env} is a buffer value, above which the readings are declared to be erroneous and then eliminated. This process is shown below:

$$\mathrm{y}_i = \begin{cases} y_i & if \quad y_i < y_{max} \\ y_{i-1} & if \quad y_i > y_{max} \end{cases}$$

where y_i is the y component of the laser reading and y_{max} is the defined threshold for the maximum value of y. After the readings have been processed, they are passed on to the next stage of the perception method.

4.4.2 Curb Candidates Detection

A curb is characterised by a sharp change in the distance readings returned by a single cycle of the laser sensor data as illustrated in Figure 4.6. The task then becomes detecting this sharp change in the readings returned by the laser sensor. Each cycle of laser range readings is investigated for sharp changes. This is done by checking the difference between consecutive distance readings in a cycle. A large difference is indicative of a sharp change. A threshold can be defined to detect sharp changes in the range readings thereby determining the presence of a curb. This means that when a range reading is found to differ from its neighbour above a detection threshold this is taken as a curb candidate. The curb candidates of a single cycle are recorded together with their corresponding range differences.

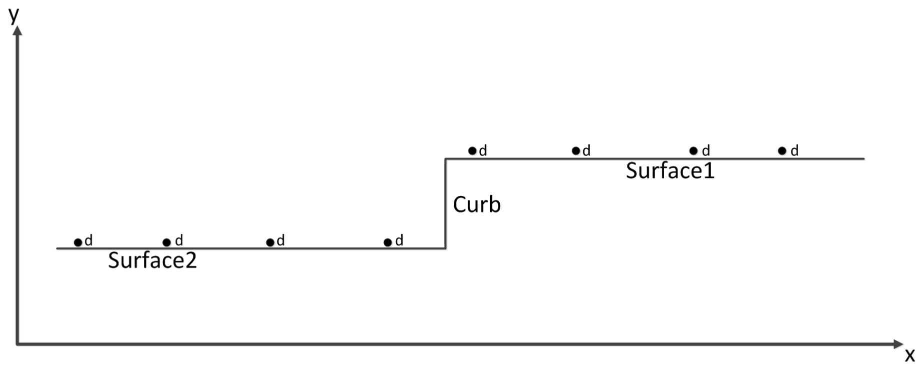


Figure 4.6: Laser range readings from down facing laser.

This threshold should be high enough to allow for the detection of the curb while discarding any differences that may arise due to the nature of the texture of the surface. The above statement can be formulated as Algorithm 1 illustrated below.

Algorithm 1 Curb Candidate Detection

1: $(r, \theta) \leftarrow [(r_1, \theta_1), ..., (r_n, \theta_n)]$
2: **for** $i \leftarrow 1, ..., n - s$ **do**
3:     **if** $(r_i - r_{i+s}) \geq C_{th}$ **then**
4:         $(r_i, \theta_i) \in CurbCandidates$
5:     **end if**
6: **end for**

r_i is the i^{th} laser range reading, n is the number of laser range readings in a single cycle, s is the number of skipped consecutive range readings and C_{th} is the curb threshold value.

4.4.3 Post-Processing

The nature of the navigation environment anticipates a singular curb detection. When multiple curb candidates are detected then some are assumed to be false detections. These false detections are caused by sensor noise. The task then becomes eliminating these false positives and remaining with the true curb location. To determine the true curb, the consistency of the candidates is tested. This is done by comparing the positions of the candidates with the previous detected candidates. The true curb will show a consistent position over multiple laser cycles.

To eliminate false positives, an assumption is made that states that the curb readings will have consistent positions in the immediate following laser cycle readings. This assumption is coupled with the property that states that the path is slow changing. Using this knowledge, a method for eliminating false positives is developed.

To improve the above algorithm, a property called curb consistency is introduced. It is a value that indicates how consistent the detected curb candidate is when compared to previous cycle readings. It indicates how often the curb candidate has appeared in previous laser cycle readings. The curb candidates are acquired using the difference in range. The curb candidates and their corresponding positions are stored. These positions are relative to the sensor position. The next cycle of readings are investigated. The curb candidate positions of this new cycle are also stored. The positions of the curb candidates from the previous cycle are compared to those of the present cycle. If a curb candidate is found to have a position that is close to that of a curb candidate from a previous cycle, then its curb consistency value is incremented. When two readings are compared, those candidates that have low consistency values and have not been updated in a while are removed from the curb candidate list. Algorithm 2 below represents the pseudo code for the algorithm.

where CP_{i1} is the curb position of the $i1^{th}$ curb candidate, $k1$ is the number of curb

Algorithm 2 Eliminating false positives

for $i1 \leftarrow 1, ..., k1$ **do**
  for $i2 \leftarrow 1, ..., k2$ **do**
    if $(CP_{t1} - CP_{t2}) \geq CP_{th}$ **then**
      $ConsistencyValue_{t1} + +$
    end if
  end for
end for

candidates in the first cycle of laser readings, $k2$ is the number of curb candidates in the second cycle of laser readings, CP_{th} is the curb position threshold value, $ConsistencyValue_{i1}$ is the consistency value for the $i1^{th}$ curb candidate.

This process is repeated with the aim of converging to the true curb. As the number of cycles increase, those candidates that have low consistency points are dropped from the list of curb candidates. The curb is detected in one cycle of laser readings and then its position is monitored in the subsequent laser cycle readings.

4.5 Dropped Curb Detection

A dropped curb is when the height of the curb reduces to a level close or equal to that of the road. This is illustrated in Figure 4.7. The curb drop allows easy wheel access to the walkway. It could also serve an important role of indicating an exit point from the walkway.

The height of the detected curb can be determined using the curb detection method introduced earlier. The difference of the curb candidate provides information about the height of the curb. The resolution of the measurement depends on the laser sensor distance resolution which is often in the millimetre range.

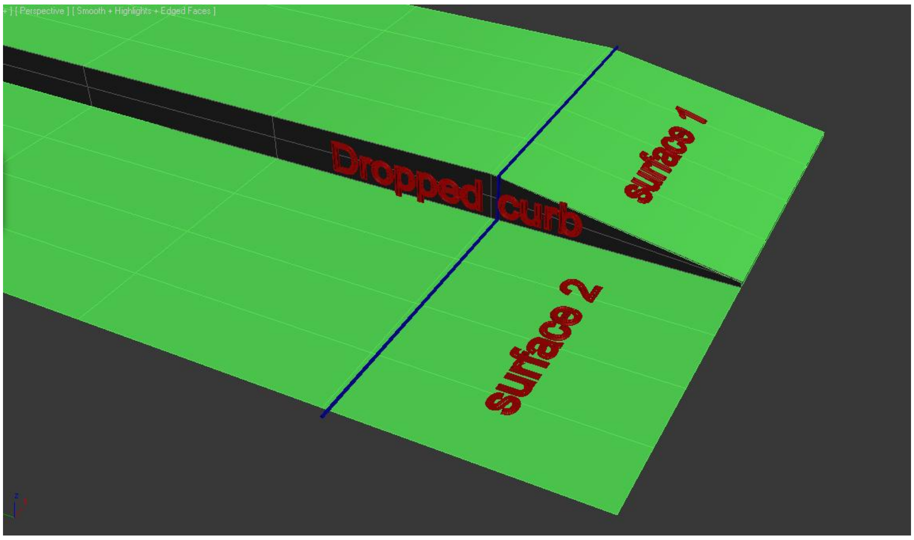


Figure 4.7: Dropped curb structure.

The height of the detected curb is important when it comes to characterising curb drops. The curb drop is when the curb is lowered to the height of the road. For this method the dropped curb is when the curb candidate difference is lower than the detection threshold. When this happens, the detected curb suddenly disappears from the immediate proceeding cycles of laser sensor readings. When the height of the curb is monitored, this curb drop can be anticipated. This is due to the nature of a curb drop. A curb drop is characterised by a gradual reduction in curb height. This means that by monitoring the curb height and checking specifically for gradual reductions, the drop can be anticipated.

With the curb drop anticipated, a method is then devised to detect the lowered curb. This is achieved by reducing the threshold for curb detection. This threshold reduction is only used for indices that are close to where the curb height reduction was detected. The threshold will gradually reduce up to the point where the curb cannot be detected. This is summarised in Algorithm 3 shown below: where i is the number of laser readings cycles, CD_{th} is the curb drop threshold

Algorithm 3 Lowering the Curb Detection Threshold

1: **if** $(CurbHeight_{i-1} - CurbHeight_i) \geq CD_{th}$ **then**
2: $\quad CP_{th} - -$
3: **end if**

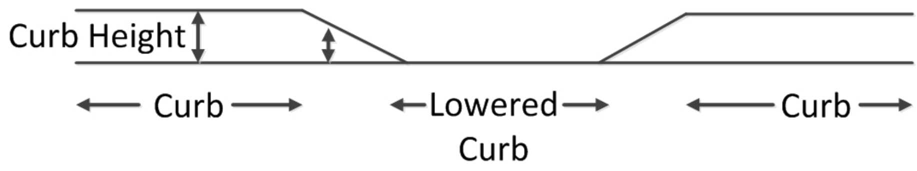


Figure 4.8: Dropped curb with height equal to the road surface.

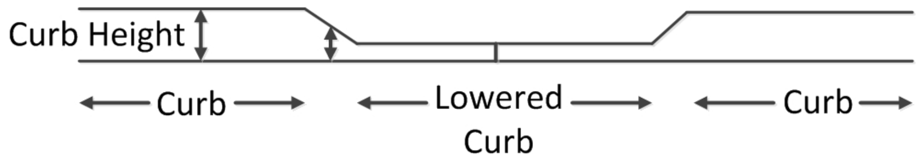


Figure 4.9: Dropped curb with height close to the road surface.

and CP_{th} is the curb position threshold value.

In some instances, the curb drops to the level of the road while in others, the curb drops to a level close to that of the road as illustrated in Figure 4.8 and Figure 4.9 respectively. In the second instance, it is possible to detect the curb in its lowered state.

4.6 Vegetation and Surface Texture Detection

Vegetation can be found bordering the walkway and it is usually grass. It, however, has a rougher texture when compared to the paved walkway as illustrated in Figure 4.10. This is not always the case, there are times when the grass is freshly mowed and it exhibits a smooth texture that rivals if not exceeds that of the walkway. However, the grass provides a feature that can be used to determine

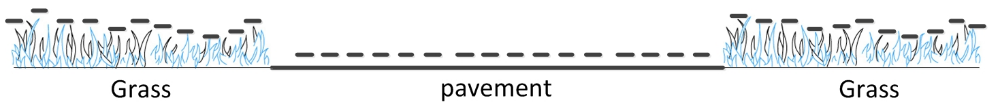


Figure 4.10: Cross section of paved walkway bordered by grass showing laser range strikes.

the dimensions of the path namely the width.

The project proposes a method that combines the techniques used by Wurm et al [54] and Wolf et al [59] to detect the pedestrian walkway. The reason for this is that although vegetation does border the walkway, there are some instances when this vegetation is not available. This may be due to season changes that leave the vegetation dry or erosion that sweeps it away leaving bare ground. The detection of the grass or vegetation that surrounds the path can be detected using range readings from a laser sensor that is pitched downwards at an angle. The range information is investigated and processed. The information from the range data shows that the vegetation would exhibit a uneven distribution of range readings compared to the paved path readings.

The sensor placement is the same as that of the curb detection. This sensor is used for the purpose of curb detection and also the detection of vegetation using range readings. Using the same dimensions, it is still evident that the trade off between the look ahead distance and the reliability of the sensors is a balance that has to be carried out.

The range readings are in the form of range and angle. They are changed into cartesian coordinates in the frame of the sensor. These coordinates show the height of each of the strikes to the ground infront of the MoS. The span is also kept the same as before when it was used for curb detection.

The sensor data that is returned by the sensor is investigated to determine if there are any erroneous readings. When an erroneous reading is detected, it is replaced by the neighbour's value. This aims to smooth the readings.

The readings are investigated for texture. This is achieved by checking the variation between consecutive range readings. This is done by finding the difference between the heights of consecutive range readings. The resolution of the range readings neighbours is varied to determine the best configuration that provides the highest detection confidence. The comparison starts with comparing range readings directly next to each other ie, the immediate neighbours are compared to each other. The comparison moves on to include a skip of consecutive range readings. The comparison of direct neighbours is anticipated to be fine resolution that is expected not to adequately represent the variation of vegetation. This is especially true for instances of freshly mowed grass where the texture of the vegetation is close to that of a paved path. The comparison yields a property that indicates the texture of the area that is covered by those particular laser points.

The texture indicator for the specific areas in a cycle are combined. The combination aims to find a smooth area next to a rough area. In this instance the smooth area would represent the paved walkway while the rough area would be the vegetation. When the readings of the indicators are combined to form an area that is representative of the strip infront of the MoS, the readings are combined once again. This recombination produces even bigger areas that show the texture of the surface. The recombination is an effort to reduce the readings and make them more uniform. The assumption is that the texture of an area will exhibit a few readings of another texture. For example the rough texture area will exhibit a few readings that will indicate the presence of a smooth texture. These conflicting readings or outliers are reduced or even eliminated by the recombination. The aim is to have a clear boundary that separates the vegetation from the walkway.

To determine the surface texture of the area infront of the scooter the distance readings from one laser cycle are examined. The difference between consecutive distance readings is calculated. The differences that are greater than a specified threshold are considered rough while values below indicate a smooth surface. The differences are arranged in order of laser data indices and the property is introduced that shows if the difference indicates smoothness or roughness. A search is performed to find high density indicators. If a particular region of the readings has a high density of indicators that show one particular texture, then that is considered the texture of that region. Algorithm 4 presents a pseudo code of the grass candidate detection method.

Algorithm 4 Grass Candidate Detection

$(r, \theta) \leftarrow [(r_1, \theta_1), ..., (r_n, \theta_n)]$
for $i \leftarrow 1, ..., n - s$ **do**
  if $(r_i - r_{i+s}) \geq G_{th}$ **then**
    $(r_i, \theta_i) \in GrassCandidates$
  end if
end for

r_i is the i^{th} laser distance reading, n is the number of laser distance readings in a single cycle, s is the number of skipped distance readings, G_{th} is the grass threshold value.

This method is not very reliable because surface texture is not very consistent. That is why it is supplemented by another technique that relies on the intensity readings of the laser sensor. The laser readings of one cycle are examined. These readings are matched to the vegetation intensity readings. The results of the match are used to deduce the presence or absence of vegetation in the environment.

4.7 Obstacle Detection

The horizontal laser sensor returns range readings that indicate the presence of obstacles that exist in the vicinity of the MoS. These obstacles that are detected are relative to the sensor position. Therefore the position of the sensor is transformed into a position that is relative to the world. The obstacle position is then saved in a list of detected obstacles.

The laser sensor is able to detect any obstacles that happen to be with in a radius of $5m$ in a span of 240^o infront of the MoS. The range data that is produced by the sensor is in the form of range and angle. This data shows the laser strikes that returned to the sensor indicating the presence of an obstacle. This information can be used to determine the position of a detected obstacle.

Received data from the laser sensor is checked for errors and any values greater than the maximum of the sensor range are removed. The absence of an obstacle is marked by a zero value from the laser sensor reading. The readings that indicate the presence of an obstacle are the non zero range values. The cartesian positions of these points are calculated. These positions are relative to the sensor. Then, these positions are converted into global coordinates. Position calculations require information about the current location and orientation. This information is provided by the localisation of the navigation system. The positions are then saved to a list. This list contains the obstacles that have been detected together with the positions. The list can also be used as a map that shows the surrounding of the MoS. The obstacles can be used to determine the shape of the path in cases where the walkway is bordered by buildings, barriers or other static environment structures.

An urban environment is littered with obstacles to navigation. Some of the obstacles include buildings, cars, people and pets. The successful detection of the obstacles allows for collision free navigation. This task is facilitated by use of a laser sensor that sweeps in a two-dimensional plane around the scooter.

The laser sensor returns distance readings of reflected beams. These distance readings are the ones that indicate the presence or absence of an obstacle on the sensor's field of detection.

The local coordinates of the detected obstacle, (x_l, y_l), can be calculated based on the length of the reflected beam, d_t, and the angle of the reflected beam, θ_t. This can be expressed as:

$$x_l = d_t \cos(\theta_t) \tag{4.7}$$

$$y_l = d_t \sin(\theta_t) \tag{4.8}$$

$$\theta_t = (n * \theta_{res}) - \frac{\pi}{6} \tag{4.9}$$

where t is time, θ_{res} is the resolution of the laser sensor and n is the index of the reflected beam.

$$x_f = d_n \cos(\theta_t) \tag{4.10}$$

$$y_f = \sqrt{L^2 - H_d^2} \tag{4.11}$$

$$\theta_t = (n * \theta_{res}) - \frac{\pi}{6} \tag{4.12}$$

Where t is time, (x_f, y_f) are the local coordinates of the detected feature, d_n is the length of the reflected beam, θ_{res} is the resolution of the laser sensor, n is the index of the reflected beam where the feature has been detected and θ_t is the angle of the reflected beam.

4.7.1 Plotting Detected Obstacles

As an alternative to the above obstacle detection technique, it is possible to pass the obstacle information to the mapping algorithm that updates the environment map. In this case an obstacle is defined as any non-zero distance reading from the horizontal laser. This method does not discriminate between static and dynamic obstacles and relies on the high frequency of map updating to track the movement of any dynamic obstacles. Location of the detected obstacles (x_t, y_t) is determined by:

$$x_g = d_t \cos(\theta_t + \beta_t)\gamma + lat_t \tag{4.13}$$

$$y_g = d_t \sin(\theta_t + \beta_t)\gamma + lon_t \tag{4.14}$$

where (x_g, y_g) are the GPS coordinates of the detected obstacle, (lon_t, lat_t) are the GPS coordinates of the scooter, β_t is the current bearing of the MoS, d_t is the length of the reflected beam, θ_t is the angle of the reflected beam and γ is the scaling constant.

$$x_f = d_n \cos(\theta_t + \beta_t)\gamma + lat_t \tag{4.15}$$

$$y_f = (\sqrt{L^2 - H_d^2})\gamma + lon_t \tag{4.16}$$

$$\theta_t = (n * \theta_{res}) - \frac{\pi}{6} \tag{4.17}$$

where (x_f, y_f) are the local coordinates of the detected feature, d_n is the length of the reflected beam, θ_{res} is the resolution of the laser sensor, n is the index of the reflected beam where the feature has been detected and θ_t is the angle of the reflected beam.

The above expressions show how the obstacles that are detected by the laser sensor are plotted onto a map. The expression requires the current location and heading of the mobility scooter. This data is acquired from the GPS receiver. This fusion of data from the two sensors allows the navigation system to construct a map of its immediate surroundings. This map can then be used by the system to plan its movement within the environment.

4.8 Road-Side Detection

The side of the road in this context refers to the side that the MoS is travelling along. Urban environments have areas at the side of the road that are reserved for pedestrians. The particular side of the road that the side walk is located, could be used to aid in the navigation of the mobility scooter. The side of the road could also be used to place boundaries onto the navigation system instructing it to consider the road as more of a hazard compared to the opposite side of the walkway.

The features that are used to determine the road side are the walkway features like the curb and the static obstacles. The following assumption is made about the environment:

- The road is located at the same side as the curb.

- The road is located at the opposite side that has the closest and most static obstacles.

- There is minimal intermittent traffic along the road.

The position of the detected curb is recorded. This position is used by the navigation system to determine where at the side of the MoS, the curb is located. This yields a variable that indicates the location of the road from the information gathered by the curb location.

Algorithm 5 Curb Position

1: **if** $(CurbIndex) > LMid$ **then**
2: $\quad RoadLeft_{cu}++$
3: **else if** $(CurbIndex) \leq LMid$ **then**
4: $\quad RoadRight_{cu}++$
5: **end if**

In Algorithm 5, $LMid$ is the mid point of the laser cycle readings, $RoadRight_{cu}$ is a variable that indicates the presence of the road on the right and $RoadLeft_{cu}$ is a variable that indicates the presence of the road on the left.

The locations of the detected obstacles are all grouped into two sections corresponding to the sides of the MoS. The environment is expected to have more obstacles to the side opposite to the road because this side is where the buildings, barriers and hedges are expected to reside. The obstacles that are further than the width of the road are assumed to reside on the opposite side of the road. These obstacles are not considered in the determination of the road side. The obstacles from both sections are compared with the section with the most obstacles determined to be opposite to the location of the road. This yields a variable that indicates the location of the road relative to the mobility scooter.

Algorithm 6 Obstacle Position

if $(ObstacleIndex) > LMid$ **then**
  $RoadLeft_{ob} + +$
else if $(ObstacleIndex) \leq LMid$ **then**
  $RoadRight_{ob} + +$
end if

Where $LMid$ is the mid point of the laser cycle readings, $RoadRight_{ob}$ is a variable that indicates the presence of the road on the right and $RoadLeft_{ob}$ is a variable that indicates the presence of the road on the left.

Two variables, $RoadSide_{cu}$ and $RoadSide_{ob}$, are used to determine the side of the road that the MoS is navigating. Due to the nature of the features used, the curb is assigned a higher weight. This is because the curb is a much more reliable indicator for road side detection than the detected obstacles. However there are instances where the curb could be occluded by another structure such as a car or the curb could be absent like in the instance of a lowered curb. In these circumstances, the detected obstacles provide a method for road side detection.

