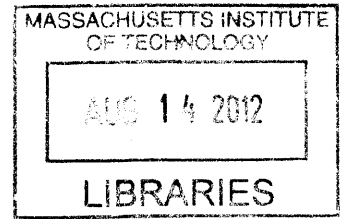


# Development of a Robot Localization and Environment Mapping System

by

Cynthia Dawn Walker Panas

Sc.B. Mechanical Engineering  
Massachusetts Institute of Technology, 2007



**ARCHIVES**

Submitted to the Department of Mechanical Engineering  
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2012

© Massachusetts Institute of Technology 2012. All rights reserved.

Author .....  
Department of Mechanical Engineering  
January 6, 2012

Certified by .....  
Kamal Youcef-Toumi  
Professor of Mechanical Engineering  
Thesis Supervisor

Accepted by .....  
David E. Hardt  
Professor of Mechanical Engineering  
Graduate Officer



# Development of a Robot Localization and Environment Mapping System

by

Cynthia Dawn Walker Panas

Submitted to the Department of Mechanical Engineering  
on January 6, 2012, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Mechanical Engineering

## Abstract

The intent of this research is to develop a robust, efficient, self-contained localization module for use in a robotic liquefied petroleum gas (LPG) tank inspection system. Inspecting large LPG tanks for defects is difficult, expensive and energy intensive. Replacing the human inspectors with a robotic inspection system will make the inspection process faster, less expensive, more reliable and safer. The sensing platform designed in this work can collect data about the environment and track the robotic inspection platform, recording the defect locations. It consists of a two axis gimballed sensing platform with a single point distance sensor placed in the manhole of the tank. A collection of algorithms were developed to use in conjunction with the sensing platform to collect and process the 3D data into a map of the environment. The algorithm's main feature is a robust and efficient method of segmenting and fitting data to a right capped cylinder that is faster and more robust to noise than current methods. The improved performance comes from a unique combination of object shape knowledge, the Gauss image and 3D histogram techniques which achieve accurate segmentation without iteration. The hardware and software were demonstrated to function robustly in a noisy environment. The unique ability of the system to work in an LPG tank allows it to be integrated into a robotic inspection system that can remove the majority of the cost and risk associated with LPG tank inspection.

Thesis Supervisor: Kamal Youcef-Toumi  
Title: Professor of Mechanical Engineering





## Acknowledgments

This work was made possible through the topic suggestion and financial support of the Qatar National Research Fund. I would like to thank Dr Uvais Qidwai for his insight into the project details and continued work on the funding paperwork.

I would like to thank my Professor, Kamal Youcef-Toumi, for his insight into difficult problems and his trust in me to carry out my work as I believe necessary. I would also like to thank him for his support during my father's illness and eventual passing. Professor Youcef-Toumi was very understanding and his flexibility with my work made that horrible time easier.

I would also like to thank my fellow graduate students, Andreas Schuh, Amith Somanath and Ethan Heller for helping with the equipment and taking the time to offer advice on a range of subjects when I was having difficulty finding a solution.

I also want to acknowledge and express my gratitude to my family for their personal support throughout my graduate experience. I want to thank my mother and sister for their unceasing willingness to listen to the complicated technical details of my work despite their lack of understanding. I also want to give a special thanks to my inlaws for their willingness to let me sit in their dining room for hour upon hour while I was writing my thesis. Thank you for continually encouraging and feeding me during this process.

Finally, it is unquestionably necessary to give huge thanks for the personal and technical support of my husband and fellow graduate student, Robert Panas. He spent countless hours helping me come up with ideas, fix bugs in my code and fabricate the hardware. Without him this would not have been possible.

This thesis is dedicated to the memory of my late father Donald Walker. His influence and unending support made this thesis possible. I will love and miss him forever. May he rest in peace.



# Contents

<b>1</b>	<b>Introduction</b>	<b>21</b>
1.1	Motivation . . . . .	21
1.2	Research Focus . . . . .	22
1.3	Background . . . . .	22
1.3.1	LPG/LNG . . . . .	22
1.3.2	Conventional Inspection . . . . .	24
1.3.3	Robotic Inspection . . . . .	25
1.3.4	Proposed System Design . . . . .	26
1.4	Prior Art . . . . .	27
1.4.1	Localization Method Prior Art . . . . .	27
1.4.2	Data Processing Prior Art . . . . .	29
1.5	Scope . . . . .	31
<b>2</b>	<b>Sensor Platform Design</b>	<b>33</b>
2.1	Sensor Platform Requirements . . . . .	33
2.2	Distance Sensing Technologies . . . . .	35
2.2.1	Vision Sensors . . . . .	35
2.2.2	GPS . . . . .	35
2.2.3	Acoustic . . . . .	36
2.2.4	Electromagnetic Sensors . . . . .	37
2.2.5	Sensor Technology Conclusion . . . . .	40
2.3	Sensor Layout . . . . .	41
2.3.1	On Robot . . . . .	42

