

Summer 6-2015

Sensor Fusion and Obstacle Avoidance for an Unmanned Ground Vehicle

Gurasis Singh
Embry-Riddle Aeronautical University

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Aerospace Engineering Commons](#)

Scholarly Commons Citation

Singh, Gurasis, "Sensor Fusion and Obstacle Avoidance for an Unmanned Ground Vehicle" (2015).
Doctoral Dissertations and Master's Theses. 248.
<https://commons.erau.edu/edt/248>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Doctoral Dissertations and Master's Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

SENSOR FUSION AND OBSTACLE AVOIDANCE FOR AN UNMANNED
GROUND VEHICLE

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

Gurasis Singh

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

June 2015

Embry-Riddle Aeronautical University

Daytona Beach, Florida

SENSOR FUSION AND OBSTACLE AVOIDANCE FOR AN UNMANNED
GROUND VEHICLE

by

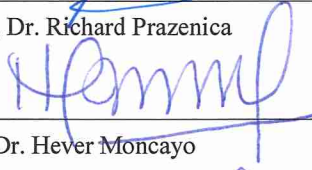
Gurasis Singh

A Thesis prepared under the direction of the candidate's committee chairman, Dr. Richard Prazenica, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

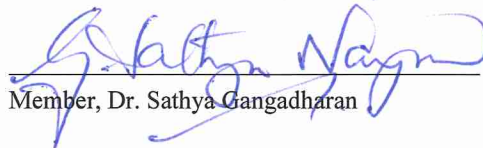
THESIS COMMITTEE



Chairman, Dr. Richard Prazenica



Member, Dr. Hever Moncayo



Member, Dr. Sathya Gangadharan



Department Chair, Dr. Anastasios Lyrintzis
or Graduate Program Coordinator, Dr. Yi Zhao

6/29/2015

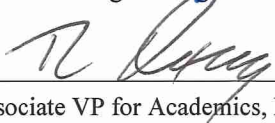
Date



Dean of College of Engineering, Dr. Maj Mirmirani

6/29/2015

Date



Associate VP for Academics, Dr. Robert Oxley

6/30/2015

Date

Acknowledgements

It provides for me colossal joy to recognize the individuals who gave me the vital directions, help and ceaseless inspiration towards the fulfillment of this preposition.

I would like to augment my appreciation and exceptionally recognize my advisor, Dr. Richard J. Prazenica for his specialized direction, supervision and consistent good backing amid the course of this postulation. He empowered me to strike the right harmony in the middle of classes and theory work, investigate the flexible character implanted in me and yield a quality output at all times.

I would like to thank Dr. Hever Moncayo for serving on the panel and for furnishing me with quality specialized information throughout the span of this postulation.

I owe my deepest gratitude to Dr. Sathya Gangadharan for serving on the panel and for giving fundamental direction and backing in the midst of his occupied calendar.

I would like to share this achievement with my mother Manjit Kaur, my father Jaswinder Singh and my sister Manmeet and thank them for their love, motivation, support and faith in me.

I would like to express deep appreciation towards my friends Navroz Lamba and Abhinandan for their support and encouragement throughout my thesis work.

Finally, I would like to share this feat with my Uncle Dr. Bp Sodhi and my aunt Sharon Sodhi and express my sincere appreciation for their never-ending love, continual motivation and strong support extended at all times.

ABSTRACT

Author: Gurasis Singh

Title: Sensor Fusion and Obstacle Avoidance for an Unmanned Ground Vehicle

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Aerospace Engineering

Year: 2015

In recent years, the capabilities and potential value of unmanned autonomous systems (UAS) to perform an extensive variety of missions have significantly increased. It is well comprehended that there are various challenges associated with the realization of autonomous operations in complex urban environments. These difficulties include the requirement for precision guidance and control in conceivably GPS-denied conditions as well as the need to sense and avoid stationary and moving obstructions within the scene. The small size of some of these vehicles restricts the size, weight and power consumption of the sensor payload and onboard computational processing that can be accommodated by UAS.

This thesis analyzes the development and implementation of terrain mapping, path planning and control algorithms on an unmanned ground vehicle. Data from GPS, IMU and LIDAR sensors are fused in order to compute and update a dense 3D point cloud that is used by an implicit terrain algorithm to provide detailed mathematical representations of

complex 3D structures generally found in urban environments. A receding horizon path planning algorithm is employed to adaptively produce a kinematically-feasible path for the unmanned ground vehicle. This path planning algorithm incorporates obstacle avoidance constraints and provides a set of waypoints to be followed by the unmanned ground vehicle. A waypoint controller is designed and implemented to enable the vehicle to follow the waypoints from the path planner. Open-loop experiments are provided with an unmanned ground vehicle in order to demonstrate terrain generation with real sensor data. Closed-loop results are then presented for a simulated ground vehicle in order to demonstrate the performance of the receding horizon path planning and control algorithms using the terrain map generated from the open-loop experiments.

Contents

1) Introduction.....	12
1.1 Terminology.....	12
1.2 Motivation.....	12
1.3 Literature Review.....	13
1.4 Existing Autonomous Vehicles.....	16
1.4.1 Military Applications.....	16
1.4.2 Consumer Applications.....	17
1.5 Technical objectives.....	18
1.6 Organization of thesis.....	21
2) Unmanned Ground Vehicle and Sensors.....	22
2.1 Unmanned Ground Vehicle.....	22
2.2GPS receiver.....	25
2.3IMU.....	27
2.4 LIDAR.....	28

3) Data Processing and Sensor fusion.....	30
3.1 Reference Frames.....	30
3.2 Coordinate Transformation.....	32
3.3 Position Calculations.....	35
3.3.1 Computing the initial GPS location in ECEF frame.....	37
3.3.2 Local NED frame coordinate transformation.....	37
3.3.3 Locating the origin of Local NED frame.....	39
3.3.4 Expressing the GPS Position in local NED frame.....	40
3.4 INS Orientation Calculations.....	40
3.5 LIDAR Point Cloud Calculations.....	41
4) Terrain Mapping and Path Planning.....	43
4.1 3D Implicit terrain representation from point cloud.....	43
4.2 Receding Horizon Path Planning Algorithm.....	47
5) Modeling and Control of Unmanned Ground Vehicles.....	54
5.1 UGV Kinematics.....	54
5.2 Controller.....	56

6) Experimental and Simulation Results.....	62
6.1 Open-Loop Experiment Measurements 1.....	62
6.2 Open-Loop Experiment Measurements 2.....	73
6.3 Closed-Loop Simulation Results.....	82
7) Conclusions.....	97
7.1 Concluding Remarks.....	94
7.2 Future Work.....	94
Bibliography.....	96

List of Figures

1.1 Overview of an autonomous System for UGV	20
2.1a Corobot Unmanned Ground Vehicle	24
2.1b Left Side Power Selector Switch and power connectors.....	24
2.1c Ride Side Power and Reset Buttons.....	24
2.1d Rear view of the Corobot.....	25
2.2 Global Sat BU-353GPS receiver.....	26
2.3 3D GX3-45 GPS Aided Inertial Navigation System.....	28
2.4 Hokuyo UTM-30 LX LIDAR.....	29
3.1 Different Frames of Reference.....	31
3.2 Rotation about axis by yaw angle.....	32
3.3 Rotation about axis by Pitch angle.....	33
3.4 Rotation about axis by roll angle.....	33
3.5a Earth Center Earth Fixed (with origin at center of Earth).....	35
3.5b the longitude from overhead view of Earth.....	36
3.5c The Geodetic Latitude and altitude from side of earth.....	36
3.6 Rotation about e axis by individual longitude.....	38
3.7 LIDAR frame of reference with LIDAR at origin.....	45

4.1 Effect of Varying the isolevel Parameters.....	47
4.2 Candidate Path Points for two dimensional path planning.....	49
4.3 Path Planning Scenario (Goal location B).....	51
5.1 Differential Drive Kinematics.....	54
5.2 Feedback control law of Ground Robot.....	57
5.3 Waypoint Controller.....	58
5.4 Position of Robot at different time intervals.....	61
6.1.1 View of overall path from Camera on UGV.....	63
6.1.2 View of the overall path taken at ERAU campus (shown in red).....	64
6.1.3 Zoomed in view of figure 5.1.2.....	64
6.1.4 Ground Vehicle Track with sample images from camera on UGV.....	65
6.1.5 LIDAR measurements with vehicle path (Shown in Red).....	66
6.1.6 GPS measurement.....	66
6.1.7 NED Position of the vehicle CM.....	67
6.1.8 Orientation Measurements.....	67
6.1.9 3D representations at different time intervals.....	69
6.1.10 Complete 3D terrain representation with Vehicle path.....	69
6.1.11 Front view of complete 3D terrain Map.....	70

6.1.12 Effect of Varying Isolevel Parameter.....	72
6.2.1 View of overall Path.....	73
6.2.2 View of Overall Path from google earth.....	74
6.2.3 Ground Vehicle Track with Sample Images from camera mounted mounted.....	74
6.2.4 LIDAR measurements with path shown in red.....	75
6.2.5 NED position of the vehicle CM.....	75
6.2.6 GPS measurements from Global Sat BU-353.....	76
6.2.7 Orientation measurements from Micro strain.....	76
6.2.8 3D terrain representation (Vehicle path shown in red).....	79
6.2.9 Overhead view of 3D terrain representation with path (shown in red).....	79
6.2.10 Oblique view of 3D terrain representation.....	80
6.2.11 3D view of terrain representation by 3D implicit terrain algorithm.....	80
6.2.12 Alternative view of the full 3D terrain representation.....	81
6.3.1 Overall Path Scenario with Target location.....	81
6.3.2 Receding Horizon Decisions at several Points.....	89
6.3.3 Simulate movement of the UGV through the at various times.....	93

CHAPTER 1 Introduction

1.1 Terminology

The meaning of autonomous is somewhat unclear and open to much debate. This work will describe an autonomous vehicle as a mobile robot that can explore and operate within an urban environment with minimal human cooperation. Terms such as robot and intelligence have a wide range of implications. These terms will be characterized in order to define the autonomous vehicle examined in this work. The term robot evokes various images. The movie industry has made robots popular and depicts the concept of robots in numerous motion pictures such as Iron man, Dhoom2 and Transformers. However these mythical cases are not an accurate representation of the robots that presently exist and work today.

For this research, the term robot will depict an unmanned ground vehicle (UGV). The autonomous UGV gathers information from onboard sensors, processes these data and then reacts to this information. To an outside observer, the robot would appear to be autonomously making decisions focused around the circumstances it experiences in its surroundings.

1.2 Motivation

Urban pursuit and related missions impose unique requirements on autonomous systems frameworks. Since an outside communication link is not always available or practical for providing control commands to the vehicle, unmanned vehicles must have the capacity to operate autonomously. In particular, the constrained payload available to small aerial systems poses a great challenge to the system design. A tradeoff between flight

execution, sensors and processing assets must be found. Since communication with these unmanned system cannot be assured at all times, sensor process and decision making must be done onboard for successful operation in obstacle strewn environments such as hilly terrain, woodlands or urban areas. These capabilities require the incorporation of sensing and control methodologies for obstacle avoidance, path planning and control that consider the complexities of the whole UAV system.

1.3 Literature Review

Over the previous decades, UAVs have been progressively utilized for an extensive array of applications [1]; for example observation, reconnaissance, surveying and mapping, spatial data securing, and geophysics investigations. Therefore, the navigation of unmanned aerial vehicles is of considerable interest. The limited payload of small aerial system poses a great challenge to the system design. The Global Positioning System (GPS) is an essential navigational sensor modality utilized for the vehicle direction and guidance [7, 8]. Nonetheless, a comprehensive study alluded to as the Volpe report [9] discusses vulnerabilities of GPS related with signal interruptions. These interruptions can be caused by RF interference, ionosphere interference, jamming or spoofing. This report inspired many strategies to mitigate the vulnerabilities of the current GPS navigation convention.

Another widely used navigation sensor used for calculating the position of the vehicle [10-13] is the IMU (inertial measurement unit). It is composed of accelerometers, magnetometers and angular rate sensors. Accelerometers measure the sum of linear acceleration, typically in three orthogonal axes, due to the vehicle motion and the acceleration due to gravity. The direction of the local magnetic field is measured by magnetometers. A measure of the three dimensional rate of rotation of the body (i.e.,

angular velocity) is provided by angular rate sensors. An orientation estimate is typically computed using the direction of the earth's gravitational and magnetic vector along with the angular rates, but there are drawbacks associated with using an IMU alone for navigation. First, there is quadratic growth in position drift error due to the double integration of acceleration data containing bias and random noise. Similarly, the integration of angular rate data to estimate orientation is also subject to the accumulation of drift error. To mitigate these errors, IMU are commonly fused with GPS in a navigation filter, providing improved accuracy. An inertial navigation system (INS) with GPS depends on the quality of the GPS signal but the GPS signal can be vulnerable to interference and spoofing as discussed earlier. Therefore, alternative navigational sensors are of interest for GPS-denied navigation. In addition, alternative sensing modes are required in order to sense and avoid obstacles within complex operational environments.

Recently, airborne and space-based laser altimetry has developed as a promising strategy to capture precise digital elevation data with LIDAR sensors. An ever increasing range of applications has taken advantage of the high accuracy and dense sampling provided by LIDAR sensors, which have the advantages of low computational load and high processing rates [22-25]. Moreover, LIDAR is not constrained by lighting conditions. These sensors, therefore, have been employed to form dense point clouds for different applications [28, 29, and 30]. A 3D point cloud can be used by different methodologies to map the environment and to provide obstacle avoidance constraints. Some collision avoidance methods for UAVs are based on potential field methods that model obstacles as repellants and waypoints as attractors [32, 33]. Considerable research has been done on the problem of obstacle avoidance for mobile ground robots [34, 35]. For collision free

robot guidance, two safety properties: passive safety (no collision can happen when the robot moves) and strong passive safety (the robot further maintains sufficient maneuvering distance from obstacles to avoid collisions) were studied [35]. In another work, the point cloud from LIDAR sensors was used to generate 3D mathematical representations that basically provide the information about the obstacles in the environment [37].

Many algorithms have been tested for path planning of autonomous vehicles. These are categorized as path planning and trajectory planning algorithms. Path planning entails generating an obstacle free path to avoid collision based on set of criteria [38, 39]. Trajectory planning plans the movement of the robot along the planned path. One path planning approach that is receiving considerable attention in the literature is receding horizon control. The receding horizon methodology has the ability to incorporate obstacle avoidance constraints as well as input and state constraints. The receding horizon algorithm solves an optimal control problem over a finite time horizon, after which a new optimal control problem is solved. This process continues until the objective is achieved. Therefore, the receding horizon algorithm provides the ability to incorporate new data into the planning and control of the vehicle. The stability of receding horizon control methods requires careful consideration compared with conventional (infinite horizon) optimal control approaches, but is now a well understood issue [40].

There are many examples in which receding horizon methods were employed to plan and control the trajectory of the autonomous vehicles [41, 42]. An important characteristic feature of the receding horizon method is that it involves re-planning after short time intervals, making it possible to consider new information frequently. In another work, this approach was employed to plan the movement of an autonomous UAV along a

2D straight line path avoiding obstacles by re-planning every time it received new data [43]. In the majority of the above studies, the obstacles are assumed to be known a priori or estimated online by an obstacle detection algorithm.

1.4 Existing Autonomous vehicles

The two main applications of UAVs or UGVs are military use and civilian use.

1.4.1 Military Applications

In the previous decade, because of world conflicts and the quick progression of innovation, there has been an extraordinary military interest in self-governing or autonomous vehicles. In the future, the U.S. military would like to have a large portion of its battle force provided by unmanned independent vehicles. Autonomous vehicles are a favored weapon system because of their productivity, information accumulation capacities and assurance of human life. UAVs have the potential to perform intelligence, surveillance and reconnaissance (ISR) missions. Their application has extended to electronic assault, strike mission, concealment and demolition of enemy air resistance (SEAD/DEAD), battle pursuit and salvage (CSAR).

Two well-known unmanned aerial vehicles are the U.S. Air Force's Predator and Global Hawk. The Predator is intended to provide steady discernment, reconnaissance and surveillance. This UAV is controlled from a ground control station (GCS), through a satellite connection, from it is given missions by a human operator. While the Predator has the ability to perform battle missions, a human operator makes an official determination to fire on a target. The Predator's capacity to assault ground targets and protect troops from danger has demonstrated its value to the U.S. military [3]. Another example, the Global Hawk, is solely an observation and surveillance UAV that was assembled for high altitude,

long endurance missions. Furnished with complex radar and sensing systems, it can supply responsive and persistent information from many places inside enemy territory, day or night, with little concern for climate [4].

Autonomous vehicles are likewise being utilized at sea by the U.S. Navy and on the ground by the U.S. Army. Providing a ground vehicle with the ability to adjust to its surrounding and maintain strategic distance from obstructions is a challenging task.

1.4.2 Consumer Applications

As innovation advances and becomes less expensive, the number of applications of autonomous vehicles will increase. As a rule with innovation, what was at one time a top of the line, top secret military application eventually enters the commercial market in a totally distinctive capacity. As an example, the global positioning system (GPS) was created in the 1970's entirely for the military but today is broadly utilized for automobile navigation, mail and bundle following and many other commercial applications. A similar trend is occurring with autonomous vehicles. For example, Roomba is an autonomous vacuum cleaner designed and made by iRobot, which costs around \$250. iRobot claims Roomba can keep a house vacuumed with next to no human intervention. Another example of a civilian autonomous robot is Robomower. Friendly Robotics asserts that a customer can schedule the Robomower to cut their yard when desired [6]. These products seem at face value to be extraordinary answers for unremarkable assignments but there are issues with both of them. Lamentably, both of these items fail to meet expectations. They are not particularly clever machines. For example, both these robots explore their surroundings in an irregular manner. They make headway until they experience an obstruction, then they make a turn arbitrarily and proceed until another obstruction is

encountered. In a basic domain, for example a rectangular room or a round yard with few obstructions, this strategy can be effective. In complex environments, however, these systems end up being wasteful and ineffective. Surveys of these robots assert that when they were utilized in complex world settings, they would end up not vacuuming the entire room or leaving portions of grass unmowed. Clearly a superior guidance approach must be concocted for these scenarios.

1.5 Technical Objectives

This thesis emphasizes the system integration of algorithms for sensor fusion, mapping, path planning and controls for an unmanned ground vehicle (UGV) operating in a complex urban environment, with future extensions to an unmanned air vehicle (UAV). A key objective of the thesis is to provide a systems-level investigation of the different component algorithms required in the closed-loop system. These algorithms include sensor fusion to generate navigation and point cloud data, 3D implicit terrain representations of the scene, receding horizon path planning with obstacle avoidance constraints and a waypoint controller to enable the vehicle to follow the planned path. The following objectives are achieved in this thesis:

- 1) Generate detailed navigation data from the unmanned ground vehicle using IMU and GPS sensors.
- 2) Create a 3D point cloud from processed LIDAR data and generate a 3D representation of the environment using 3D implicit terrain algorithms.
- 3) Implement a receding horizon algorithm to adaptively plan a path through the scene while enforcing obstacle avoidance constraints provided by the 3D terrain map.

- 4) Develop and implement a UGV controller to enable the UGV to follow the planned waypoints.
- 5) Perform open-loop experiments with data collected from a UGV operating in a complex environment as well as closed-loop simulations of a UGV model navigating in a complex scene.

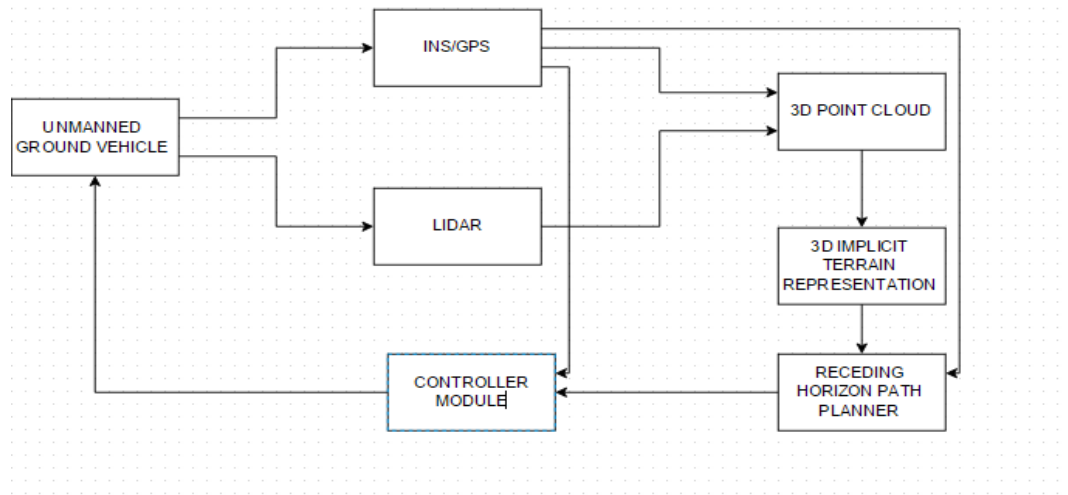


Figure 1.1: - Overview of an Autonomous System for UGV

The vehicle would commonly be instrumented with one or more traditional navigation sensors such as inertial measurement unit (IMU) (i.e. accelerometers and rate gyros) and GPS unit. A typical UAS payload might also incorporate a laser rangefinder or a line-scan LIDAR unit. Because the vehicle has an onboard LIDAR or rangefinder, a point cloud can be produced from LIDAR measurements and an estimate of the inertial position and orientation of the vehicle. The 3D point cloud, which is computed in an inertial reference frame, is utilized to produce a detailed 3D terrain map, which gives obstacle avoidance constraints to receding horizon path planner. The receding horizon module computes the path points that minimize a cost functional over a finite time horizon

while incorporating obstacle avoidance constraints. The control is connected over a subset of this horizon after which another optimization problem is performed to generate the next set of path points for the vehicle. Along these lines, the receding horizon methodology can incorporate new data from the updated obstacle map in the next path planning iteration. A controller is designed to drive the robot to the points planned by the receding horizon algorithm to reach its target point. The controller receives new waypoints at every time interval until the goal is achieved.

1.6 Organization of this Thesis

This thesis is organized as follows:

- Chapter 2 discusses the unmanned ground vehicle (UGV) and sensors used in this research.
- Chapter 3 provides details about the different reference frames and coordinates transformation required for navigation and the sensor fusion algorithms used to generate a LIDAR based point cloud.

Chapter 4 discusses the implicit terrain algorithms used to represent 3D urban terrain features based on LIDAR measurements and also provides the formulation of the receding horizon path planning algorithm with obstacle avoidance constraints.

- Chapter 5 introduces a simulation model of the ground robot and develops a waypoint controller in order to navigate the vehicle to follow the planned path.
- Chapter 6 provides open-loop experimental results in which these algorithms have been applied to data gathered from the UGV sensors as it traveled through

a complex scene. Closed-loop simulations are then provided and the performance of the algorithms is discussed

- Chapter 7 provides conclusions and future recommendations.

CHAPTER-2

Unmanned Ground Vehicle and Sensors

This chapter provides a detailed description of the unmanned ground vehicle and onboard sensor suite that was employed in this research. The sensor suite includes a MicroStrain 3DM-GX3-45 inertial navigation system (INS), a Hokuyo UTM-30LX line-scan LIDAR unit, and a Global sat BU-353 GPS receiver. Data from these sensors were fused to estimate the inertial position and orientation of the vehicle and to compute a 3D point cloud from which a terrain map of the environment was adaptively generated.

2.1 Unmanned Ground Vehicle

The unmanned ground vehicle is a completely robotized vehicle that can travel on a specific predefined course with minimal human intercession. The ground vehicle used in this work is a Corobot four wheeled robot that has an onboard computer. Figure 2.1 shows several views of the Corobot unmanned ground vehicle.



Figure 2.1a Corobot Unmanned Ground Vehicle

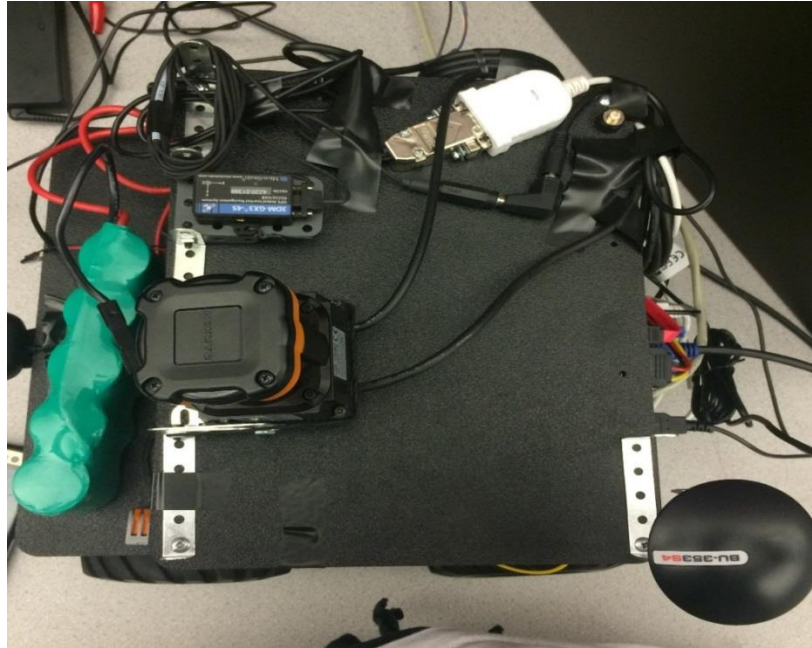


Figure 2.1b: Top view of the Corobot with sensors

The Corobot is instrumented with sensors, which are integrated with its hardware in order to record and process the sensor measurements. The ultimate goal is to enable the Corobot to operate autonomously in urban environments using the current position and desired waypoints generated by the receding horizon path planning algorithm. A controller (discussed later) is designed to generate commands to the motors in order to move the robot towards the next desired waypoint. The Corobot is Linux and Windows compatible, and Linux is required to be set up for the initial use. The sensors are integrated with the Corobot using Robot Operating System (ROS). The Corobot is intended for simple access, dismantling and reassembly for including and changing of equipment and parts. Because of its open structure, the Corobot is sensitive to debris. Operation under outdoor lighting conditions should represent no issue to the Corobot; however water will cause damage because the unit is not waterproof. The Corobot can connect to a local wireless network, which enables communication with the robot from a networked desktop, laptop or,

depending on network connections, from anywhere on the Internet. To drive the Corobot, the Corobot control panel is interfaced with a keyboard or joystick. Figure 2.1b depicts an overhead view of the Corobot with the integrated sensor suite, and Figures 2.1c through 2.1e provide several other views of the vehicle.