4.9 Conclusion

The pedestrian walkway in an urban environment is characterised by features that are detected by laser range sensors. Among these features is the curb that offers the most reliable feature detection characteristic. The reliability has made it the most sought after feature in the environment. However, the curb is not always present and this led to the development of other feature detection methods for the other features that are used to indicate the presence of a walkway. Features like the grass that is found bordering paths is detected by using range readings from a laser sensor. The obstacles like walls, hedges and barriers are also detected using a range sensor.

After detection of the path, other characteristics are determined using the detected features. Using the relative location of the detected features, the shape of the walkway can be determined and also the side of the road on which the walkway may be situated. This information is used by the navigation system to move the mobility scooter safely in an urban environment.

Chapter 5

Localisation

5.1 Introduction

This chapter introduces the method used for the localisation of the MoS in an outdoor environment. It shows how the question of "Where am I?" is answered. Global Positioning System (GPS) is used to provide both position and direction of travel. It provides a simplified method for localisation in an urban outdoor environment. The chapter shows how the precision of GPS was tested. Then it moves on to show how an algorithm was developed to navigate between GPS waypoints. It concludes by showing that the GPS precision and accuracy is not enough to localise blindly but does serve to monitor progress along an outdoor route

5.2 Global Positioning System Data

Localisation is the process of determining the location and position of the MoS in its environment. This information allows the navigation system of the MoS to determine the required actions that the MoS needs to perform to achieve its navigational goals. It also allows the MoS to determine its current progress as it navigates towards its final destination.

For outdoor navigation, GPS is the preferred method of localisation. It provides a cheap and simple method for localisation outdoors provided the GPS receiver has a clear view of the sky. A typical GPS receiver is accurate up to 3 meters. This accuracy however quickly degrades in the presence of signal blockages like tall buildings that are a common sight in urban environments.

GPS receivers communicate using a protocol that is defined by the National Marine Electronics Association (NMEA). The satellite information is relayed in the form of NMEA sentences. These sentences provide information like the position and velocity of the receiver calculated from the satellite signal. These sentences also provide information about the health of the received satellite signal. The sentences begin with a '$' symbol and end with a checksum after the '*' symbol. In the sentences, the different fields like the latitude and longitude are separated by commas. For the GPS receiver the sentences start with '$GP' followed by three letters that identify the sentence. For example $GPRMC is the GPS NMEA sentence that displays the recommended minimum data for GPS.

The GPS receiver outputs the following three NMEA sentences every second.

- GPGGA which contains the satellite fix information.
- GPGSA which contains the overall satellite data.

- GPRMC which contains the recommended minimum data for GPS.

These sentences have information about the calculated position, time and health of the satellite signals. The position and bearing information is required for this project. GPRMC sentence which stands for Recommended Minimum Data for GPS and is shown below contains the required information:

GPRMC,190006.657,A,5254.6595,N,00111.0739,W,4.29,271.08,220113,jjj.j,E*77
where

- RMC is the Recommended Minimum sentence C,
- 190006 is the time of fix which is 19:00:06 UTC,
- A is Status and can be A=active or V=Void,
- 5254.6595,N is the Latitude 52 degrees 54.6595' North,
- 00111.0739,W is the Longitude 001 degrees 11.0739' West,
- 4.29 is the Speed over the ground in knots,
- 271.08 is the bearing angle in degrees True,
- 220113 is the Date,
- jjj.j,E is the Magnetic Variation E can be W or E,
- *77 is The checksum data, always begins with *.

5.3 Localisation Data Extraction

The extraction of the location data starts by identifying the $GPRMC sentence. The location data is then extracted from the sentence after verifying the validity of the sentence. With the extraction done, the process is repeated when the GPS data is updated and this is usually updated at a rate of 1 Hz.

5.3.1 Sentence Extraction

The GPS receiver produces NMEA sentences as a string of characters. This string of characters is checked for the '$GPRMC' string of characters. These characters signify the start of the $GPRMC sentence. It is from this sentence that the longitude, latitude and bearing of the MoS are extracted. The end of the sentence is signified by the checksum and this is detected by checking for the '*' symbol after the $GPRMC characters. So when the $GPRMC string of characters is detected, an array is opened and then the following stream of characters are saved to this array. The characters are saved until the '*' symbol is detected. Then this character array contains the $GPRMC sentence and it is from this array of characters that the location data is extracted.

The sentence is first checked to find out if it is valid. This information is contained by the character that is after the time fix field of the sentence with each field being separated by commas. Using this information, the validity character is identified and checked. When the character 'V' is shown, this means that the sentence is invalid. This could be due to loss of the GPS signal. When this happens, the sentence is discarded and the method waits for the next update of the sentence. For instances when the sentence is valid, the character shows 'A'.

5.3.2 Latitude, Longitude and Bearing Extraction

With the validity of the sentence determined, the next step is to extract the location data. The location data that is extracted is the latitude, longitude and the bearing. The bearing is extracted only when the MoS is in motion. This is due to the fact that the GPS receiver does produce bearing values when the receiver is not in motion. The bearing values are calculated due to the variation in position data that in these instances is due to GPS signal noise.

From the information gathered from the structure of the $GPRMC sentence, the latitude is the third data field of the sentence while the longitude is the fourth field and the bearing is the sixth field. Then using the commas in the sentence to determine the beginning and end of data fields, the latitude, longitude and bearing are located. The latitude, longitude and bearing are then saved into variables as characters. These characters have to be converted from strings to floats before they can be used for navigation. The latitude and longitude character strings are split into two parts with the first part being the degrees and the second being the minutes. This is due to the fact that the latitude and longitude data are displayed in the sentence with the degrees and minutes combined into one string. This means that a direct conversion of the string to a float would produce a false value GPS coordinate value.

After making the appropriate separation of the latitude and longitude strings, the strings are converted into floats and integers. The degrees of the latitude and longitude strings are converted into integers with latitude varying between 0 and 90 and longitude varying between 0 and 180. The minutes of the latitude and the longitude strings are converted into floats and they vary between 0 and 60 with a precision of up to four decimal places. The bearing string of characters are converted to a float and it is expected to vary between 0 and 360 with a precision of up to two decimal places.

When latitude, longitude and bearing data are extracted and converted, they are stored as variables. These variables are updated when a new $GPRMC sentence is identified from the NMEA character stream that is produced by the GPS receiver. These stored variables are then used in the navigation system to provide location data for the MoS.

5.4 Bearing Following

There are times when the UAS is required to follow a specified bearing while in motion. This could be to reach a position or a waypoint that is ahead. To achieve this function an algorithm to follow a specified bearing is developed. This algorithm uses the bearing of the MoS provided by the GPS receiver to calculate a difference in bearing between the required bearing and the MoS bearing. The calculated difference in bearing is then sent to the motion controller of the MoS allowing it to perform a correction maneuver. This correction manuever allows the MoS bearing to be close to that of the required bearing. Then using sensor feedback control the MoS bearing is kept close to that of the required bearing allowing it to maintain the bearing as it moves.

5.4.1 Bearing Acquisition

First the required bearing β_r is acquired. It is the bearing that the MoS is supposed to have to achieve its goal. This goal could be to move to a waypoint, follow a path or avoid an obstacle. This could be provided by another function or it could be hardcoded in the navigation system. Before the current bearing of the MoS is acquired a check is made to verify that the MoS is in motion. This is to ensure that the bearing value that is provided by the GPS receiver is due to the motion of the MoS. To achieve this a check of the movement status of the MoS is performed. With the movement status of the MoS verified the current bearing of the MoS β_m provided by the GPS data is then acquired and recorded.

5.4.2 Bearing Difference Calculation

The current bearing and the required bearing are compared to determine the difference in bearing, β_d. For two distinct values of bearing, there are two differences β_{d1} and β_{d2} between them as illustrated in Figure 5.1. They differ in size although in some instances they could have the same size. To determine the difference, a comparison is made to determine which of the two bearings is greater. When the required bearing is greater than the current MoS bearing the differences are calculated using the following expressions:

$$\beta_{d1} = \beta_r - \beta_m \tag{5.1}$$

$$\beta_{d2} = 360 - (\beta_r - \beta_m) \tag{5.2}$$

Then when the MoS bearing is greater than the required bearing the differences are calculated using the following expressions.

$$\beta_{d1} = \beta_m - \beta_r \tag{5.3}$$

$$\beta_{d2} = 360 - (\beta_m - \beta_r) \tag{5.4}$$

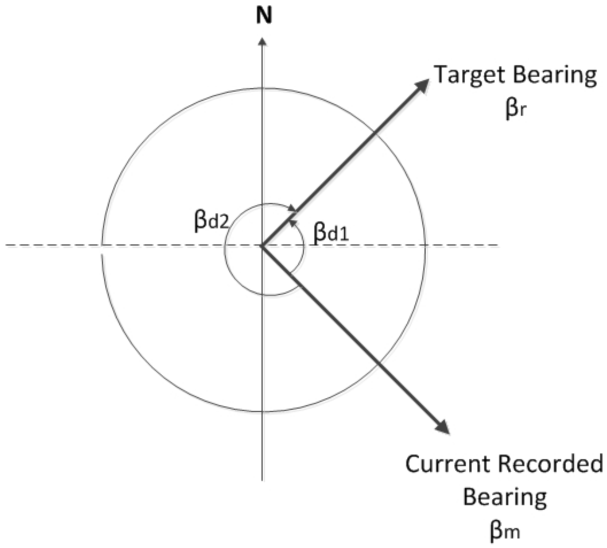


Figure 5.1: Illustration of the difference between the MoS bearing and the required bearing.

These differences are the two values illustrated in Figure 5.1. A comparison is then made to determine which one of these differences has the least value. The difference with the least value is taken as the difference in bearing and is then used for the correction motion of the MoS. This means that preferrance is awarded to turning through the least bearing.

This difference is the amount of bearing change that is required to have the UAS bearing the same as the required bearing. The UAS then has to turn through this amount of bearing. This function is performed by the motion controller. When the UAS has been turned through this bearing, it resumes moving forward. Then the current UAS bearing is recorded and this is again compared to the required bearing. This process is meant to ensure that the bearing turn did infact alter the UAS bearing to the required bearing. When the UAS bearing is found to be close to the required bearing then it is maintained. Otherwise, it is turned through the bearing difference. The closeness to the required bearing

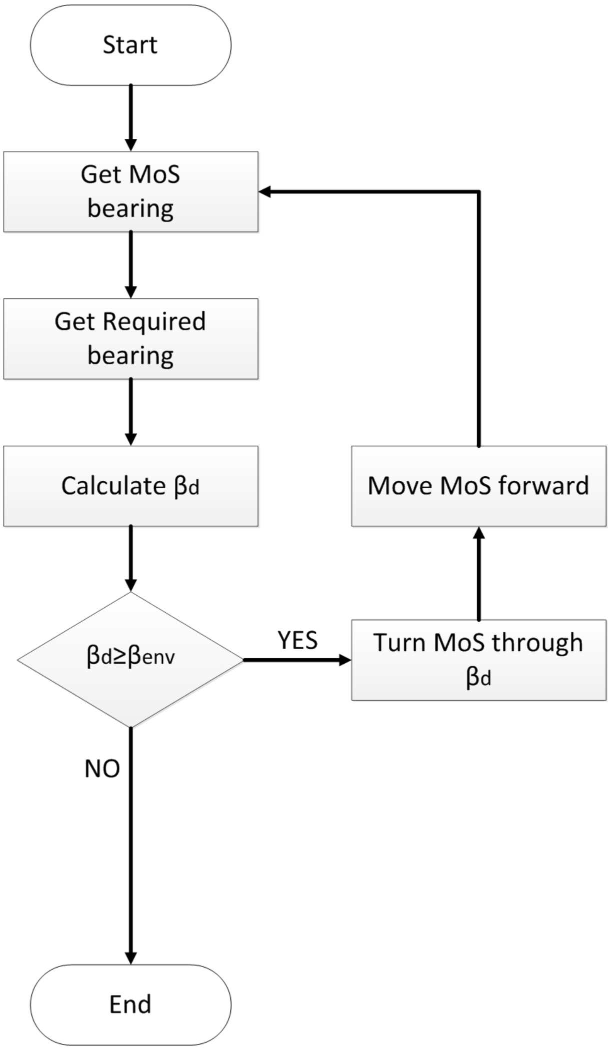


Figure 5.2: Flowchart diagram of the algorithm to follow a specified bearing.

is defined by the bearing envelop shown as β_{env}. This is the accepted difference between the UAS bearing and the required bearing. Due to the noise in the GPS signal, it is unreasonable to expect the UAS bearing to match the required bearing precisely. There is variation in the readings that is caused by the signal accuracy. If there wasn't an envelop then the UAS would be constantly trying to adjust its bearing so that it can match the required bearing.

In summary the UAS when required to follow a specified bearing, it records the difference between the current and the required bearing. When this difference is outside the acceptable range a correction movement is carried out. This correction movement moves the bearing of the UAS to that of the required one. This new bearing is maintained using feedback control. The receiver provides updates of the UAS bearing and this is compared to the required bearing. The flow chart diagram of the algorithm to follow a defined bearing is illustrated in Figure 5.2

5.5 Waypoint Following

An algorithm to move the MoS to a GPS waypoint is developed. This algorithm uses the current position of the MoS to calculate the distance and bearing to the waypoint. These values are used to provide control commands to the MoS motion controller that enable it to reach the waypoint. The MoS is made to follow the bearing of the waypoint while monitoring the distance to the waypoint. The distance to the waypoint is used to indicate the progress of the MoS as it moves towards the destination waypoint.

First the destination waypoint (x_w, y_w) is acquired and recorded. This waypoint is in the form of a GPS coordinate and it could be generated from another algorithm that is part of the navigation system of the MoS. Then the current position of the MoS (x_m, y_m) is acquired from the GPS data produced by the receiver. The distance, d_w, of the waypoint from the MoS is calculated using the following expression:

$$d_w = \sqrt{(y_m - y_w)^2 + (x_m - x_w)^2} \tag{5.5}$$

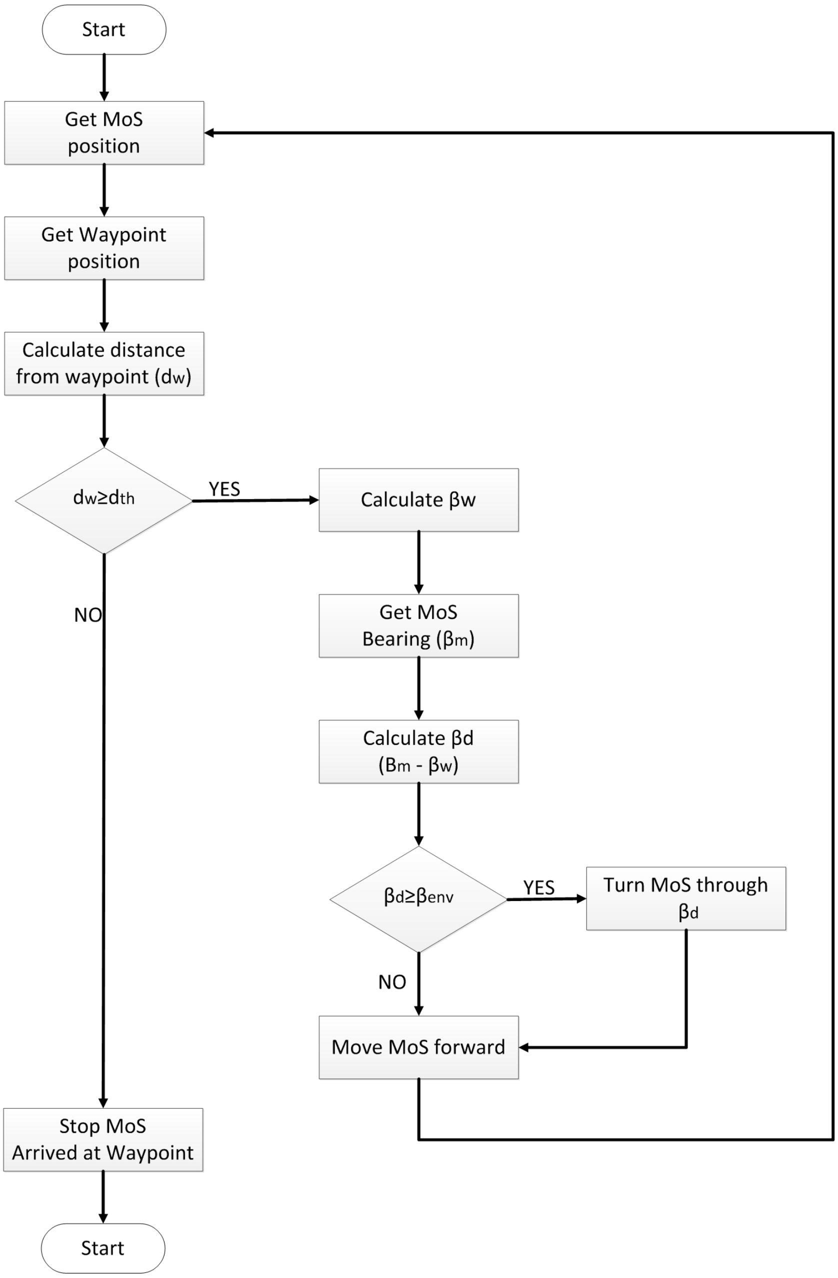


Figure 5.3: Flowchart diagram of the algorithm to move to a specified waypoint.

Using this calculated distance, a check is made to determine if the MoS is near the waypoint. This aims to ensure that the MoS is not at the destination waypoint. To acheive this the distance is compared to a defined minimum distance threshold, d_{th}. If the value of the distance is found to be below this threshold then the navigation system concludes that the MoS has arrived at the waypoint. However, if the distance is above the threshold, the bearing of the waypoint β_w from the MoS is calculated.

$$\beta_w = \arctan(\frac{y_m - y_w}{x_m - x_w}) \tag{5.6}$$

This bearing is the direction that the MoS has to follow to reach the destination waypoint. After the bearing has been calculated, it is then stored as the required bearing, β_r. At this point, the bearing following algorithm is implemented with β_w as β_r. Then using the bearing following algorithm, the MoS follows the bearing of the destination waypoint. The progress of the MoS is determined by monitoring the distance between the MoS and the waypoint. When the distance gets below the predefined threshold, the algorithm concludes that the MoS has arrived at the waypoint.

In summary, when the MoS is required to navigate to a waypoint specified by the GPS coordinates, position of the MoS is recorded. Then, using the position of the MoS and the waypoint the bearing from the MoS to the waypoint is calculated. The MoS is then made to follow this bearing until it reaches the waypoint. A flow chart diagram of this algorithm is illustrated in Figure 5.3.

5.6 Route Following

An algorithm to move the MoS through a route that is defined by GPS waypoints is developed. A route in this case is a sequence of GPS waypoints with the last waypoint being the final destination of the route. The algorithm moves the MoS through the route by navigating the MoS to each of the waypoints in the order that they appear in the route.

First the route $[w_1, ..., w_i, ..., w_n]$ where $w_i = (x_{wi}, y_{wi})$ that contains the set of waypoints is acquired. The number of waypoints in the route is determined to ensure that the route contains more than one waypoint. If the route is determined to contain only one waypoint then this algorithm switches to the waypoint following algorithm. When the route is found to contain more than one waypoint the distance, d_i, between consecutive route waypoints is calculated.

$$d_i = \sqrt{(x_{wi} - x_{w(i+1)})^2 + (y_{wi} - y_{w(i+1)})^2} \tag{5.7}$$

This is to ensure that the distance between the waypoints is above the minimum distance threshold d_{th}. If the distance between two consecutive waypoints is found to be below the distance threshold, one of the waypoints is eliminated and the distance is calculated again. The route is considered to be valid when the distance between the consecutive waypoints and the number of waypoints are above the respective thresholds.