2.3.2	Off Robot . . . . .	46
2.3.3	Concept Comparisons . . . . .	52
2.3.4	Concept Selection . . . . .	54
<b>3</b>	<b>Mapping Algorithm</b>	<b>57</b>
3.1	Algorithm Requirements . . . . .	57
3.2	Algorithm Overview . . . . .	60
3.2.1	Data Collection and Normal Calculation Steps . . . . .	65
3.2.2	Segmentation . . . . .	76
3.2.3	Fitting . . . . .	83
3.3	Algorithm Advantages and Limitations . . . . .	86
<b>4</b>	<b>Results</b>	<b>87</b>
4.1	Evaluation Through Simulation . . . . .	87
4.2	Prototype Hardware . . . . .	91
4.2.1	Prototype Tank . . . . .	92
4.2.2	Prototype Operating Code . . . . .	93
4.3	Evaluation Through Hardware Testing . . . . .	93
<b>5</b>	<b>Conclusion</b>	<b>97</b>
5.1	Future Work . . . . .	98
5.1.1	Hardware Future Work . . . . .	98
5.1.2	Software Future Work . . . . .	99
5.1.3	Inspection System Future Work . . . . .	100
<b>A</b>	<b>Matlab Files</b>	<b>101</b>
A.1	Main Tank Mapping File . . . . .	102
A.2	Cylinder Fitting Function . . . . .	107
A.3	Plotting Function . . . . .	115
A.4	Plane Fitting Function . . . . .	120
A.5	Circle Fitting Function . . . . .	121
A.6	Normal Collection Simulation Function . . . . .	122

A.7	Distance Sensor Simulation Function . . . . .	128
A.8	Tip Tilt Function . . . . .	131
<b>B</b>	<b>Labview Code</b>	<b>133</b>
B.1	Main Function . . . . .	134
B.2	Point of Interest Calculation subVI . . . . .	136
B.3	Neighborhood Property Calculations subVI . . . . .	137
B.4	Neighbor Location Calculations subVI . . . . .	138
B.5	Radian to Quadrature Count and Quadrature Count to Radian subVIs	139
B.6	Motor Movement subVIs . . . . .	140
B.7	Read Distance subVI . . . . .	141
<b>C</b>	<b>Outlier Removal Method</b>	<b>143</b>



# List of Figures

1-1	LPG Storage Tanks. (a) Dominion Cove Point LNG facility located in Lusby, Maryland. Photo courtesy of Dominion Corporation [1]. (b) Two LPG storage tanks. Photo courtesy of Lusas Corporation [2]. (c) Row of LPG storage tanks. Photo courtesy of Gauging Systems Inc [3].	23
1-2	LPG Tank Cross-section. Multilayer construction of steel plating, insulation and support walls. Domed carbon steel top has suspended aluminum deck. Concrete base layers and support walls make external inspection impossible. . . . .	23
1-3	Inspection Robots. (a) External crawling inspection robot [4]. (b) Maverick, a remote-controlled, submersible robot for inspection of gasoline tanks [5]. (c) Neptune, a fuel storage tank inspection robot [6]. (d) ICM climbing robot with installed NDT sensors [7]. (e) Techcorr cleaning and inspection robot [8]. . . . .	25
1-4	Proposed Design of Inspection System. Internal mobile inspection robot scans inside surfaces of tank for defects. Localization sensor platform locates and tracks position of inspection robot. Controlling computer is located outside the tank, away from the hazardous environment inside the tank. . . . .	26
1-5	Triangulation. Distance from sensor creates a sphere(or circle when in 2D) of possible locations of the point of interest. The intersection of the spheres(or circles) pinpoints the location of the point of interest. .	28

- 2-1 Concept 1: A set of scanning distance sensors on the robot in a plane parallel to floor measure to the walls and a drop in locating feature. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed arrows represent the measurements taken by the sensors and the dashed line circles show the possible locations of the robot based on the measurements. The circle intersections are the possible locations of the robot. These possible locations are narrowed down to the actual location based on the integration of previous data. . . . . 43
- 2-2 On Robot Concept 2: A set of scanning distance sensors on the robot in a plane parallel to floor with 1 single point distance sensor measuring to a reflector in the manhole. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed arrows represent the measurements taken by the sensors and the dashed line circles show the possible locations of the robot based on the measurements. The circle intersections are the possible locations of the robot. The location is finalized by the measurement to the reflector. 44
- 2-3 Concept 3: A set of single point distance sensors on the robot in a plane parallel to floor with 1 single point distance sensor measuring to a reflector in the manhole. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed arrows represent the measurements taken by the sensors and the dashed line circles show the possible locations of the robot based on the measurements. The circle intersections are the possible locations of the robot. The location is finalized by the measurement to the reflector. 45



2-4 Concept 4: A separate sensor module consisting of a set of scanning distance sensors dropped into the tank measuring the distance to walls and robot. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot, while the red dot represents the sensor platform. The dashed arrows represent the measurements taken by the sensors. . . . . 46

2-5 Concept 5: A physical tether to the robot from the manhole. The orientation and the length of the tether determine position. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed line circle shows the possible locations of the robot based on the length of the tether. The two angle measurements determine where the robot is on the circle defined by the tether length. . . . . 48

2-6 Concept 6: A local GPS-like system consisting of distance measuring sensors placed on walls by robot as it heads into tank. The large black circle represents the tank walls. The small black circle with a central dot is the manhole. The larger black dot is the center of the tank. The green rectangle represents the robot. The dashed line circles show the possible locations of the robot based on the distance measurements from each sensor. The circles' intersection is the location of the robot. 49

2-7	<p>Concept 7: A local GPS-like system consisting of distance measuring sensors on a expanding scaffolding hanging from the manhole. The large black circle represents the tank walls. The small black circle with a central dot is the manhole. The larger black dot is the center of the tank. The green rectangle represents the robot. The orange triangle represents the expanding network with sensors on each corner. The dashed line circles show the possible locations of the robot based on the distance measurements from each sensor. The circles' intersection is the location of the robot. . . . .</p>	50
2-8	<p>Concept 8: A sensor module consisting of a single point distance sensor and two axis angle measurement devices placed in the manhole. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The measured distance and the angle of the measurement fully define the location of the robot. Measurements can also be taken to the walls to collect data for a map of the tank. .</p>	51
2-9	<p>Two surveying total robotic systems. Left: Trimble S8 surveying total station [9]. Right: Leica TPS1200+ surveying total station[10]. . . .</p>	54
3-1	<p>How the tank inspection application drives the algorithm requirements. The three application needs of minimizing time, being tolerant of technician variation and error and being tolerant of noise induced by the environment drive the six algorithm requirements of minimizing iteration, minimizing the amount of data, determining the parameters automatically, being sensor location and orientation independent, outputting an easy to understand tank map and being robust to noise. .</p>	58