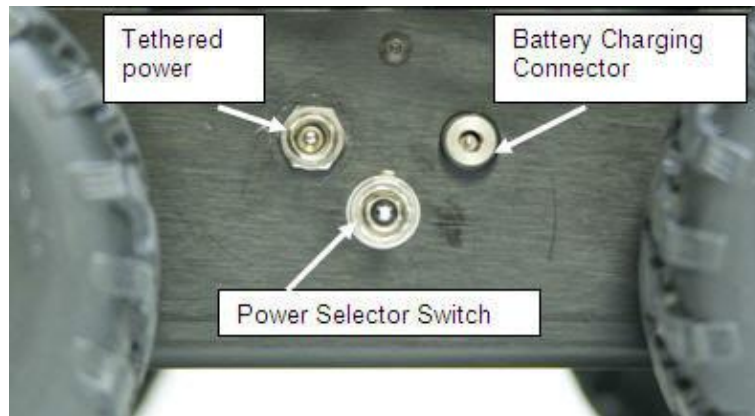


Figure 2.1c: Left side power selector switch and power connectors

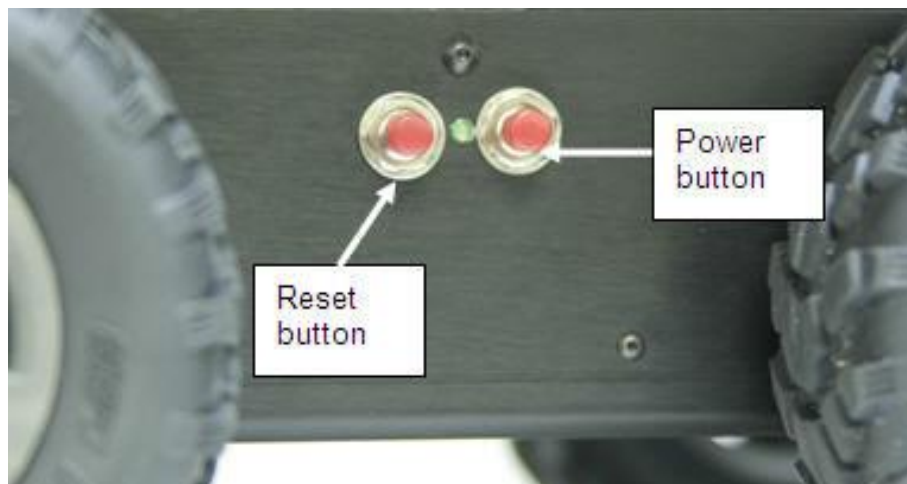


Figure 2.1d: Right side power and reset buttons



Figure 2.1e: Rear view of the Corobot

2.2 GPS Receiver

The global positioning system (GPS) is a space based global navigation satellite system that provides reliable positioning and time information in all weather and anywhere on or near the earth. GPS was established in 1973 by the U.S. Department of Defense. The GPS consists of a constellation of 27 earth orbiting satellites (24 in operation and three extra in case one fails). A GPS receiver locates four or more of these satellites, computes the distance to each satellite, and uses this information to deduce its location. The advantages of using GPS for navigation include its accuracy in determining position and the fact that GPS receivers are lightweight and inexpensive. A shortcoming of GPS is that the signal is often unavailable in urban and mountain areas, as well as indoor and underground environments. GPS signals are also vulnerable to interference and spoofing. They have a low update frequency, which is normally 1-10 Hz, which is 1-2 orders of magnitude lower than that of a standard INS.



Figure 2.2 GlobalSat BU-353 GPS receiver

The GPS receiver used for this work is the GlobalSat BU-353. This GPS receiver has the following features:

- 1) Its bottom is non slippery and waterproof
- 2) It has the capability to predict satellite position for up to 3 days in advance
- 3) Low power consumption
- 4) It has a built in patch antenna

Each row of the GPS data file corresponds to a position measurement. The data file has 4 columns corresponding to [Time stamp (ms), Geodetic Latitude λ_D (deg), Longitude φ_D (deg), Altitude h (m)]. Note that the GPS measurement provides position relative to the WGS-84 ellipsoid. Therefore, the altitude is approximately equivalent to height above sea level.

2.3 Inertial Navigation System

An inertial navigation system (INS) is an electronic device that uses a combination of accelerometers, gyroscopes, GPS, and magnetometers to measure the vehicle's position, velocity, orientation and angular velocity. The INS contains an inertial measurement unit (IMU), which measures acceleration utilizing one or more accelerometers and measures

rotation using one or more rate gyroscopes. Furthermore, many INS incorporate magnetometer data in a navigation filter in order to reduce drift in the orientation (i.e., roll, pitch, and yaw) angles that are calculated from integrating noisy rate gyro data. The position of the vehicle can be calculated directly from IMU data in a process known as dead reckoning, but typically GPS data are incorporated into a navigation filter to reduce drift in the position estimates. An advantage of IMU sensors is that their data sampling rate is high (on the order of 50 Hz) compared to that of GPS (on the order of 1-10 Hz).

For this thesis research, a Microstrain 3DM-GX3-45 INS sensor, as shown in Figure 2.3, was used. This INS consists of a triaxial accelerometer, a triaxial gyro, a triaxial magnetometer and a temperature sensor. It additionally has an installed processor that runs a navigation filter to provide static and dynamic orientation and inertial position estimates.



Figure 2.3 Microstrain 3DM-GX3-45 GPS Aided Inertial Navigation System

The 3D-GX3-45 provides a range of navigation related output quantities. Fully calibrated inertial measurements include acceleration, angular rates, magnetic field, delta theta and delta velocity vectors, Euler angles (pitch, roll and heading), rotation matrix and quaternion.

2.4 LIDAR

LIDAR (light detection and ranging) is a remote sensing innovation that measures 3D point clouds of the earth's surface. Some airborne LIDAR instrumentation utilize a laser scanner with up to 400,000 pulses of light per second. At the point when an airborne laser is pointed at a focused region on the ground, the light emission is reflected by the surface. A sensor records this reflected light to measure the range. Laser range measurements are fused with vehicle position and orientation information in order to generate a dense, detail-rich collection of 3D elevation points known as a "point cloud."

The LIDAR used in this work was the Hokuyo UTM-30 LX line-scan LIDAR, pictured in Figure 2.4. This line-scan sensor rotates the laser about a single axis in order to produce planar scans of the surrounding environment.



Figure 2.4 Hokuyo UTM-30LX Line-Scan LIDAR

Features of the Hokuyo UTM-30LX LIDAR include

- 1) Minimum range of 10 mm
- 2) Maximum range of 60 m

- 3) Scan angle of 270 degrees
- 4) Scan rate of 25 ms (40 scans/sec)
- 5) Mass of 210g
- 6) Angular resolution of 0.25 degrees

Chapter -3

Data Processing and Sensor Fusion

This chapter discusses the algorithms required for processing the GPS, INS, and LIDAR data in order to compute the position and orientation of the vehicle as well as generating a 3D point cloud. In order to construct a terrain map, the LIDAR measurements must be expressed in a common inertial reference frame. The computation of this LIDAR-based point cloud entails transforming the LIDAR measurements, which are collected with respect to the LIDAR sensor, to the inertial frame using the estimated position and orientation of the ground vehicle. The reference frames and coordinate transformations required for this analysis are also provided in this chapter.

3.1 Reference Frames

Several different reference frames are utilized in the sensor fusion algorithms discussed in this chapter. These reference frames, illustrated in Figure 3.1, include a local geodetic (North-East-Down) frame \mathbf{G} , which for the purposes of this ground robot application serves as an inertial reference frame, a body reference frame \mathbf{B} that is fixed within the vehicle (i.e., the coordinate axes move with the vehicle), and a sensor frame \mathbf{S} that is fixed within each sensor.

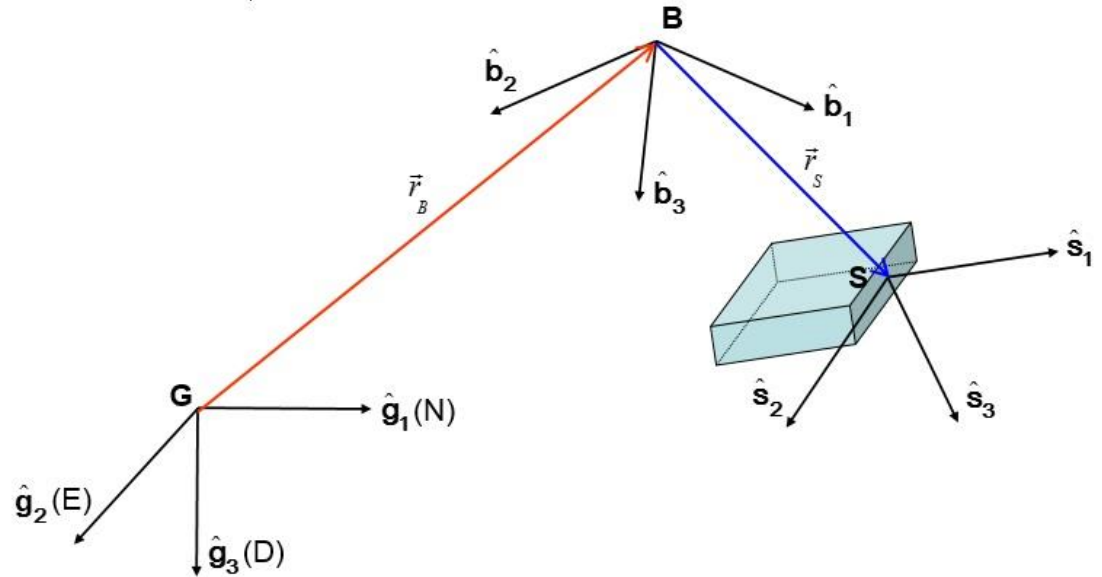


Figure 3.1 Geodetic (NED), Body, and Sensor Reference Frames

In Figure 3.1, \vec{r}_b represents the position of the vehicle center of mass (CM) relative to the origin of the inertial frame and \vec{r}_s denotes the position of the sensor relative to the vehicle CM. The coordinate axes of the geodetic, body, and sensor frames are defined as follows.

Local Geodetic (Inertial) Frame G

- \hat{g}_1 corresponds to local North
- \hat{g}_2 corresponds to local East
- \hat{g}_3 corresponds to local Down

Body Fixed Frame B

- Origin is located at the vehicle CM
- \hat{b}_1 is pointed out the front of the vehicle
- \hat{b}_2 is pointed out the right side of the vehicle
- \hat{b}_3 is pointed downward, completing the orthogonal triad

Sensor Fixed Frame S

- Origin is collocated with the LIDAR sensor
- \hat{s}_1 is pointed out the front of the sensor
- \hat{s}_2 is pointed out the right side of the sensor
- \hat{s}_3 is pointed downward, completing the orthogonal triad

3.2 Coordinate Transformations

Given the reference frames defined in the previous section, it is necessary to define the coordinate transformations required to express data in these different reference frames. These coordinate transformations take the form of rotation matrices that transform data representations from one reference frame to another.

Given the measured roll φ , pitch θ , and yaw ψ of the vehicle, the rotation matrix R_G^B defining the transformation from the geodetic (inertial) reference frame to the body-fixed reference frame is defined in terms of 3 single-axis rotations. The first of these is a rotation about the \hat{g}_3 axis by the yaw angle, as shown in Figure 3.2. This rotation uses the usual convention that yaw is measured clockwise from the North.

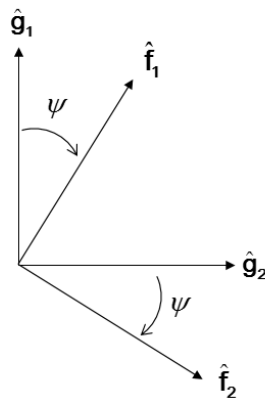


Figure 3.2 Rotation about the \hat{g}_3 (local down) axis by the yaw angle

The transformation matrix defining this rotation is given by

$$\begin{Bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \end{Bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{g}_1 \\ \hat{g}_2 \\ \hat{g}_3 \end{Bmatrix}$$

The next rotation is about the \hat{f}_2 axis by the pitch angle, as shown in Figure 3.3. This rotation uses the convention that an upward pitch angle is considered positive.

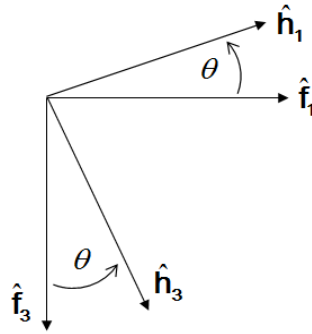


Figure 3.3 Rotation about the \hat{f}_1 axis by the pitch angle Θ

The transformation matrix defining this rotation is given by

$$\begin{Bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \end{Bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{Bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \end{Bmatrix}$$

The final rotation is a rotation about the \hat{f}_2 axis by the roll angle, as shown in Figure 3.4.

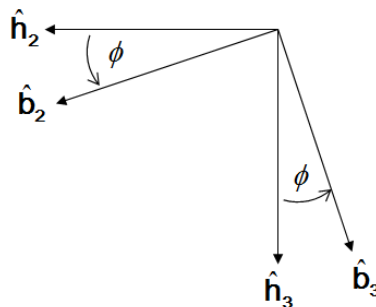


Figure 3.4 Rotation about the \hat{h}_1 axis by the roll angle ϕ

The transformation matrix defining this rotation is given by

$$\begin{Bmatrix} \hat{\mathbf{b}}_1 \\ \hat{\mathbf{b}}_2 \\ \hat{\mathbf{b}}_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{h}}_1 \\ \hat{\mathbf{h}}_2 \\ \hat{\mathbf{h}}_3 \end{Bmatrix}$$

The transformation matrix R_G^B from the local geodetic (inertial) frame to the body-fixed frame is then composed from the three single-axis rotations as follows:

$$R_G^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The inverse rotation matrix (i.e., rotation from the body fixed frame to the inertial reference frame) is defined as R_B^G , which is given by $R_B^G = (R_G^B)^T$.

Similarly, given the roll, pitch and yaw (ϕ_s, θ_s, ψ_s) of the sensor frame relative to body fixed frame, which are functions of how the sensors are mounted on the vehicle, the rotation matrix defining coordinate transformation from the body fixed frame to the sensor frame is calculated as

$$R_B^S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_s & \sin \phi_s \\ 0 & -\sin \phi_s & \cos \phi_s \end{bmatrix} \begin{bmatrix} \cos \theta_s & 0 & -\sin \theta_s \\ 0 & 1 & 0 \\ \sin \theta_s & 0 & \cos \theta_s \end{bmatrix} \begin{bmatrix} \cos \psi_s & \sin \psi_s & 0 \\ -\sin \psi_s & \cos \psi_s & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In general each sensor has a unique position and orientation relative to the origin of the sensor frame. These are function of the how individual sensors are mounted. We define the following relative position vectors.

\vec{r}_L = position of the LIDAR relative to the sensor frame origin

\vec{r}_{INS} = position of the INS sensor relative to the sensor frame origin

\vec{r}_{GA} = position of the GPS receiver antenna to the sensor frame origin

Defining references frame fixed to each individual sensor, the rotation matrices that define the transformation from the sensor fixed reference frame to the individual sensor frames can be defined in terms of the roll, pitch and yaw of each sensor relative to the sensor frame. These angles are a function of how each sensor is mounted. Therefore, we define the following rotation matrices.

$$R_S^L(\varphi_L, \theta_L, \psi_L) = \text{orientation of the LIDAR relative to the sensor frame}$$

$$R_S^{INS}(\varphi_{INS}, \theta_{INS}, \psi_{INS}) = \text{orientation of the INS sensor relative to the sensor frame}$$

$$R_S^{GA}(\varphi_{GA}, \theta_{GA}, \psi_{GA}) = \text{orientation of the GPS antenna relative to the sensor frame}$$

In this ground vehicle application, the rotation matrix R_B^S (rotation of body fixed frame to the sensor frame) reduces to the identity matrix because the sensors are mounted on the vehicle so they are aligned with the body fixed axes. For this work, the origin of sensor frame was selected to be collocated with the origin of the LIDAR. The LIDAR, INS, and GPS sensors are mounted with the same alignment; therefore, the rotation matrices ($R_S^L, R_S^{INS}, R_S^{GA}$) all reduce to the identity matrix.

3.3 Position Calculations

The latitude, longitude and altitude are measured with respect to geodetic coordinates providing the position of the GPS receiver relative to the WGS-84 ellipsoid. An earth centered, earth fixed reference frame \mathbf{E} can be defined at the center of the earth as follows:

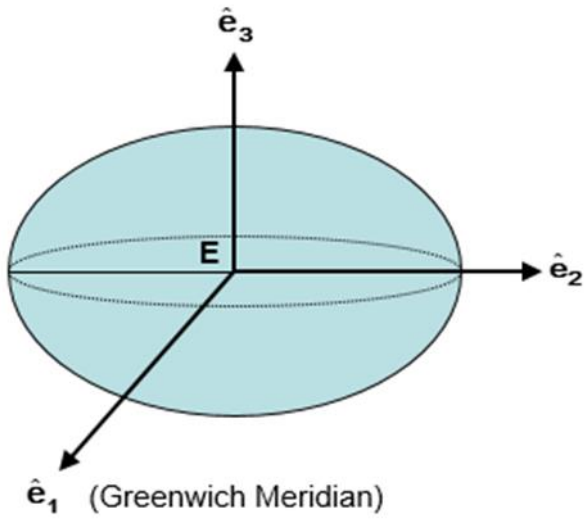


Figure 3.5a Earth centered, earth fixed frame \mathbf{E} (origin at center of Earth)

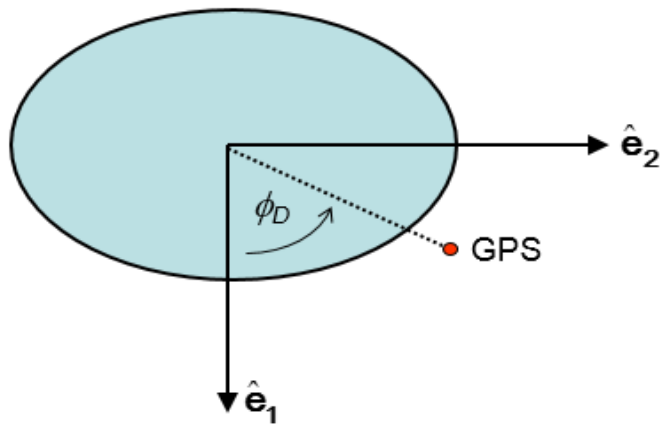


Figure 3.5b The longitude ϕ_D from an overhead view of the Earth

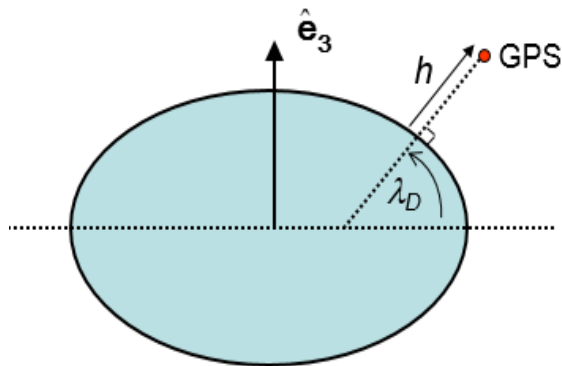


Figure 3.5c The geodetic latitude λ_D and altitude h from a side view of Earth

Given the GPS measurements (geodetic latitude, longitude and altitude), it is desirable to compute the vehicle position with respect to a local geodetic (North, East, Down) reference frame. A NED reference frame can be defined from the initial GPS measurement $(\lambda_{D,0}, \varphi_{D,0}, h_0)$ by first expressing the initial GPS location in the Earth-centered, Earth-fixed (ECEF) frame. Then the rotation matrix that transforms the ECEF frame into a local NED frame can be defined, and the origin of the NED frame is located at the initial ECEF location of the vehicle center of mass. All subsequent GPS measurements can then be referenced to this local NED frame.

3.3.1 Computing the initial GPS position in the ECEF frame

The initial GPS location can be expressed in terms of the ECEF reference frame as follows:

$$X_{0,GPS,ECEF} = \left(\frac{a}{\chi} + h_0 \right) \cos(\lambda_{D,0}) \cos(\varphi_{D,0})$$

$$Y_{0,GPS,ECEF} = \left(\frac{a}{\chi} + h_0 \right) \cos(\lambda_{D,0}) \sin(\varphi_{D,0})$$

$$Z_{0,GPS,ECEF} = \left(\frac{a}{\chi} (1 - e^2) + h_0 \right) \sin(\lambda_{D,0})$$

where

$a = 6378137$ m (semi-major axis of the WGS-84 ellipsoid)

$e^2 = 0.00669438$ (square of the eccentricity of the WGS-84 ellipsoid)

$$\chi = \sqrt{1 - e^2 \sin^2(\lambda_{D,0})}$$

3.3.2 Local NED Frame Coordinate Transformation

The rotation matrix R_E^G defining the transformation from the ECEF reference frame to a local NED reference frame, where the North-East plane is tangent to the surface of the WGS-84 ellipsoid at the origin, can be defined in terms of three single-axis rotations. The first rotation is about the ECEF \hat{e}_3 axis by the initial longitude $\varphi_{D,0}$ as shown in Figure 3.6.

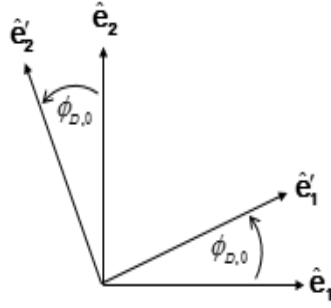


Figure 3.6 Rotation about \hat{e}_3 axis by initial longitude $\phi_{D,0}$

The rotation about the \hat{e}_3 axis by the initial longitude $\varphi_{D,0}$ is given by the following matrix:

$$\begin{Bmatrix} \hat{e}'_1 \\ \hat{e}'_2 \\ \hat{e}'_3 \end{Bmatrix} = \begin{bmatrix} \cos\varphi_{D,0} & \sin\varphi_{D,0} & 0 \\ -\sin\varphi_{D,0} & \cos\varphi_{D,0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{Bmatrix}$$

The next rotation corresponds to a rotation about the \hat{e}'_2 axis by the initial latitude $\lambda_{D,0}$, which is given by the matrix

$$\begin{Bmatrix} \hat{e}''_1 \\ \hat{e}''_2 \\ \hat{e}''_3 \end{Bmatrix} = \begin{bmatrix} \cos\lambda_{D,0} & 0 & \sin\lambda_{D,0} \\ 0 & 1 & 0 \\ -\sin\lambda_{D,0} & 0 & \cos\lambda_{D,0} \end{bmatrix} \begin{Bmatrix} \hat{e}'_1 \\ \hat{e}'_2 \\ \hat{e}'_3 \end{Bmatrix}$$

After the second rotation, we have an Up North East reference frame. In order to obtain a North East Down frame, the direction of the upward axis must be reversed and the

coordinate axes must be reordered. These operations are achieved by the following transformation:

$$\begin{Bmatrix} \hat{\mathbf{g}}_1 \\ \hat{\mathbf{g}}_2 \\ \hat{\mathbf{g}}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{e}}''_1 \\ \hat{\mathbf{e}}''_2 \\ \hat{\mathbf{e}}''_3 \end{Bmatrix}$$

3.3.3 Locating the Origin of the Local NED Frame

The origin of the local NED frame is located at the position of the vehicle CM when the initial GPS measurement is taken. The position in the ECEF reference frame can be computed as

$$\mathbf{r}_{0,\text{cm}}^{\text{ECEF}} = \begin{Bmatrix} X_{0,\text{cm}} \\ Y_{0,\text{cm}} \\ Z_{0,\text{cm}} \end{Bmatrix}_{\text{ECEF}} = \begin{Bmatrix} X_{0,\text{GPS}} \\ Y_{0,\text{GPS}} \\ Z_{0,\text{GPS}} \end{Bmatrix}_{\text{ECEF}} - \underbrace{\left(\mathbf{R}_G^{\text{E}} \mathbf{R}_{0,\text{B}}^{\text{G}} \mathbf{r}_S^{\text{B}} + \mathbf{R}_G^{\text{E}} \mathbf{R}_{0,\text{B}}^{\text{G}} \mathbf{R}_S^{\text{B}} \mathbf{r}_{\text{GA}}^{\text{S}} \right)}_{\text{Position of GPS w.r.t vehicle CM}}$$

where

$\mathbf{R}_{0,\text{B}}^{\text{G}}$ = initial rotation from body-fixed to local NED reference frame (derived from the initial orientation angles)

\mathbf{r}_S^{B} = position of the sensor frame origin relative to the CM, expressed in the body-fixed reference frame.

\mathbf{R}_S^{B} = rotation from sensor frame to body-fixed reference frame (computed from the mounting angles of the sensor box)

$\mathbf{r}_{\text{GA}}^{\text{S}}$ = position of the GPS antenna relative to the sensor frame origin, expressed in the sensor reference frame.