After the validation of the first waypoint (x_{w1}, y_{w1}) in the sequence is considered as the destination waypoint. The waypoint following algorithm is then implemented. The distance d_{w1} and the bearing β_{wi} to the waypoint from the current MoS location are calculated i.e;

$$\begin{aligned} d_{w1} &= \sqrt{(x_m - x_{w1})^2 + (y_m - y_{w1})^2} && (5.8) \\ \beta_{w1} &= \arctan(\frac{y_m - y_{w1}}{x_m - x_{w1}}) && (5.9) \end{aligned}$$

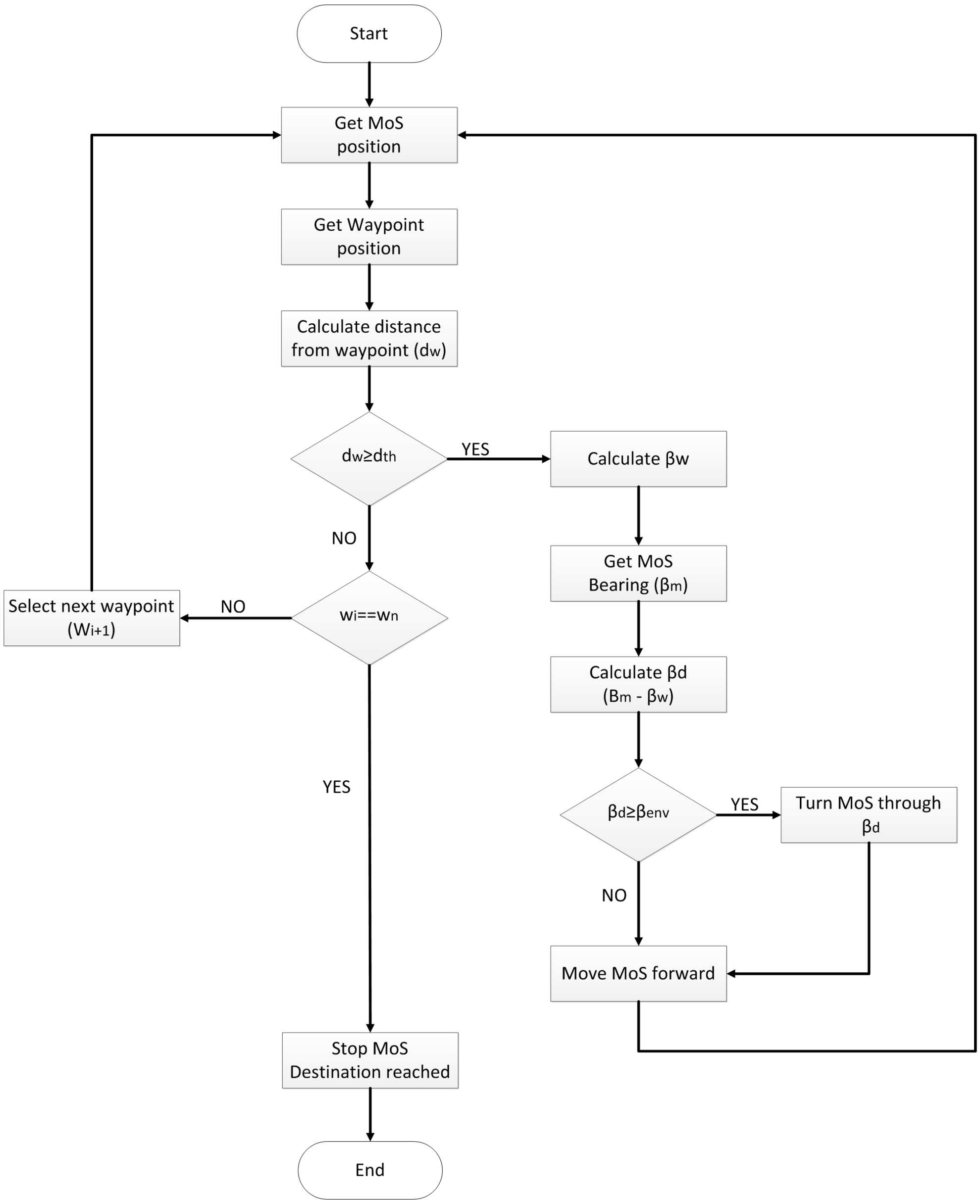


Figure 5.4: Flowchart diagram of the algorithm to follow route specified by GPS waypoints.

The MoS follows the bearing to the waypoint. When the MoS arrives at the position of the waypoint which is determined by the minimum distance threshold, the next waypoint in the sequence is chosen as the destination. Everytime the

MoS arrives at the waypoint the process is repeated until the MoS arrives at the position of the final waypoint in the route. When this happens the algorithm determines if the MoS has arrived at the location of the final destination.

In summary, when the MoS is required to move through a route that is defined by GPS waypoints, first the route is validated. This involves checking to find if the route contains the minimum number of waypoints and that these waypoints are far apart. Then the MoS is made to navigate to each of the waypoints in the route in sequence until it arrives at the final waypoint. This algorithm is illustrated in Figure 5.4.

5.7 Conclusion

The localisation of the MoS is performed by a GPS receiver. This receiver produces NMEA sentences that contain location information and in particular the latitude, longitude and bearing of the MoS. Using this information, a bearing following algorithm is developed that allows the MoS to follow a specific direction. The bearing following algorithm is extended into an algorithm that moves the MoS to a waypoint. Then an algorithm that allows the MoS to follow a route that is defined by GPS waypoints is developed. These algorithms enable the MoS to navigate in an outdoor environment using feedback from the GPS receiver.

Due to the precision of this localisation, it is used more as a method to monitor the progress of the mobility scooter along a route. It is also used to provide navigation for short sections of the route.

Chapter 6

Route Navigation

6.1 Introduction

This chapter introduces the overall structure for navigation of the mobility scooter in an urban outdoor environment. The chapter starts by showing how a route is generated given a start and final destination. It then moves on to how this generated route is processed to mark any potential areas for road crossing. The chapter then shows how the UAS handles crossing the road to move to the opposite sidewalk.

6.2 Route Generation

To generate a route like the one illustrated in Figure 6.2 in an outdoor urban environment, a route generation method is developed. This method uses the Google API specifically the "get directions function". This function is part of the Google API and it takes direction requests in the form of HTTP requests. The

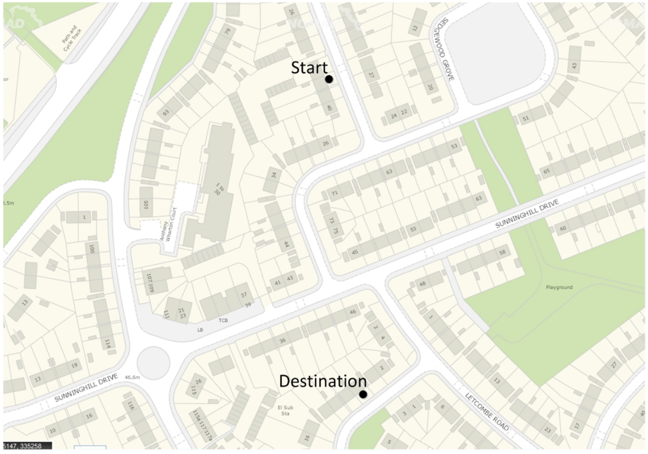
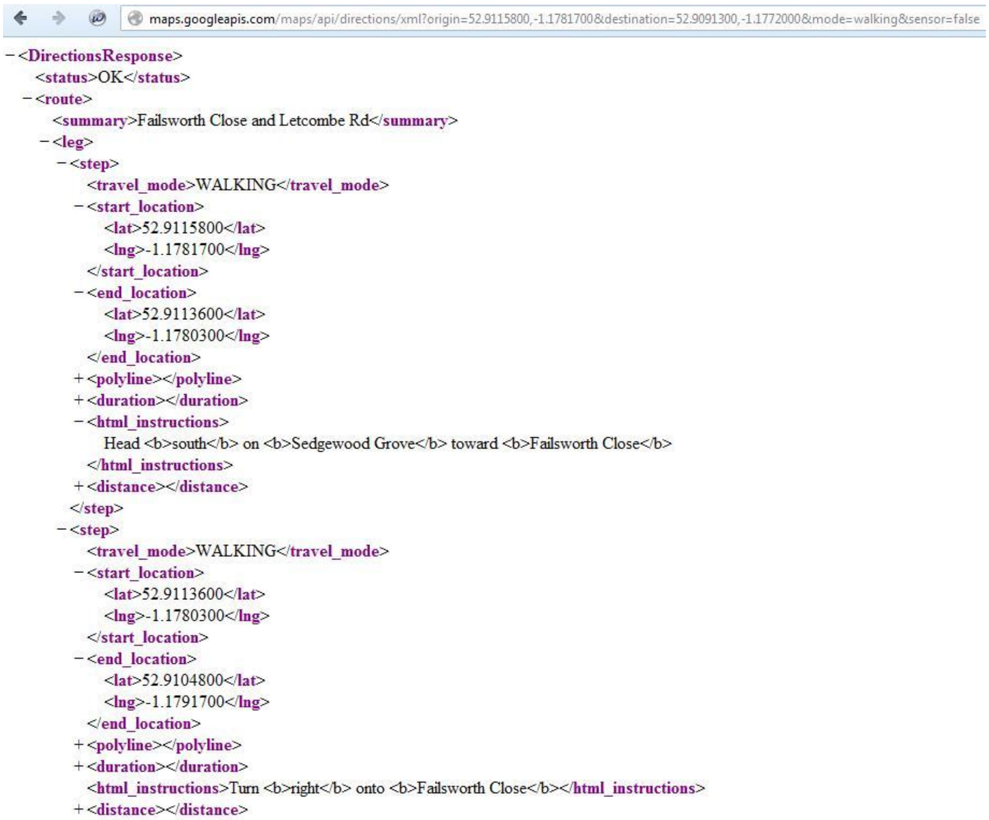


Figure 6.1: Map showing start and destination.

Figure 6.2: Directions generated for route to destination.

request is in the form of a URL, the start and final destination can be defined as either strings or GPS coordinates. The mode of transportation can also be defined as either car, bicycle or walking. The response of the request can be either JavaScript Object Notation (JSON) or an XML document. Which is also set in the request.

The output of the function is a document that contains the intermediate way-points that lie between the start and end destination. This output also contains




```
- <DirectionsResponse>
    <status>OK</status>
  - <route>
      <summary>Failsworth Close and Letcombe Rd</summary>
    - <leg>
      - <step>
          <travel_mode>WALKING</travel_mode>
        - <start_location>
            <lat>52.9115800</lat>
            <lng>-1.1781700</lng>
          </start_location>
        - <end_location>
            <lat>52.9113600</lat>
            <lng>-1.1780300</lng>
          </end_location>
        + <polyline></polyline>
        + <duration></duration>
        - <html_instructions>
            Head <b>south</b> on <b>Sedgewood Grove</b> toward <b>Failsworth Close</b>
          </html_instructions>
        + <distance></distance>
        </step>
      - <step>
          <travel_mode>WALKING</travel_mode>
        - <start_location>
            <lat>52.9113600</lat>
            <lng>-1.1780300</lng>
          </start_location>
        - <end_location>
            <lat>52.9104800</lat>
            <lng>-1.1791700</lng>
          </end_location>
        + <polyline></polyline>
        + <duration></duration>
          <html_instructions>Turn <b>right</b> onto <b>Failsworth Close</b></html_instructions>
        + <distance></distance>
```


Figure 6.3: The HTTP directions request and the XML response.

directions in the form of strings.

This project uses GPS coordinates to define the start and the end destinations of the journey. The transportation mode is set as “walking” although other transportation modes could be used since the route directions that are produced are similar. Due to the directions request being made by an HTTP request, the technique requires that the navigation system have access to the internet.

The response XML document, such as the one shown in Figure 6.3, is saved in a cache so that it can be processed. This processing starts by extracting the GPS coordinates of the intermediate waypoints. The coordinates are labelled in the XML document using $<lat>$ and $<lng>$ for latitude and longitude respectively. By searching the document for the strings that appear after these labels, the waypoints can be identified and saved as strings. These strings are then converted into floats enabling them to be used by the navigation system. The next information that is extracted from the XML document are the direction instructions. These are navigation instructions that are meant to be human readable. They include statements like "*turn left at the next corner*". Though they are meant for human consumption, they still provide useful information about the route that can be used by the navigation system. The instructions are labelled in the XML document by the *<html_instructions>* label. Using these labels, the instructions are extracted from the document as a series of strings. The navigations system uses a lookup table to determine the meaning of these instructions and then uses them to improve its own route information.

The route that is produced contains intermediate waypoints. These are GPS coordinates of points that lie between the start and final destination of the journey. The waypoints are points in the journey where a change in direction movement is required; an example of this could be at a corner or a junction. The direction instructions provide information about how to handle these waypoints such as what direction to take at the waypoint. Although the navigation system is able to calculate the direction changes required at these waypoints, the information from the route instructions provides extra data that can be used to improve the certainty of the navigation system.

The route that is extracted from the response, such as the one shown in Figure 6.4, is a set of GPS waypoints together with the corresponding direction changes at those points. Each waypoint has two bearing changes, the change from the XML document and the one calculated by the navigation system. The

```
step1
start: 52.9115800, -1.1781700
destination: 52.9113600, -1.1780300
Head south on Sedgewood Grove toward Failsworth Close
Route Code: 3

step2
start: 52.9113600, -1.1780300
destination: 52.9104800, -1.1791700
Turn right onto Failsworth Close
Route Code: 6

step3
start: 52.9104800, -1.1791700
destination: 52.9105700, -1.1788100
Turn left onto Sunninghill Dr
Route Code: 5

step4
start: 52.9105700, -1.1788100
destination: 52.9095300, -1.1770600
Turn right onto Letcombe Rd
Route Code: 6

step5
start: 52.9095300, -1.1770600
destination: 52.9091300, -1.1772000
Turn right onto Woodsford Grove
Route Code: 6

routesteps = 5
```

Figure 6.4: The route extracted from the processed XML response.

bearing change produced by the navigation system is calculated by using the GPS coordinates of the waypoints.

6.2.1 Nature of the Returned Data

The XML file contains the directions that should be followed to get to the final destination. It provides a breakdown of the route into sections. Each section represents an area where the UAS is on a stretch of the same road or path. When a corner or change in direction of travel is required this is when one section ends and another section begins. It is at these points that the waypoints are placed.

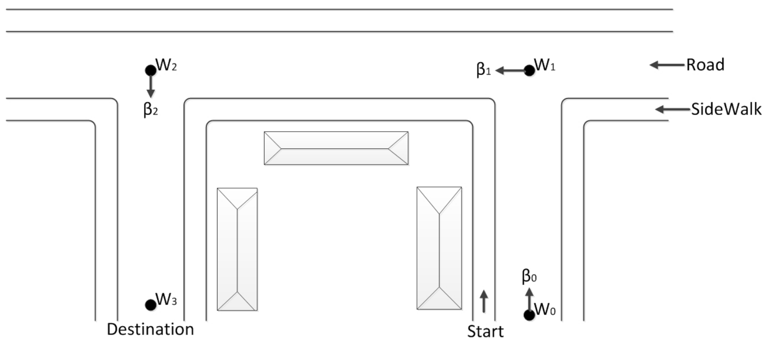
So the file contains these waypoints and the nature of direction of change required. For example, the file could state that take a left at waypoint 3. As stated before, this information is meant for human understanding but it does provide a general bearing that can be passed on to the scooter navigation system.

6.3 Route Processing

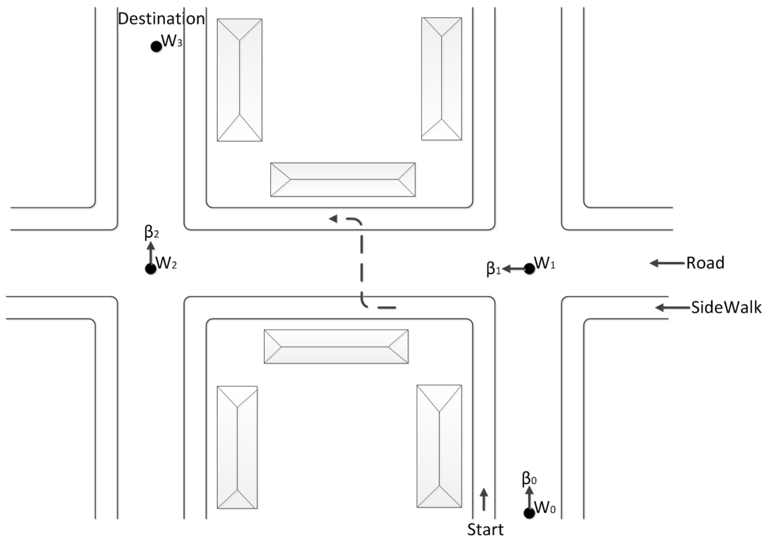
The processing of the route is done in two phases. The first phase extracts data from the XML page. The second phase performs calculations on the extracted data to determine other characteristics of the route like corner directions. The data that is extracted includes waypoint coordinates and the general direction change that the scooter should follow at these waypoints. The data is stored in a route array. So the route at this point contains waypoints with the corresponding direction change. This direction change is either left or right.

The second phase of the processing aims to determine sections of the route that necessitate road crossing. This is important because road crossing places the UAS in a hazardous situation. By identifying these sections early, the navigation system can aim to complete this task before reaching the waypoint where the hazard is potentially higher. To identify these sections, the route waypoint direction changes are compared consecutively. When two consecutive points have similar turn directions then there is no need for crossing as illustrated in Figure 6.5(a).

When two consecutive points have different turn directions this means that the section requires a road cross as illustrated in Figure 6.5(b).
where c_i is the nature of the bearing change at the $(i+1)^{th}$ waypoint. So the complete route contains the coordinates of the next waypoint w_i, the bearing of



(a)



(b)

Figure 6.5: Typical outdoor routes (a) no road crossing required (b) road crossing required.

that waypoint β_i from the previous waypoint and the nature of the approaching corner c_i. This could be either left, right or straight through.

The route can be defined by the following waypoints; w_0 , w_1 , w_2 , w_3, the corresponding bearings that are to be maintained between the waypoints as; β_0 , β_1 , β_2, and the bearing change at the waypoints as; c_1 , c_2.

Algorithm 7 Road Crossing Determination

if $c_i \neq c_{i+1}$ **then**
  Road Cross Required
else
  Stay On Walkway
end if

6.4 Road Crossing

In an urban outdoor environment there are occasions when the MoS even though having reserved areas of travel has to venture out into the road. The road crossing could be vital for the MoS to access a final destination. It could also assist the MoS to get into a suitable position for the next section of the route. An example of this could be if the route data indicated that the MoS should turn right at the next waypoint and it was located at the left side of the road as shown in Figure 6.6. Another reason for crossing could be to avoid a walkway blockage that may inhibit navigation of the path. And last, this crossing could provide a method of the MoS to perform a U-Turn. This mainly is driven by the fact that walkways are not known to be wide enough to accommodate such a manoeuvre.

This then requires the MoS to cross the road which contains motor vehicles that provide a hazard to both the user and the MoS. To enable the MoS to cross the road, a road crossing algorithm is developed. This algorithm uses information from the route to determine if there is a need for crossing the road. Then using the onboard sensors, the navigation system identifies a suitable location in the environment where it can cross the road. The system monitors the progress of the MoS as it crosses the road and then determines when the MoS has arrived at the opposite sidewalk. To facilitate this crossing algorithm, the navigation system makes the following assumptions about the environment:

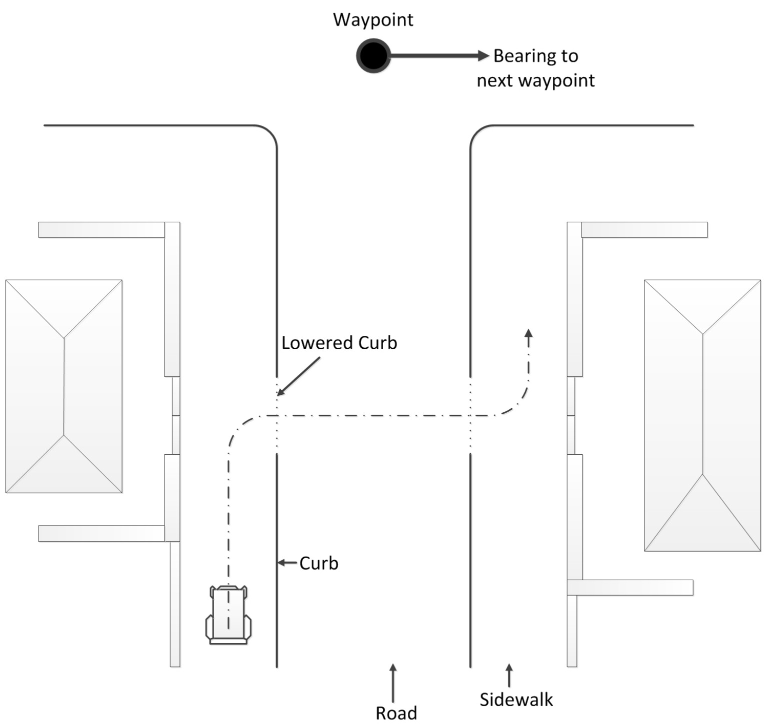


Figure 6.6: A road crossing scenario.

- There is a sidewalk on the opposite side of the road.

- There is a dropped curb or ramp that allows wheeled access of the mobility scooter onto the opposite sidewalk that is aligned or closely aligned to the current one.

- There is not a central refuge in the middle of the road.

The flowchart in Figure 6.7 shows the steps involved in the road cross algorithm. First, the need to cross is determined by the waypoints in the route. Each member in the route set contains the GPS coordinates of the waypoints and the corresponding change in bearing. This change in bearing or direction is stored in

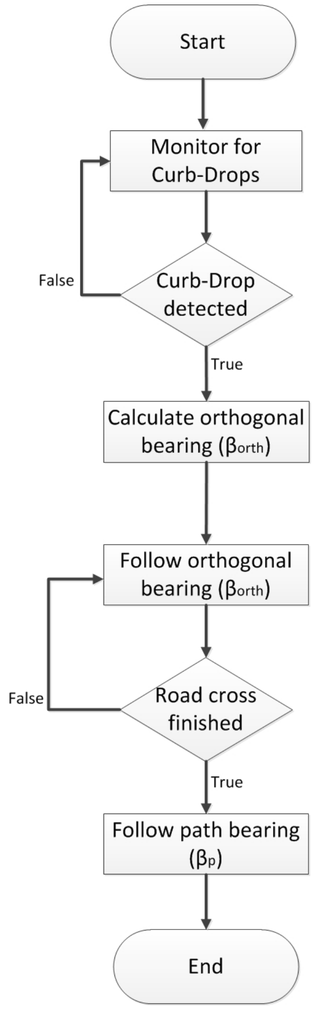


Figure 6.7: Flowchart of the road crossing algorithm.

the format of a float value and a string description. An example could be when the bearing change states that the MoS should turn through 90 degrees or turn left at the corresponding waypoint. The need to cross arises when two consecutive waypoints have bearing change instructions that differ. This then means that the MoS needs to cross to the opposite side walk as it moves between these waypoints.