3-2	Right capped cylinder with tank and sensor coordinate frames. The cylinder coordinate frame is designated by $XYZ$ where $Z$ is aligned with the center axis of the cylinder. The sensor has two coordinate frames, $\theta\phi z$ and $xyz$ , where $z$ is aligned with the starting direction of the sensor for both. . . . .	61
3-3	Resulting data from different sensor platform orientations upon data collection startup. Tank rotation is dependent upon sensor starting orientation. . . . .	62
3-4	Algorithm processing flow. Process movement is shown by black arrows Note the repeating steps within the data collection process. Data passed from one step to the next is shown by large arrows. . . . .	64
3-5	Possible layouts for neighborhood points collection. a) a checkerboard layout with points evenly distributed along both axes b) a circular layout with points evenly distributed around the circle perimeter. . . . .	66
3-6	Standard Deviation in Normal Angle, STD, vs the number of points (N) and characteristic length (L) of a circular point layout and square checkerboard layout for noise standard deviation of 0.02. The equations for the planar fits are shown in Equ. (3.7) and Equ. (3.8). . . . .	67
3-7	Encoder resolution effect on distance measurement error. The actual position of the axis can be anywhere within the step size of the encoder resulting in an angular error and mis-measurement of the distance, $e_e$	69
3-8	Simulated average error in cylinder fits vs the neighborhood radius as a fraction of either the diameter or height, whichever is smallest. This was simulated without noise to isolate the effects due to smoothing. A safe upper limit for neighborhood radius size is approximately 20% of the smaller tank dimension . . . . .	70
3-9	Derivation of allowable angular error, $\theta_e$ from input allowable accuracy, a. Difference between $h_m$ and $h_a$ must be equal to or less than $a$ . . . . .	71

3-10	Effect of sensor placement on data collection. As shown in a, c and d, data points collect around the sensor when a portion of the sensor view is obstructed. The best data is evenly distributed about the cylinder. The best place for the sensor is to be placed in the center of the tank as shown in b. The worst place is in the corner of the tank as shown in a. . . . .	73
3-11	Using $n_n$ and $r_n$ to find commands for sensor platform. a) The projection of the desired neighborhood circle onto a unit sphere surrounding the sensor. b)The vectors used to calculate the neighbor locations. . .	74
3-12	Gaussian mapping of an uncapped cylinder (a) and a plane (b) onto the Gaussian sphere. . . . .	77
3-13	(a) Simulated raw data with arrow indicating floor normal. (b) 3D Histogram of the Gaussian mapping of a capped cylinder. Note the one point (at the end of the arrow) where the color difference indicates many normals is in line with the expected floor normal . . . . .	78
3-14	Deriving bin size from $\theta_e$ . Worst case error is from the center of the bin to its corner. Bins are arranged at a unit distance from the center of the sphere, thus the arm length is 1. . . . .	79
3-15	Isolation of the floor, roof and wall data sets. The only points excluded are those from the corners that are ambiguous as to which group they belong . . . . .	81
3-16	Derivation of $\delta z$ and $\delta r$ . The tank is tipped by $\theta_e$ . $\Delta d$ and $\Delta z$ are the extents of the data in the corresponding direction. . . . .	82
3-17	Cylinder Fitting Steps and Final Completed Fitting. The white stars are the date points, while the black is the fitted plane/cylinder. In the top row are the floor and roof fits from left to right. The bottom left is the cylinder fit. The bottom right is the trimmed and combined fit.	85

4-1	Raw data, fits and fit accuracies of varying simulated tests. The simulated sensor was rotated by 15, 30, 45, 60 and 70 deg with varying tank heights, diameters and manhole locations. The manhole locations are indicated by the black circle in the raw data image. . . . .	88
4-2	Skewed circle data. The blue line indicates the expected circle, while the black shows the resulting circle fit from the algorithm. The line passing through the points where the two circle intersect is the axis the cylinder is rotated. . . . .	89
4-3	Accuracy versus sensor noise. The desired accuracy for the tests that make up the chart was 0.05m. The accuracy is within tolerance for noise levels up to the desired accuracy and sometimes above. . . . .	90
4-4	Algorithm segmentation and fitting time as a function of the number of points. It shows an exponential increase. Even above 10000 points, the time to process the data points is only a few seconds . . . . .	91
4-5	Prototype Gimbal. Consists of a nested U shaped structure driven by a Maxon EC-Max 40 and RE 50 motor, controlled by Maxon EPOS2 controllers. The rotational sensors are SICK DFS60A 65536 count encoders. The distance sensor is a 17m range SICK DT500. . . . .	92
4-6	Prototype Tank. Made of a PVC frame covered by stretched fabric. The tank was placed on a rubber floor. . . . .	93
4-7	Test Run Data. a) The raw test run data using the tank and sensor platform described above. b) The same data rotated with the outliers removed and its fit. The roof data is very angled, most likely due to reflection problems between the laser and stretched fabric. . . . .	94
B-1	Main Program Front Panel . . . . .	134
B-2	Main Program Block Diagram . . . . .	135
B-3	POI Calculation SubVI Connector . . . . .	136
B-4	POI Calculation SubVI Block Diagram . . . . .	136
B-5	Neighborhood Property Calculations SubVI Connector . . . . .	137