3.3.4 Expressing the GPS position in the Local NED Frame

The local NED position of the vehicle CM at each time step is computed by subtracting the position of the GPS antenna relative to the CM:

$$\mathbf{r}_{\text{CM}}^G = \begin{Bmatrix} \mathbf{X}_{\text{cm}} \\ \mathbf{Y}_{\text{cm}} \\ \mathbf{Z}_{\text{cm}} \end{Bmatrix}_G = \mathbf{r}_{\text{GPS}}^G - \left(\mathbf{R}_B^G \mathbf{r}_S^B + \mathbf{R}_B^G \mathbf{R}_S^B \mathbf{r}_{\text{GA}}^S \right)$$

where R_B^G denotes the rotation from the body-fixed to the local NED reference frame, which is derived from the vehicle orientation angles at each time step.

3.4 INS Orientation Calculations

Orientation measurements are obtained from the Microstrain 3DM-GX3-45 INS sensor. The measurements $(\varphi_m, \theta_m, \psi_m)$ provide the orientation of the INS sensor relative to the local geodetic (inertial) frame, from which we can then compute the orientation of the vehicle relative to the geodetic frame. The rotation matrix defining the transformation from the geodetic frame to the orientation sensor frame is computed as

$$R_G^{INS} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_m & \sin\varphi_m \\ 0 & -\sin\varphi_m & \cos\varphi_m \end{bmatrix} \begin{bmatrix} \cos\theta_m & 0 & -\sin\theta_m \\ 0 & 1 & 0 \\ \sin\theta_m & 0 & \cos\theta_m \end{bmatrix} \begin{bmatrix} \cos\psi_m & \sin\psi_m & 0 \\ -\sin\psi_m & \cos\psi_m & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As mentioned previously, the origin of the sensor frame is collocated with the origin of the LIDAR sensor frame. Therefore, given the rotation matrices R_S^{INS} (orientation of the INS frame relative to the sensor frame) and R_B^S (orientation of the sensor frame relative to the body fixed frame), the rotation matrix describing the orientation of the body fixed frame relative to the local geodetic (NED) frame is given by

$$R_G^B = R_S^B R_{INS}^S R_G^{INS}$$

For this work, the INS sensor is aligned with the LIDAR, and the LIDAR is aligned with the vehicle. Therefore,

$$R_S^B = R_{INS}^S = I$$

$$R_G^B = R_G^{INS}$$

Hence, $\varphi = \varphi_m$, $\theta = \theta_m$, and $\psi = \psi_m$. That is, the vehicle orientation angles are equivalent to the orientation angles measured by the INS sensor.

3.5 LIDAR Point Cloud Calculations

The LIDAR reference frame is defined with the origin located on the LIDAR sensor. The LIDAR x axis (\hat{l}_1 axis in Figure 3.5) is fixed along the sensor boresight, the y axis (\hat{l}_2 axis in Figure 3.7) is pointed 90 degrees to the right of the sensor and the z axis is pointed downward (i.e., into the page).

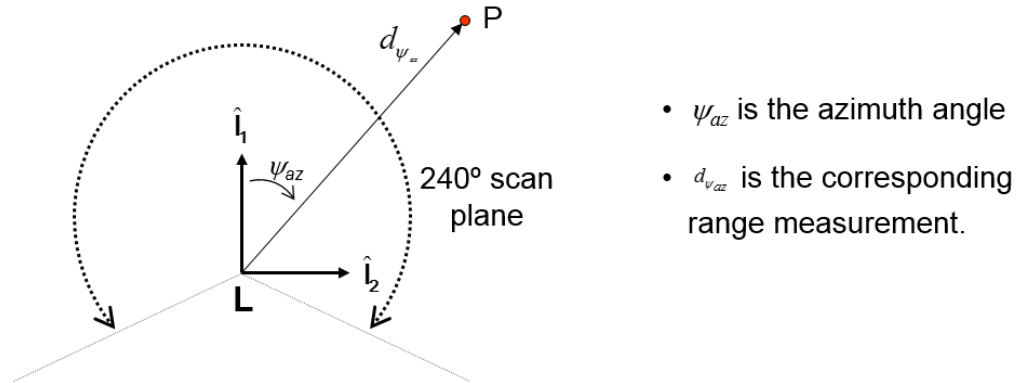


Figure 3.7 LIDAR Reference Frame

Since the Hokuyo UTM-30LX is a line-scan LIDAR, all the measurements are in the \hat{l}_1 - \hat{l}_2 plane shown in Figure 3.7. The position vector of a LIDAR point measurement is given in the LIDAR reference frame as

$$\{\mathbf{r}_P\}_L = \begin{Bmatrix} x_P \\ y_P \\ z_P \end{Bmatrix}_L = \begin{Bmatrix} d_{\psi_{az}} \cos(\psi_{az}) \\ d_{\psi_{az}} \sin(\psi_{az}) \\ 0 \end{Bmatrix}$$

The LIDAR point measurement in the local geodetic (NED) frame is then computed as

$$\mathbf{r}_P^G = \begin{Bmatrix} X_P \\ Y_P \\ Z_P \end{Bmatrix}_G = \mathbf{r}_{Cm}^G + \mathbf{R}_B^G \mathbf{r}_S^B + \mathbf{R}_B^G \mathbf{R}_S^B \mathbf{r}_L^S + \mathbf{R}_B^G \mathbf{R}_S^B \mathbf{R}_L^S \mathbf{r}_P^L$$

where

\mathbf{r}_{Cm}^G = position of the vehicle CM, expressed in the local geodetic (NED) reference frame

\mathbf{r}_L^S = position of the LIDAR relative to the sensor frame origin, expressed in the sensor reference frame (note that this is zero since the LIDAR frame is collocated with the sensor frame in this application)

R_L^S = rotation from the LIDAR reference frame to the sensor reference frame, derived from how the LIDAR is mounted relative to the sensor frame.

In this manner, each LIDAR point measurement is expressed in terms of a common reference frame, the local geodetic (NED) frame. This generates a dense collection of data points in the inertial frame, known as a point cloud, from which a 3D terrain map of the environment can be constructed.

Chapter 4

Terrain Mapping and Path Planning

4.1 3D Terrain Representations from Point Cloud Data

The point cloud generated from LIDAR measurements is used to generate a terrain representation of the environment. This thesis considers two methodologies that are based on mathematical learning theory developed by researchers at the University of South Carolina. Mathematical learning theory is a zone of parametric statistics that can be viewed as a converging of nonlinear approximation theory, adaptive tree algorithms and statistical estimation. A vital characteristic of learning algorithms is that they do not require information regarding the underlying probability distribution describing the full scenario. The first terrain mapping approach utilizes a mathematical learning algorithm to build adaptive, multiresolution terrain representations from point cloud data [36]. This approach generates a piecewise constant terrain representation over rectangular or triangular sub domains, but linear interpolation can be applied to obtain continuous piecewise linear representations. This approach is well suited for real time implementations because it is fast and recursive in nature. The only disadvantage with this approach is that it has tendency to cover the point cloud data with a terrain skin, which provides a functional representation that is not sufficient to represent certain 3D terrain features.

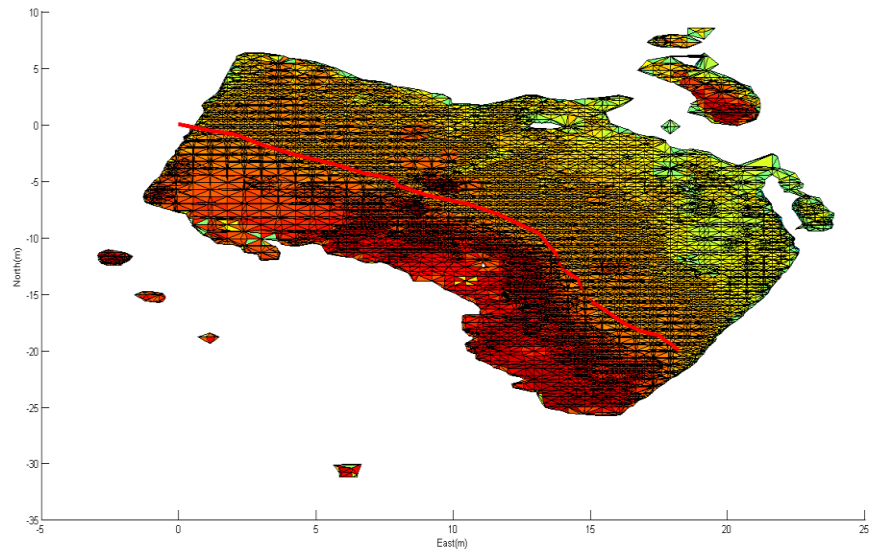
The second methodology uses implicit terrain algorithms to represent 3D features in the environment. It is able to represent 3D surfaces with a high level of detail; however, this algorithm is not recursive in nature (that is, the whole terrain representation must be regenerated every time new point cloud data are obtained). Basically, the implicit terrain algorithm fits a 3D isosurface around the point cloud data. The computation of the

isosurface is achieved by first producing a tetrahedral occupancy grid depending on the data distribution. An unsigned distance is then calculated from the cell vertices to the data within the cell in order to generate a 3D surface that embodies the data for each tetrahedral occupancy cell. The implicit terrain algorithms can be applied in navigation systems in order to generate representations of complex 3D terrain features. These algorithms are used to provide obstacle avoidance constraints to a receding horizon path planning algorithm.

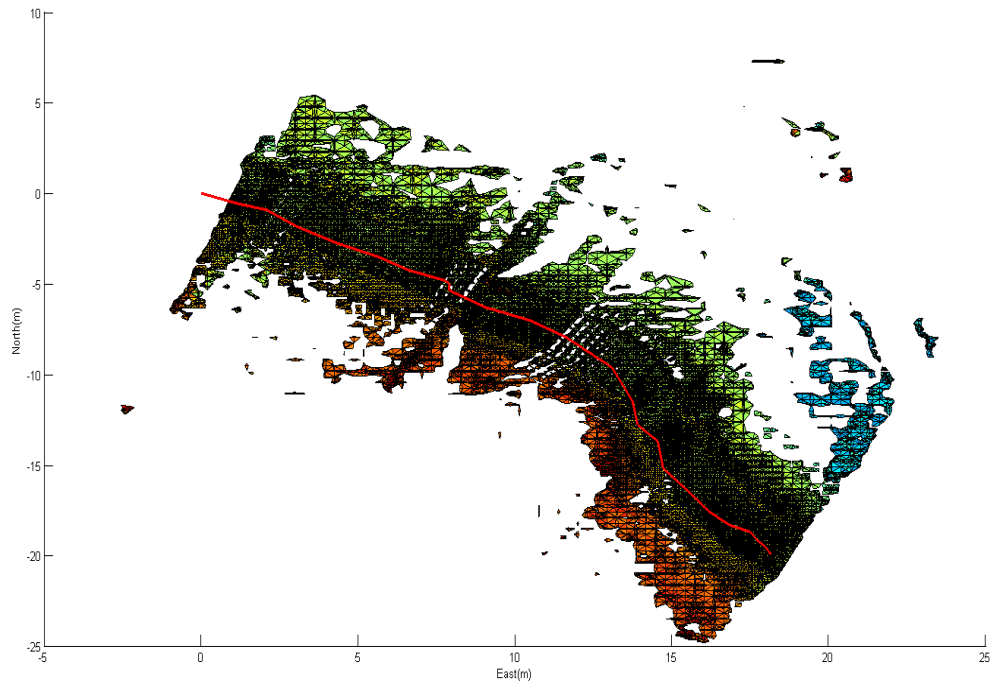
The representation of the environment by these implicit terrain algorithms depends on three key parameters: the depth parameter, occupancy parameter and isolevel parameter. The depth parameter is used to describe the minimum size of the occupancy cells that will be used for the representation of the 3D terrain. The occupancy parameter defines the minimum number of data points that must occupy a cell before it is further subdivided into smaller cells. The isolevel parameter determines the distance of isosurface from the point cloud data. In other words, the isolevel parameter basically specifies the size of the isosurface that encapsulates the data. The isolevel parameter can be varied according to the resolution required to capture specific 3D terrain features since it provides the resolution of the complex 3D structure (that is, it specifies to what level of detail a particular feature can be resolved). A lower value of the isolevel parameter is better for the resolution of small features like people or vehicles, but it often results in holes or gaps in the terrain representations of larger structures such as buildings. Hence a higher value of the isolevel parameter is required for larger structures because it produces a smoother and more continuous surface, which best fits larger complex structures.

A test was performed to investigate the effect of varying the isolevel parameter on a specific point cloud data set, the results of which can be seen in Figure 4.1. This figure

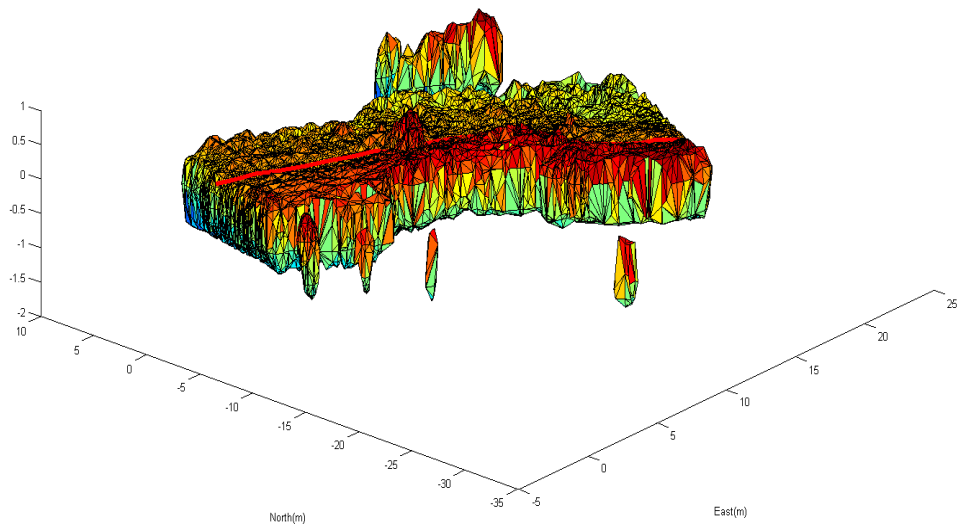
shows the terrain representations obtained from LIDAR data collected from the unmanned ground vehicle as it traversed a path over relatively flat terrain. During this experiment, the vehicle passes between two orange traffic cones on either side of the path. The point cloud data were processed using the implicit terrain algorithm with isolevel parameters of 0.5 and 0.1.



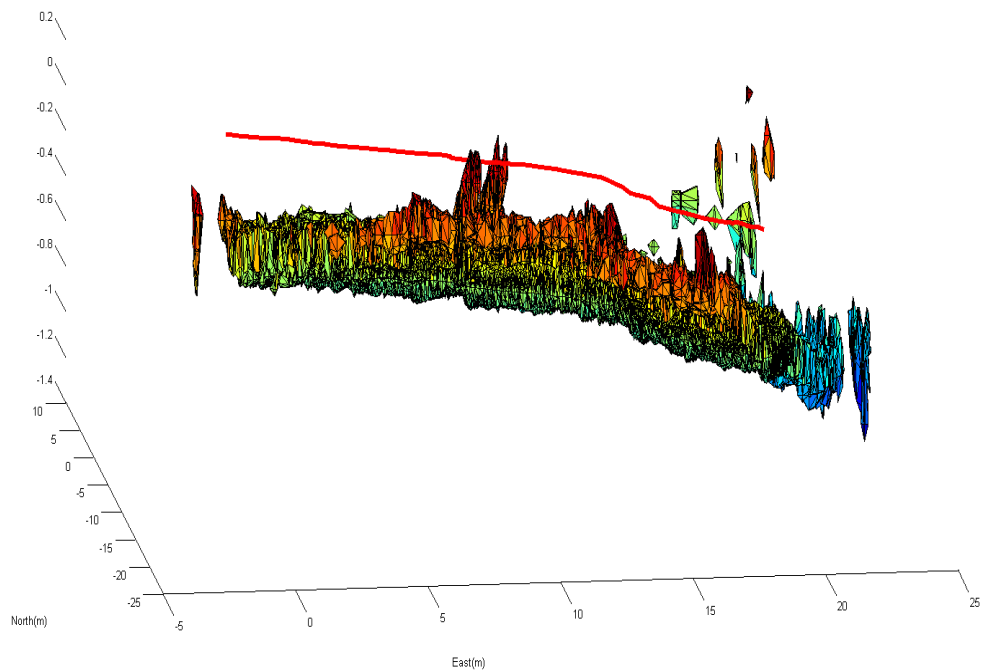
(a) Terrain: isolevel 0.5



(b) Terrain: Isolevel 0.1



(c) Terrain: Isolevel 0.5 (Different View)



(d) Terrain: Isolevel 1.0 (Different view)

Figure 4.1 Effect of Varying the Isolevel Parameter

It is clear from Figure 4.1 (c) and 4.1 (d) that a better resolution for small objects (cones) is obtained using a smaller value of the isolevel parameter, but the smaller isolevel parameter also results in gaps in the representation of the ground terrain (see Figure 4.1(b)). The larger isolevel provides better representations of larger structures or the full environment because it provides a smoother and more continuous terrain surface.

4.2 Receding Horizon Path Planning Algorithm

Receding horizon control (RHC), otherwise called model predictive control (MPC), is a modern control algorithm that started to receive considerable attention in the 1980's. Essentially, with RHC, an optimization problem is solved over a finite time interval to determine a plan of action over a fixed time horizon. In RHC, the planner incorporates the

goal and constraints as components of an optimal control problem. RHC can also incorporate new information, constraints, future references and assessments of future disturbances. In this work, RHC is implemented as a path planning algorithm, which computes path points for the UGV that are passed to a waypoint controller.

The receding horizon path planning algorithm employed in this work computes a path that minimizes a mission based cost functional and incorporates obstacle avoidance constraints derived from the 3D terrain representation [43]. Therefore, with this approach, the vehicle travels a path that minimizes a cost function (low fuel consumption, minimum time of travel, etc.) subject to obstacle avoidance constraints. As discussed earlier, the RHC algorithm computes an optimal path over a finite time horizon. A part of this path is executed and meanwhile new information is accumulated by the sensors, the 3D terrain map is upgraded and a new optimal path is then computed. In this way, the vehicle path is planned over a sliding window in time until the vehicle attains its goal, which is a specified destination waypoint in this study. The autonomous navigation system is designed in such a manner that it generates the points or planned path to the control system, which in return generates the required control commands for the vehicle to follow the desired trajectory. Hence, in this work, RHC is used to generate a path that minimizes a cost functional subject to obstacle avoidance constraints derived from the 3D terrain map so that the vehicle can attain a target final waypoint.

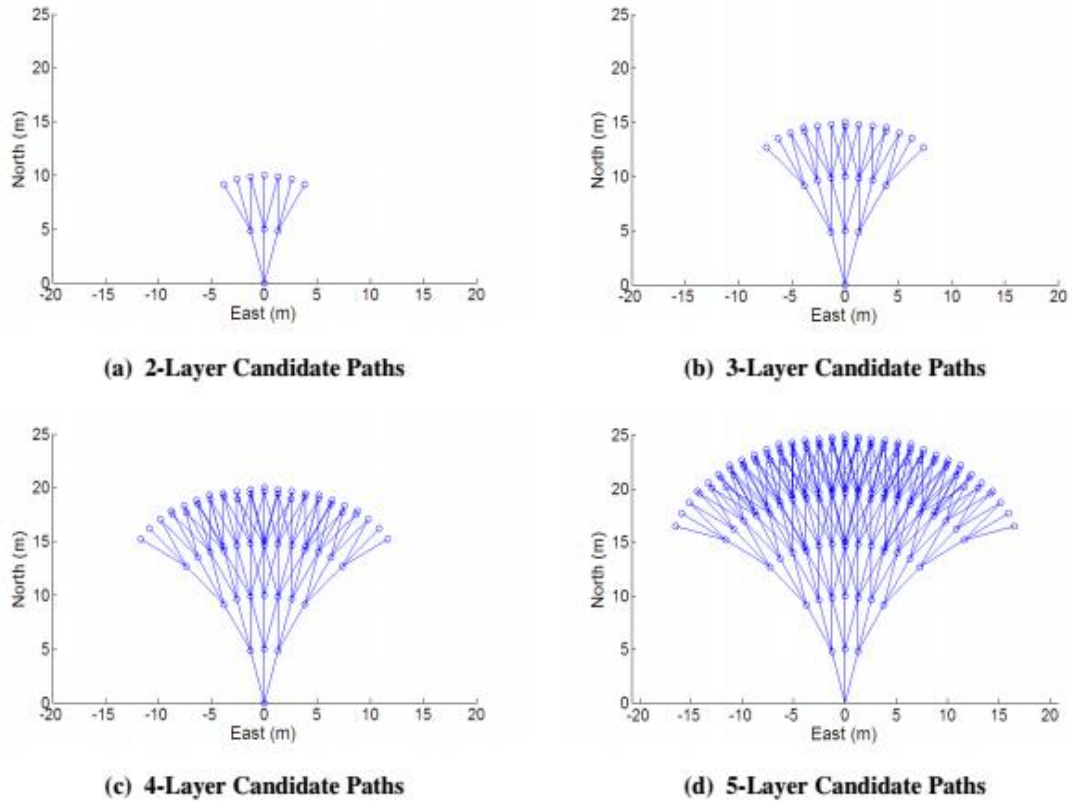


Figure 4.2 Candidate Path Points for Two-Dimensional Path Planning

The receding horizon path planning algorithm involves producing an arrangement of candidate path points for a certain interval of time and then determining the optimal sequence of path points within that horizon. Basically, the candidate points are arranged in a tree like structure such that there are different layers of points, and for moving from a point in one layer to another layer, there are multiple options available. In two dimensions, the important parameters are N_p , N_c , ψ_h and d , where N_p is number of planned points, N_c is number of path points to be followed, ψ_h is the maximum horizontal angle between the candidate points in the next layer and d is the distance between points in consecutive layers. Figure 4.2 shows the multiple layers of candidate 2D paths radiating from a vehicle situated at the origin and oriented due north. In this example, for every point in a given layer, there are 3 choices for moving to the following layer corresponding to heading changes of -15° ,

0° , and 15° . Therefore, in this example, there is a 30° maximum horizontal angle (ψ_h) between candidate points in the next layer with an angular resolution of 15° , and the distance d is 5 meters between points in consecutive layers. With these parameters, there are a total of 3^{Np} candidate two-dimensional paths.

The augmentation of this approach to 3D is straightforward and involves selecting a maximum vertical path angle ψ_v and a vertical angular resolution. If these two parameters are set equal to the horizontal angles as in the above scenario, there are 9 decisions from a given path point to the next and a total of 9^{Np} candidate paths. An important point to be noted here is that vehicle kinematic constraints can be applied through an appropriate choice of the path planning parameters. That is, the parameters can be chosen so that all candidate paths are within the maneuvering capabilities of the vehicle.

The mission objective of navigating the unmanned ground vehicle from a starting point A to a target location B can be cast as an optimization problem subject to path constraints for obstacle avoidance. The UGV equations of motion can be written as a system of nonlinear, coupled ordinary differential equations:

$$\dot{X}(t) = F(X(t), U(t), t) \quad (1)$$

$$X(0) = X_0$$

The vector $X(t)$ represents the ground vehicle state vector and $U(t)$ denotes the control input. Obstacles (i.e., the terrain map) are estimated from sensor measurements $Y(t)$, given by

$$Y(t) = O(X(t); \chi, t) \quad (2)$$

The observation operator O defines the output Y and its dependence on the vehicle states and the surrounding environment. The trajectory of the ground vehicle is constrained such

that it does intersect any obstacle within the environment. The symbol χ is used to express the subset of R^3 , three-dimensional space, occupied by obstacles. We denote

$$\begin{aligned}\chi(X, Y, Z) &= 1 && \text{if } (x, y, z) \in \chi \\ \chi(X, Y, Z) &= 0 && \text{otherwise}\end{aligned}\quad (3)$$

Therefore, the path of the vehicle is subject to the constraint

$$(PX)(t) = (X_c(t), Y_c(t), Z_c(t)) \notin \chi \quad (4)$$

In Eq. (4), P is an operator that selects specific states from the state vector X. This constraint enforces the requirement that the center of mass of the vehicle (X_c, Y_c, Z_c) , which is subset of the vehicle state vector, must not intersect any of the obstacles.

The receding horizon path planning algorithm associated with the objective of navigating the UGV from a starting point (X_A, Y_A, Z_A) to a goal location (X_B, Y_B, Z_B) is then formulated as the following optimization problem:

Find the path points $\{P_k^i = (X_k^i, Y_k^i, Z_k^i), k = 1, \dots, N_H\}$ that minimize the cost functional

$$J_i = \left\{ (X_{N_s+1}^i - X_{N_s}^i)^2 + (Y_{N_s+1}^i - Y_{N_s}^i)^2 + (Z_{N_s+1}^i - Z_{N_s}^i)^2 \right\}$$

subject to the constraints

$$\pi(P_k^i, P_{k-1}^i) \notin \chi \quad k = 1, \dots, N_H \quad (\text{obstacle avoidance})$$

$$P_k^i \in \Gamma_k^i \quad k = 1, \dots, N_H \quad (\text{set of candidate path points})$$

$$P_0^i = (X_A, Y_A, Z_A) \quad (\text{initial condition})$$

This optimization problem is solved from $i = 0, \dots, N$, where N is the number of path planning steps required to reach the goal location B. After each path planning optimization, which provides N_H path points, the first $N_C < N_H$ path points are passed to a waypoint

controller, which computes the control inputs required to drive the UGV through the path points. The UGV travels through the N_C path points, during which time it collects new sensor data, updates the terrain map, and then solves a new optimization problem to compute the next set of path points. This process continues until the UGV reaches the goal location B.