If the MoS determines that it needs to perform a road cross between waypoints

w_i and w_{i+1}, it first waits until it has passed the first waypoint w_i before it begins the crossing procedure. After the MoS has arrived at w_i and performed the bearing change it moves along the sidewalk while looking for points where it can cross the road. In this instance the MoS first determines the side of the road that the current sidewalk is located. This information could have been acquired earlier since the side of the road that the sidewalk is located on remains the same and only changes on rare occasions like road crossings.

After the MoS has determined the side of the road that is located on, the focus then moves on to identifying points on the side walk that allow for safe passage off the sidewalk. This is due to the fact that sidewalks are usually demarcated from the bordering road by a curb which is a raised step. The MoS is unable to safely traverse this curb (which can be as high as 220mm) due to its low ground clearance. A curb drop or lowered curb on the other hand is a feature that is found on sections of the sidewalk and it helps enable wheel access on or off the sidewalk. At these sections of the sidewalk, the height of the curb is lowered to a height that is close to that of the road. So the MoS having determined a need to cross the road, moves along the sidewalk looking for dropped curbs. The MoS uses the information about the side of the road that the sidewalk is located to anticipate the location the curb drop will be detected. For example if the MoS determines that it is located on a sidewalk that is located on the left side of the road, it can expect the curb drop to be detected on its right side as it travels along the sidewalk.

When the curb drop is located, the current bearing of the sidewalk is calculated and recorded. The bearing of the path is attained by considering the previous bearing readings that were provided by the GPS receiver. They are assumed to be fairly constant due to the assumption that the path in an urban environment is slow changing. The average of these previous values is taken as the bearing of the path after determining that they have a low variation. Also to ensure that the recorded bearing is indeed that of the path, only the most recent fairly constant bearing readings are used. From this bearing a crossing bearing is calculated. This bearing is the direction that the MoS has to take so that it can reach to the opposite side of the road. The aim is to have the MoS set off from the sidewalk and then cross the road arriving at a point that is directly opposite to the side that it set off. This follows the assumption that dropped curbs are aligned or closely aligned at opposite sides of the road.

Using the perception system of the navigation system, the MoS determines when it has reached the dropped curb. Then the MoS changes its bearing to the crossing bearing which is the orthogonal bearing of the path bearing. The MoS then maintains this bearing as it crosses the road. As the MoS follows this crossing bearing, it uses the onboard lasers to detect when it has completed the road crossing.

Environmental cues are used to determine when the MoS has reached the opposite sidewalk. One of these cues is the appearance of a wall or barrier directly infront of the MoS and has an orientation that is orthogonal to the current direction of travel. when a wall or barrier is detected infront of the MoS the navigation system concludes that the MoS has arrived at the opposite sidewalk. The MoS then turns into the sidewalk path. This is done using the previously recorded bearing of the path. This assumes that the opposite sidewalk is also travelling along the road in the same direction as the previous path. Using the GPS data, the MoS turns to the bearing of the path which is orthogonal to the crossing bearing. The MoS verifies that it is indeed following the sidewalk by using the perception system to detect features that indicate the presence of a walkway. These features include a curb, wall or vegetation all of which are aligned in the direction of the path that is being followed.

This algorithm relies on the dropped curb on both sides of the sidewalk to be aligned at opposite sides of the road. The next assumption is that there is a minimal amount of road traffic and it is slow approaching giving the algorithm time to detect and take the neccessary actions. The algorithm also assumes that there is a wall or barrier at the opposite side of the sidewalk that demarcates the sidewalk from the surrounding environments.

6.5 Behavioural Navigation

Behavioural navigation controls the MoS by coupling sensor data directly to the actuators. A behaviour links the perception of a particular feature to the motion controller of the MoS. This is predominantly used to follow a feature that has been detected by the perception system. The assumption is that, by following a feature like a curb that borders the walkway, the MoS can in turn follow the walkway allowing it to reach its destination.

6.5.1 Curb following behaviour

The curb following behaviour couples the detected curb feature with the motion controller of the MoS. The behaviour uses the information about the perceived curb to maintain an appropriately safe distance from the curb while travelling along a walkway.

6.5.2 Wall following behaviour

The wall following behaviour couples the detected wall feature with the motion controller of the MoS. The behaviour uses the information about the perceived wall to maintain an appropriately safe distance from the wall while travelling along a walkway.

6.5.3 Grass following behaviour

The grass following behaviour couples the detected grass feature with the motion controller of the MoS. The behaviour uses the information about the perceived grass to maintain an appropriately safe distance from the grass bordering the walkway while travelling along a walkway.

6.5.4 GPS trajectory following behaviour

The GPS trajectory following behaviour implements the waypoint following algorithm. The behaviour enables the MoS to move between the GPS waypoints.

6.5.5 Obstacle avoidance behaviour

The obstacle avoidance behaviour couples the detected obstacle feature with the motion controller. The behaviour uses the information about the perceived obstacle to plan and execute a path that avoids collision while still following the walkway.

6.6 Behavioural architecture

The behaviours are arranged in a hierarchy as shown in Figure 6.8. The top-most behaviour is the obstacle avoidance behaviour since it is the most critical behaviour. The rest of the other behaviours are arranged according to the reliability of the behaviours to navigate the MoS along a walkway.

A behaviour becomes active when the feature that corresponds to it is detected by the perception system. The active behaviour that is highest in the hierarchy is the one that is responsible for the motion of the MoS. This behaviour is called the control behaviour. When a behaviour that is higher than the control behaviour changes from inactive to active, there develops a need to switch the control behaviour to the newly activated behaviour. The instant the higher behaviour becomes active, a timer is started. This timer measures how long the newly activated behaviour has been active.

The control behaviour is switched to the activated behaviour after it has been active for a specified amount of time. This is done so as to avoid the unneccessary switching of the control behaviours which is caused by errors in the sensor readings. These errors can lead to false detections in the perception system leading to intermittent activation of behaviours. The delayed switching also helps keep the frequency of switching low which leads to smoother movement trajectories.

For example if the wall following behaviour is active and the curb following behaviour is inactive, the control behaviour will be the wall following behaviour despite the fact that it is lower than the curb following behaviour in the hierarchy. However, when the perception system detects the presence of a curb, the curb

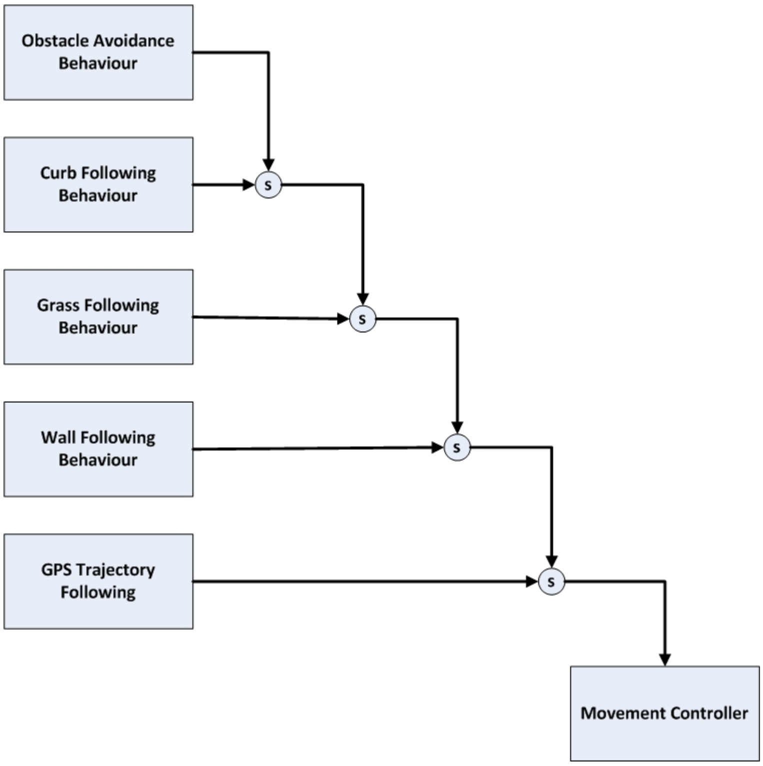


Figure 6.8: Behavioural navigation hierarchy.

following behaviour is activated. At this point, both the curb following and wall following behaviours are active. But since the curb following behaviour is higher than the wall following behaviour in the hierarchy, it is selected as the control behaviour.

The hierarchy shown in Figure 6.8 has obstacle avoidance as the behaviour that supersedes all other behaviours. This is because safety of the pedestrians, user, MoS and the environment infrastructure is a critical aspect of the navigation system. The lowest behaviour is the GPS trajectory following behaviour. This is because it is the least precise navigation behaviour. It is also the one behaviour that is always active and can be counted on to remain active even in the absence of all other features. Additionally, the GPS signal which is the critical component of the GPS trajectory following behaviour is stronger in outdoor wide open spaces. These environments are characterised by a lack of detectable features such as obstacles, walls, curbs and vegetation. On the other hand, the GPS signal is unreliable in cluttered environments which could include features like building walls. This means that by placing the GPS trajectory following behaviour at the bottom of the hierarchy, it will control the MoS motion in situations when it is most reliable.

These behaviours control the movement of the MoS in the environment. With the behaviours organised in a hierarchy, scenarios are presented to evaluate the functionality of the navigation system. In the first scenario, there is a walkway that is bordered by a wall on one side and a curb on the opposite side. The curb however, is not continuous like the wall. It has a section, b, in the middle where it drops to the level of the road next to it.

6.7 Conclusion

With the route generator, processor and crossing algorithm the navigation system is able to navigate from a starting point to a final destination in an urban outdoor environment. The navigation system would navigate in a residential area as this provides repetitive structures that could provide a challenge for an individual that is not familiar with the particular area. Such areas have less traffic compared to city and shopping centers. This means that the obstacles that are encountered by the navigation system would be expected to adhere to a specific set of rules. An example would be pedestrians yielding to the MoS.

The route generator provides routes that are not necessarily the most optimal.

This is common in residential areas that have pedestrian paths that are not properly documented or may be seasonal. This could be countered by having the navigation system recording routes while the mobility scooter is under manual control and then saving these routes. When it comes to route generation, the saved routes could be compared to the generated routes and the most optimum route could be used for navigation.

Chapter 7

Experiments

7.1 Introduction

The various components of the developed navigations system are evaluated and tested in a real world outdoor environment and a simulated environment. The outdoor environment is used predominantly to evaluate the perception methods of the navigation system while the simulated environment is used to evaluate the route navigation component of the system. Both the real and simulated environments used for the experiments contain features that are typical of an urban outdoor pedestrian environment allowing for experiments that are representative of real world situations. As the various components of the developed navigation system are tested in the experiments, some limitations are identified leading to the modification of some algorithms.

7.2 Outdoor Environment Setup

The various components of the developed navigations system are evaluated and tested in an outdoor environment. This environment is a university campus that provides a number of features that are typical of an urban outdoor environment. A university campus environment is used as opposed to an actual urban environment because it is easy to control. This means that safety precautions can easily be implemented and maintained while carrying out experiments.

The outdoor environment selected for carrying out the experiments has got a paved walkway. This walkway has got a curb on one side that demarcates the path from the road next to it. The height of the curb is approximately $10cm$ although this value varies due to surface degradation and road height variations. Towards the end of the walkway, the height of the curb gradually drops to the level of the road forming a dropped curb feature as shown in Figure 7.1(d). At this section of the walkway, the curb slopes towards the level of the road while turning forming a corner that is defined by the lowered curb. On the opposite side of the walkway there is a section that has grass, Figure 7.1(a), and another section where the walkway joins onto an open pavement area, Figure 7.1(c). The length of the walkway is approximately $50m$. The section of the walkway that is bordered by a curb to one side and grass on the other has got an approximate width of $1m$. This walkway has a straight section, Figure 7.1(a), and one that turns both right and left, Figure 7.1(b). This walkway lies in the middle of two buildings forming a typical scenario of an urban canyon.

To collect data, the MoS experiment platform is moved along the path using remote motion control. The MoS is moved through the path in a trajectory that is

(a)

(b)

(c)

(d)

Figure 7.1: Outdoor environment used to test navigation components.

typical of a user. This means that the trajectory is smooth and does not contain any unnecessary maneuvers. The data gathered includes the laser range data from the down facing and horizontal laser sensors, and the location and bearing data from the GPS receiver. The laser sensors produce 681 range readings at a frequency of $10Hz$. The GPS receiver produces three NMEA sentences from which 3 values are extracted at a frequency of $1Hz$.

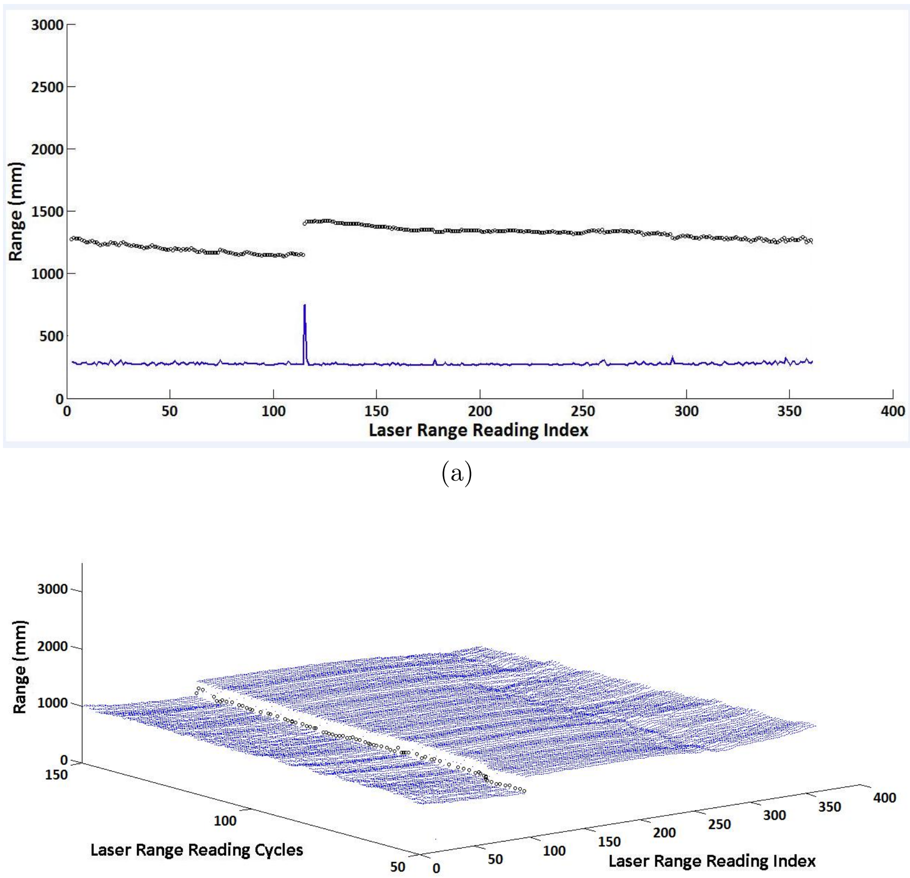
7.3 Curb Detection Experiment

To evaluate the developed curb detection method, a curb detection experiment is carried out. This experiment aims to determine if the threshold is sufficient for detection of a typical curb. The experiment also aims to determine the reliability of the curb detection method by comparing the location of the detected curb with the ground truth.

This experiment involves controlling the MoS along a walkway that is bordered by a curb to one side while the downfacing laser sensor acquires range data about the surface. The acquired range data of the down facing laser is loaded into Matlab so that the data can be visualised. The range readings of a single laser cycle are plotted in a 2D graph as shown in Figure 7.2(a).This information visualises the surface height profile of a narrow strip of area infront of the MoS. From this graph the change in height which indicates the presence of a curb is visible when an appropriate scale is used.

The curb detection method is then used to process the range data. The curb detection threshold is set to $5cm$ and the number of skipped range readings is set to 5. The method detects sharp changes in height together with their corresponding positions in the range data.

To evaluate the performance of this curb detection technique, the range data is plotted on a graph with the positions of the detected curb superimposed on the same graph as shown in Figure 7.2(a) and Figure 7.2(b).This is to compare the position of the curb detected by the method with the actual location of the curb that is provided by the range data. In Figure 7.2(a) the graph below the range readings is a plot of the differences of the consecutive laser sensor readings. The



(a)

(b)

Figure 7.2: Laser range data with detected curb candidates (a) single cycle laser range data (b) laser range data from multiple cycles.

spike indicates a high difference between the neighbouring laser range readings. It is at this point that the curb is perceived to be located. Figure 7.3 shows the curb detections of multiple laser cycles forming a map that shows the relative locations of the curbs.

This curb detection method is able to detect the presence of a curb when the height of the curb is greater than $5cm$. When the height is above this threshold, the difference between consecutive range readings that are at the curb is high enough to be distinguished from other high differences that may be due to uneven

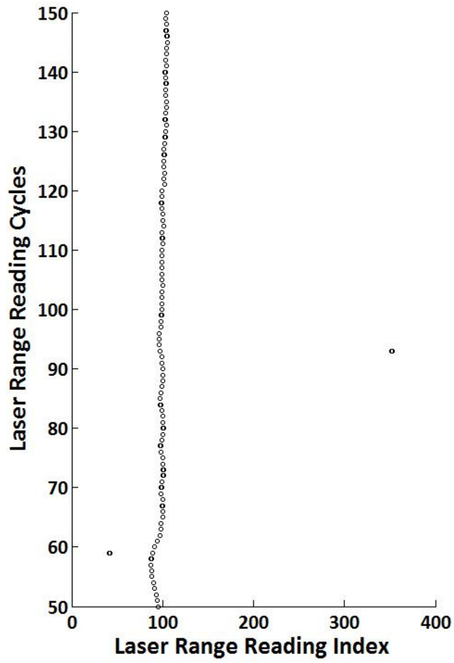


Figure 7.3: Detected curb candidates over multiple laser range cycles.

surface texture. The curb detection method is also able to perform when the walkway is relatively smooth. This smooth surface texture reduces the number of false positives since the method achieves detection by investigating the nature of the surface texture using range readings. The other condition that is necessary for the performance of the curb detection method is that the ground needs to be sufficiently dry. This is because of the reflective properties of the water. Water diffuses the reflections of the laser causing errors in range calculations due to incoherent reflections.

7.4 Dropped Curb Detection Experiment

To evaluate the developed dropped curb detection method, a dropped curb detection experiment is carried out. This experiment aims to determine the method's ability to detect variations in the height of the curb over multiple range cycle readings. The experiment also aims to investigate how lowering the detection threshold affects the reliability of the dropped curb detection method.

This experiment involves controlling the MoS along the end section of the walkway while the downfacing laser sensor acquires range data about the surface. At this section, the curb that borders the walkway lowers to a height close to that of the road while forming a corner. The acquired range data of the down facing laser is loaded into Matlab so that the data can be visualised and the dropped curb detection method can be implemented on this data. The detection threshold of the dropped curb detection method is set to vary from $5cm$ to $3cm$. The number of skipped consecutive range readings is set as 5.

From Figure 7.4, the variation of the detected curb can be seen. This variation shows how the height of the curb lowers to indicate the dropped curb feature. The dropped curb shown in these readings still maintains a height above the ground. To detect this height, the detection threshold is lowered to $3cm$. By lowering this threshold, the corner that is formed by the lowered curb can be detected in the range readings as shown in Figure 7.5. However, lowering the detection threshold increases the number of false detections as shown in Figure 7.5, making this detection method unreliable.

7.5 Grass Detection Experiments

The grass detection method is evaluated by processing sensor data collected from two different experimental environments. The first location, Figure 7.6(a), is a walkway that is bordered by a curb on one side and grass on the other while the

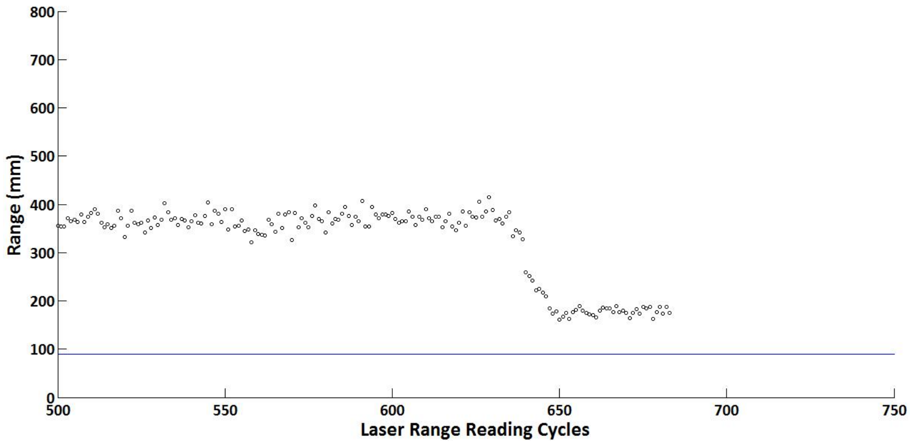


Figure 7.4: Variation in the detected curb heights over multiple laser range reading cycles.