B-6	Neighborhood Property Calculations Block Diagram . . . . .	137
B-7	Neighbor Location Calculations SubVI Connector . . . . .	138
B-8	Neighbor Location Calculations Block Diagram . . . . .	138
B-9	Radian to Quadrature Count SubVI Connector . . . . .	139
B-10	Quadrature Count to Radian SubVI Connector . . . . .	139
B-11	Radian to Quadrature Count SubVI Block Diagram . . . . .	139
B-12	Quadrature Count to Radian SubVI Block Diagram . . . . .	139
B-13	Motor Movement SubVI Connector . . . . .	140
B-14	Motor Movement SubVI Block Diagram . . . . .	140
B-15	Read Distance SubVI Connector . . . . .	141
B-16	Read Distance SubVI Block Diagram . . . . .	141
C-1	Clipping process in the z direction. $z_5$ is the 5th percentile of the data. $z_{95}$ is the 95th percentile of the data. $z_0$ and $z_{100}$ are the modified extents containing all valid data but excluding outliers. $l$ is the actual height of the data. . . . .	143
C-2	Clipping process in the radial direction. . . . .	145

# List of Tables

2.1	Sensor Platform Functional Requirements . . . . .	33
2.2	Pugh Chart of Sensor Technology Options . . . . .	40
2.3	Pew Chart of Sensor Placement Concepts . . . . .	53
3.1	Required User Inputs . . . . .	68
3.2	Optional or Derived User Inputs . . . . .	68
4.1	Estimated Parameters Using Various Numbers of Neighborhoods . . .	94
4.2	Estimated Parameters Using Various Desired Accuracies . . . . .	95





# Chapter 1

## Introduction

### 1.1 Motivation

Inspecting large liquefied petroleum gas (LPG) tanks for defects is difficult, expensive and energy intensive. LPG is stored in steel alloy tanks at approximately -160C. These tanks are periodically inspected for cracks, corrosion and other defects, despite the significant hazards to the inspector [11]. External inspection is not possible for the lowest 1m of the walls and the floor of the tanks due to structural concrete in these areas. The current inspection process poses serious safety challenges as it involves sending human inspectors into the emptied tanks. The tanks need to be taken out of service and warmed for 10-14 days in order to make them safe for human ingress [12]. The cost of this shutdown is approximately 15 million dollars per day and significant energy is required to warm the tanks up and then cool them back down.

A robotic inspection system will make the inspection of LPG tanks more efficient. The down time can be shortened because the tank will not need to be as warm for the robot inspector as it needs to be for the human inspector. This reduces inspection cost and there will be less energy used to cool the tank back down to operating temperatures. The inspection process itself can be done more quickly and thoroughly because the robots will move along prescribed paths more accurately than human inspectors. The risk is also diminished because robots are not sensitive to the same hazards as human inspectors.

## 1.2 Research Focus

The purpose of this research is to develop a robust, efficient, self-contained localization module for use in a robotic liquefied petroleum gas tank inspection system. This module consists of the hardware sensing platform and the corresponding algorithms for mapping the environment. It will be used with a new inspection robot being developed in a joint project between MIT and Qatar University.

The first part of the work is designing a sensor hardware platform. The design process involves determining the best sensing technology, followed by determining the optimal layout of the sensors in the LPG tanks. These choices must take into consideration the environment of the tank as well as the lack of any distinguishing features besides the presence of a manhole in the walls or roof.

The second part of the work focuses on fitting the 3D data from the sensor platform to a capped cylinder. The resulting fit creates a map of the environment to be used in inspection robot path planning and defect localization. The algorithm needs to be robust to noise because the environment and interaction between the sensor and tank walls may result in significant error. It also needs to be faster than current fitting methods to decrease inspection time.

## 1.3 Background

### 1.3.1 LPG/LNG

Liquefied Petroleum Gas, or LPG, refers to a class of hydrocarbon fuels composed mostly of propane and butane [13]. LPG is gaseous at atmospheric temperature and pressure, but it can be cooled or compressed into a liquid. LPG is commonly used as fuel in heating, cooking and even vehicles. It is a byproduct of natural gas or crude oil refining so large tanks are often found on-site at petroleum processing plants [13]. These tanks must be periodically inspected for corrosion, cracking and other defects. Figure 1-1 shows a few examples of the large LPG storage tanks.

The configuration and internal environment of the LPG tank pose challenges to



Figure 1-1: LPG Storage Tanks. (a) Dominion Cove Point LNG facility located in Lusby, Maryland. Photo courtesy of Dominion Corporation [1]. (b) Two LPG storage tanks. Photo courtesy of Lusas Corporation [2]. (c) Row of LPG storage tanks. Photo courtesy of Gauging Systems Inc [3].



Figure 1-2: LPG Tank Cross-section. Multilayer construction of steel plating, insulation and support walls. Domed carbon steel top has suspended aluminum deck. Concrete base layers and support walls make external inspection impossible.

inspection. LPG tanks are carbon or duplex steel approximately 98m in diameter and 60m tall. The tanks sit on concrete slabs with their walls reinforced by concrete for at least a meter up from the floor. The details of the tank structure are shown in Figure 1-2. External inspection of the floor and walls behind the concrete is impossible, so internal inspection is necessary. The temperature inside is cryogenic even after draining. There is no source of light in the tank and any light source needs to be low power in order to avoid ignition of residual particles. The residual particles also impose limitations on other energy imparting systems, such as EM radiation. The low temperatures, residual flammable particles and lack of light need to be taken into consideration when designing an inspection system.

### **1.3.2 Conventional Inspection**

Conventional inspection is completed by human inspectors inside the tank with specialized equipment. The tank must be drained, warmed to habitable temperatures and cleaned before the inspector is sent in [11]. The inspector collects a sparse amount of data due to the slow speed of human inspection and limited allowable exposure time. After inspection the tank must be cooled to cryogenic temperatures before being put back into service.