Figure 4.3 Path Planning Scenario (Goal Location B)

This path planning algorithm was implemented by planning a series of trajectories using the receding horizon algorithm. This implementation minimizes a cost function that incorporates weighted terms for approaching the objective mission point B subject to obstacle avoidance constraints based on the 3D implicit terrain map. The obstacle avoidance constraints are implemented by first checking that, for every available path, the path points are no less than a distance d_s from the from the 3D terrain surface, which indicates that the candidate path points are clear of obstacles by at least a margin of safety d_s . Then, all the paths that fulfill this requirement are ranked in order of increasing cost. Beginning with the minimal cost path, a higher resolution set of points is created along the

straight segments connecting the path points, and then the obstacle avoidance constraints are checked for each of these intermediate points. This procedure addresses the possibility that an obstacle may exist between two successive path points. If by chance none of the candidate paths fulfill these constraints, the algorithm returns a command to the vehicle to either stay in place to gather extra information or to change direction in an attempt to find a path that satisfies the requirements.

Chapter 5

Modeling and Control of the Unmanned Ground Vehicle

5.1 UGV Kinematics

The unmanned ground vehicle (UGV) used for this work is a differential drive robot with four wheels. The movement of the wheels on the same side of the vehicle is the same; hence for the ease of calculation, the equations of motions can be calculated for two tires representing either side of the vehicle. The two wheels are modeled as mounted on a common axis and can be moved independently in both the forward and backward directions. For the rolling motion of the robot, the velocity of each wheel can be varied but there must be common point for both wheels about which the robot can rotate. This point is called the instantaneous center of curvature.

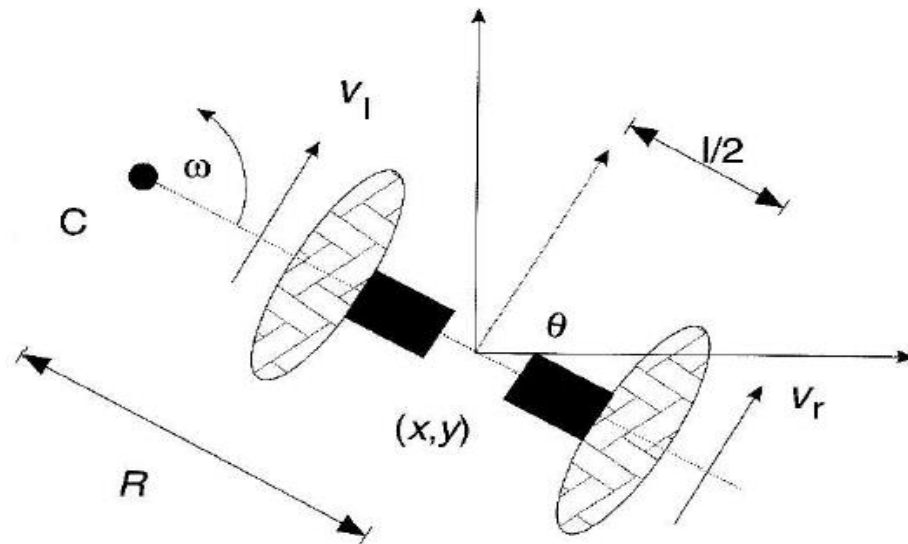


Figure 5.1 Differential Drive Kinematics for the Unmanned Ground Vehicle

The rate of rotation ω about the instantaneous center of curvature must be same for both the wheels. Hence we can write:

$$\omega \left(R + \frac{l}{2} \right) = V_r \quad 5.1$$

$$\omega (R - l/2) = V_l \quad 5.2$$

where

l = distance between the center of the two wheels

V_r = translational velocity of the right wheel

V_l = translational velocity of the left wheel

R = distance from the instantaneous center of curvature to the midpoint between the wheels

c = instantaneous center of curvature

From equation 5.1 and 5.2,

$$R = \frac{\left(\frac{l}{2}\right)(V_l + V_r)}{(V_l - V_r)} \quad 5.3$$

$$\omega = (V_l - V_r)/l \quad 5.4$$

From these equations, four cases can be derived:

1) If $V_l = V_r$, the robot moves in a straight line (linear motion) with no rotation.

In this case, we obtain $R = \infty$ and $\omega = 0$.

2) If $V_l = -V_r$, the vehicle rotates about the midpoint of the wheel axis.

3) If $V_l = 0$, the vehicle rotates about the left wheel in which case $R = l/2$.

4) If $V_r = 0$, the vehicle rotates about the right wheel in which case $R = l/2$.

The current position of the robot is denoted as (x, y) and the heading angle θ is defined as the angle between the position vector and the x axis. The values of V_l and V_r

can be used to vary the position and orientation of the robot. The location of the instantaneous center of curvature c can be determined as

$$c = [x - R \sin(\theta), y + R \cos(\theta)]$$

At time $t + dt$, the pose of the robot is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\omega dt) & -\sin(\omega dt) & 0 \\ \sin(\omega dt) & \cos(\omega dt) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - c_x \\ y - c_y \\ \theta \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \\ \omega dt \end{bmatrix}$$

5.2 Controller

As discussed earlier, the receding horizon algorithm is used to plan the path to be followed by the ground robot. Waypoints along the planned path are then passed to the UGV controller, which drives the vehicle through each waypoint. Hence, the controller was designed to accept waypoints from the RHC path planner and generate the control commands required to travel through those waypoints. The controller designed for this work outputs left speed, right speed and time to the robot. This controller was implemented and tested in Simulink using a simulation model of a differential drive robot similar to the Corobot UGV.

Figure 5.3 shows the Waypoint Controller generating left and right speed. The PI controller is fed with the information from RHC, INS and GPS. So once it has RHC path points, UGV heading and UGV position, it outputs the Left speed, right speed and time (seconds). Further this information is sent to Torque Actuator Command which in turns generate right and left torque(to the UGV wheels).

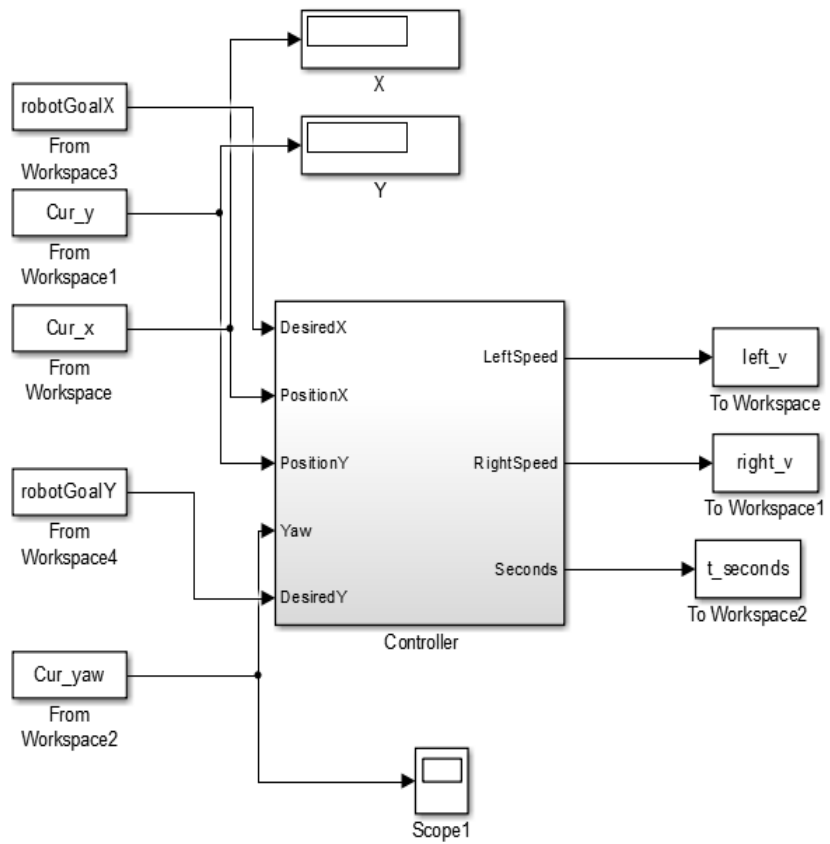


Figure 5.2 Feedback Control Law for the Unmanned Ground Vehicle

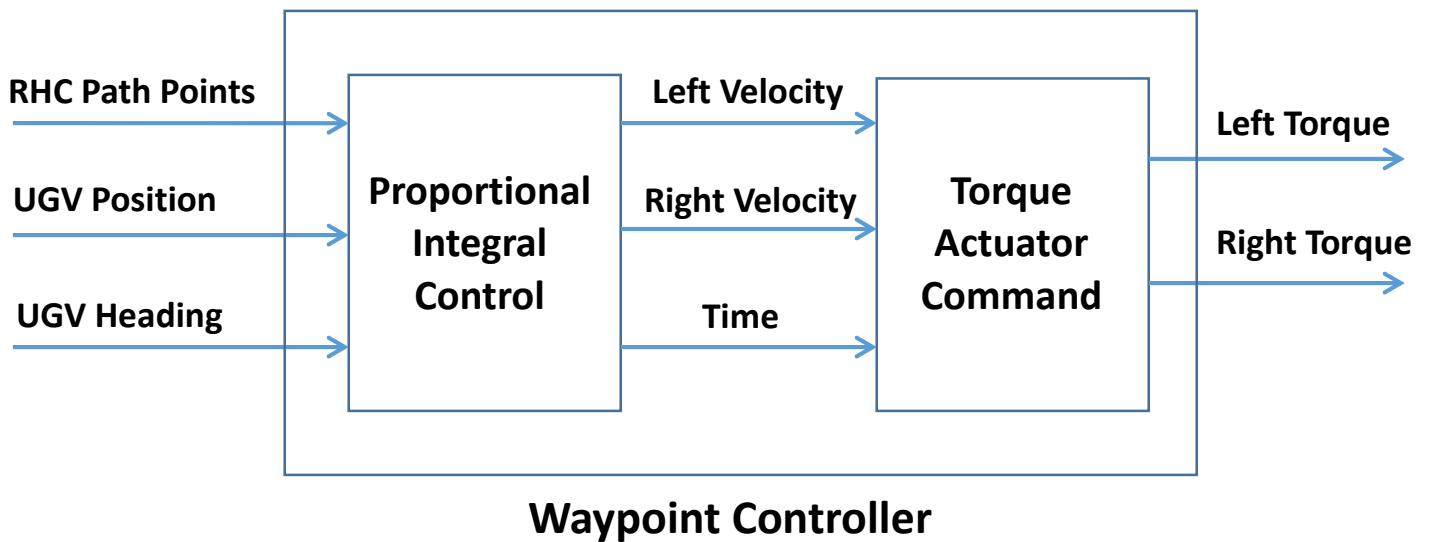
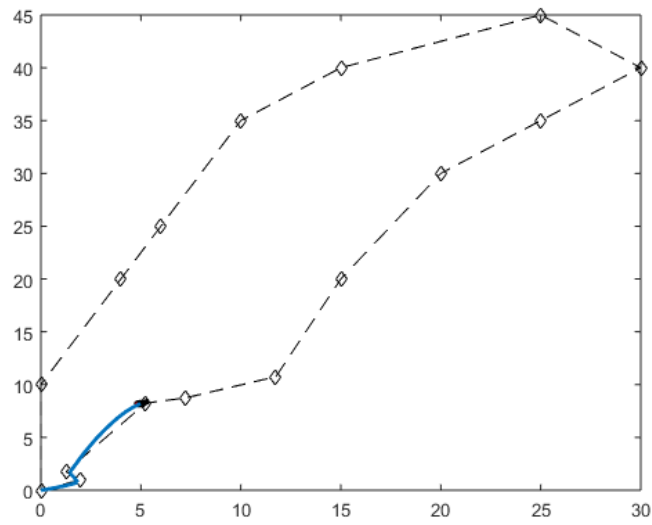
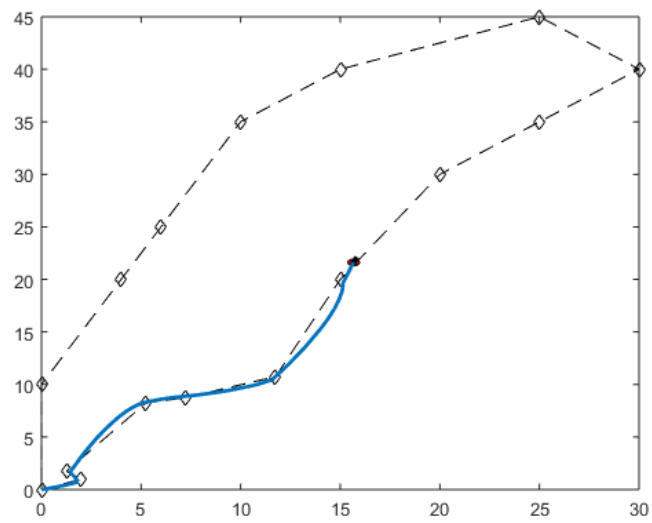


Figure 5.3 Waypoint Controller Generating Left and right torque

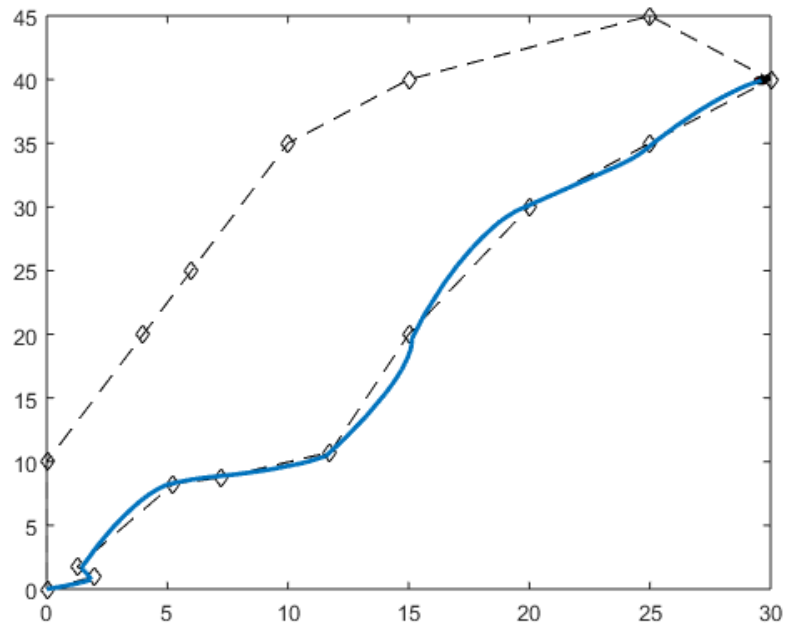
To check the functionality of the controller and simulation model, a set of path points was passed to the controller and simulation model in order to simulate the ability of the controller to drive the vehicle through the waypoints. (In practice, these points would be provided by the receding horizon path planning algorithm). The simulation results, shown in Figure 5.5, demonstrate that the vehicle successfully travels through the waypoints.



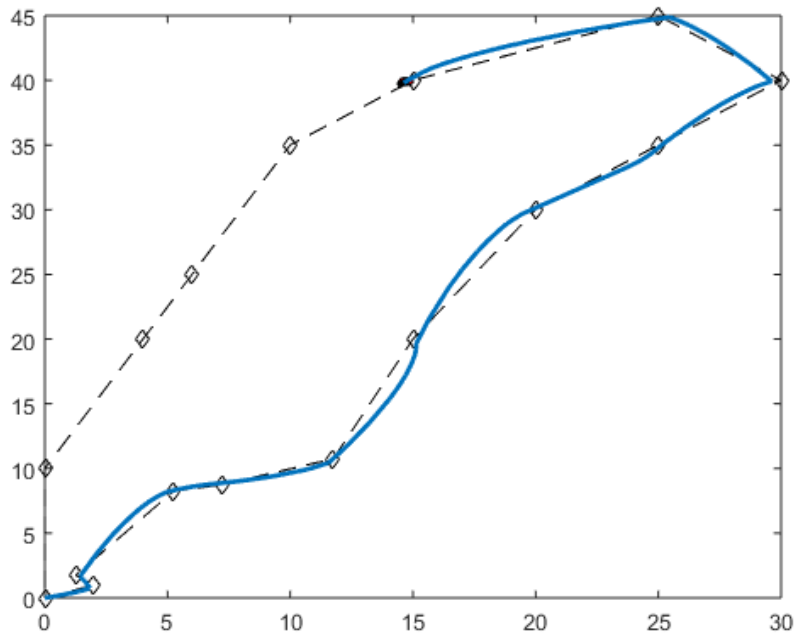
(a) Position of Vehicle at $t = 20$ s



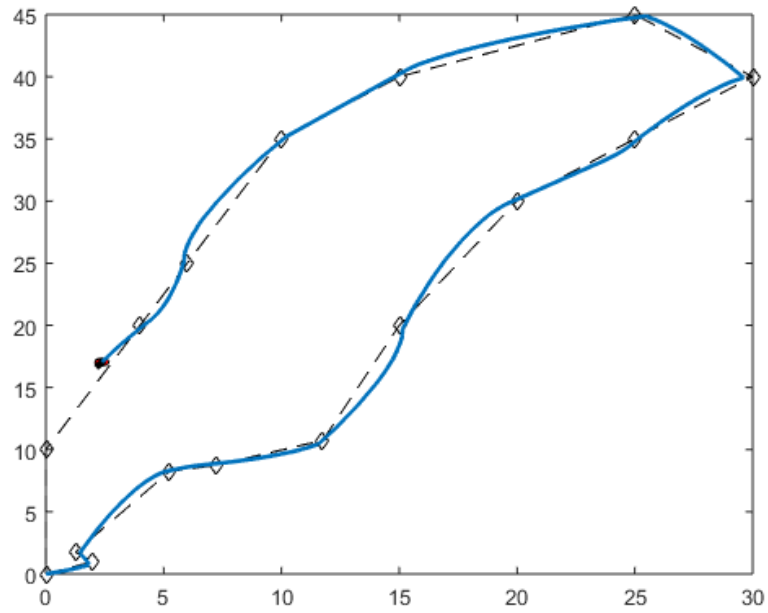
(b) Position of Vehicle at $t = 40$ s



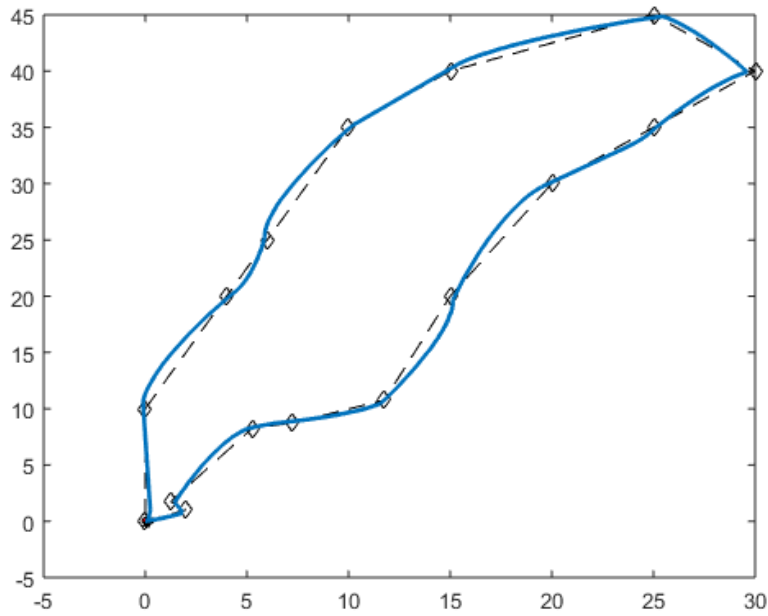
(c) Position of Vehicle at t = 60 s



(d) Position of Vehicle at t = 80 s



(e) Position of Vehicle at $t = 100$ s



(f) Position of Vehicle at $t = 120$ s

Figure 5.5 Position of Ground Vehicle at Different Time Intervals.

Chapter 6

Experimental and Simulation Results

A series of experimental and simulation studies were performed in order to demonstrate and evaluate the performance of the GNC algorithms applied to the unmanned ground vehicle. An initial open-loop experiment was conducted in which the UGV was manually controlled to drive a straight path on level terrain with two cones on either side of the path serving as obstacles. The purpose of this experiment was to confirm the functionality of the terrain mapping algorithms for a relatively simple test case. Then, a second open-loop experiment was conducted in which the robot slowly traveled on a curved path with multiple obstacles in the area in order to test the terrain mapping algorithms on more complex terrain. Finally, a set of closed-loop simulations was performed in which the UGV planned a path and autonomously drove through the planned waypoints to arrive at a target location. In these simulations, the terrain data derived from the second set open-loop experiment was used to provide the obstacle avoidance constraints for the path planning algorithm.

6.1 Open-Loop Experiment #1

Two open-loop experiments were run with the unmanned ground vehicle (UGV) driven around different locations on the Embry-Riddle Aeronautical University campus. The first experiment was a relatively simple case in which the UGV was driven along a straight path with two cones serving as obstacles. The two cones were placed on the side of the sidewalk and the UGV was driven between them. This relatively simple case was used to verify the proper functioning of the sensor fusion and terrain algorithms. Figures 6.1.1 – 6.1.3 show the UGV path and the cones that were placed in the scene. Figure 6.1.1

shows the view from an onboard camera while Figures 6.1.2 and 6.1.3 provide an overhead view of the path. In these figures, the red line shows the path driven by the UGV, and two X marks in Figure 6.1.3 denote the places where cones were placed.



Figure 6.1.1: View of Overall Path from the Camera on the UGV



Figure 6.1.2: Overhead View of Path taken by the UGV on the ERAU Campus (red line shows the path)

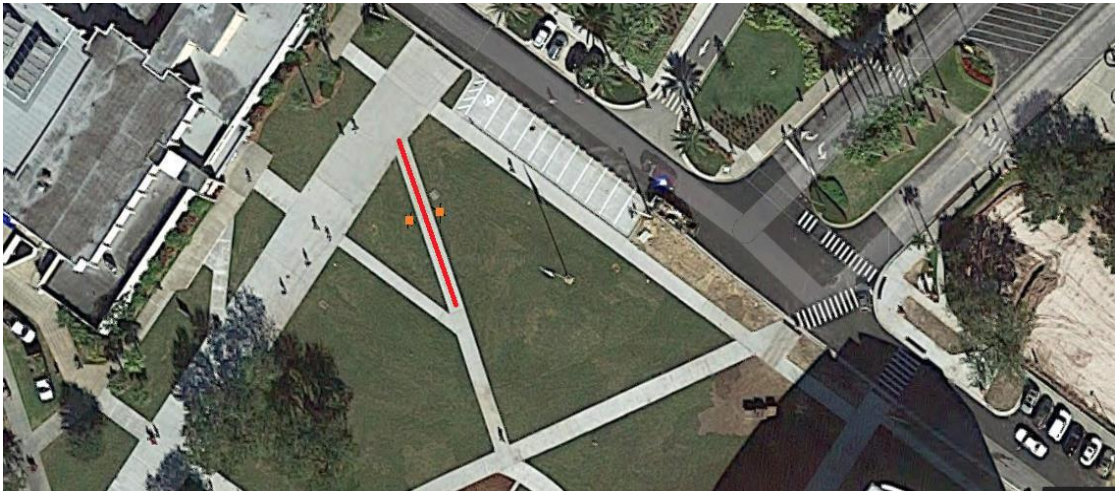


Figure 6.1.3: Zoomed-in View of UGV Path

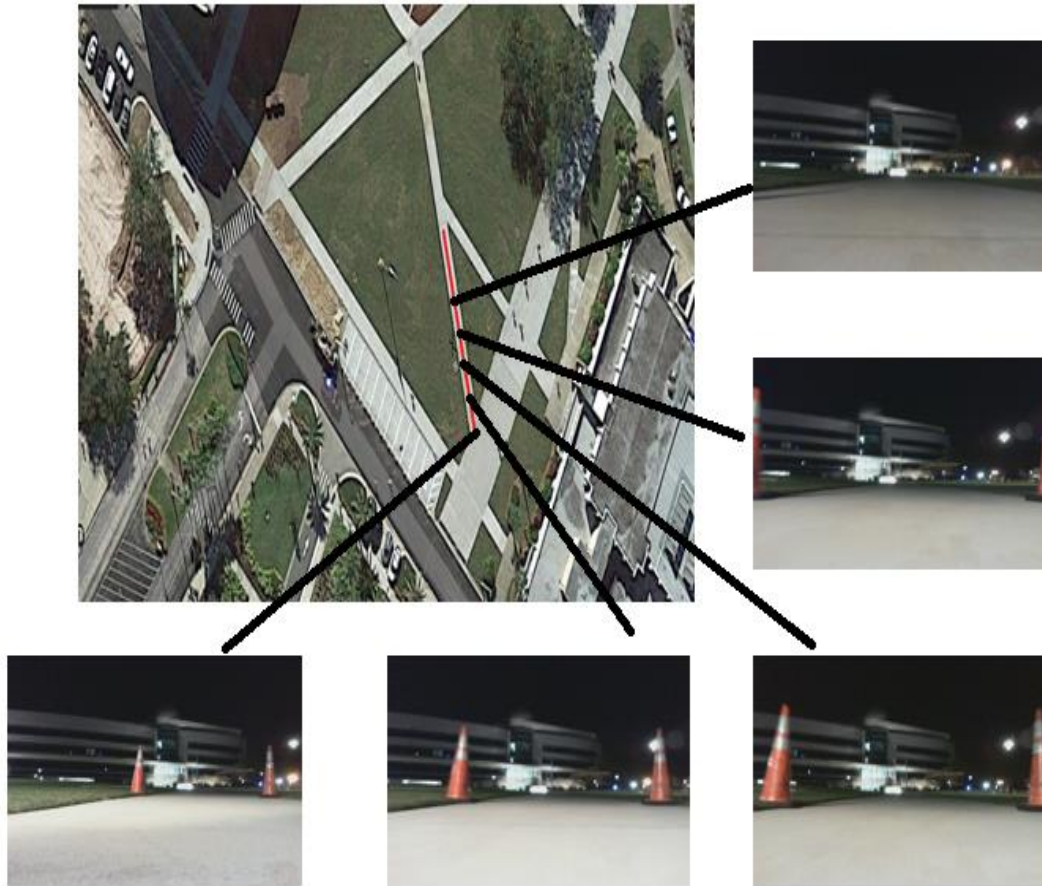


Figure 6.1.4: Ground Vehicle Track with Sample Images from the UGV Camera

Figure 6.1.4 provides a detailed view of the path taken by the UGV. The sample images were collected by the camera mounted on the UGV. It is important to note that, for this work, the images taken by the camera on the UGV are not used for navigation but are instead just used for giving a detailed view of the scenario as shown in the figure.