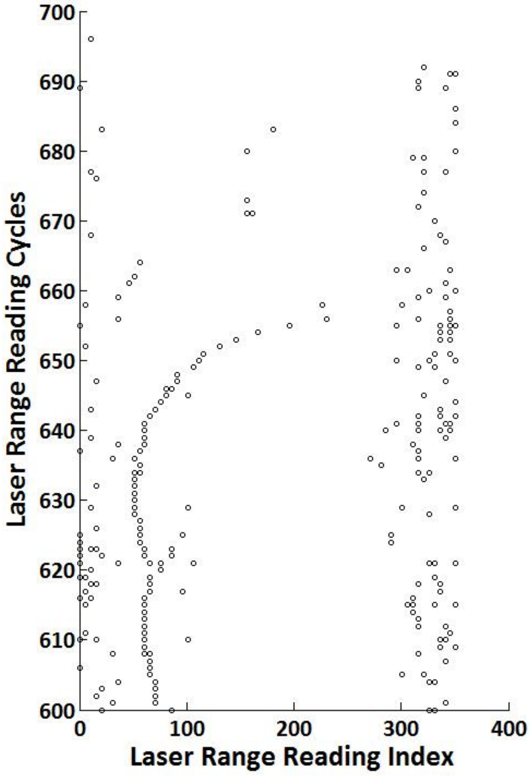


Figure 7.5: Detected dropped curb candidates over multiple laser range cycles.

(a)

(b)

Figure 7.6: Experiment environments with the walkway bordered by grass (a) curb and grass (b) wall and grass.

second location shown in Figure 7.6(b), is a walkway that has gravel and a wall on one side and grass and a hedge on the other side.

The graph in Figure 7.7 shows the performance of the grass detection method. The detection method uses the non uniformity of the range data that is returned by rough surfaces to infer the presence of grass. Due to the low threshold for the difference detection, this method generates a large number of candidates each cycle with some lying on the walkway. This is because of the fact that sometimes there are obstacles such as leaves or papers on the walkway that the method detects as being part of the walkway concluding that the path has a rough texture. This method is also sensitive to noise in the range readings that occurs at the edges of the laser detection span.

The graph in Figure 7.8 shows the grass detection method that is extended to consider the density of the detected grass candidates. The candidates that are close to other candidates form a group that defines the area in-between as being part of the grass region. Then those candidates that do not have any neighbours close to them are taken as false positives and discarded from the grass candidates set.

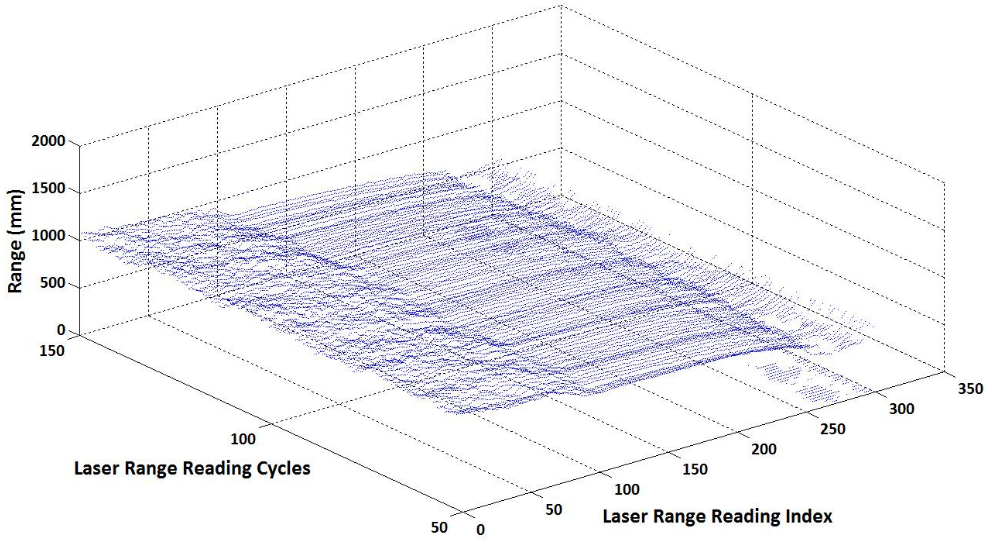


Figure 7.7: Laser range readings of walkway bordered by grass.

7.6 Wall and Hedge Detection Experiment

The graph in Figure 7.9 shows range readings from a horizontal laser sensor. The left shows a hedge or plants that have been detected. The distribution of the laser strikes is uneven which is due to the uneven nature of the leaves and branches of the plants. Though the readings are uneven, it can still be noticed that they follow a general direction that is similar to that of the walkway. In this same graph, the right shows a wall that is detected by the laser sensor. The laser range readings that form this detected wall appear to be evenly distributed compared to those of the hedge. The readings form a line that follows the geometry of the wall and since the wall is aligned with the walkway, the geometry of this line can be used to provide information on the direction of the path.

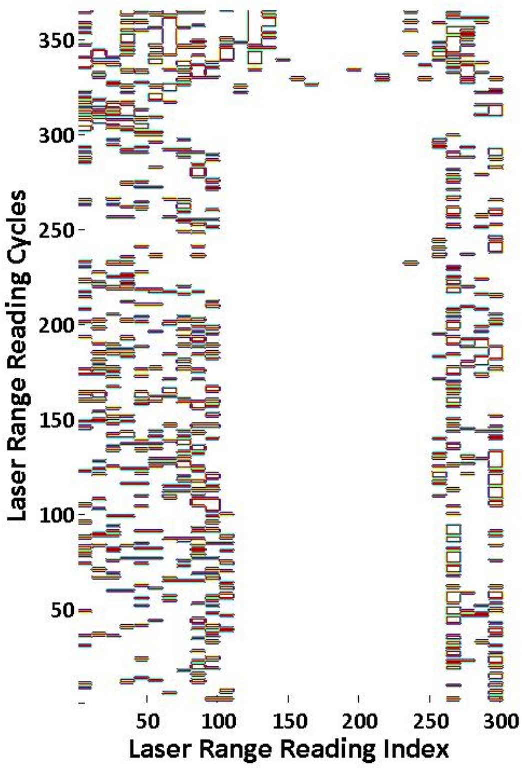


Figure 7.8: Detected grass candidates over multiple laser range cycles.

7.7 GPS Precision Experiments

To check for the precision of GPS data, a series of tests are carried out. These tests have the mobility scooter move repeatedly on the same route while logging the GPS data. The first set of data has the mobility scooter move in a straight line route. The second has the scooter move in a route that contains two right angled turns The logged GPS data is plotted on a graph to show the variation of both the position and bearing values.

The graph in Figure 7.10 shows the position information that was logged as the MoS was remote controlled along an outdoor straight route. To determine the precision of the GPS data from the receiver, the MoS was moved along the

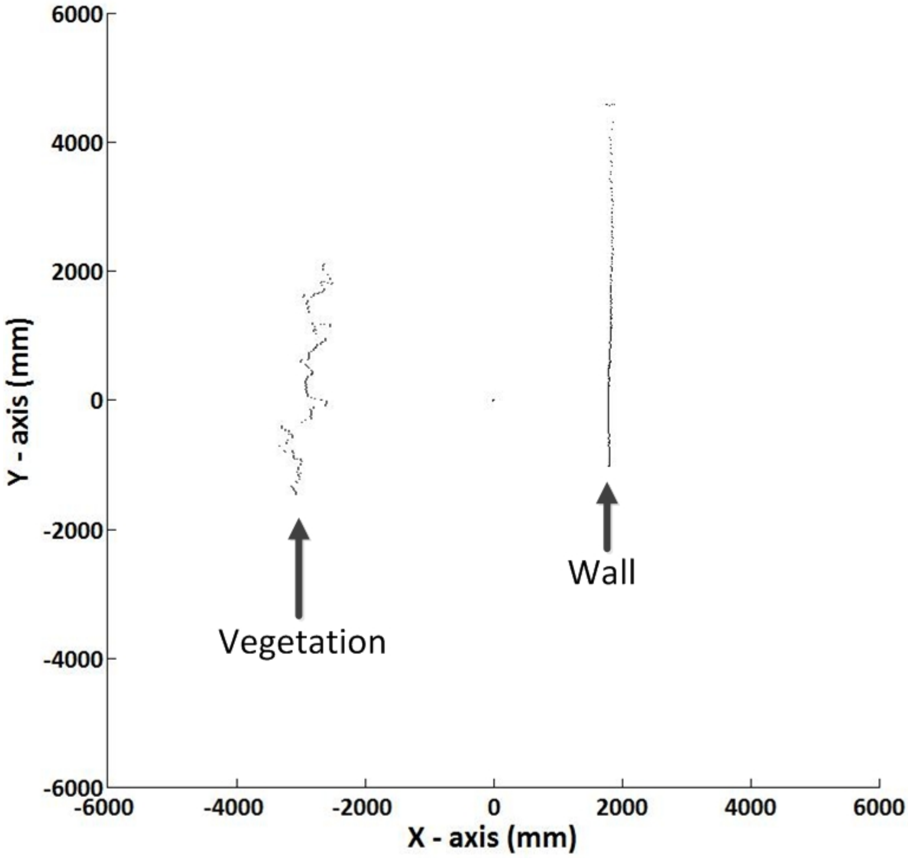


Figure 7.9: Range readings of horizontal laser sensor showing a detected wall and hedge.

road and then it was moved along the sidewalk that was beside the road. The distance between the two routes was approximately $3m$. The GPS location data is plotted on the same graph. It can be seen that the precision of the GPS allows for these two routes that are $3m$ apart to be distinguished. To determine the repeatability of these results, the MoS is moved over these routes multiple times all within a time span of 4 hours. The graph shows that although the routes can be distinguished, there are instances when the routes appear close together. This is due to the drift that exists in GPS readings. Due to this drift, GPS data is usually supplemented with other navigational methods to improve the overall

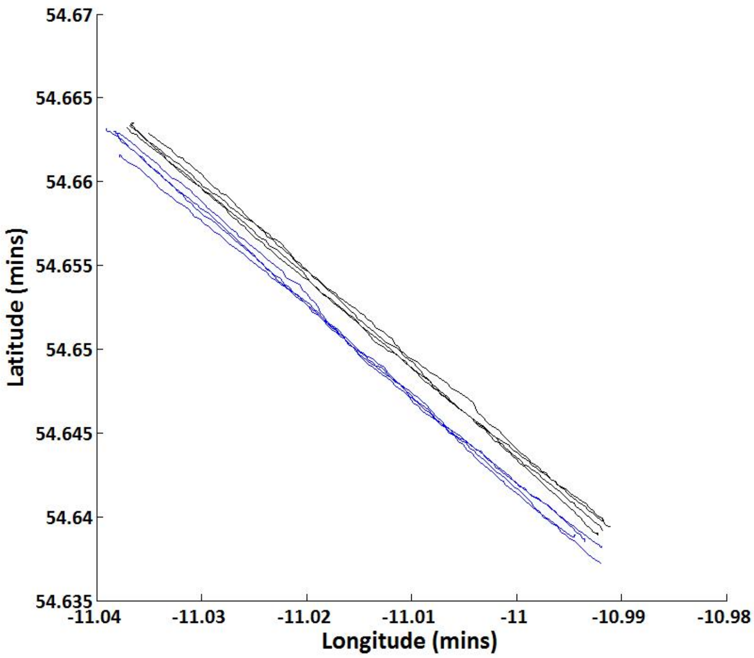


Figure 7.10: GPS location data logged while MoS moves along a road and a sidewalk.

reliability of the navigation system.

The accuracy of GPS cannot be calculated from the readings produced by the receiver. This is due to the fact that we are unable to determine the ground truth of the MoS position. Without this ground truth, the error which is the difference between the measured value and the ground truth cannot be calculated. A reasonable compromise is the calculation of the variation in the GPS location readings. For the route data presented in Figure 7.10, a line of best fit could be calculated for points produced after multiple runs on the same route. Then the shortest distance of the points from this line could be used to provide a characterisation of the precision and accuracy of the GPS readings.

Then the MoS is moved along a route that has two right angled turns while the GPS position and bearing / heading data is logged. The GPS location data

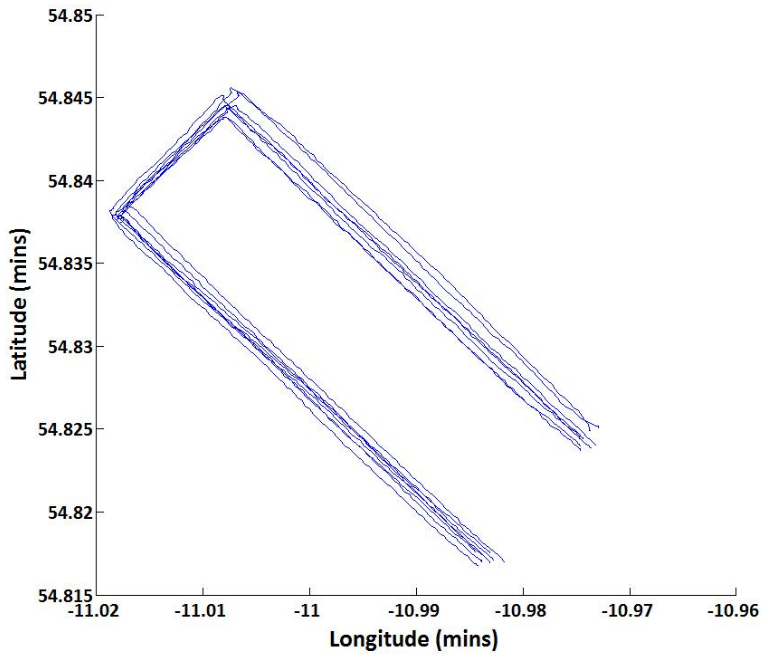


Figure 7.11: GPS location data logged while MoS moves along a route that has two right-angled turns.

gathered during multiple runs of the MoS along this route is logged and plotted in a graph shown in Figure 7.11. From the graph, it can be seen that the shape of the route corresponds to that of the real world despite the fact that the readings contain measurement drifts caused by GPS signal errors.

The graph in Figure 7.11 shows the variation of the bearing of the MoS for a single route. The straight sections of the route produce fairly steady bearing readings while the turns produce distinctive changes in bearing. This property can be used to gather information about a route through the observation of the variation of the bearing values during navigation.

To demonstrate the property of ascertaining information from variation of bearing values during movement of the MoS, an experiment is carried out that has the MoS move along a sidewalk then cross a road to the opposite sidewalk and

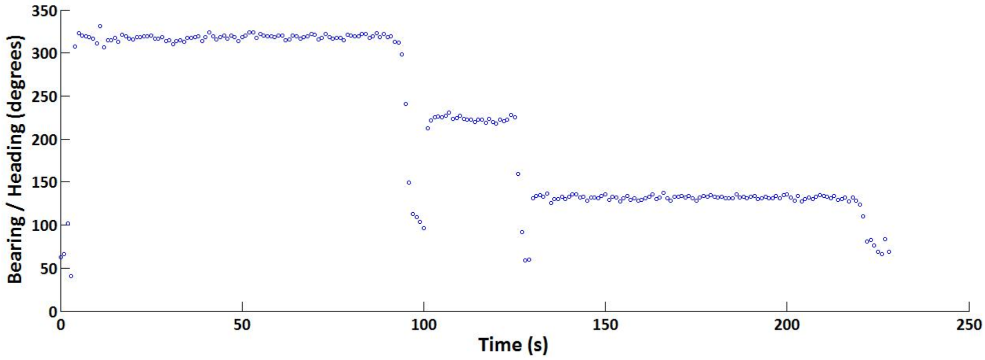
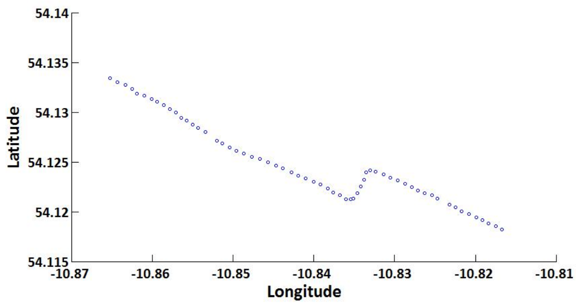
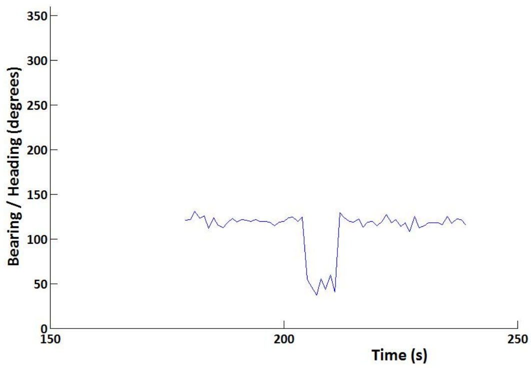


Figure 7.12: GPS bearing data logged while MoS moves along a route that has two right-angled turns.



(a)



(b)

Figure 7.13: GPS location data logged while MoS crosses a road (a) location data (b) bearing / heading data.

then keep moving. The GPS data of this outdoor road crossing is shown in Figure 7.13(a) and Figure 7.13(b). The graph in Figure 7.13(a) shows the position data with the sections of the course changes visible. When this same information is represented in Figure 7.13(b) showing the bearing variation, the bearing changes that correspond to the course changes are also visible. To demonstrate the drift, another graph is shown with multiple readings of the same crossing route plotted together. from this graph it can be seen that although the drift makes the route appear in different locations, the overall shape of the route is retained.

The real world environment is used to conduct experiments that characterise the precision of the localisation data produced by the GPS receiver. The bearing following and waypoint navigation algorithms on the other hand are tested in a virtual environment. This is due to the fact that the virtual environments offer scenarios that could otherwise be difficult to setup in the real world. The characteristics of the GPS receiver determined from the real-world experiments are used in the virtual environment to provide a more consistent set of results.

7.8 Virtual MoS in V-REP

A virtual model of the MoS is used in the simulation experiments. The virtual model of the experimental platform has got dimensions that are approximately equal to those of its real life counter part. This model too has a differential drive system. The model is fitted with laser sensors that produce range readings similar to those in the real world. The placement of the sensors on the virtual MoS corresponds to that of the real world experimental platform. This MoS is placed in a virtual environment that is constructed to have characteristics similar to those of a typical urban outdoor environment. The navigation algorithms are programmed into this MoS and the simulator is run to evaluate the developed

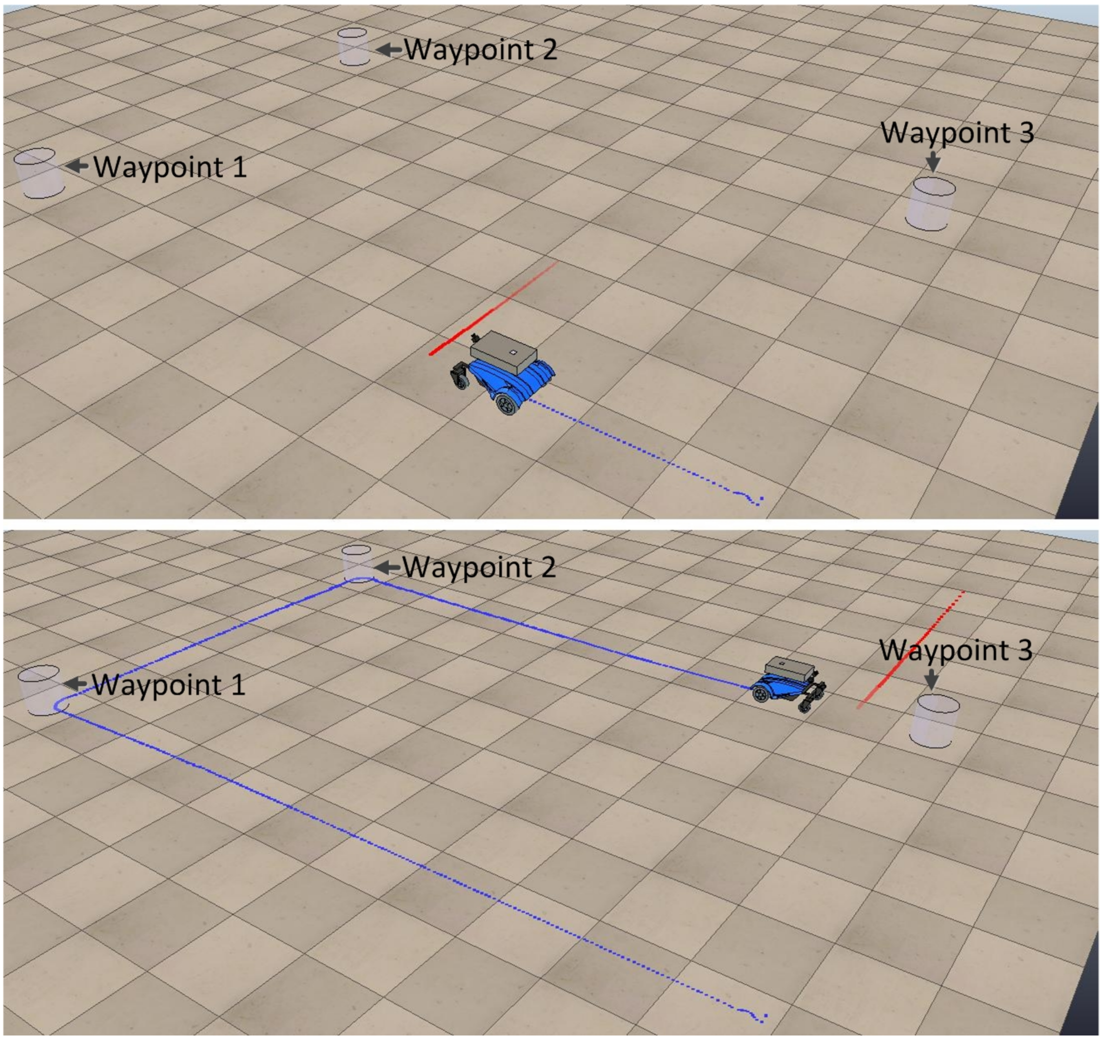


Figure 7.14: GPS navigation of MoS simulated in V-REP.

assistive navigation system.