Conventional inspection has many negatives including high cost, being hazardous to the inspection staff, and negative environmental impacts. Taking the tanks out of service and warming them to habitable temperatures is very costly, approximately 15 million dollars per day of down time for up to two weeks. In addition to the high cost of inspection, there are significant safety and health challenges to manned inspection due to the confined space and residual material [12]. Cleaning the tanks for human ingress involves degassing the tank and cleaning sludge from the bottom. The release of gas into the environment and the disposal of sludge is hazardous to the environment. Developing a robotic inspection system that can withstand colder temperatures and does not require complete degassing or cleaning of the tank floor removes the safety and health issues and greatly reduces the environmental impact of tank inspection.

### 1.3.3 Robotic Inspection

Robotic inspection has the advantages of not exposing human inspectors to the tank environment, reducing the cost and environmental impact and being more effective than human inspection. The temperature of the tank will not need to be raised to as high a temperature because the robot habitable temperature is much lower than human habitable temperature. This will reduce the energy needs and down time resulting in a less expensive inspection. A robot inspection system would not require complete degassing and cleaning, thereby reducing the environmental impact of inspection. The robot can also be designed to clean the tank during the inspection routine much like the Techcorr robot in Figure 1-3 (e). Robot inspection removes the risk to inspectors because they are not needed in the tank. Finally, robotic inspection will result in a complete inspection of the tank because the robot can be exposed to the hazardous environment for longer than a human inspector.

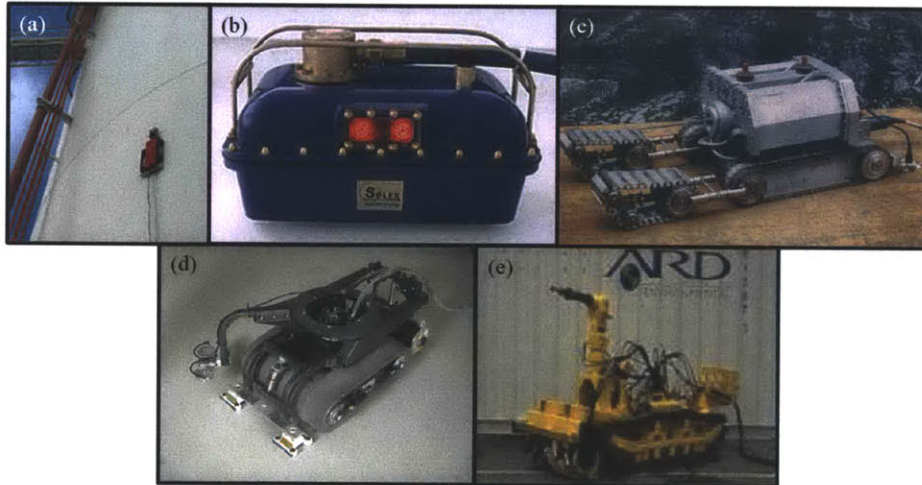


Figure 1-3: Inspection Robots. (a) External crawling inspection robot [4]. (b) Maverick, a remote-controlled, submersible robot for inspection of gasoline tanks [5]. (c) Neptune, a fuel storage tank inspection robot [6]. (d) ICM climbing robot with installed NDT sensors [7]. (e) Techcorr cleaning and inspection robot [8].

Current inspection systems are unable to replace human inspectors for the bottom of LPG tanks.. Robotic inspection of storage tanks is not a novel idea. Some current inspection robots are shown in Figure 1-3. Most inspection robots can attach to and inspect the tank externally, such as the robot in Figure 1-3 (a) [14][15]. These robots



can only inspect the walls above the structural concrete with no way to inspect the floor of the tank. There are some robots, such as those depicted in Figure 1-3 (b), (c), (d) and (e), that can access the floor of tank through internal inspection, but they are designed for tanks storing other substances [11][16]. These robots are designed for use in tanks storing products that are in a liquid state and stored at room temperature, namely kerosene, gasoline, jet-fuel, etc. One of the unique challenges of designing an inspection robot for LPG tanks is that the tank is held at cryogenic temperatures. None of the previous designs are applicable to use in this environment, so a system needs to be developed to specifically address LPG tank inspection.

### 1.3.4 Proposed System Design

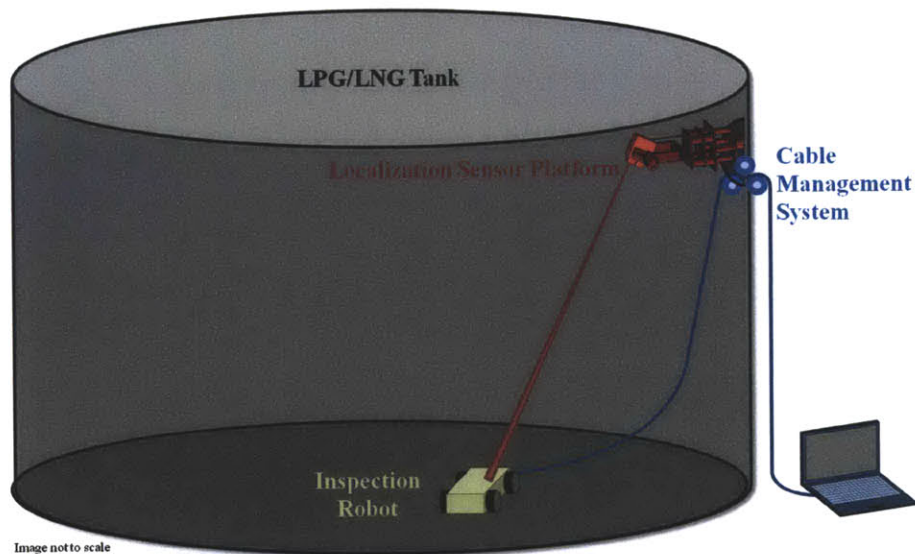


Figure 1-4: Proposed Design of Inspection System. Internal mobile inspection robot scans inside surfaces of tank for defects. Localization sensor platform locates and tracks position of inspection robot. Controlling computer is located outside the tank, away from the hazardous environment inside the tank.