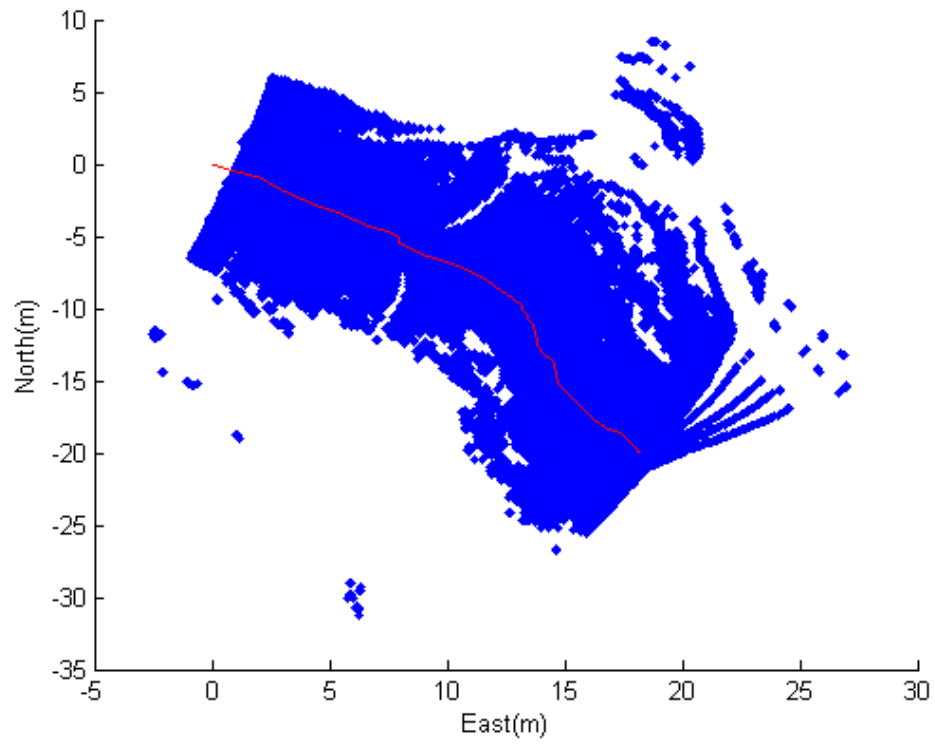


Figure 6.1.5: LIDAR Measurements with UGV Path (shown in red)

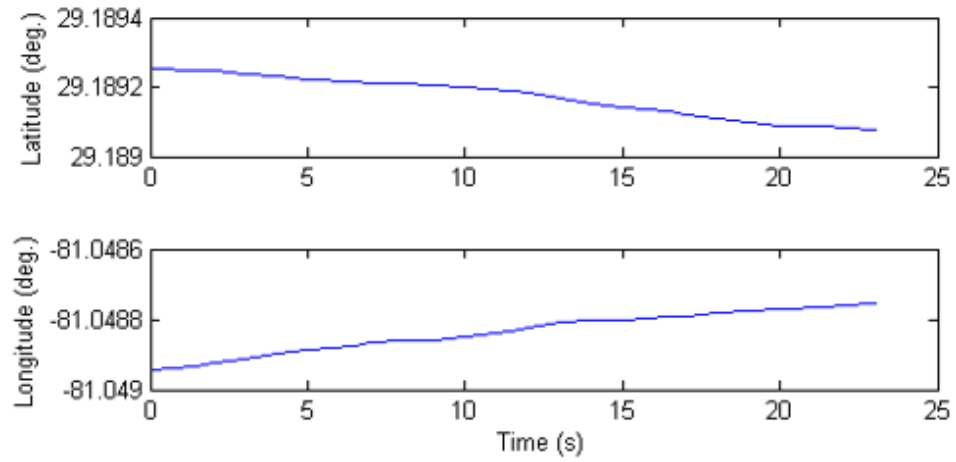


Figure 6.1.6: GPS Latitude and Longitude Measurements

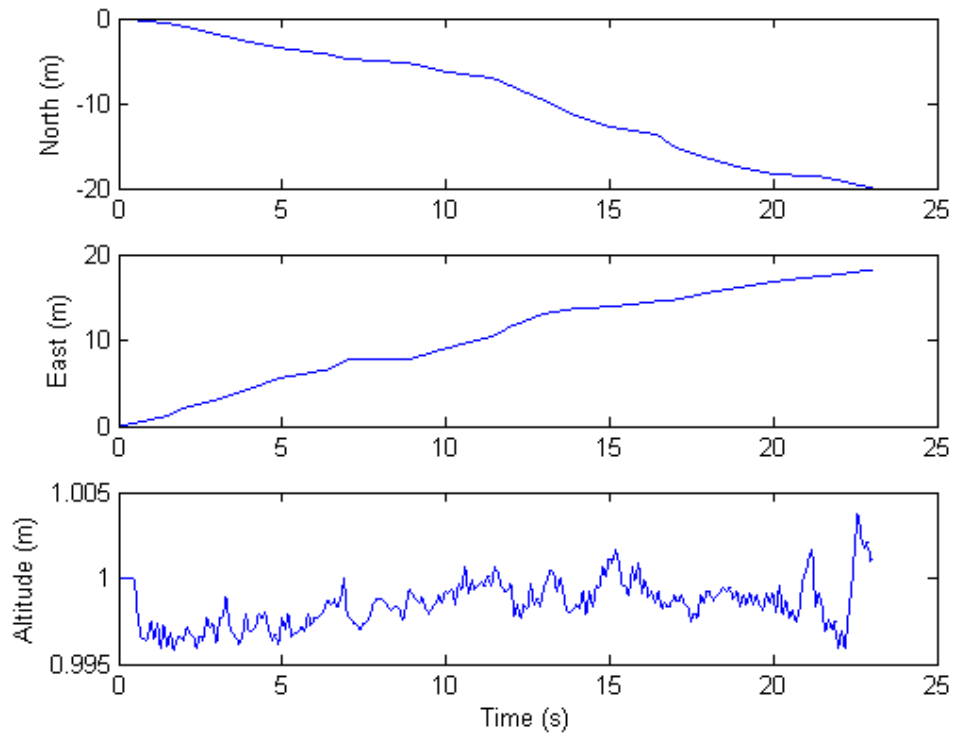


Figure 6.1.7: NED Position of the Vehicle CM

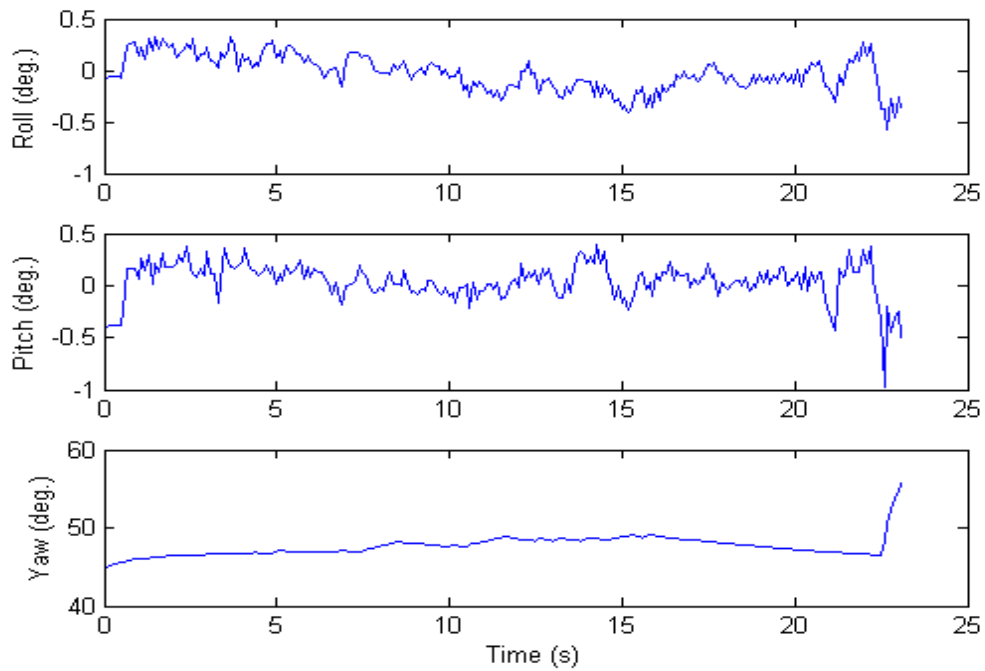
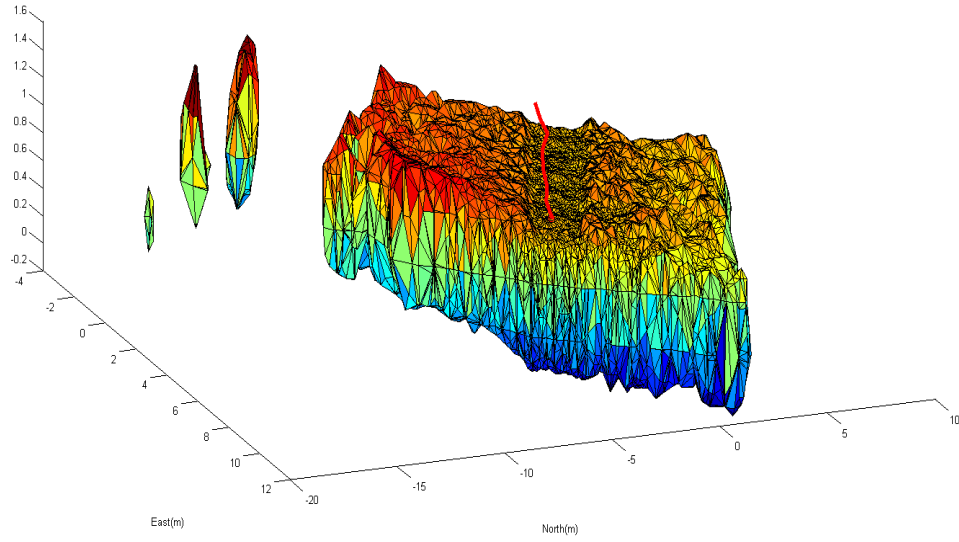
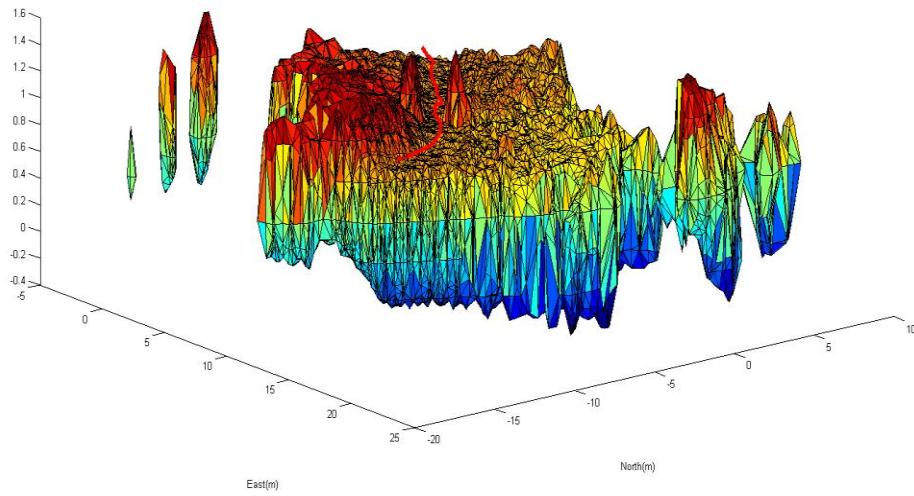


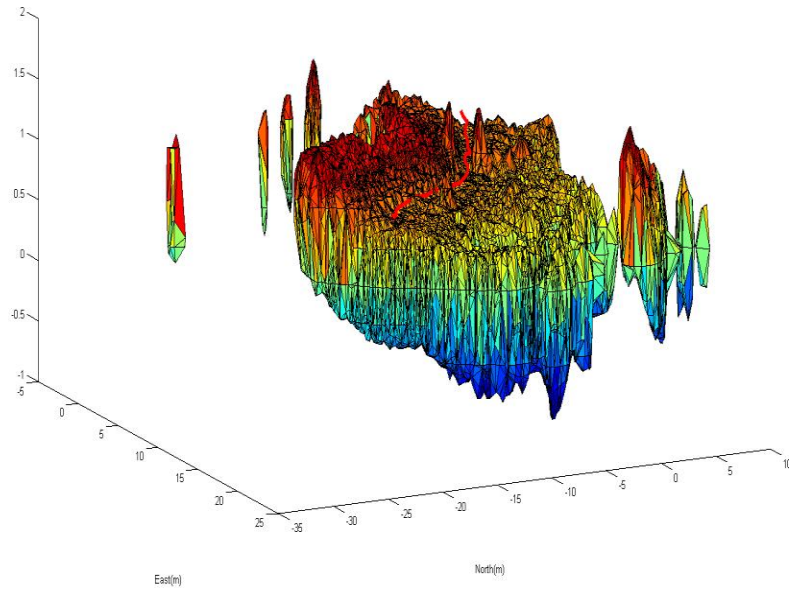
Figure 6.1.8: Orientation (Roll, Pitch, and Yaw) Measurements



(a) 3D Terrain Representation at $t = 10$ sec



(b) 3D Terrain Representation at $t = 20$ sec



(c) 3D Terrain Representation at $t = 40$ sec

Figure 6.1.9: 3D Terrain Representation at Different Time Intervals

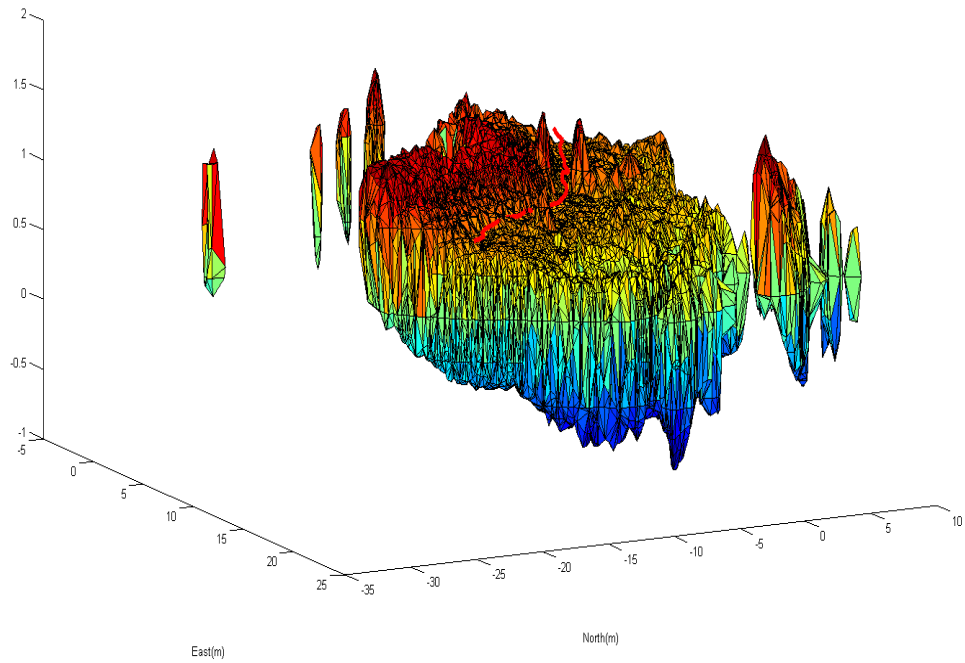


Figure 6.1.10: Complete 3D Terrain Representation with Vehicle Path (shown in red)

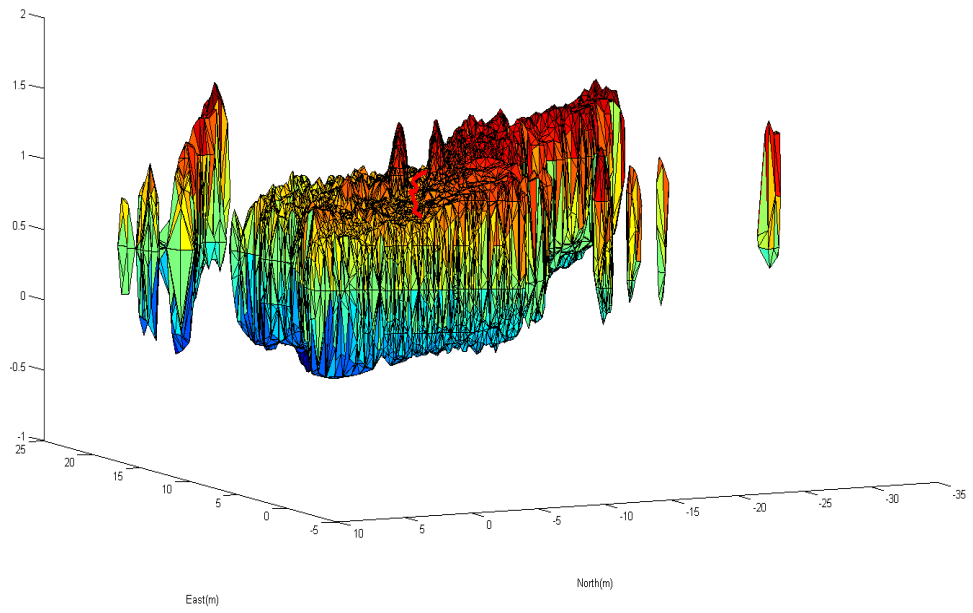
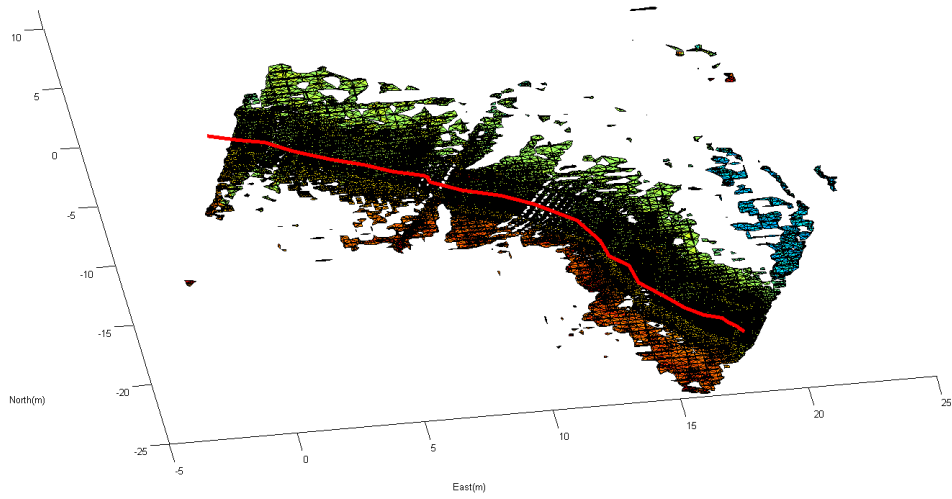
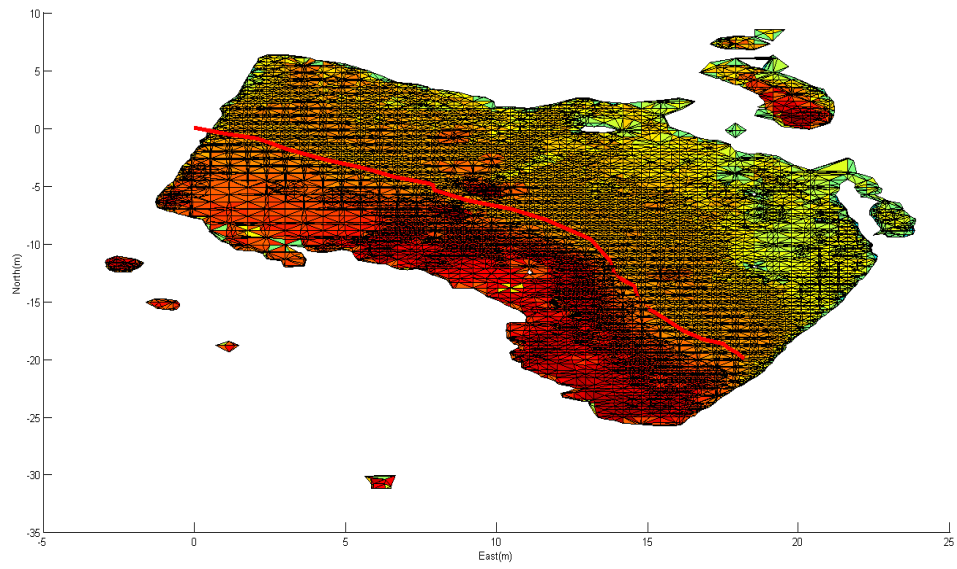


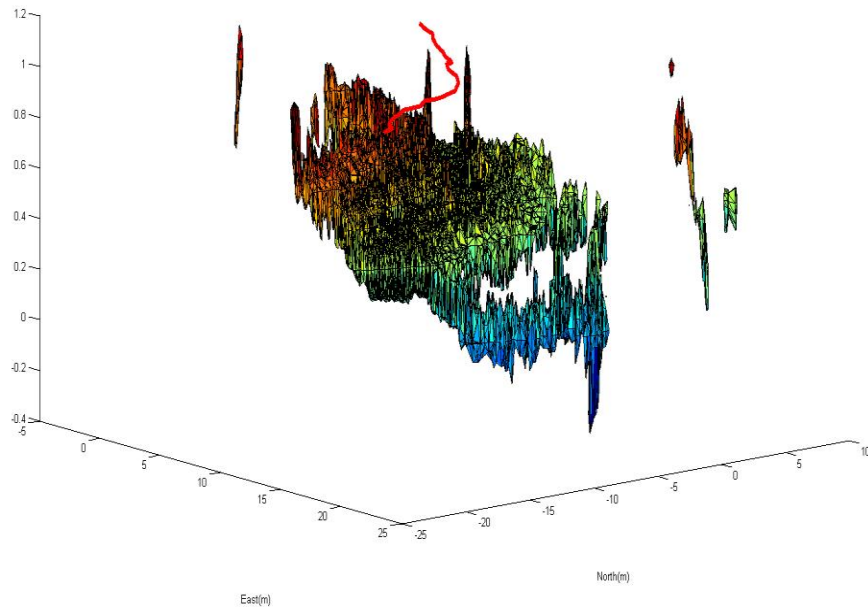
Figure 6.1.11: Front View of Complete 3D Terrain Map



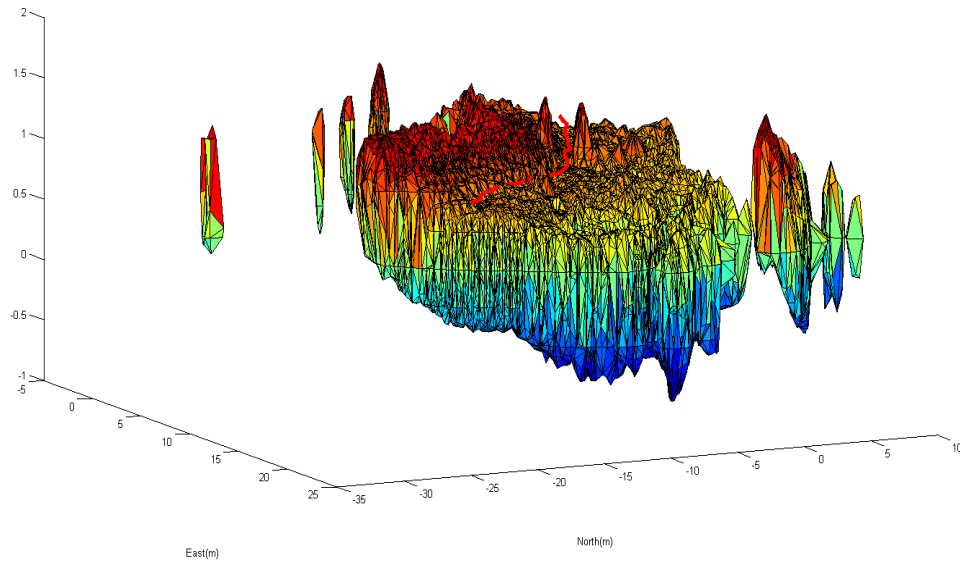
(a) Terrain: Isolevel 0.1



(b) Terrain: Isolevel 0.5



(c) Terrain: Isolevel 0.1



(d) Terrain: Isolevel 0.5

Figure 6.1.12: Effect of Varying Isolevel Parameter on Terrain Representation

Figures 6.1.5 – 6.1.8 illustrate the GPS, LIDAR, NED position, and orientation measurements obtained during the open-loop experiment. It includes computing the location and orientation of the vehicle relative to a local NED (inertial) reference frame., and the LIDAR data in Figure 6.1.5 has been registered to the NED frame. All the data were recorded with a global time stamp but at different sample rates. Therefore, resampling of the data was required in order to perform the analysis. The computations were based on the derivations in Chapter 3. The LIDAR measurements with the vehicle path in red is shown in Figure 6.1.5. The LIDAR points were registered to the local NED reference frame using the GPS and orientation data that were resampled at at the same interval as the LIDAR data. Figure 6.1.6 shows the GPS latitude and longitude measurements. GPS data were sampled at approximately 8 Hz. A local NED reference frame was defined at the

initial location of the vehicle CM, which was computed from the initial GPS measurement as shown in Figure 6.1.7. The NED position was computed from the GPS and orientation data, after resampling these data via linear interpolation. Figure 6.1.8 shows the orientation measurements. As the data were collected from the ground vehicle, there was not much variation in roll and pitch measurements. Figure 6.1.9 illustrates the 3D terrain representation at different time intervals. It shows the build up of 3D terrain as the data are collected during the run. In each figure, the path is shown in red and the terrain has been constructed from the LIDAR points collected up to that point. Figure 6.1.12 shows the variation in 3D terrain representation as a result of varying the isosurface parameter. As discussed previously, this parameter determines the distance of isosurface from the point cloud data. In other words, the isosurface parameter basically specifies the size of the isosurface that encapsulates the data.