7.9 Waypoint Navigation Experiments

The bearing and waypoint following algorithms are evaluated in both the real world and in simulation. However, due to the ease of setting up experiments, these algorithms are tested more extensively in simulation than in the real world environment. Figure 7.14 and Figure 7.15 shows the simulation environment in which the simulated model of the MoS uses the GPS navigation algorithms to follow a route. This route is defined by three waypoints whose positions are marked visually in the simulator by cylinders. In this simulation, the bearing envelop value is set to 10^o while the minimum distance threshold is set to $0.5m$. The MoS navigates to each of the waypoints in the the specific order that they appear in the route data. The trajectory of the MoS as it navigates this route is shown in the simulator and it clearly shows that the GPS navigation algorithms are able to lead the MoS to each of the waypoints that are defined in the route set.

7.10 Curb Following Experiments

A curb following experiment is setup in V-REP This experiment is aimed at evaluating the developed curb detection and following technique. The results of the experiment are investigated to determine if the MoS maintained a safe distance from the curb as it followed the walkway.

The virtual environment constructed in V-REP includes a walkway that is bordered by a road on one side and a clear field on the other side. This walkway has a straight section that leads into another section that is curved with a radius of $6m$. At the border between the walkway and the road, there is a curb with a height of $10cm$. The curb following algorithm is programmed into the virtual MoS. The fringe regions are set to lie between laser indices 200 and 220 for the right side and 450 and 470 for the left side. A function is added to the program of the MoS that instructs the simulation to draw distinctly coloured points at each

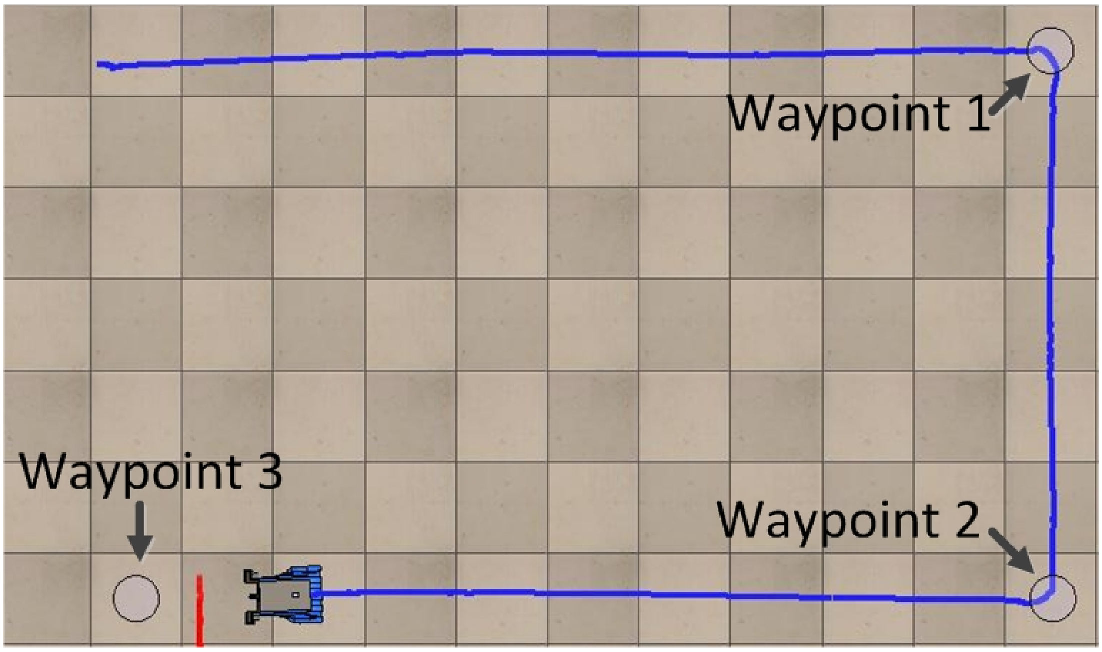


Figure 7.15: Overhead view showing simulated model of MoS navigating a GPS defined route in a virtual outdoor environment constructed in V-REP.

of the positions that the MoS occupies during simulation. These points represent the trajectory of the MoS as it moves in the environment during simulation.

The simulation is started and made to run for an appropriate time duration. During the simulation, the MoS identifies and follows the curb that borders the walkway. The MoS positions itself at a safe distance from the edge of the walkway by ensuring that the detected curb is within the fringe region as programmed by the algorithm. Figure 7.16 and Figure 7.17 show the MoS following the walkway with the trajectory indicating the performance of the curb following algorithm in the simulated environment. The trajectory of the MoS shown in Figure 7.17 demonstrates the precision of this curb following algorithm. Figure 7.18 shows how the MoS uses sensor feedback motion control to maintain the position of the detected curb within the defined fringe region.

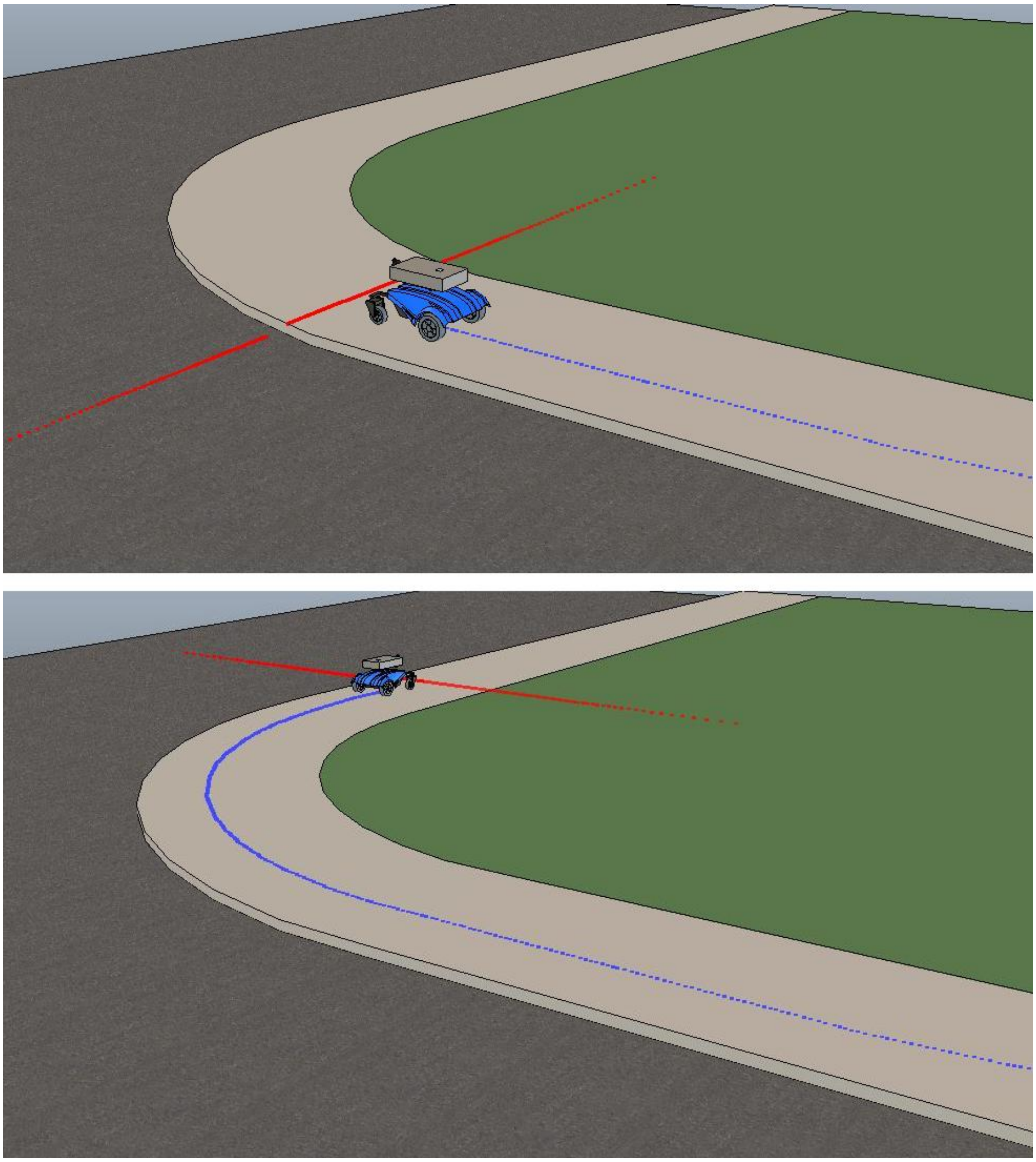


Figure 7.16: Curb following navigation behaviour of a simulated MoS model in a virtual environment constructed in V-REP.

The simulation of the curb following algorithm in a virtual environment shows the feasibility of the curb following algorithm. However, this algorithm relies on the MoS being placed at an appropriate position at the start of the simulation.

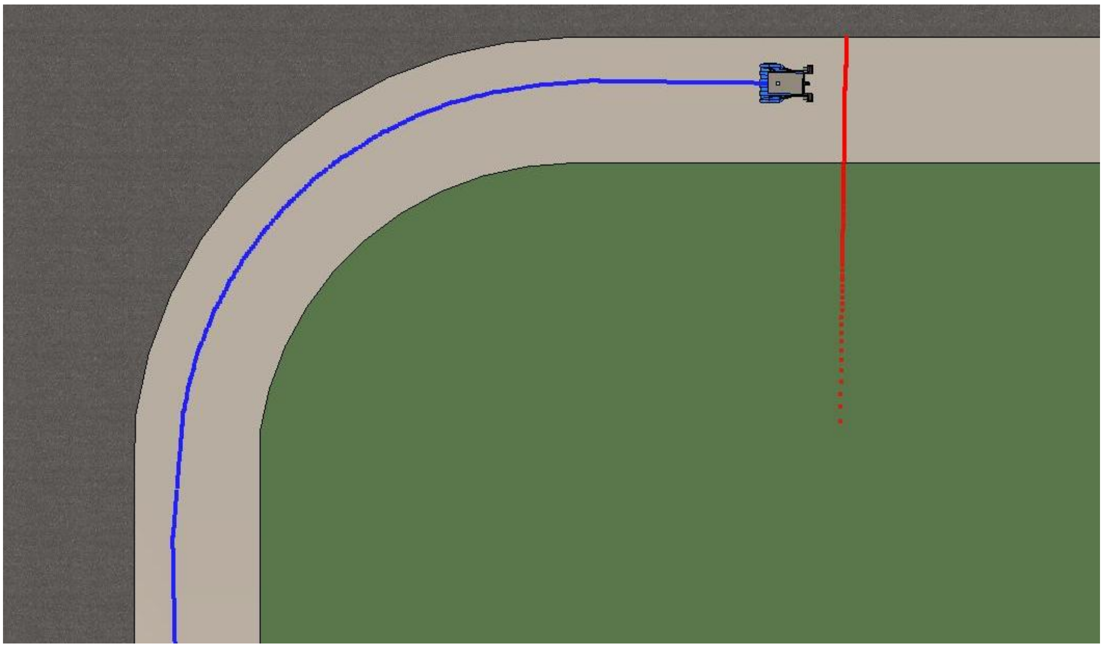


Figure 7.17: Overhead view of curb following navigation behaviour of a simulated MoS model in a virtual environment constructed in V-REP.

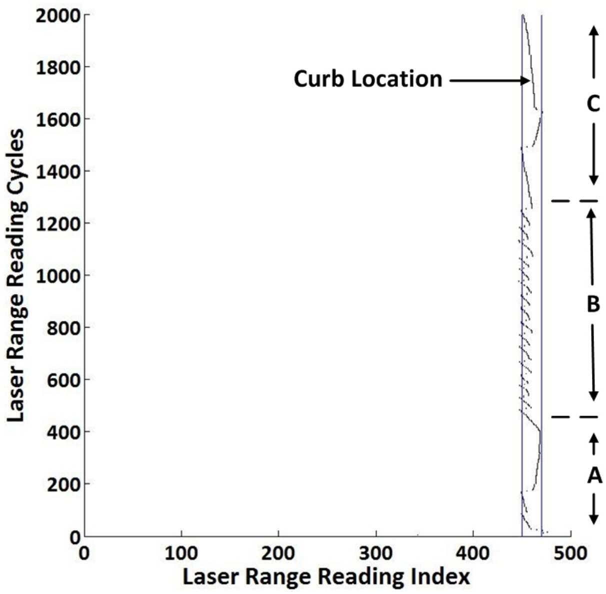


Figure 7.18: Location of the detected curb in the laser readings as the MoS navigates using the curb following behaviour

This means that the MoS has to be facing in the direction of the path and it should be a reasonable distance from the curb. The simulated environment also has curb edges that are geometrically perfect while in the real world these edges are subject to wear and tear that degrades the structures. The sensors in the simulated environment return perfect noiseless laser range readings while in the real world the sensor readings contain noise that is caused by artefacts like mirrors and puddles of water that are typical of an urban outdoor environment. Despite these factors, the experiment run in the simulated environment does provide a verification of the functionality of the curb following algorithm.

7.11 Wall Following Experiments

The wall following experiment is setup in V-REP. This experiment aims to evaluate the wall following algorithm of the MoS. This is done by checking to see if the MoS is able follow a wall at a specified minimum distance.

The virtual environment used for this evaluation consists of a wall that is next to a road. This wall has a straight section and a another section that is curved with an approximate radius of $4m$. The wall is solid and continuous with no spaces anywhere inbetween. The wall following algorithm is programmed into the Virtual MoS with the safety distance set to $1m$. This distance is allowed to vary up to $1.1m$. The MoS is positioned in an appropriate position with its front facing the same direction as the wall.

The simulation is run for an appropriate time to allow for the MoS to move in the virtual environment. Figure 7.19 shows the MoS as it moves in the virtual environment. In this figure, it can be seen that the simulator draws a trajectory

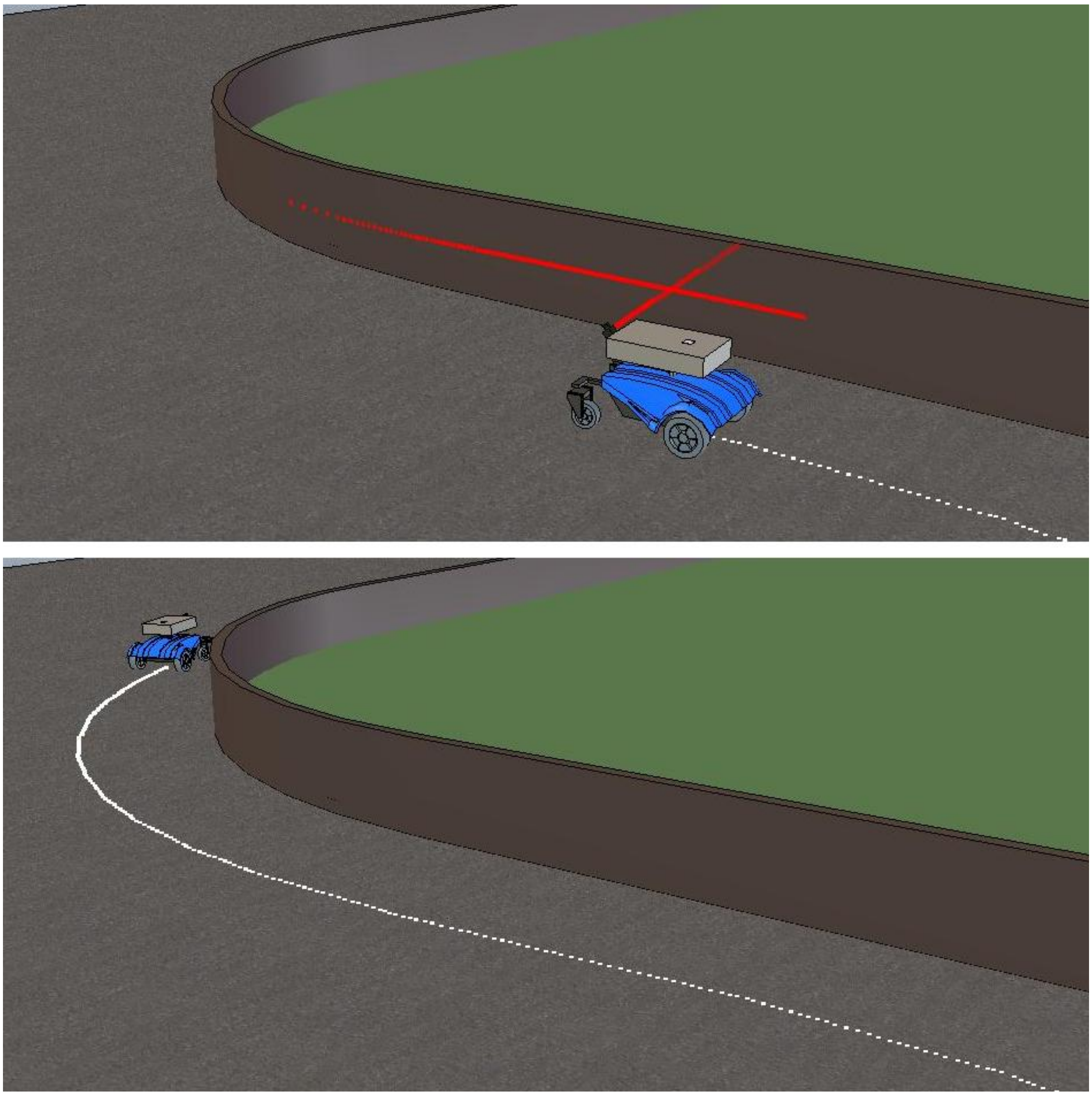


Figure 7.19: Wall following navigation behaviour of a simulated MoS model in a virtual environment constructed in V-REP.

that indicates the previous positions of the MoS. This algorithm uses the horizontal laser sensor predominantly and this can be seen in the figure as a strip of light reflected off the wall.

Figure 7.20 shows the performance of the wall following algorithm in the virtual environment. In this figure, the trajectory of the MoS shows how the MoS

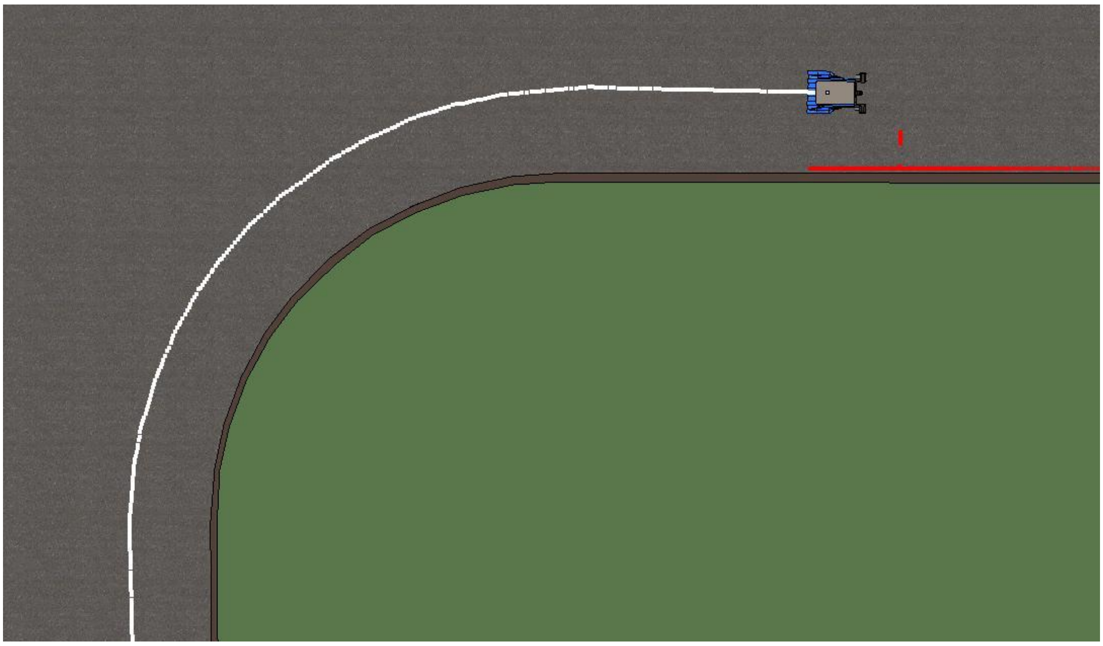


Figure 7.20: Overhead view of wall following navigation behaviour of a simulated MoS model in a virtual environment constructed in V-REP.