The proposed inspection system design consists of a mobile platform, an ingress and egress mechanism, non-destructive testing equipment, an external controlling computer, and a localization system. The layout of these components are shown in Figure 1-4. The NDT equipment will be mounted on the mobile platform that enters the tank via the ingress and egress mechanism. A controlling computer will be located

outside the tank and will communicate with the robot, either through an umbilical cord or a wireless system. Placing the controlling computer outside the tank reduces the explosion risk by keeping the majority of the electronic switching outside the explosive environment. A localization system will be placed in the manhole to create a reference frame from which a map of the tank will be created.

This thesis encompasses the work done on the design of the localization system for the robotic LPG tank inspection system. The first component of the work is the sensor platform design, including the concept definition and hardware selection. The second part of the work consists of the development and first implementation of data collection and tank mapping algorithms. The fitting algorithm detailed in this work is specific to cylindrical tanks with planar caps because it is the most common LPG storage tank shape.

## **1.4 Prior Art**

The two components of the inspection system detailed in this thesis are the design of the localization system hardware and the processing of the collected data into a useful map. This section discusses prior work done on localization systems, followed by a discussion of the prior work done on fitting a capped cylinder to 3D data.

### **1.4.1 Localization Method Prior Art**

There are four methods of locating a mobile robot discussed here: triangulation, vision, landmark interaction, and dead reckoning. These three methods are the most common means of localizing a robot in its environment. The method of localization and limitations of each of these methods are described below.

One of the most common methods for locating mobile robots is triangulation. Three or more sensors with known locations send out signals in all directions. The signals are received and returned by transceivers on the robot [11][17]. The distance to the robot from each sensor can be calculated from the time of flight of the sensor signal to and from the robot. The distance from the sensor defines a sphere of possible

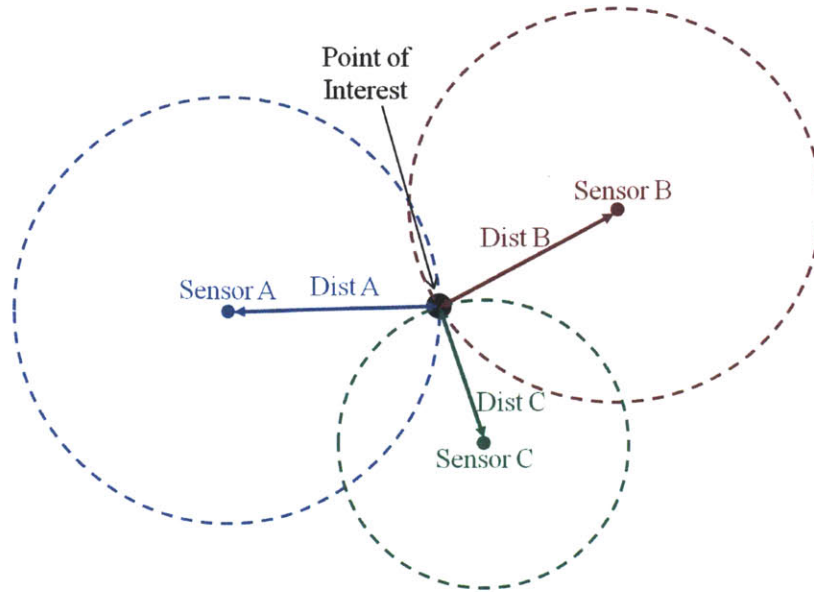


Figure 1-5: Triangulation. Distance from sensor creates a sphere(or circle when in 2D) of possible locations of the point of interest. The intersection of the spheres(or circles) pinpoints the location of the point of interest.

robot locations. The sphere's intersection pinpoints the location of the robot as shown in Figure 1-5. A method of repeatedly and accurately placing triangulation sensors will need to be developed before triangulation is a feasible option for the inspection localization system.

A second localization method is to use vision sensors to image the environment [18]. This is most similar to the human method of localization. A robot uses a camera, instead of eyes, to record its surroundings and then extracts details of the environment to orient itself by. There are two major difficulties with vision localization in an LPG tank. The first difficulty is the lack of guaranteed features in the tank. There may not be any features in the tank that can be extracted from images to orient the robot with. The second difficulty is the lack of light in the tank. There is no light entering the tank, and only a low power light source can be used without risking explosion.

Another method that is similar to a vision based system is a method that uses landmark sightings [19]. Landmarks are placed at known intervals in the environment. These landmarks can be visual or can use another technology. They are detected by the robot and its location can be found based on the relative location of the



landmarks. The difficulty with this method is that the landmarks would need to be placed accurately. This limits the transportability of the system in the same way that placing triangulation sensors do.

The above methods are usually combined with dead reckoning [19]. Dead reckoning uses sensors that measure the motion of the robot wheels or tracks. The amount of rotation is directly related to the distance traveled. Errors in the measurement of the distance traveled occur when the wheels or tracks slip. Slipping is more likely to occur on uneven ground, which the tank may have due to uneven build up of particulates. The slipping error keeps dead reckoning from being a stand alone method for localization, but it can provide supplemental information when used in conjunction with another localization method.