6.2 Open-Loop Experiment #2

In the second open-loop case, the UGV path was curved with numerous obstacles in the scene. Figures 6.2.1 – 6.2.3 depict the path taken by the UGV during this open-loop experiment.



Figure 6.2.1: View of Overall Path



Figure 6.2.2: Overhead View of UGV Path from Google Earth (path shown in red)

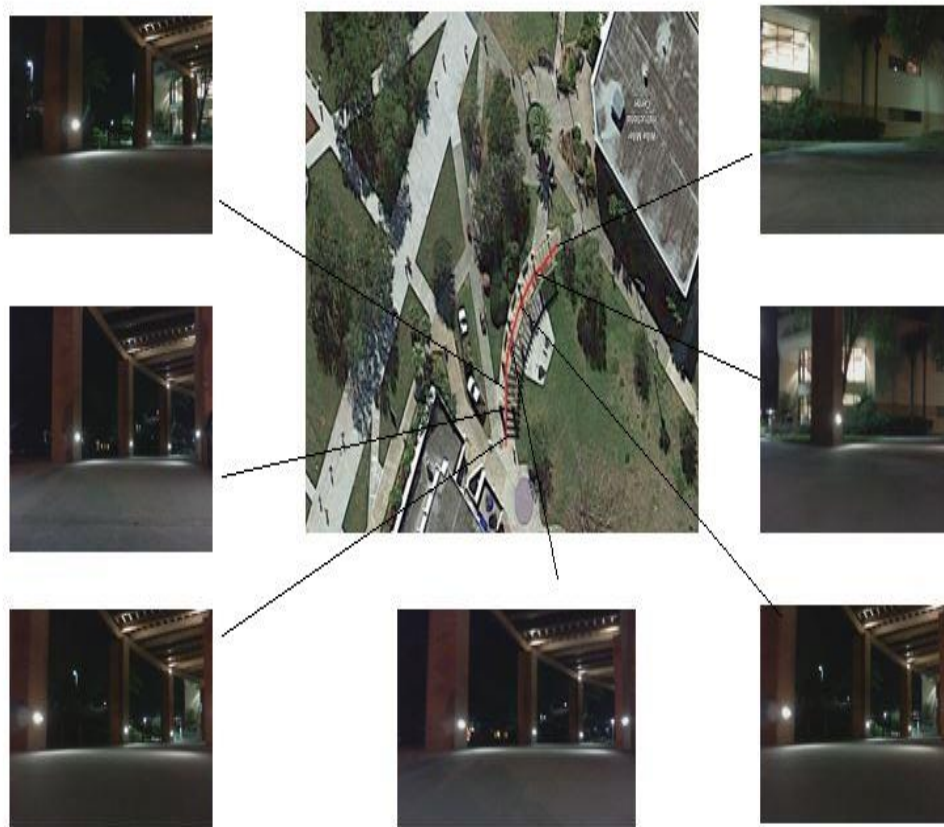


Figure 6.2.3: UGV Track with Sample Images from the Onboard Camera

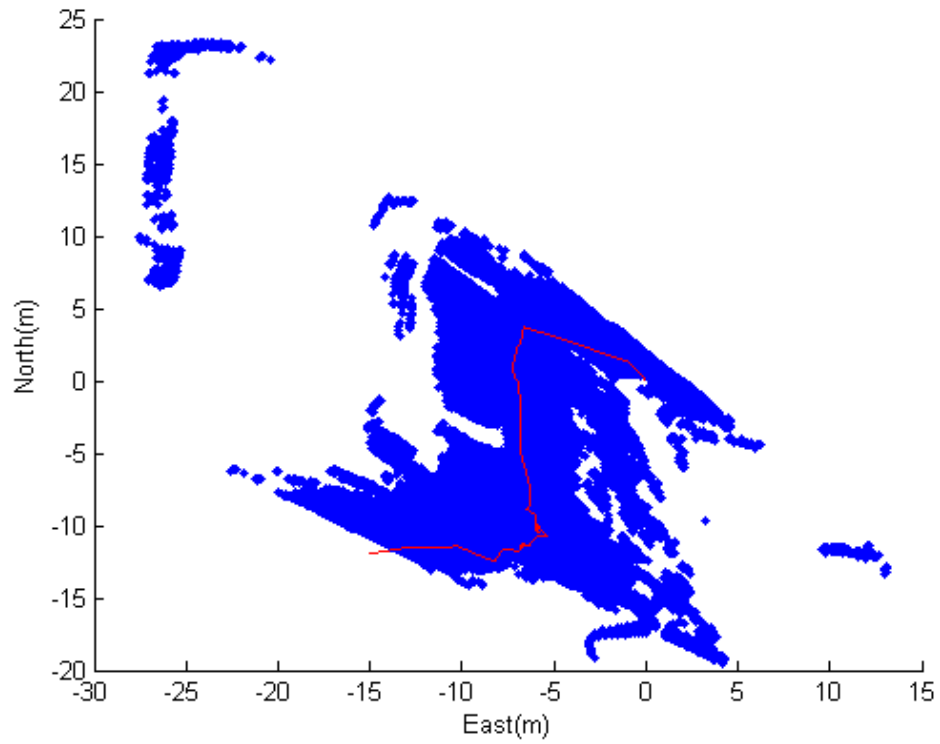


Figure 6.2.4: LIDAR Measurements with UGV Path (shown in red)

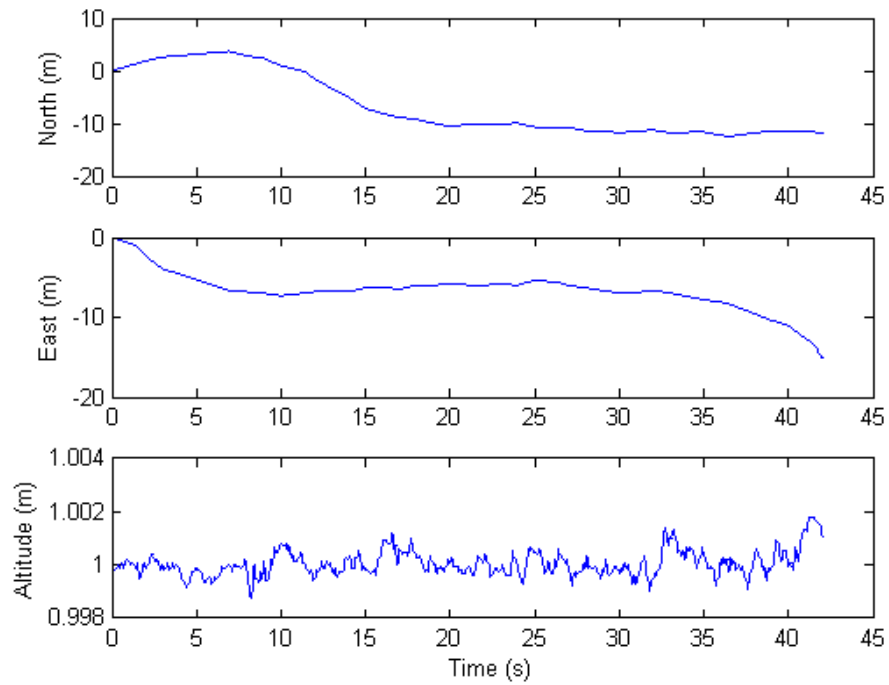


Figure 6.2.5: Inertial Position of the Vehicle CM

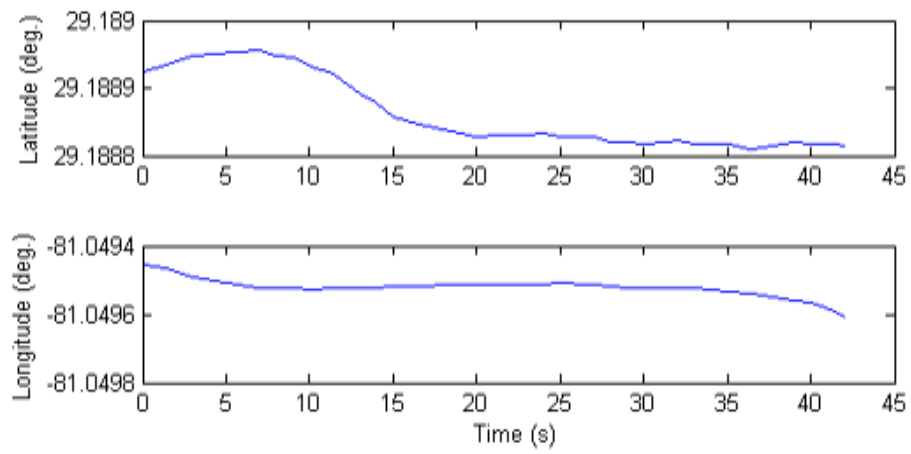


Figure 6.2.6: Latitude and Longitude GPS Measurements

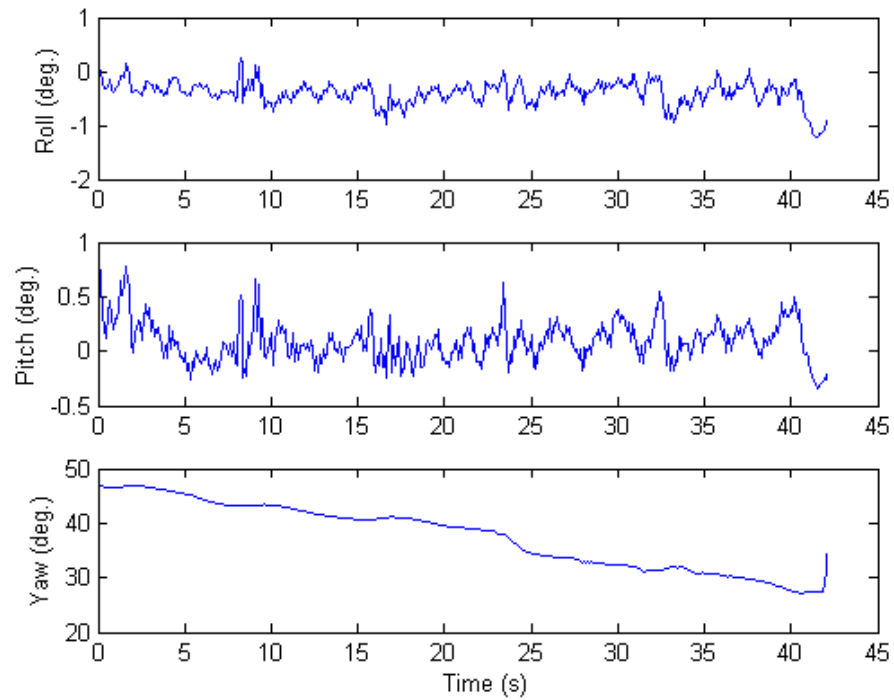
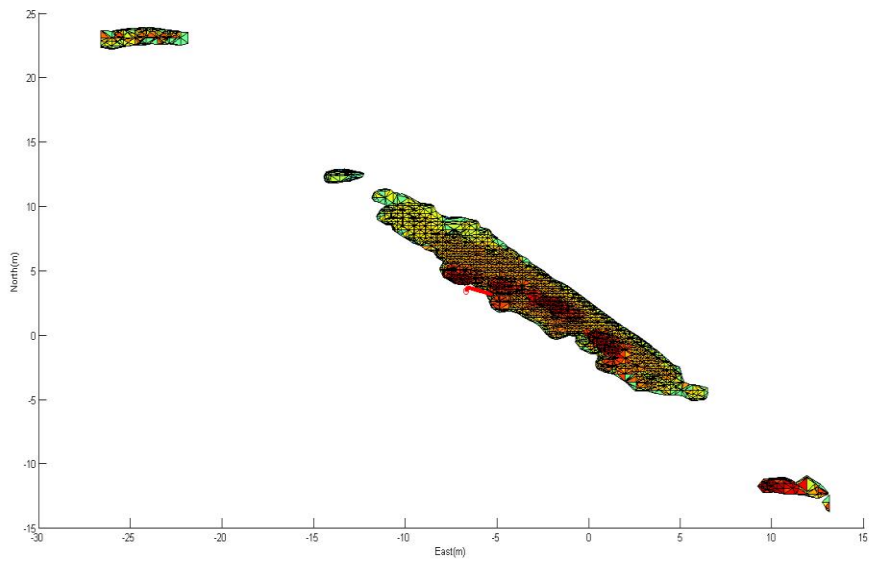
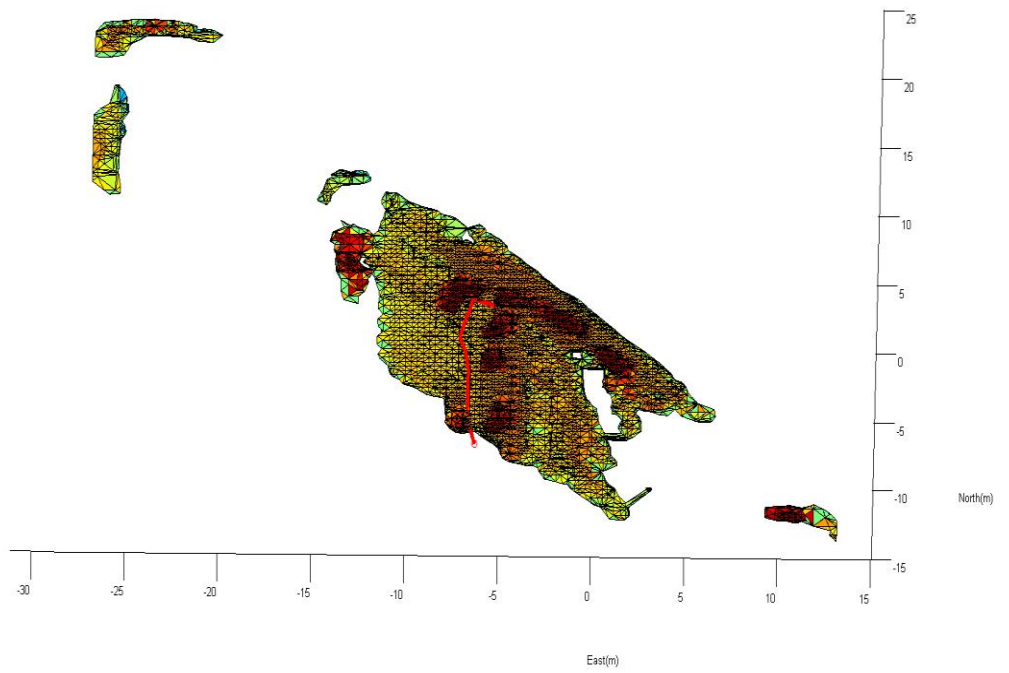


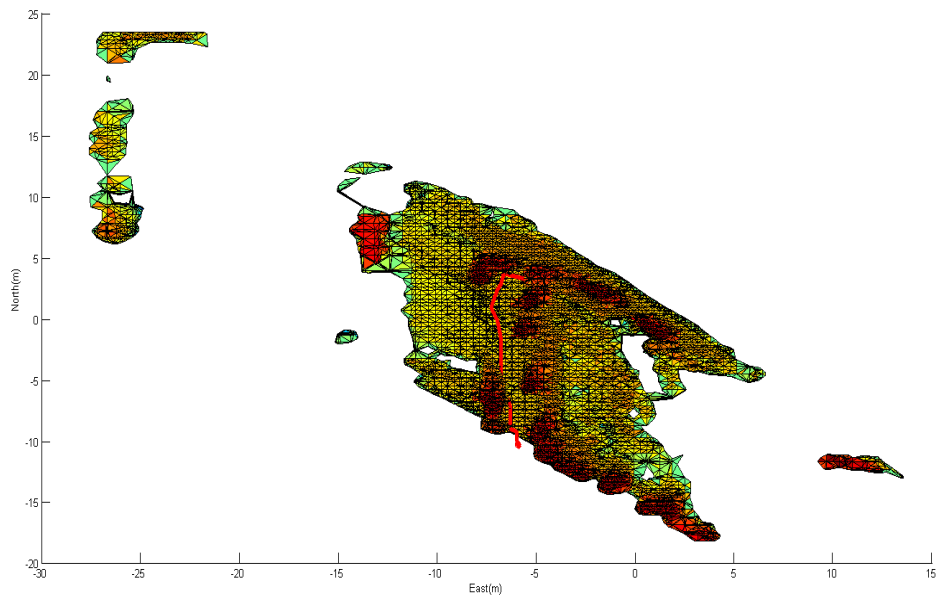
Figure 6.2.7: Orientation Measurements



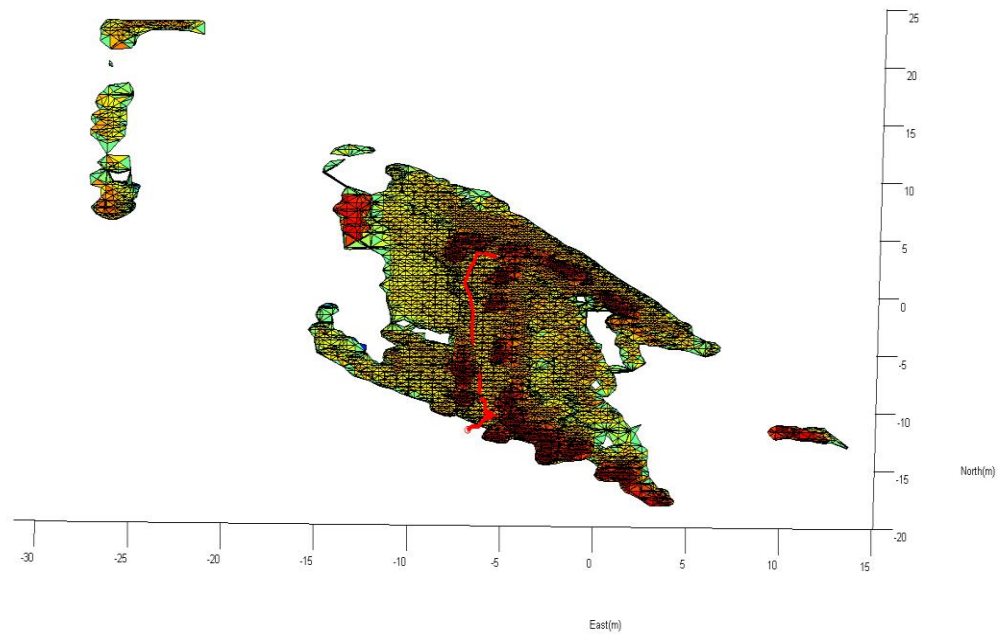
(a) 3D Terrain Representation at $t = 7$ sec



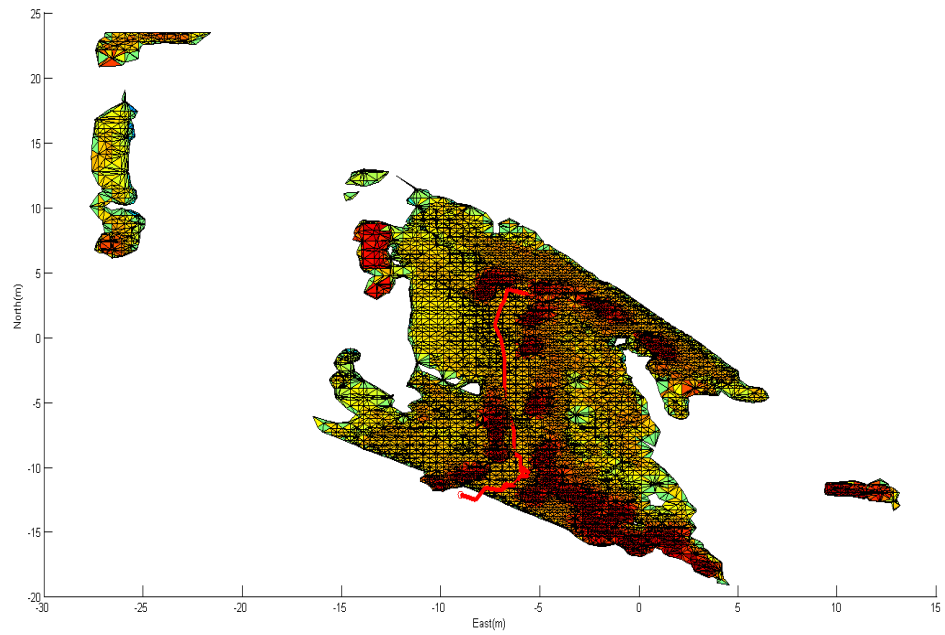
(b) 3D Terrain Representation at $t = 15$ sec



(c) 3D Terrain Representation at $t = 25$ sec



(d) 3D Terrain Representation at $t = 35$ sec



(e) 3D Terrain Representation at $t = 40$ sec

Figure 6.2.8: 3D Terrain Representation at Different Time Intervals (vehicle path shown in red)

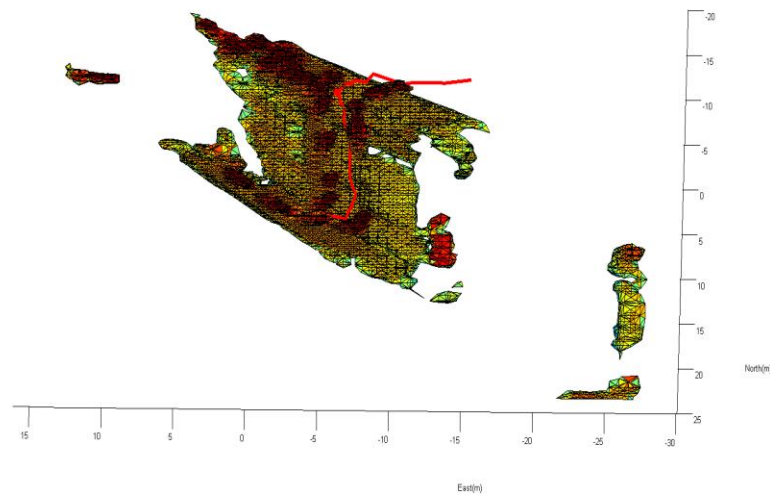


Figure 6.2.9: Overhead View of 3D Terrain Representation with UGV Path (shown in red)

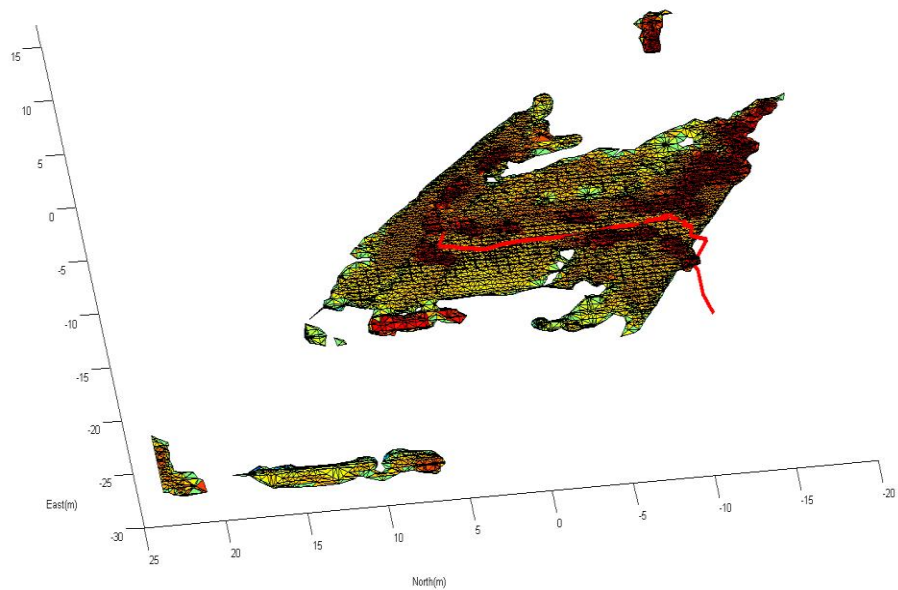


Figure 6.2.10: Oblique View of 3D Terrain Representation

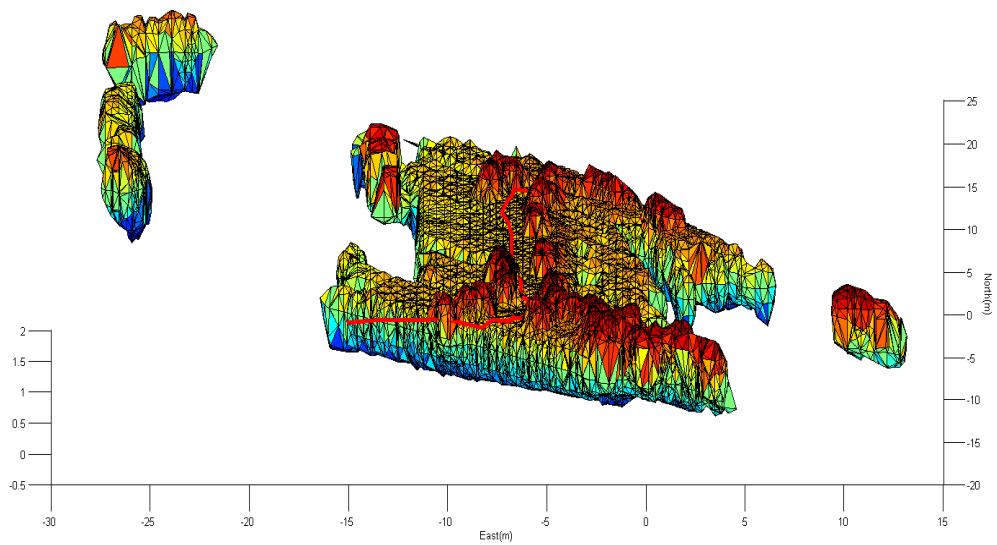


Figure 6.2.11: Alternative View of 3D Terrain Representation

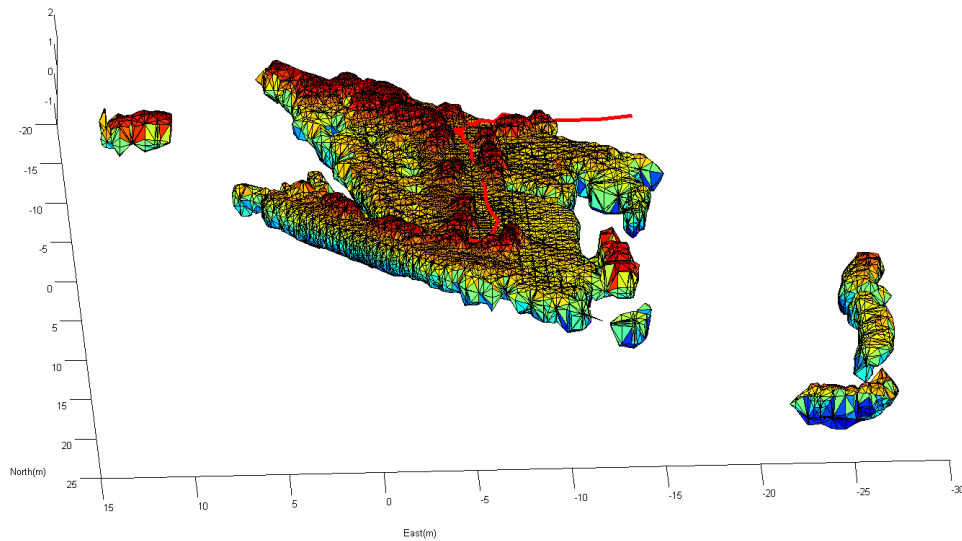


Figure 6.2.12: Alternative View of 3D Terrain Representation

Figure 6.2.4 – 6.2.7 illustrate the GPS, LIDAR and orientation measurements. It includes computing the location and orientation of the vehicle relative to a local NED (inertial) reference frame, and registering the LIDAR data to the NED frame. All the data were recorded with global time stamp but at different sample rates. Therefore, resampling of the data was required in order to perform the analysis. The computations were based on the derivations in the Chapter 3. The LIDAR measurements with path in depicted in red is shown Figure 6.2.4. The LIDAR points were registered to the local NED reference frame using the GPS and orientation data that were resampled at at the same interval as the LIDAR data. Figure 6.2.6 shows the GPS latitude and longitude measurements, which were sampled at approximately 8 Hz. A local NED reference frame was defined at the initial location of the vehicle CM, which was computed from the initial GPS measurement as shown in Figure 6.2.5. The NED position was computed from the GPS and orientaion data, after resampling these data via linear interpolation. Figure 6.2.7 shows the orientation

measurements. As the data were collected from the ground vehicle, there is not much variation in roll and pitch measurements. Figure 6.2.8 illustrate the 3D representation at different time intervals. It shows the build up of 3D terrain as the data are collected during the run. Each figure shows the vehicle path in red and the terrain map that has been constructed from the LIDAR points collected up to that point. Figures 6.2.9 – 6.2.12 show overhead, oblique and alternative views of the 3D terrain representation.