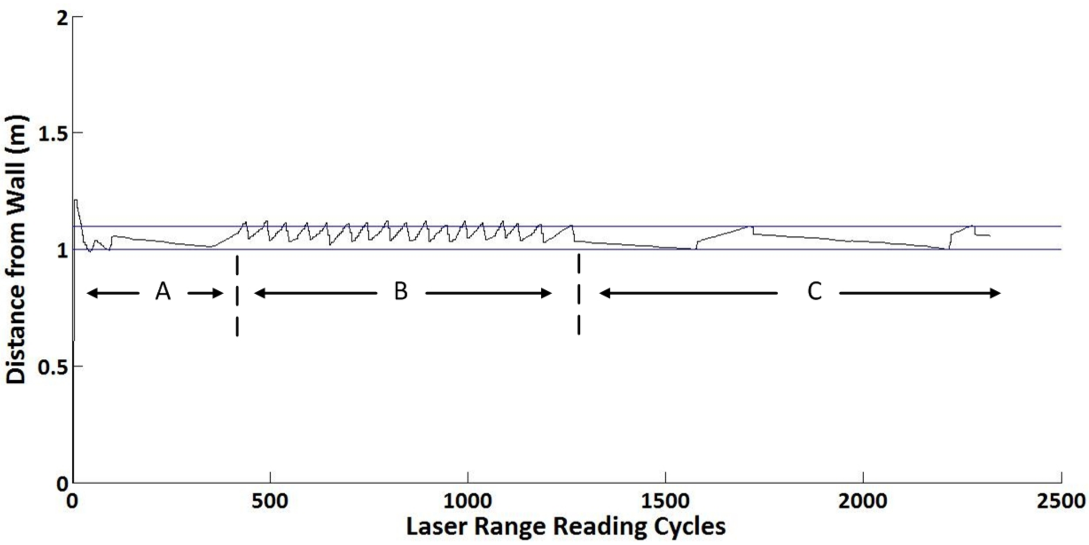


Figure 7.21: Distance of the MoS from the wall as it executes the wall following behaviour

is able to follow the wall while maintaining an approximately constant distance from the wall. Figure 7.21 shows how the MoS uses sensor feedback motion control to maintain the specified distance from the wall.

The virtual environment demonstrates the functionality of the wall following algorithm. It should be noted that, in the virtual environment, the sensor noise that is common in real world scenarios is absent. This means that the performance of the wall following algorithm might not be as high as it is in the simulation. An example of this is the variation of the safety distance during navigation, which is anticipated to vary more in the real world due to sensor noise and noise in the actuation system of the UAS. Despite these factors, the simulation does provide an adequate evaluation of the functionality of the wall following algorithm.

7.12 Behaviour Interaction Experiments

The navigation architecture uses reactive components that allow it to respond appropriately to any changes to the environment during navigation. The way in which the different navigation behaviours interact is investigated in experiments that are carried out in both the real world and the simulated world. The first scenario considered is the situation where the walkway has a lowered curb. In this instance, the MoS has to switch from following the walkway using the curb following behaviour to following the walkway using another behaviour. The other behaviour is dictated by the features that are available, for example if there is a wall present then the MoS will switch to the wall following behaviour or if there isn't a wall then the MoS will switch to the GPS following behaviour.

In the simulation shown in Figure 7.22 and Figure 7.23, the walkway has a lowered curb and a wall. The MoS trajectory in Figure 7.24 shows how the MoS keeps on following the walkway despite the lack of a curb. In this instance, the

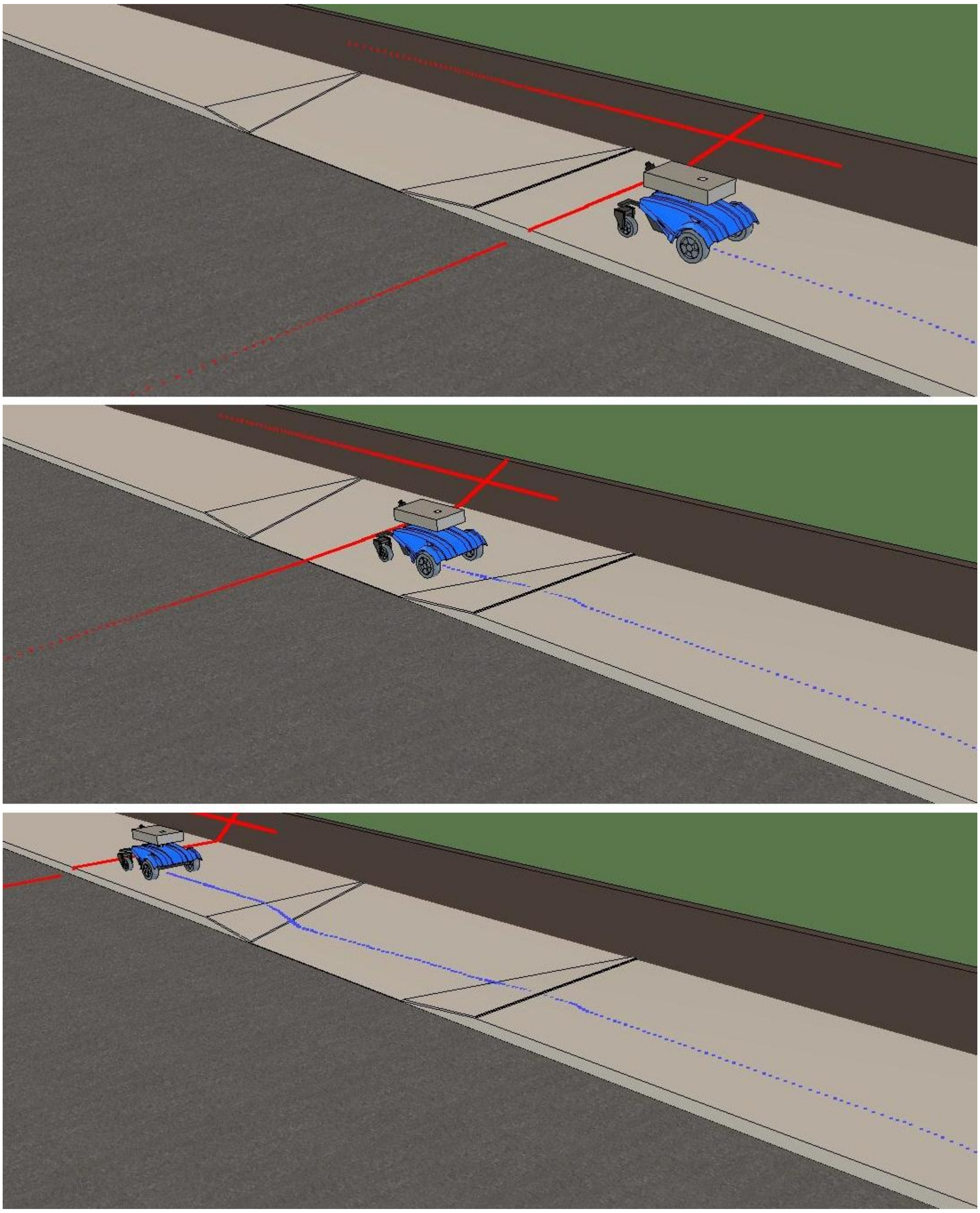


Figure 7.22: Navigation system switching behaviours from curb following to wall following and then back again to curb following in V-REP.

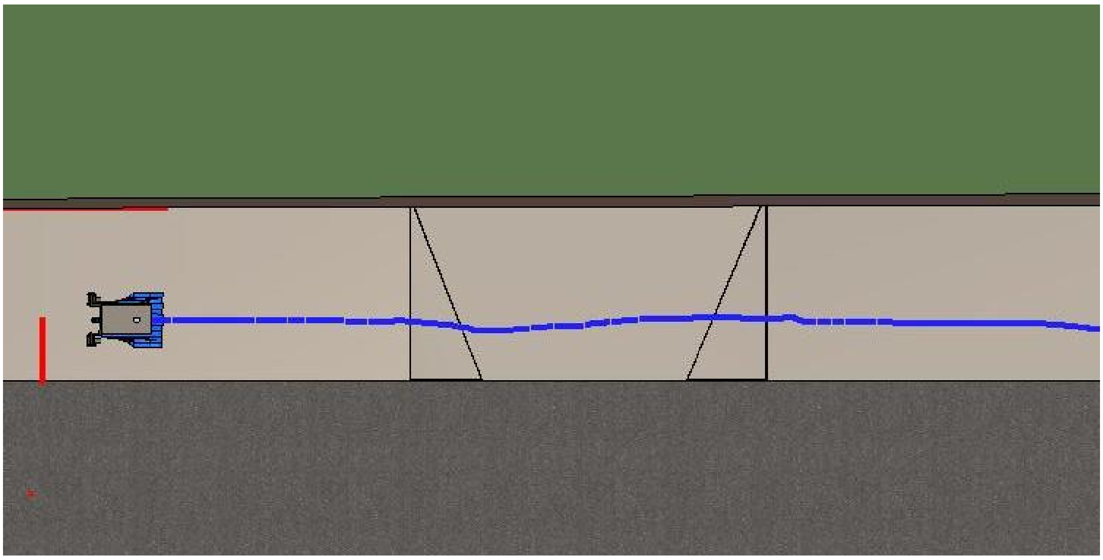


Figure 7.23: Overhead view of navigation system switching behaviours from curb following to wall following and then back again to curb following in V-REP.

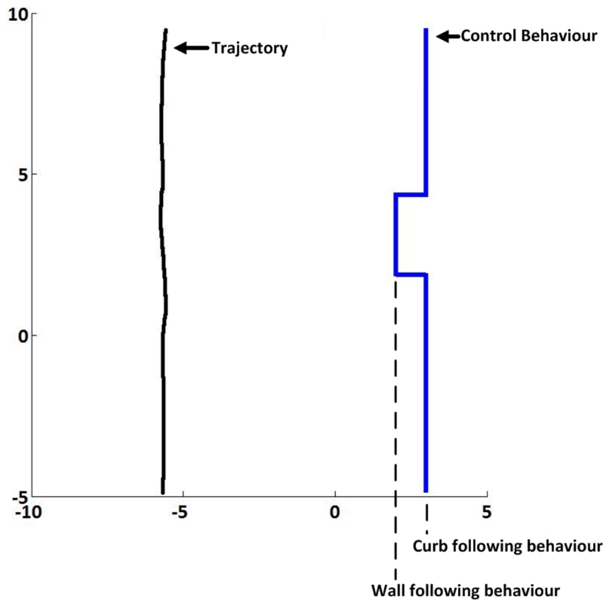


Figure 7.24: Switching the navigation control behaviour as the MoS navigates along a walkway

MoS switches to wall following and this allows it to acquire the direction of the walkway and then follow that direction. Then when the curb reappears after the

MoS navigates past the lowered curb, the MoS switches back to the curb following behaviour.

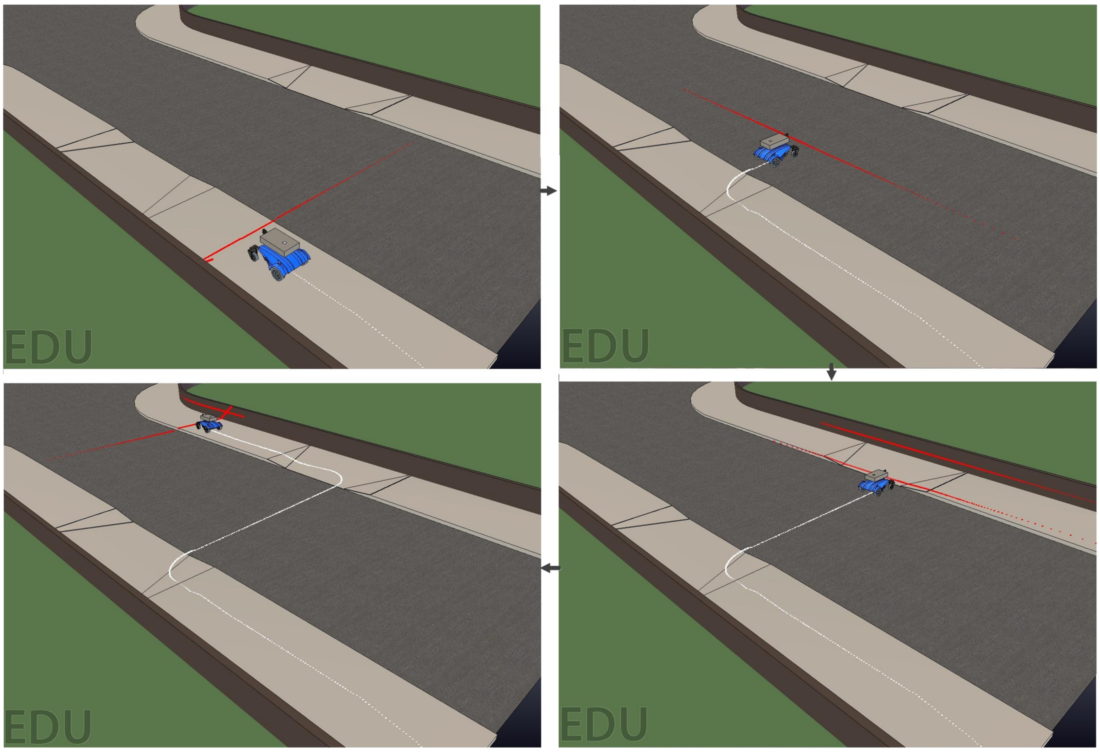


Figure 7.25: Simulated model of the MoS crossing the road to get to the opposite sidewalk in a virtual outdoor environment constructed in V-REP.

7.13 Road Crossing Experiments

The road crossing algorithm of the MoS is evaluated in simulation using V-REP software. In this simulation, a road that has sidewalks located on opposite sides is constructed. Each of these sidewalks has lowered curbs that are aligned on opposite sides of the road. The sidewalks are both bordered by walls on the side that is opposite to that of the road. This virtual outdoor environment is constructed with dimensions that closely approximate to those of a typical outdoor urban environment. The width of the road is $5m$, the width of both walkways is $2m$ and the width of the lowered curb is $2m$.

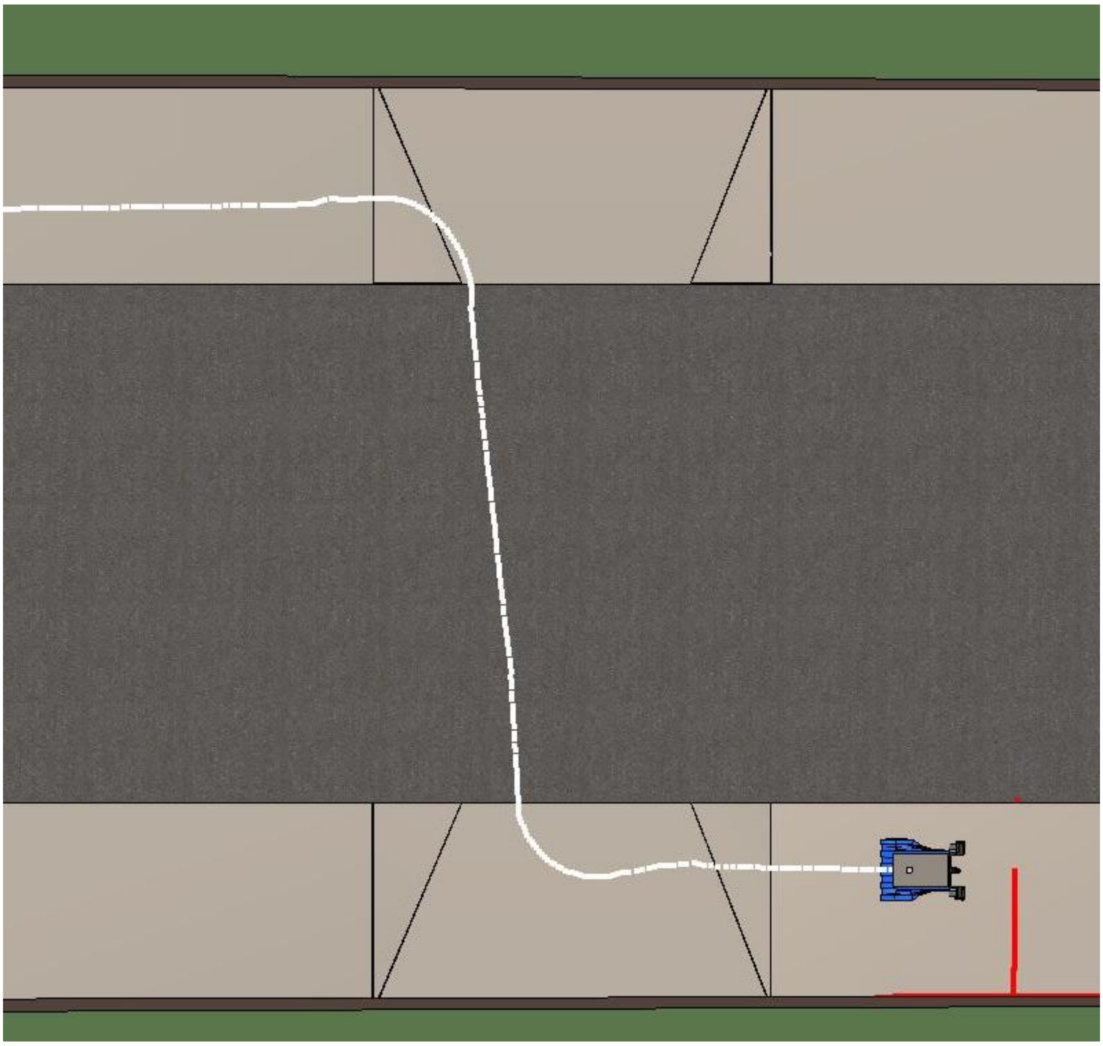


Figure 7.26: Overhead view of the MoS crossing the road to get to the opposite sidewalk in a virtual outdoor environment constructed in V-REP.

Figure 7.25 and Figure 7.26 show the performance of the road crossing algorithm. The MoS follows the curb as it searches for a suitable point from which to leave the sidewalk. When the MoS detects a lowered curb, it leaves the sidewalk following a bearing that is orthogonal to that of the sidewalk. This bearing allows it to cross the road and when it reaches the opposite sidewalk, the MoS detects the wall and then switches its bearing to that of the sidewalk. The MoS follows the sidewalk by following the curb that is bordering the sidewalk. This is shown by the trajectory of the MoS as it navigates the environment.

7.14 Conclusion

The experiments are carried in a real environment and a virtual environment. Though these environments are different, they do provide opportunities to evaluate the developed navigation components in an efficient manner. The real world environment provides valid evaluations for the perception of the real world features that exist in pedestrian walkways. The virtual environment allows for experiments that would have otherwise been time consuming to setup in the real world.

From the experiments, it can be seen that of the features found in an outdoor urban environment, the curb is the most distinct feature. This is then followed by the detection of walls and obstacles. It can also be seen that the grass detection method and the lowered curb methods are not reliable. This unreliability is due to the fact that these methods require the lowering of the detection threshold which increases the number of noisy range readings.

The waypoint navigation experiments display the ability of the MoS to follow a route that is defined by GPS coordinates. However, due to the precision limitation of the GPS signal this navigation strategy needs to be supplemented to achieve navigation on a walkway.

The behaviour navigation experiments display the ability of the MoS to navigate along a walkway that is bordered by various features.

Chapter 8

Conclusion and Future Work

8.1 Introduction

Autonomous navigation in any environment is a challenging task. For an outdoor urban environment, this task is no less challenging but a few factors make it more tractable. The first factor is the presence of geometric structures in the environment. These structures like curbs, sidewalks and barriers follow a set of rules that are governed by the building and construction codes of the area. By assimilating this information into the navigation system, the MoS is able to make some assumptions about the environment like the geometry of the path.

8.2 Ultrasound Sensors

For obstacle detection, the MoS uses laser range sensors. These offer a higher accuracy and resolution compared to ultrasound sensors. However, the laser sensor does not detect the presence of glass. Due to the nature of laser sensors, this material is invisible to the sensor. Glass is a common obstacle in urban outdoor environments. It is used by buildings that want to display their interior structure to the pedestrians for example shops. In such instances the MoS would be unable to detect the glass wall of the building.

To aid with this limitation, ultrasound sensors can be used for obstacle detection. They can be used to supplement the laser sensors and provide range readings in situations where the laser sensors fail to detect the presence of glass. Ultrasound sensors are cheaper than laser sensors so their addition would not strongly impact the hardware development costs.

8.3 Dynamic Obstacles

Urban outdoor structured environments contain a significant amount of dynamic obstacles. These usually include vehicles, people and pets. However due to the nature of the laser range sensor data, these obstacles are indistinguishable from static obstacles unless further processing of data is carried out. Detection of dynamic obstacles by the navigation system would lead the MoS to make better navigation decisions. These decisions could include whether to wait for a dynamic obstacle to clear out of the path or to plan an alternative path to the goal destination.

8.3.1 Pedestrians

Pedestrians share the same path as the MoS. This makes them the most likely obstacles that the MoS will detect as blocking the navigation path. If the navigation system is able to detect and identify obstacles as pedestrians, then it could assume that the path is not blocked and as such there is no need for planning an alternative route.

Detection of pedestrians could be done in two phases. The first phase could be to detect the presence of an obstacle that lay in the boundaries of the walkway. The second could then determine if the detected obstacle was moving. This way the pedestrian could be detected as an obstacle within the bounds of the walkway that is in motion.

This method assumes that a pedestrian is constantly in motion. When this assumption fails to hold then the pedestrian is classified as a static obstacle. Although this method is limited, it provides a classification for a dynamic obstacle using the same sensor data that is used to detect other obstacles. This ensures that there is no need for installing a new sensor type that is dedicated to pedestrian detection, thereby maintaining the original hardware development costs.