All of the localization methods listed above have significant difficulties when applied to an LPG tank inspection system. A method needs to be designed that can work in the LPG tank conditions. The requirements and design of a localization sensor platform that can overcome the difficulties of the environment are described in Chapter 2.

### **1.4.2 Data Processing Prior Art**

Processing environment data involves segmenting it into sections that are then fit to geometric primitives such as planes and circles [20][21]. The edges of the shape are located at the intersection of the geometric primitives. All fitting is done using a batch least squares optimization method. This thesis focuses on developing a new segmentation method that is faster and more robust to noise than the current methods.

There are four categories of segmentation methods: Clustering, Seed and Grow, Edge detection and Hough Transform. In each of these methods, properties of points on the surface of the object of interest are used. Usually these properties are the position and orientation of the surface normal at the point of interest [22]. Each of these methods is described below.

In clustering, the properties of each data point are compared and points with

similar properties are grouped together. Points are considered to be part of a cluster when the properties of the point of interest and the properties defining the cluster are within a user defined similarity threshold. When a point does not belong to any of the existing clusters, a new cluster is formed. The comparison process iterates through each point on the surface until all points have been sorted into a cluster. Each point now lies in a cluster and each cluster corresponds to a surface.

Seed and Grow is similar to clustering in that it groups together points with similar properties. The difference is that random seeds are selected and compared to their neighbors instead of to a set of clusters. Neighboring points are compared and collected into a group until the discrepancy between the properties of the current point and its neighbor exceeds a user defined threshold. Once all the neighboring points next to the grown section fall outside of a threshold for similarity, another seed is chosen. The process repeats until all points are inside a group. A final sweep combines homogeneous regions [21][22].

Edge detection also compares neighboring points, but in this method points defining the edges of the surfaces are collected instead of points within the surfaces. The properties of each point are compared to the properties of all its neighbors. An edge is indicated when discrepancies reach a defined threshold. These edges are collected and used to define the extents of each section [23].

The Hough Transform relies on a voting procedure to find the most common geometric parameters for a given area. Several points are chosen and all possible planes (or whatever shape is of interest) that intersect these points are determined. A vote is cast for each of the possible solutions. The solution with the most votes is assumed to be the true fit [24][25]. Multiple surfaces will result in multiple solutions with a large number of votes. Each point's properties are then compared to the solutions and grouped according to the best fit.

These methods are flexible in the shapes they are able to segment and fit, but there is room for improvement when segmenting inspection environment data. These methods are flexible because they assume nothing about the environment from which the data is being extracted. No external information is needed and the solution

is unconstrained. This flexibility comes at the expense of computation power and processing time. All the methods mentioned above require iteration to converge on a solution. The computation expense and processing time can become significant when the size of the data point cloud is large [26]. Improvement on processing time and computation needs can be achieved by integrating some knowledge of the solution into the algorithm. Fortunately, the general shape of the inspection environment is known in the case of inspection robots. The shape knowledge can be used to remove iteration which is the major contributor to the time and computational expense. The knowledge can also be used to reject noise errors in the data, making the fit more robust to noisy data than the methods mentioned above.

## 1.5 Scope

This thesis describes the design of a localization sensor platform and corresponding algorithms for an LPG tank inspection localization system. Chapter 2 discusses the qualitative design process and selection of technologies for the sensing platform. Chapter 3 details the data collection and mapping algorithm. Chapter 4 describes simulated and real world testing of the algorithm and hardware design. Chapter 5 summarizes the work completed on the localization system and provides a discussion of future work that could be done to improve the system.



# Chapter 2

## Sensor Platform Design

The first step in designing a robot localization system is to determine the best hardware design given the constraints imposed by the environment. This chapter describes the qualitative design process used to design the localization platform for a robotic LPG tank inspection system. It begins with a discussion of the system requirements followed by an evaluation of the available sensing technologies. Eight sensor layouts are evaluated and the final design concept is discussed in detail.

### 2.1 Sensor Platform Requirements

The requirements for a localization sensor platform in an LPG tank are driven by three factors: the environment in which the platform will be operating, the need for the system to be portable between tanks, and the funding limitations. The requirements are listed in Table 2.1.

Table 2.1: Sensor Platform Functional Requirements

Property	Required	Desired	Units
Sensing Distance Range	0 to 98	0 to 150	m
Operating Temperature	-100 to 50	-100 to 50	C
Size (width x length)	<0.5 x 0.7	0.1 x 0.1	m
Energy Released	<0.48	<0.25	mJ
Tank Independent	Yes	Yes	—
Commercial Availability	Available	Readily Available	—
Cost	<20,000	<10,000	\$

The tank environment influences the sensing distance range, the operating temperature, the size and the energy released requirements of the sensing platform. The sensing distance range is determined by the size of the tank. The diameter of the tank is 98 m so the sensor's minimum required range is 0 to 98 m with a desired range up to 115 m, the diagonal distance across the tank. The system must be small enough to fit through the 0.5 m diameter manhole. The environment in the tank can be down to -100 C, while the temperature outside the tank can reach up to 50 C. The sensor system may need to operate in both environments, so the required and desired operating temperature ranges are -100 to 50 C. The tank is also dark so any sensor system needs to be able to operate in low light or use a low power light source. The sensors must also emit only a low amount of power due to the explosive nature of the fumes in the the air. The minimum ignition energy for propane in air is 0.25-0.48 mJ [27]. Any sensor must be emitting lower than 0.48 mJ energy, preferably less than 0.25 mJ. A final but unlisted requirement driven by the metal tank wall is that only systems sending signals from within the tank are acceptable. The metal tank acts as a Faraday cage, thus signals from outside the tank are unreliable.