6.3 Closed-Loop Simulation Results

Closed-loop simulations were performed using the UGV simulation model and waypoint controller discussed in Chapter 5. The environment used for the closed-loop simulations was chosen to be equivalent to the location used for the second set of open-loop experiments. The 3D terrain map generated from processed LIDAR data from the second open-loop experiment was used to provide the obstacle avoidance constraints for the path planning algorithm. A starting point and target location were set within the terrain map generated from open-loop measurements, and the receding horizon path planner was used to plan a path for the UGV to travel to the target. Path points from the receding horizon algorithm were passed to the waypoint controller, and the simulation continued until the goal location was achieved.

The receding horizon algorithm was implemented on the set of terrain data generated from the second open-loop experiment. In an initial test, the path planner was employed without the waypoint controller in order to demonstrate the functionality of the receding horizon algorithm. For implementing this algorithm, a goal location B was placed in the scene as shown in Figure 6.3.1. The path planner was run at different points along

the open-loop UGV trajectory from the second open-loop experiment. The purpose of this test was to confirm that the path planning algorithm would select appropriate waypoints to guide the UGV to the target location based on different positions along the open-loop path.

In this implementation, the receding horizon parameters were set so that, for each path planning step, 3 points were planned at a distance of 1 m apart with the objective of navigating the ground robot to the goal location. Figure 6.3.1 represents a sample in which the planned and executed points are shown with a goal location selected at $[-10, -10, 0]$ m. The 3D terrain representation of the environment was produced from open-loop LIDAR measurements, which provided obstacle avoidance constraints for the path planner. The receding horizon algorithm was used to plan waypoints in order for the UGV to achieve the mission, which was to reach point B. The number of planned points in each path planning interval was set to $N_p = 4$, and the number of control points (i.e., the number of points to be executed before replanning) was set to $N_c = 2$. Hence the path planner would plan 3 points and execute 2 of them followed by the planning of 3 points again and executing 2. This process continued until the goal location was achieved.

Figure 6.3.2 provides an overhead perspective of several of the path planning steps used. In each of the plots, the red points indicate the points that the UGV has already traveled through, yellow points denote the set of 3 planned points, and the green points denote the path points that would be executed by the waypoint controller (a subset of the 3 planned points). The values of the other receding horizon parameters are $d = 1$ m, $\varphi_h = 90^\circ$, $\Delta\varphi_h = 15^\circ$, where d is the distance between the planned points, φ_h is the maximum horizontal cone angle (which limits the maximum allowable angle between consecutive

path points), and $\Delta\varphi_h$ is the horizontal angular resolution (which determines the angular spacing between path options).

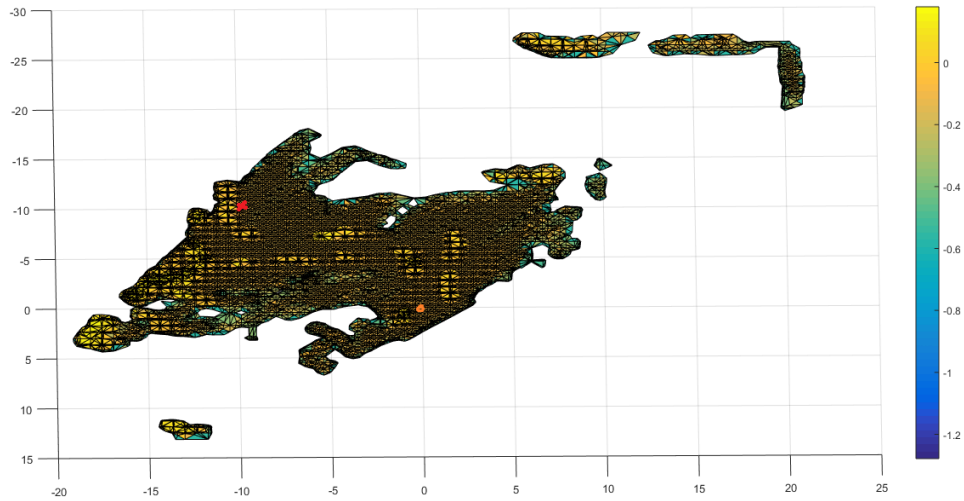
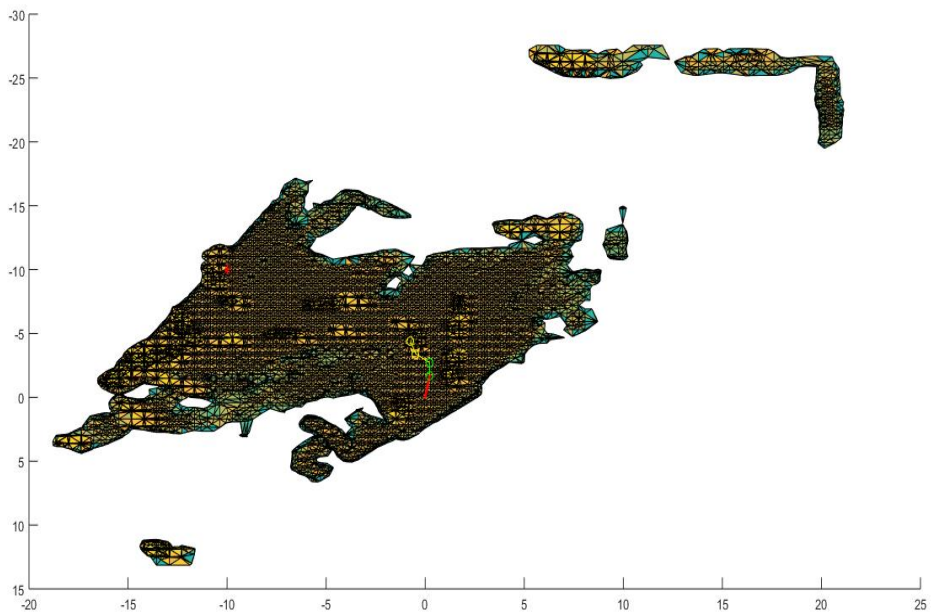
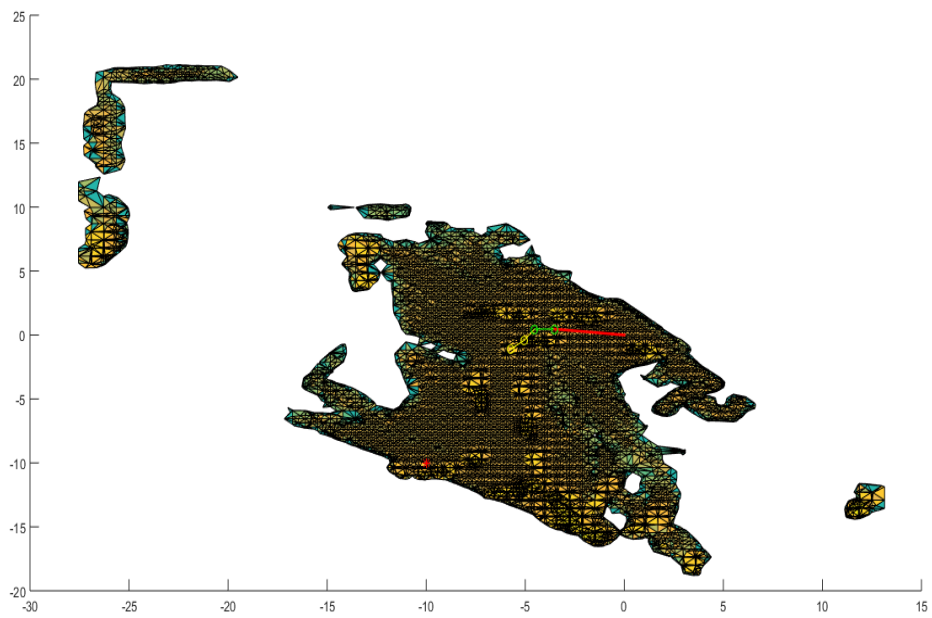


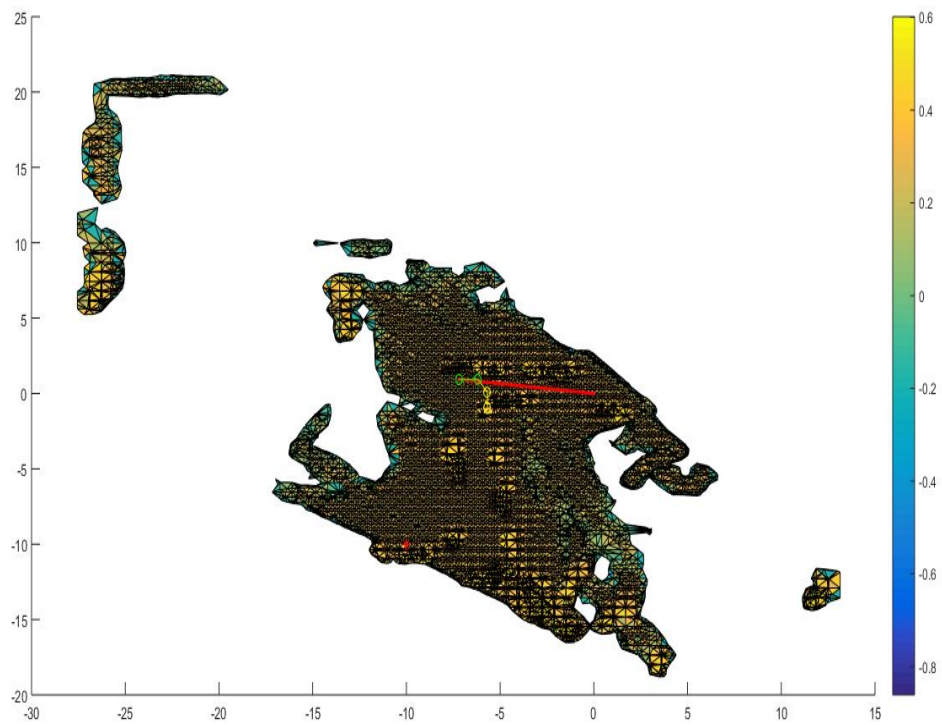
Figure 6.3.1: Overall Path Scenario with Target Set at $[-10, -10, 0]$



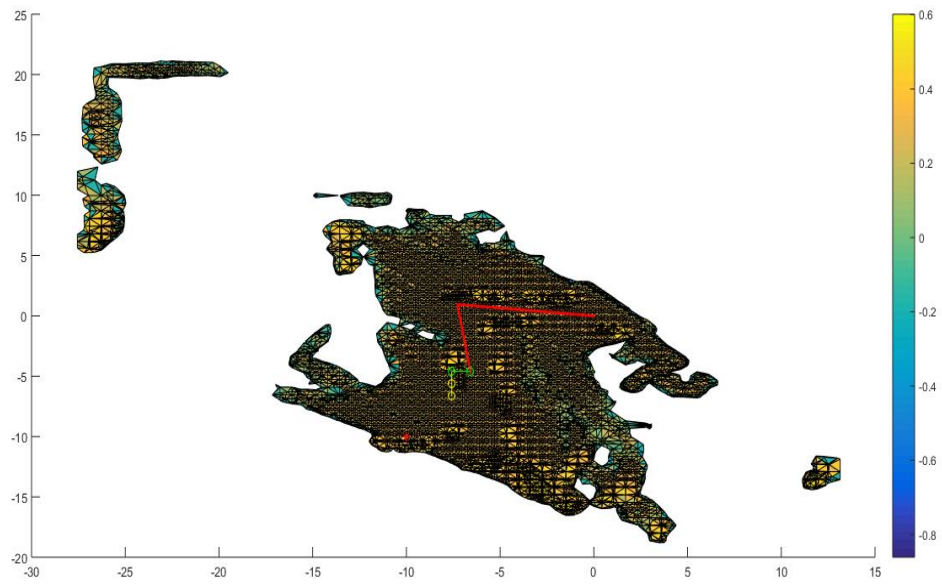
(a) Path Planning Decision at $t = 10$ sec



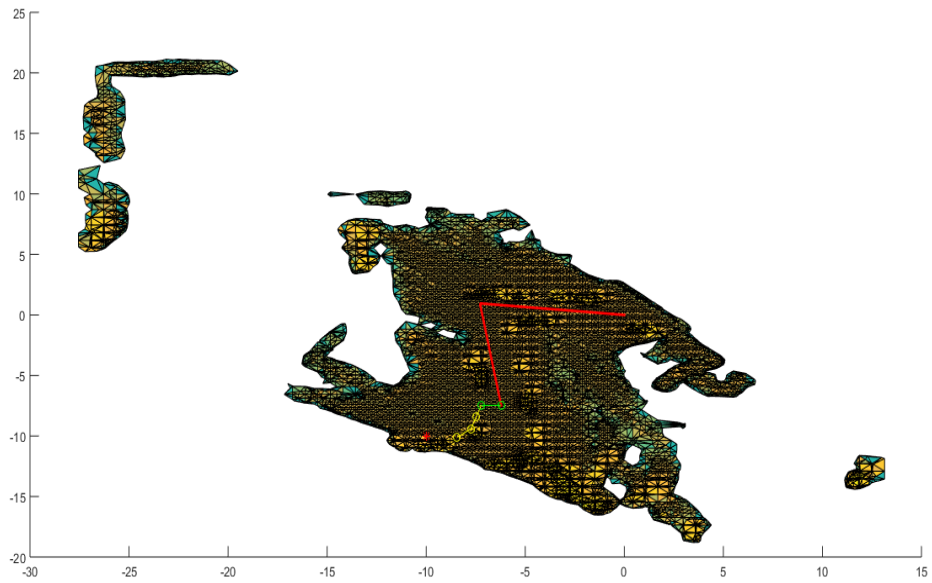
(b) Path Planning Decision at $t = 20$ sec



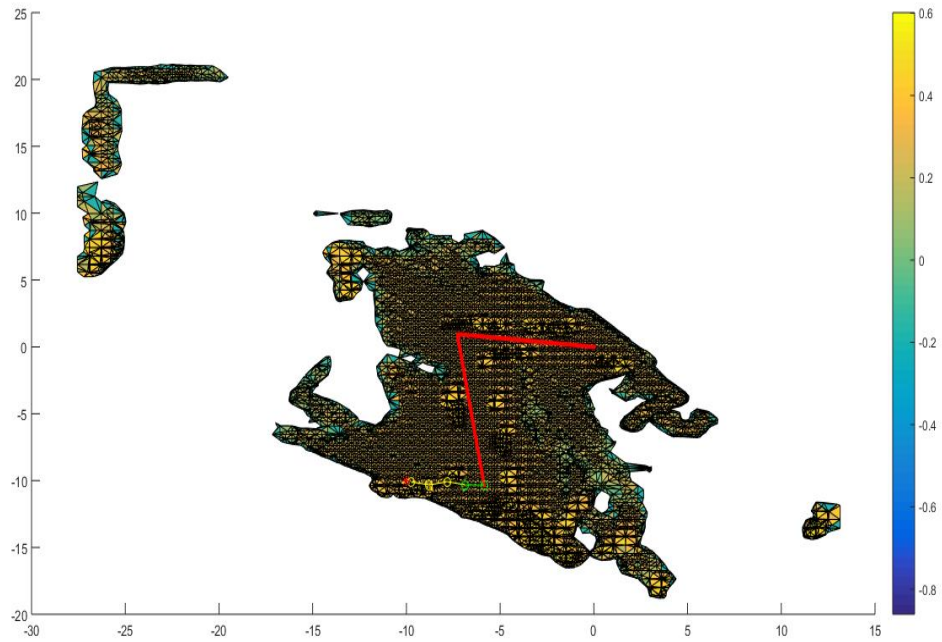
(c) Path Planning Decision at $t = 30$ sec



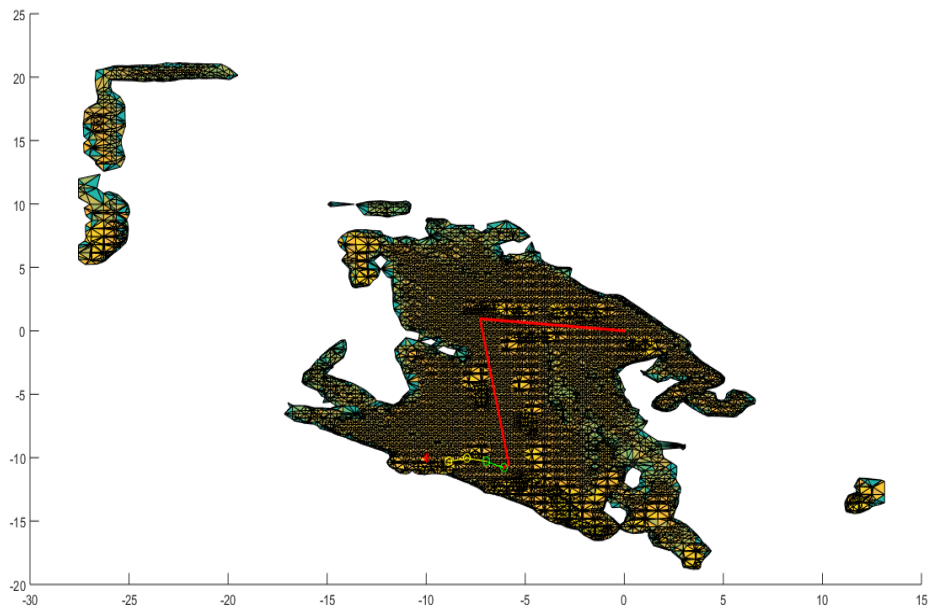
(d) Path Planning Decision at $t = 40$ sec



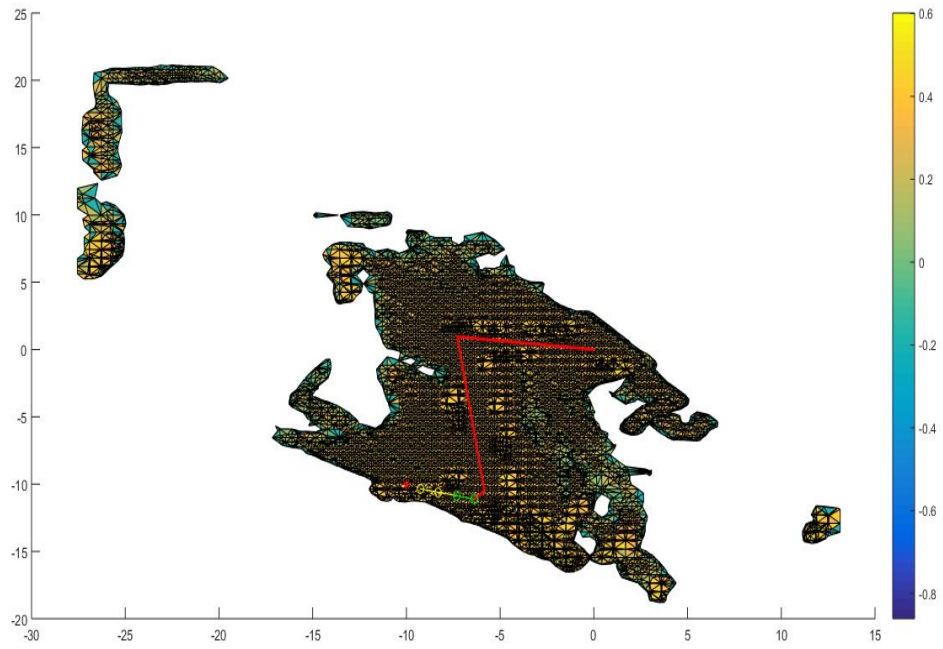
(e) Path Planning Decision at $t = 50$ sec



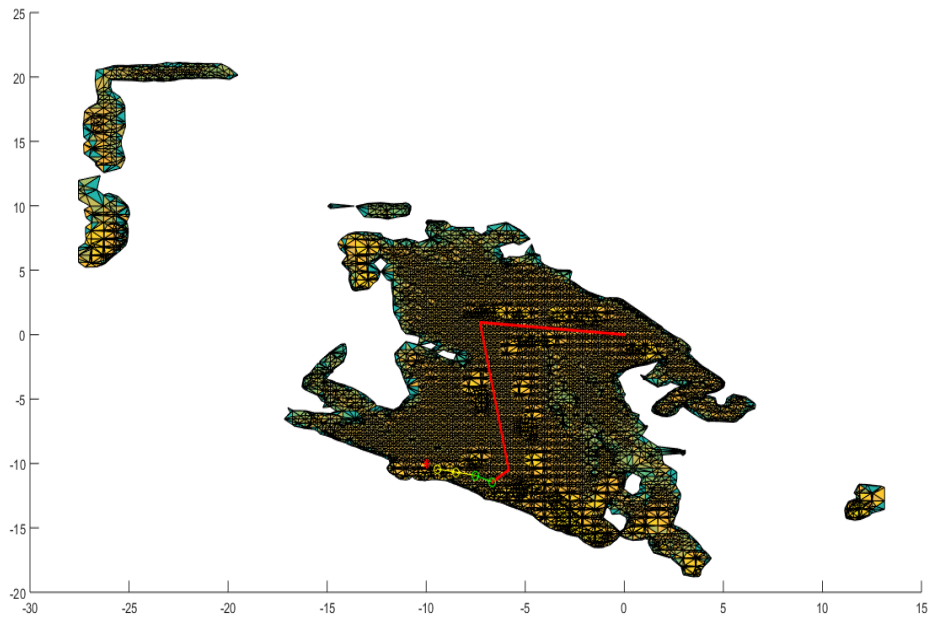
(f) Path Planning Decision at $t = 60$ sec