8.3.2 Cars

For the MoS, the detection of road vehicles is not enormously vital. They can be simply detected as obstacles without the added classification. There are times however, when classification can be used to gain a better understanding of the environment by the navigation system. An instance could be when the MoS is crossing the road and it needs to detect the presence of an oncoming car.

The detection could use the assumption that all cars are on the road and those that are not on the road are stationary and should be considered as ordinary obstacles. Then using the range data, any obstacles that would be detected as located on the road could be classified as cars. Given the location of the MoS on the walkway and the characteristics of the laser range sensors, it is not possible to determine the shape of the car.

8.4 Assistive Technology

Due to the unpredictability of the environment, it is not entirely possible to make a fully autonomous navigation system. But even though a system was made to be fully autonomous, there are other huddles to over come like the user not willing to relinquish complete control to the system. The user will always need to be part of the control loop of the MoS. The compromise is to make a system that assists the user to navigate in the environment by offering warnings or corrections to their commands. This puts the problem of autonomous navigation into the realm of assistive technologies. With assistive technology, one of the tasks involves determining what parts of the navigation problem should be allocated to the navigation system and which parts should be left to the user.

8.5 Concluding Remarks

Concluding Remarks: The rise in MoS usage has led to a need for improved safety of both the MoS and the pedestrians that share the environment with it. The project aimed to meet this need by investigating the feasibility of a UAS capable automating some of the functionality of a typical MoS.

The investigation of the research found that the autonomous navigation problem in an outdoor urban environment is made more tractable by the fact that it is structured, and that rules govern the movement of the dynamic obstacles within the environment. This allows for the movement of the dynamic obstacles to be anticipated. Also the structure of the outdoor urban environment allows for a destination to be reached by following a walkway thereby further simplifying the navigation task.

This project uses the strategy of performing outdoor navigation in the environment by identifying and following a walkway inorder to get to the destination. In a typical outdoor urban environment, the walkway is made easily distinguishable to allow the pedestrians and other environment users to clearly identify it. This is made possible by the presence of distinct features that are placed on or around the walkway enabling it to stand out from the rest of the surrounding environment. Some of these distinguishing features enable it to be detected by laser range sensors. Of all these features, the curb is the one that is most closely associated with the presence of a walkway. There is research available that was used to detect the curb but for road navigation. This research can still be used for walkway identification and navigation. The project relies on identifying the boundaries of the walkway which does not work well in open spaces that are common of shopping areas.

The localisation in an outdoor environment is made easier due to two factors. The first factor is the availability of digital maps. The second factor is the availability of GPS. Though this localisation method is not very precise, it does serve as an overall global localisation strategy. This localisation strategy is supplemented by a local navigation strategy that uses local detected features.

The navigation architecture uses periodic planning which is used to calculate a global route. The movement in the environment is controlled by a reactive architecture. Due to the structured environment, some of the interactions of the MoS can be anticipated and then responses can be programmed that enable it to handle them. This reduces the need for deliberative functions in the navigations architecture. Also, the fact that it an assistive system means that it can transfer control to the user if it identifies a scenario that is does not know how to handle. The task is getting the MoS to correctly identify those scenarios that it is capable of handling and those that are out of its programmed abilities parameters.

8.6 Future Works

Future Works: The obstacle detection component of the navigation system could be modified to include the detection of dynamic obstacles. The detection strategy can use the multiple cycles range readings of the laser sensors to compare the positions of obstacles over time. The dynamic obstacles can then be classified as those obstacles that are change positions over time indicating that they are in motion.

Another localisation method can be added to supplement gps. This could include an inertial measurement unit and wheel encoders for odometry. GPS has a low update frequency which limits the localisation functions of the navigation system. IMU and odometry provide update rates of 50Hz and this enables the navigation system to determine its position inbetween GPS updates. IMU are easier to install compared to wheel encoders that require mechanical installation on the MoS.

A local path planner can be developed for local navigation. This path planner can then be compared to the reactive navigation strategy. The comparison would aim to determine which of the two local navigation strategies produced smoother movements by examining the trajectories produced by the MoS.

Testing the overall navigation algorithm in an outdoor urban environment. This is to ensure that the algorithms that work in the simulation work adequately in the real environment too. Different feature variations can be identified and used to test and evaluate the robustness of the feature detection methods. Various outdoor urban environment layouts can also be identified and used to test the navigation algorithms.

A user interface could be developed for the MoS that allows the user to input navigational commands. The interface could also be useful in conveying information to the user about the status of the MoS navigation. The design of the interface needs to take into account the fact that the MoS is used outdoors, which means that the information needs to be conveyed to the user regardless of the illumination conditions. The interface design also needs to take into account the user-base of the MoS and provide clear and intuitive controls and messages.

A study could be conducted to determine the usability of the MoS. This study could start by identifying the different categories of users of mobility scooters. Then a sample from each category could be asked for their view on a User Assisted Mobility Scooter. These questions would aim to understand the expectations of the users from a system that has the capability of autonomous navigation and whether they viewed it as a welcome functionality. Then the participants would be provided with a controlled demonstration of the UAS, that allowed them to experience the functionalities of the UAS first hand. After this demonstrations, the participants would provide feedback on their experience. This feedback and the rest of the other usability information gathered in the study could then be used to provide further avenues of development of the UAS allowing it to provide a more complete user assistive functionality.

References

[1] United Nations department of economic and social affairs. Population division, "Population ageing and development: Ten years after madrid", 2012. 1

[2] Ministry of Justice, "Uk road traffic act 1988", http://www.legislation.gov.uk/ukpga/1988/52/contents, 1988. 2, 21

[3] Transport and Travel Research Ltd, "Review of class 2 and class 3 powered wheelchairs and powered scooters (invalid carriages)", Tech. Rep., Department for Transport, 2006. 3

[4] House of Commons Transport Committee, "Mobility scooters, ninth report of session 2009-10", 2010. 3

[5] M. A. Porta Garcia, Oscar Montiel, and et al, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation", *Applied Soft Computing*, vol. 9, 2009. 11

[6] Katsumi Moriwaki and Katsuyuki Tanaka, "Navigation control for electric vehicles using nonlinear state feedback h control", *Nonlinear Analysis: Theory, Methods and Applications*, vol. 71, pp. e2920–e2933, 2009. 12

[7] Robin R. Murphy, *An Introduction to AI Robotics*, The MIT Press, 2000. 12

[8] Roland Siegwart and Illah R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Bradford Company, 2004. 12, 17, 18, 20

[9] Patrick Reignier, "Fuzzy logic techniques for mobile robot obstacle avoidance", *Robotics and Autonomous Systems*, vol. 12, 1994. 12

[10] Komatsu Ltd, "Komatsu: Autonomous haulage system", http://www.komatsu.com/ce/currenttopics/v09212/. 13

[11] Science Daily, "Self-steering automated tractor offers more precision in the field", http://www.sciencedaily.com/releases/2011/09/110919103743.htm. 14

[12] iRobot, "irobot - roomba", http://www.irobot.com/global/en/store/Roomba.aspx. 14

[13] DARPA, "Urban challenge", http://archive.darpa.mil/grandchallenge/. 14

[14] European Robotics, "Elrob the european robot trial", http://www.elrob.org/objectives.html. 14

[15] Kyeong-Hwan Lee and Reza Ehsani, "Comparison of two 2d laser scanners for sensing object distances, shapes, and surface patterns", *Computers and Electronics in Agriculture*, vol. 60, pp. 250–262, 2008. 16, 50, 65

[16] Carlos Fernandez, Vidal Moreno, and et al, "Clustering and line detection in laser range measurements", *Robotics and Autonomous Systems*, vol. 58, pp. 720–726, 2010. 16

[17] Mark Sheehan, Alastair Harrison, and Paul Newman, "Self-calibration for a 3d laser", *Int. J. Rob. Res.*, vol. 31, pp. 675 – 687, 2012. 16

[18] Sahapong Kruapech and Joewono Widjaja, "Laser range finder using gaussian beam range equation", *Optics & Laser Technology*, pp. 749 – 754, 2010. 16

[19] Zheng Liu and Dennis Krys, "The use of laser range finder on a robotic platform for pipe inspection", *Mechanical Systems and Signal Processing*, pp. 246 – 257, 2012. 16

[20] SICK AG Sensor Intelligence, "Lms 200 / lms 211 / lms 220 / lms 221 / lms 291 laser measurement system", http://www.sick-automation.ru/images/File/pdf/LMS%20Technical%20Description.pdf, Jan. 2003. 16

[21] Hokuyo Automatic Company Limited, "Scanning laser range finder urg-04lx specification", http://www.hokuyo-aut.jp/02sensor/07scanner/download/products/urg-04lx-ug01/data/URG-04LX_UG01_spec_en.pdf, Jan. 2006. 16, 50

[22] Velodyne Inc, "Velodyne lidar", http://velodynelidar.com/lidar/lidar.aspx. 16

[23] F. Moosmann and C. Stiller, "Velodyne slam", in *Intelligent Vehicles Symposium (IV), IEEE*, 2011, pp. 393 –398. 16

[24] Gangqiang Zhao, Xuhong Xiao, and Junsong Yuan, "Fusion of velodyne and camera data for scene parsing", in *Information Fusion (FUSION), Int Conf on*, 2012, pp. 1172 –1179. 16

[25] N. Wojke and M. Haselich, "Moving vehicle detection and tracking in unstructured environments", in *Robotics and Automation (ICRA), IEEE Int Conf on*, 2012, pp. 3082 –3087. 16

[26] Velodyne Inc, "Hdl-64e s2 and s2.1 users manual and programming guide", http://www.velodynelidar.com/lidar/products/manual/HDL-64E%20Manual.pdf. 16

[27] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in computational stereo", *Pattern Analysis and Machine Intelligence, IEEE Trans on*, pp. 993 – 1008, 2003. 18

[28] S. Thrun1, D. Fox, and et al, "Robust monte carlo localization for mobile robots", *Artificial Intelligence*, vol. 128, 2000. 18

[29] Kristian Ambrosch and Wilfried Kubinger, "Accurate hardware-based stereo vision", *Computer Vision and Image Understanding*, pp. 1303 – 1316, 2010. 18

[30] R.J. Valkenburg and A.M. McIvor, "Accurate 3d measurement using a structured light system", *Image and Vision Computing*, pp. 99 – 110, 1998. 18

[31] T. Dutta and G.R. Fernie, "Utilization of ultrasound sensors for anti-collision systems of powered wheelchairs", *Neural Systems and Rehabilitation Engineering, IEEE Trans on*, pp. 24 –32, 2005. 18

[32] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors", *Robotics and Automation, IEEE Journal of*, pp. 213 –218, 1988. 18

[33] United States of America Department of Defence, "Global positioning system standard positioning service performance standard", `http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf`. 19, 29

[34] Mamoun F. Abdel-Hafez, "On the gps/imu sensors noise estimation for enhanced navigation integrity", *Mathematics and Computers in Simulation*. 19

[35] M. G. Amin and Wei Sun, "A novel interference suppression scheme for global navigation satellite systems using antenna array", *IEEE J.Sel. A. Commun.*, 2006. 19

[36] Chen Shuxin, Wang Yongsheng, and Chen Fei, "A study of differential gps positioning accuracy", in *Microwave and Millimeter Wave Technology ICMMT, Int Conf on*, 2002, pp. 361 – 364. 19

[37] G.J. Morgan-Owen and G.T. Johnston, "Differential gps positioning", *Electronics Communication Engineering Journal*, pp. 11 – 21, 1995. 19

[38] M. Kara and Kun mean Hou, "Low cost differential gps receivers (lcd-gps): The differential correction function", in *New Technologies, Mobility and Security, NTMS.*, 2008, pp. 1 – 6. 19

[39] M. Hwangbo, J.-S. Kim, and T. Kanade, "Imu self-calibration using factorization", *Robotics, IEEE Transactions on*, pp. 1 – 15, 2013. 20

[40] A. Amirsadri, Jonghyuk Kim, L. Petersson, and J. Trumpf, "Practical considerations in precise calibration of a low-cost mems imu for road-mapping applications", in *American Control Conference, (ACC).*, 2012, pp. 881 –888. 20

[41] O. Shigeta, S. Kagami, and K. Hashimoto, "Identifying a moving object with an accelerometer in a camera view", in *Intelligent Robots and Systems, IEEE/RSJ Int Conf on*, 2008, pp. 3872 –3877. 20

[42] O. Gallo, R. Manduchi, and A. Rafii, "Robust curb and ramp detection for safe parking using the canesta tof camera", in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, 2008, pp. 1 – 8. 22

[43] Albert S. Huang and Seth Teller, "Lane boundary and curb estimation with lateral uncertainties", in *IROS'09*. 2009, pp. 1729–1734, IEEE Press. 22

[44] Aufrere Romuald, Mertz Christoph, and Thorpe Chuck, "Multiple sensor fusion for detecting location of curbs, walls, and barriers", in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2003)*, 2003. 22

[45] J. Siegemund, U. Franke, and W. Forstner, "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields", in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011, pp. 637 – 642. 23

[46] W.S Wijesoma, K.R.S. Kodagoda, and et al, "Road curb tracking in an urban environment", in *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, 2003, pp. 261 – 268. 23

[47] Philipp Kuettel, "Curb detection in urban outdoor environments", 2011. 23

[48] F. Oniga, S. Nedevschi, and M.M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios", in *Intelligent Transportation Systems, (ITSC). IEEE Intl. Conf. on*, 2008, pp. 67 – 72. 23

[49] Youjin Shin, Changbae Jung, and Woojin Chung, "Drivable road region detection using a single laser range finder for outdoor patrol robots", in *Intelligent Vehicles Symposium (IV), IEEE*, 2010, pp. 877 – 882. 24

[50] Wende Zhang, "Lidar-based road and road-edge detection", in *Intelligent Vehicles Symposium (IV), IEEE*, june 2010, pp. 845 –848. 24

[51] C. Unsalan and K.L. Boyer, "Linearized vegetation indices using a formal statistical framework", in *Geoscience and Remote Sensing Symposium IGARSS, IEEE Int*, 2003, pp. 982 – 984. 25

[52] J.W. Rouse and et al, *Monitoring the Vernal Advancement and Retrogradation (greenwave Effect) of Natural Vegetation*, Texas A&M University, Remote Sensing Center. 25

[53] F. J. Kriegler, W. A. Malila, and et al, "Preprocessing transformations and their effects on multispectral recognition", in *Proceedings of the Sixth International Symposium on Remote Sensing of Environment*, 1969. 25

[54] K.M. Wurm, R. Kummerle, and et al, "Improving robot navigation in structured outdoor environments by identifying vegetation from laser data", in *Intelligent Robots and Systems, (IROS). IEEE Intl. Conf. on*, 2009, pp. 1217 –1222. 25, 76

[55] David M. Bradley, Ranjith Unnikrishnan, and James Bagnell, "Vegetation detection for driving in complex environments", in *IEEE Int. Conf. on Robotics and Automation*, 2007. 25

[56] Martial Hebert and Nicolas Vandapel, "terrain classification techniques from ladar data for autonomous navigation", in *Collaborative Technology Alliances conference*, May 2003. 26

[57] Jean francois Lalonde, Nicolas V, Daniel F. Huber, and Martial Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility", *Journal of Field Robotics*, pp. 839–861, 2006. 26

[58] Jose Macedo, Roberto Manduchi, and Larry Matthies, "Ladar-based discrimination of grass from obstacles for autonomous navigation", in *Experimental Robotics VII*. 2000, pp. 111–120, Springer. 26

[59] D.F. Wolf, G.S. Sukhatme, and et al, "Autonomous terrain mapping and classification using hidden markov models", in *Robotics and Automation, (ICRA). IEEE Intl. Conf. on*, 2005, pp. 2026 – 2031. 26, 76

[60] Ali Siadat, Axel Kaske, and et al, "An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation", 1997. 27

[61] Geovany Araujo Borges and Marie-Jose Aldon, "Line extraction in 2d range images for mobile robotics", *J. Intell. Robotics Syst.*, pp. 267–297, 2004. 28

[62] Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons Inc, 1973. 28

[63] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, pp. 381–395, 1981. 28

[64] Gamini Dissanayake, Salah Sukkarieh, and et al, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications", *IEEE Transactions on Robotics*, pp. 731–747, 2001. 30

[65] L.M. Wachter and L.E. Ray, "Gps-ins state estimation for multi-robot systems with computational resource constraints", in *American Control Conference, ACC.*, 2009, pp. 561 –567. 30

[66] Kai-Wei Chiang and Yun-Wen Huang, "An intelligent navigator for seamless ins/gps integrated land vehicle navigation applications", *Applied Soft Computing*, vol. 8, 2008. 30

[67] Shesheng Gao, Yongmin Zhong, Xueyuan Zhang, and Bijan Shirinzadeh, "Multi-sensor optimal data fusion for ins/gps/sar integrated navigation system", *Aerospace Science and Technology*, vol. 13, 2009. 30

[68] Aboelmagd Noureldin, Ahmed El-Shafie, and Mohamed Bayoumi, "Gps/ins integration utilizing dynamic neural networks for vehicular navigation", *Information Fusion*, vol. 12, 2011. 30

[69] David Schleicher, Luis M. Bergasa, Manuel Ocaa, Rafael Barea, and Elena Lpez, "Low-cost gps sensor improvement using stereovision fusion", *Signal Processing*, vol. 90, 2010. 31

[70] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Simultaneous localization and map building: a global topological model with local metric maps", in *Intelligent Robots and Systems, IEEE/RSJ Int Conf on*, 2001, pp. 421 – 426. 31

[71] Ming Liu and R. Siegwart, "Dp-fact: Towards topological mapping and scene recognition with color for omnidirectional camera", in *Robotics and Automation (ICRA), IEEE Int Conf on*, 2012, pp. 3503 – 3508. 31

[72] Ming Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs", in *Robotics and Automation (ICRA), IEEE Int Conf on*, 2011, pp. 3269 – 3274. 31

[73] S. Simhon and G. Dudek, "A global topological map formed by local metric maps", in *Intelligent Robots and Systems, IEEE/RSJ Int Conf on*, 1998, pp. 1708 – 1714. 31

[74] Feng Lu and Evangelos Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans", *Journal of Intelligent Robotic Systems*, pp. 249–275, 1997. 32

[75] F. Amigoni, S. Gasparini, and M. Gini, "Building segment-based maps without pose information", *Proceedings of the IEEE*, vol. 94, pp. 1340 – 1359, 2006. 32

[76] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration", in *Robotics and Automation ICRA, IEEE Int Conf on*, 2000. 32

[77] Sebastian Thrun, Wolfram Burgard, Dieter Fox, Henry Hexmoor, and Maja Mataric, "A probabilistic approach to concurrent mapping and localization for mobile robots", in *Machine Learning*, 1998, pp. 29 – 53. 32

[78] U. Frese and G. Hirzinger, "Simultaneous localization and mapping - a discussion", 2001. 32

[79] J.E. Guivant and E.M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation", *Robotics and Automation, IEEE Trans on*, pp. 242 –257, 2001. 33

[80] Juan Nieto, Tim Bailey, and Eduardo Nebot, "Recursive scan-matching slam", *Robotics and Autonomous Systems*, vol. 55, pp. 39–49, 2007. 33

[81] Christopher Mei, Gabe Sibley, and et al, "Rslam: A system for large-scale mapping in constant-time using stereo.", *International Journal of Computer Vision*, pp. 198–214, 2011. 33

[82] R. Brooks, "A robust layered control system for a mobile robot", *IEEE Journal on Robotics and Automation*, pp. 14 – 23, 1986. 34

[83] D. Payton, "An architecture for reflexive autonomous vehicle control", in *Robotics and Automation. Proceedings. IEEE Int Conf on*, 1986, pp. 1838 – 1845. 35

[84] Hans Moravec, M. Kadonoff, and et al, "Arbitration of multiple control strategies for mobile robots", in *SPIE Proceedings: Advances in Intelligent Robotics Systems*, 1986, vol. 727. 35

[85] R. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior", in *Robotics and Automation. Proceedings. IEEE Int Conf on*, 1987, vol. 4, pp. 264 – 271. 35

[86] Nils J. Nilsson, "Shakey the robot", Tech. Rep., AI Center, SRI International, Apr 1984. 36

[87] H.P. Moravec, "The stanford cart and the cmu rover", *Proceedings of the IEEE*, pp. 872 – 884, 1983. 36

[88] Ronald C. Arkin, "Towards the unification of navigational planning and reactive control", in *AAAI*, 1989. 37

[89] J.H. Connell, "Sss: a hybrid architecture applied to robot navigation", in *Robotics and Automation, IEEE Intl. Conf. on*, 1992. 37

[90] Pride Mobility Products Corporation, "Jazzy power chairs - http://www.pridemobility.com/jazzy/index.asp", `http://www.pridemobility.com/jazzy/index.asp`, Jan. 2013. 42

[91] Hokuyo Automatic Company Limited, "Communication protocol specification for scip 2.0 standard", `http://www.hokuyo-aut.jp/02sensor/07scanner/download/data/URG_SCIP20.pdf`, Jan. 2006. 49

[92] Coppelia Robotics, "V-rep virtual robot experimentation platform", http://www.v-rep.eu/v-repSpecifications.pdf, Jan. 2013. 53

[93] Transport for London, "Pedestrian comfort level guidance", http://www.tfl.gov.uk/assets/downloads/pedestrian-comfort-guidance-technical-guide.pdf, 2010. 62