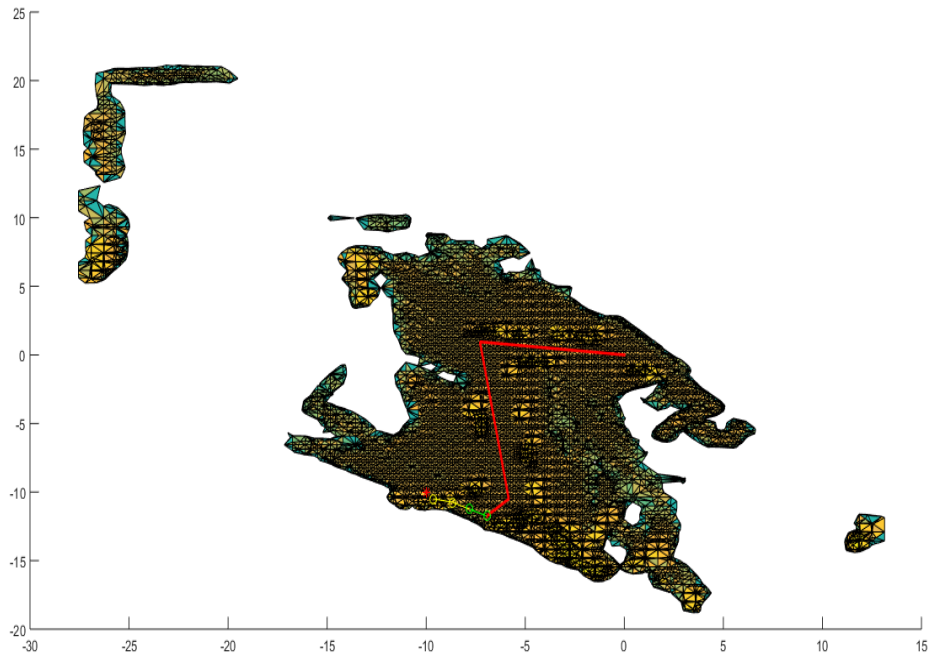
(g) Path Planning Decision at $t = 70$ sec



(h) Path Planning Decision at $t = 80$ sec



(i) Path Planning Decision at $t = 90$ sec

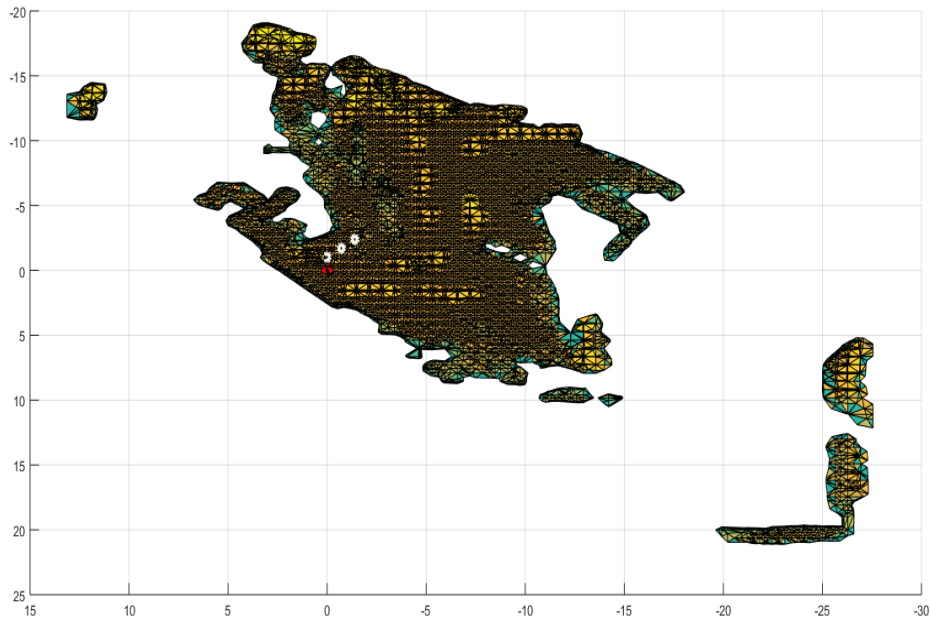


(j) Path Planning Decision at $t = 100$ sec

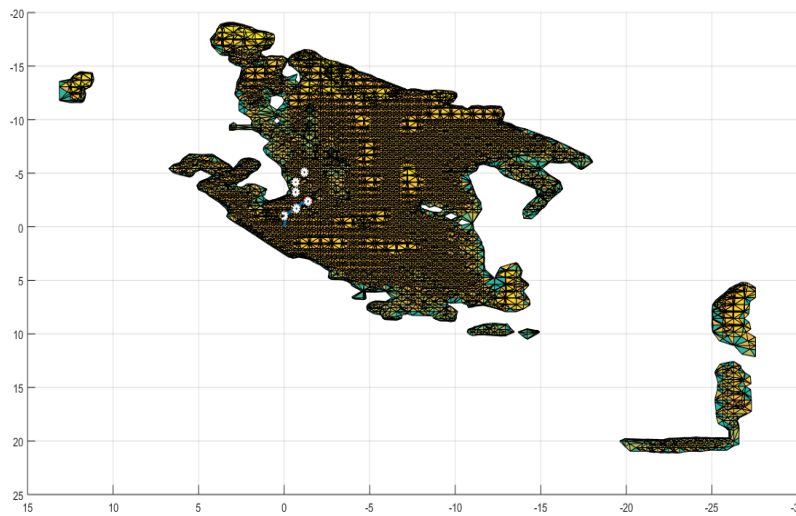
Figure 6.3.2: Receding Horizon Path Planning Decisions at Several Points

The results in Figure 6.3.2 demonstrate that the receding horizon algorithm was functioning properly. A closed-loop simulation was then performed in which a starting location A and a goal location B were selected within the environment. The receding horizon algorithm was used to derive the waypoints that would be required to drive the UGV towards the goal while avoiding obstacles as provided by the 3D terrain map. The waypoint controller was used to control the vehicle so that it passed through the required waypoints. In this example, the starting point A was set to be the origin and the goal location B was set at $[-10, -10, 0]$ m. Figure 6.3.3 shows the simulated UGV path and waypoints at different time intervals during the simulation. In this case, the receding horizon algorithm planned 3 points and 2 of these were executed by the controller before a

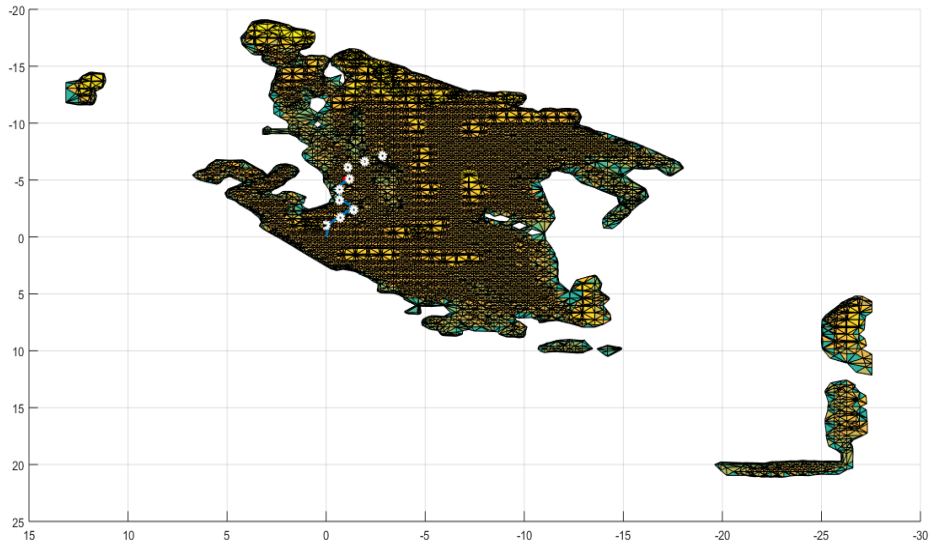
new set of points were planned. The results in Figure 6.3.3 show only the planned points that were followed by the waypoint controller at each time interval.



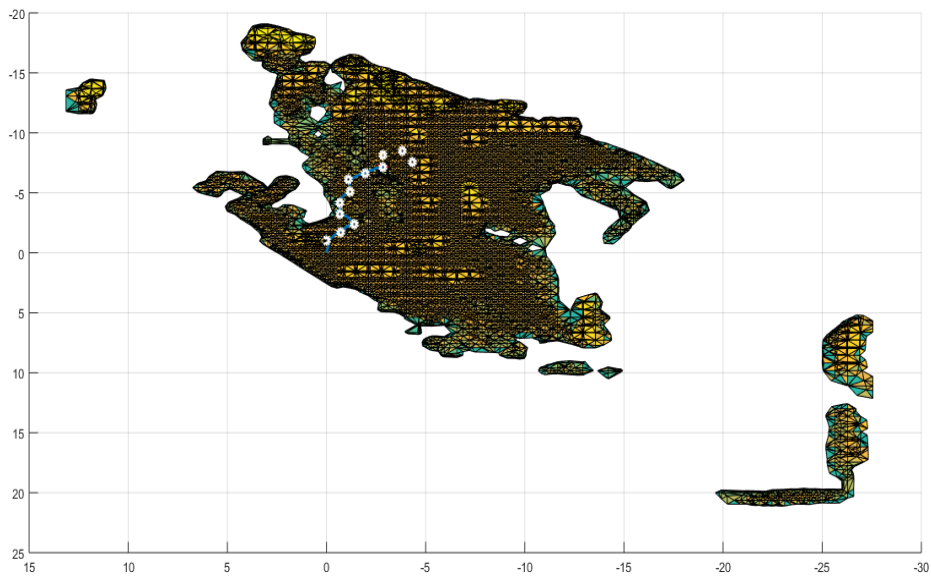
(a) Initial Three Points Planned by RHC to Reach the Target ($t = 10$ sec)



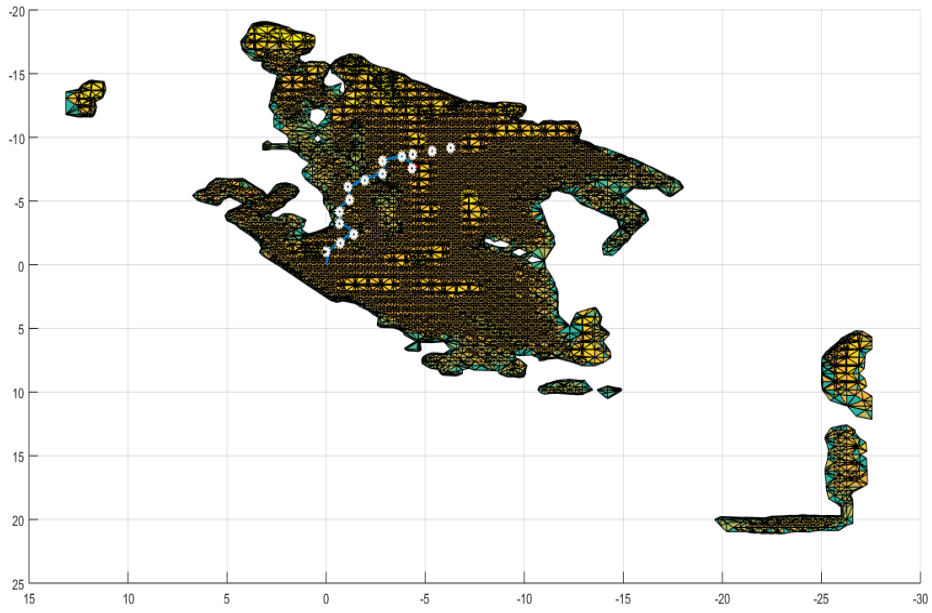
(b) Path Planning at $t = 20$ sec



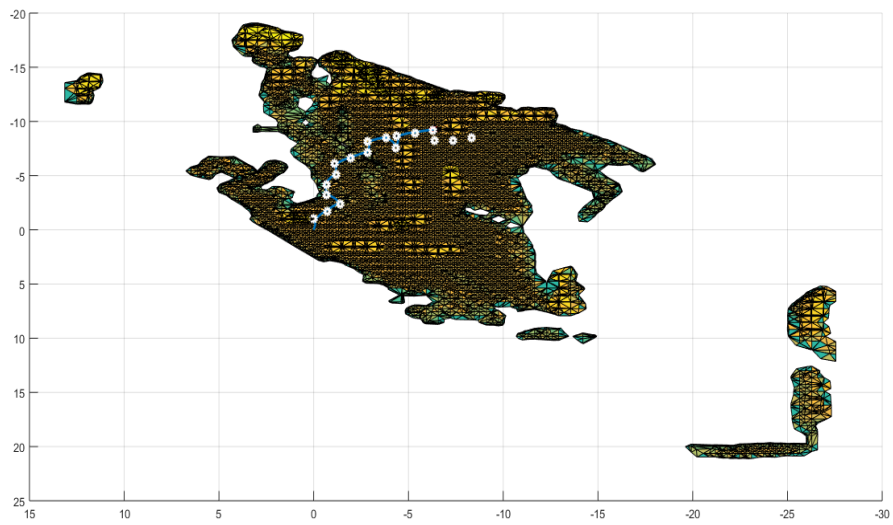
(c) Path Planning at $t = 40$ sec



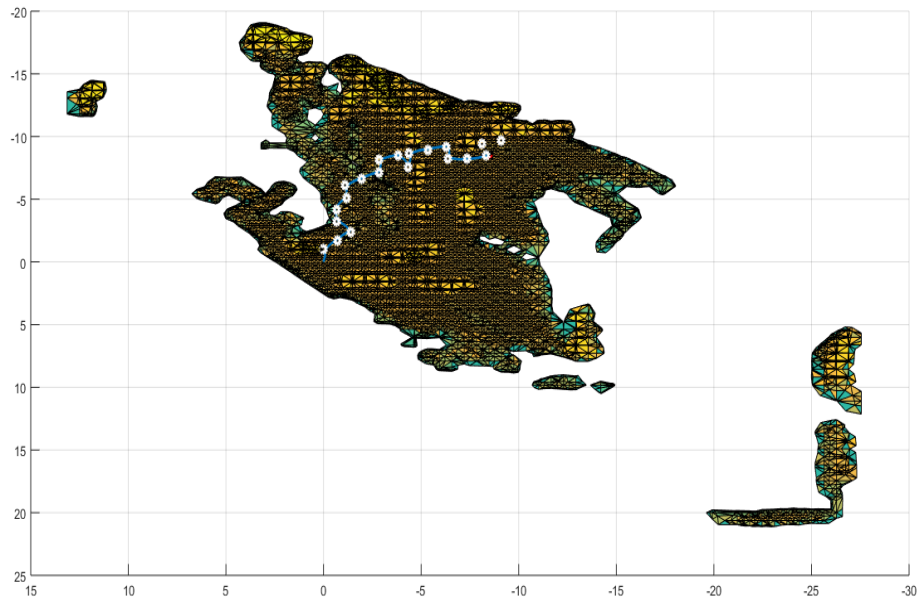
(d) Path Planning at $t = 60$ sec



(e) Path Planning at $t = 80$ sec



(f) Path Planning at $t = 100$ sec



(g) Path Planning at $t = 110$ sec

Figure 6.3.3: Simulated Movement of the UGV through the RHC Points at Various Times.

It is clear from Figure 6.3.3 that the target location was achieved successfully by the simulated UGV following the waypoints generated by the receding horizon path planner. The target was achieved without the UGV colliding with any obstruction in the scene. The yellow terrain regions in the figure show the obstacles in the terrain. White points are the points planned by receding horizon algorithm towards the goal.

Chapter 7

Conclusions

7.1 Concluding Remarks

This thesis has effectively investigated the technical objectives, which were focused on the implementation of guidance, navigation, and control algorithms to enable the autonomous operation of an unmanned ground vehicle (UGV) in complex environments. The details of the hardware and sensors required to implement these algorithms were discussed. Sensor fusion and integration of sensors on the UGV to estimate the vehicle position and orientation and to compute a LIDAR-based point cloud were successfully completed. The main contributions of this work are the testing and implementation of 3D implicit terrain algorithms to represent the environment based on the point cloud generated by processed sensor data and receding horizon path planning algorithms, which incorporated obstacles avoidance constraints derived by these terrain representations. The testing of these algorithms was done experimentally by mounting the sensors on a Corobot UGV and driving the UGV around campus. Finally, a simulation model of this ground vehicle was developed to check the functionality of the controller designed to generate the commands to navigate the vehicle to follow the waypoints provided by the receding horizon path planning algorithm. The results presented in this work, which are based on open-loop experiments and closed-loop simulation, demonstrate the successful systems level integration of these algorithms into a GNC system and the performance of this GNC system for the UGV application.

7.2 Future Recommendations

Although the results presented in this work have demonstrated the effectiveness of the 3D terrain mapping and receding horizon path planning algorithms, this research could be further developed in a number of ways.

- 1) The full closed-loop control system designed in this thesis can be implemented experimentally on an unmanned ground vehicle.
- 2) These GNC algorithms should be investigated for real-time implementation, which is particularly important for fixed wing UAVs that do not have the ability to stop and wait for data to be processed. Towards this goal, the learning algorithm proposed in [37] can be further researched as an alternative to the 3D implicit terrain algorithms for representing urban environments. This learning algorithm is recursive and fast, which making it ideal for real-time applications.
- 3) This research can be extended to consider moving obstacles, which requires the development of a different class of reactive collision avoidance control laws.
- 4) The testing and implementation of these algorithms can be extended to quadcopters or micro air vehicles.

The requirements for real-time applications present interesting trade-offs to be investigated between the onboard sensor payload and processing that can be accommodated by a particular vehicle with the complexity of GNC algorithms that can be implemented in real time. The challenging task of providing robust navigation solutions for an unmanned vehicle in situations where obstacles are present using environmental mapping and path planning would benefit a diverse range of applications requiring UAS navigation in remote, distant and undiscovered territories.

References

- [1] Valavanis, K., “Advances in unmanned aerial vehicles: state of the art and the road to autonomy.” *Intelligent Systems, Control and Automation: Science and Engineering* 33, 2007.
- [2] Dao, James, Andrew C.Revkin. “Machines are filling in for troops.” *New York times*, (16 April 2002), <http://www.cybracero.com/articles/machinetroops1.html>.
- [3] Pike, John, “RQ-1 Predator MAE UAV,” <http://www.fas.org/irp/program/collect/predator.htm>, (Federation of American Scientists, 6 November 2002).
- [4] Pike, John, “RQ RQ-4A Global Hawk,” http://www.fas.org/irp/program/collect/global_hawk.htm, (Federation of American scientists, 6 November 2002)
- [5] IRobot, “Romba Robotic Floor Vac,” <http://www.irobot.com/consumer/>
- [6] Friendly Robotics, Inc., “RoboMower,” <http://www.robomowerusa.com/>.
- [7] Oliver Montenbruck, Takuji Ebinuma, E. Glenn Lightsey, Suuny Leung. “A Real-time Kinematic GPS sensor for spacecraft relative navigation”, *Aerospace Science and technology* 6 (2002) 435-449.

- [8] T.Puls, T.Kemper, R.Kuke and A.Hein."GPS-based Position Control and Waypoint Navigation System for Quadrocopters" IEEE International Conference on Intelligent Robots and Systems October 11-15, 2009 St. Louis, USA
- [9] J.A.V.N.T.S. Center. "Vulnerability assessment of the transport infrastructure relying on the global positioning system." Office of the Assistant Secretary for Transportation Policy, U.S. Department of Transportation, Report, August 2001.
- [10] M.Morgado, P.Batista, P.Oliveira, and C.Silvestre. "Position and velocity USBL/IMU Sensor-based Navigation Filter." Institute for Sysytems and Robotics, 1049-001, Lisbon, Portugal.
- [11] David Vincet. "Accurate Position Tracking Using Inertial Measurement Units", PNI White Paper in collaboration with Miami University, February 2013.
- [12] Brandon McCarron. "Low-Cost IMU implementation via Sensor Fusion Algorithms in the Arduino Environment." California Polytechnic State University, San Luis Obispo
- [13] Md. Syedul Amin, Mamun Bin Ibne Reaz, Salwa Sheikh Nasir, Mohammad Arif Sobhan Bhuiyan, Mohammad Alauddin Mohd. Ali. "A Novel Vehicle Stationary Detection Utilizing Map Matching and IMU sensors." The Scientific World Journal Volume 2014, Article ID 597180. September 7, 2014
- [14] D.Hazry, M.Sofian, A.Zul Azfar. "Study of Inertial Meaurement Unit Sensor." Proceedings of the International Conference on mna Machine Systems (ICoMMS). October 11-13 2009, Batu Ferringhi, Penang, Malaysia.

- [15] S. Sukakarieh, "Aided Inertial Navigation Systems for Autonomous Land Vehicles," Phd thesis', Australian centre for Field Robotics, the University of Sydney, 1999.
- [16] Alison K. Brown, "Test Results of a GPS/Inertial Navigation System Using Low cost MEMS IMU", NAVSYS corporation, 14960 Woodcarver Road, Colorado Springs, CO 80921 USA.
- [17] Sandy Kennedy, Hamilton Jason, Martell Hugh. "GPS/INS integration with the iMAR-FAS IMU". XXIII FIG Congress. Munich, Germany, October 8-13, 2006.
- [18] Yueming Zhao. "GPS/IMU Integrated System for Land Vehicle Navigation based on MEMS" Thesis. Royal Institute of technology (KTH). Division of Geodesy and GEoinformatics. 10044 Stockholm Sweden. September 2011.
- [19] Hone-Jay Chu, Gurang-Je Tsai, Kai-Wei Chiang, Thanh-trung Duong. "GPS/MEMS INS Data Fusion And Map Matching in urban Areas" Department of Geomatics, National Cheng Kung University, No. 1, University Road, Tainan 701, Taiwan. August 23 2013.
- [20] Bruno Sinopli, Mario Micheli, gianluca Donato, T.John Koo. "Vision Based Navigation for an Unmanned Aerial vehicle". International Conference on Robotics and Automation. Seoul Korea. May 21-26, 2001.
- [21] Jixian Zhang. "Multi-source remote sensing data fusion: status and trends" international Journal of image and Data fusion. Vol. 1, No.1, March 2010, 5-24.
- [22] Shifei Liu, Mohamed Maherr Atia, Tashfeeb B. Karamat, Aboelmaged Noureldin. "A LIDAR-aided indoor Navigatio system for UGVs". The Journal of Navigation (2015), 68, 253-273.

- [23] Adams, M.D. "LIDAR design, use, and calibration concepts for correct environmental detection" IEEE Transactions on Robotics and Automation, 16(6), 753-61
- [24] Barber, D., Mills, J. and Smith-Voysey, S. "Geometric validation of a ground-based mobile laser scanning system. ISPRS Journal of Photogrammetry and Remote Sensing, 63(1), 128-141
- [25] Diosi, A. and Kleeman, L. "Uncertainty of Line Segments Extracted from Static SICK PLS Laser Scans" Proceedings of Australian Conference on Robotics and Automation. 2003.
- [26] Csatho B., T. Schenk. "Multisensor data fusion for automatic scene interpretation". ISPRS Intern. Archives, vil.32, part3/1, pp. 429-434.
- [27] Csatho. B, W. Krabill, J.Lucas, T.Schenk. "A multisensory data set of an urban and coastal scene. ISPRS intern. Archives, vol.32, part 3/2, pp. 15-21.
- [28] Wozencraft Jennifer, Millar David. "Airborne Lidar and integrated Technologies for coastal Mapping and nautical charting." Marine Technology Society Journal. Volume 39, number 3, fall 2005, pp.27-35(9)
- [29] Brent Schwarz. "LIDAR Mapping the world in 3D" Industry Perspective and Technology Focus.
- [30] Luke Wallace, Arko Lucieer, Christopher Watson, Darren Turner. "Development of a UAV-Lidar System with Application to forest Inventory." Remote Sensing Journal. Volume 4, issue 6. May 25 2012

- [31] Subramaniam Ramasamy, Roberto Sabatini, Alessandro Gardi. "Avionics Sensor Fusion for Small Size Unmanned Aircraft Sense and Aviod" IEEE. RMIT University-SAMME, Melbourne, Australia.
- [33] S. Lenser, M. Velso. "Visual Sonar: Fast obstacle Avoidance using Monocular Vision." IEEE/RSJ international Conference on Intelligent Robots and Systems. 886-891, 2003.
- [34] Arthur Richards, John Bellingham, Michael Tillerson, Jonathan How. "Coordination and control of multiple UAVs." AIAA Guidance, Navigation and Control Conference and Exhibit. Monterey, California. August 8, 2002.
- [35] Stefan Mitsch, Khalil Ghorbal, Andre Platzer. "On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles." Carnegie Mellon University, Computer Science Department, Pittsburgh PA 15213 USA.
- [36] P. Binev, A. Cohen, W. Dahmen, R. Devore, V. Temlyakov. "Universal Algorithms for learning theory, Part-1: Piecewise Constant Functions," Journal of machine Learning Research, Vol. 6, September 2005, pp. 1297-1321
- [37] Richard J. Prazhenica, Matthew Hielsberg, Robert C. Sharpley, Andrew J. Kurdila. "3-D Implicit Terrain Mapping and Path Planning for Autonomous MAV Flight in Urban Environments," American Institute of Aeronautics and Astronautics.
- [38] A. Howard, M.J Mataric, G.S Sukhatme. "An incremental Self-Deployment Algorithm for mobile Sensor Networks, autonomous robots, special issue on intelligent Embedded Systems, 13(2), September 2012, pp-113-126

- [39] S. Florczyk. "Robot Vision based Indoor Exploration with Autonomous and Mobile Robots, WILEY-VCH Verlag GmbH & co. KGaA, Weinheim, 2005.
- [40] Mayne D., Rawlings J., Rao C., "Constrained Model Predictive Control Stability and Optimality," *Automatica*, Vol. 36, 2000.
- [41] Y. Kuwata, A. Richards, T. Schouwenaars, J. How. "Decentralized Robust Receding Horizon Control for Multi-Vehicle Guidance." *Proceedings of the 2006 American Control Conference*, MN, June 2006, pp. 2047-2052
- [42] Y. Kuwata, J. How. "Three Dimensional Receding Horizon Control for UAVs" *AIAA Guidance, Navigation and control Conference and Exhibit*, Providence, RI, August 2004.
- [43] Richard J. Prazenica, Andrew J. Kurdila. "Vision Based Receding Horizon Control for Micro Air Vehicles in Urban Environments." *Journal of Guidance, Control and Dynamics*. February 12, 2007.