The second driving factor on the requirements is the need for the system to be portable which influences the size and tank independent requirements. The system needs to be small enough to be carried by a single technician so that it can be easily moved between tanks. The system also needs to be independent of which tank in which it is being used, thus nothing can be permanently installed in the tank. Finally the system needs to be tolerant of installation variations.

The final driving factors are the logistical and financial considerations due to funding limitations. The time limitation of the funding source does not allow for the design and testing of new technology. Any technology used in the platform needs to be composed of off-the-shelf components. In addition, the cost of the system should be reasonable. The cost limitations of \$20,000 and \$10,000 are order of magnitude estimates based on the cost of the robot and the amount of the project funding. These limitations only eliminate expensive technologies that could dominate the system cost.

## 2.2 Distance Sensing Technologies

A range of sensing technologies were examined before the optimal technology was identified. The first technology examined was a vision sensor system as it mimics how the human inspectors would navigate around the tank. A local GPS system was considered, followed by Acoustics, otherwise referred to as SONAR. Finally, two types of EM sensors were considered: RADAR and LIDAR.

### 2.2.1 Vision Sensors

A vision system was initially considered but immediately discarded. There are three issues that made the abandonment of vision sensors reasonable. The first is the lack of available light in the tank. Any vision sensor would have to be able to take images with very little to no light and the addition of a light source is almost impossible due to the risk of explosion. The second issue is the lack of distinguishing features in the tank. Even if images were able to be recorded, there are no features that could be used as references for locating the robot. The third reason for discarding the vision system is that it would require a high bandwidth connection to the controlling computer. Communication between the robot and the controlling computer is going to be limited so a low bandwidth sensor is preferred.

### 2.2.2 GPS

Localization via GPS was also immediately discarded. GPS, Global Positioning Systems, operate by receiving signals from satellites. The location of the sensor is found relative to the satellites' known locations using triangulation [28]. There are two problems with this technology when being applied to localization within the tank. The first is that even high resolution GPS is only accurate to within a few meters [29]. It is desirable to know the robot localization within a few centimeters which is orders of magnitude better resolution than the GPS can supply. The second problem is that the metal tank creates a Faraday cage that will prevent clean signal reception from external sensors [30].

### 2.2.3 Acoustic

A third technology considered for the distance sensor was acoustic sensing, otherwise known as sound navigation and ranging or SONAR. In these sensors a series of acoustic waves, usually in the ultrasonic range, are generated by a transducer and travel through the environment. When the waves hit an object, they are reflected back toward the receiver. The distance of the object from the sensor is directly proportional to the time from transmission to detection and the properties of the medium through which the wave is traveling [31]. The sound waves can be concentrated in a beam to generate directional information in addition to the distance measurement.

The maximum distance and measurement accuracy of sonar heavily depends on three factors: the operating frequency, the acoustic impedance of the medium and the acoustic impedance of sensors. High frequency waves result in better accuracy, but are attenuated more quickly than low frequency waves. Low impedance environments, such as air, need larger amplitude waves to achieve enough acoustic power to travel. To get these large amplitudes low frequencies would be needed. A rule of thumb is that it is only possible to image a feature that is larger than the wavelength of the signal. The application needs good resolution so it would need a high frequency signal, but the high frequency signal would be too attenuated at 98 m to be able to return to the transducer resulting in conflicting signal needs for the desired resolution and sensing range. Conduction between the transducer and the environment also impact the measurement. Conduction is best when the impedance of the sensor matches that of the environment. The most common transducer is made of piezoelectric material which has an impedance of 35 MPas/m [32]. The impedance of air is approximately 420 Pas/m [33]. The piezoelectric transducer is mismatched to air by about  $10^5$ . This large impedance mismatch indicates that the sensor will not generate good signals in air. The conflicting signal needs for the resolution and range combine with the mismatch in impedance make conventional transducers poor sensors for an LPG tank localization system.

There are no commercially available acoustic sensor that will work in air over the



distances needed to sense the tank. Acoustic sensors for use in air have a max range of a few meters due to the mismatched impedances. A new transducer would need to be developed using a material with the same impedance as air to achieve the desired accuracy and the necessary range. As mentioned in the requirements section, the development of a new sensor is out of the scope of the project, thus SONAR sensors are not a good option for the main distance sensor.

## **2.2.4 Electromagnetic Sensors**

The last set of technologies that were explored were electromagnetic sensors. Electromagnetic distance sensors use a process similar to that of acoustic sensing where a wave pulse is emitted into the environment and the distance to the object it hits can be calculated from the time to receive the reflected returning signal. The main difference is the type of wave used. Conventional radio detection and ranging, RADAR, uses electromagnetic waves in the radio frequency range where the wavelength ranges from 1 cm to 100 m [34]. Millimeter wave radar uses waves with 1 mm to 7.5 mm wavelength [34]. LIDAR, light detection and ranging, uses visible or infrared waves, generally with 600-1000 nm wavelengths [34].

### **Conventional Radar**

Conventional RADAR is a common technology in marine and air vehicle detection, but has significant issues that limit its applicability as an LPG localization sensor. Radar waves are reflected when there is a large change in dielectric or diamagnetic constants, which is the case for metal vehicles in the air or water [34]. For the tank inspection application, the boost in signal for metal in air is a hindrance because the large metallic wall will result in a strong signal that may wash out any signal returned by the small robot. Another reason that radar is used in vehicle detection is that the radar sensing range is large due to its long wavelength. While long wavelengths are a benefit in long range detecting, they are problematic in terms of accuracy. It is generally considered to only be possible to image a feature about the same size or

larger than the wavelength [34]. This means that most traditional radar sets would not be able to sense our robot due to its small size. The minimum range for sensing is also dependent on the wavelength, with longer wavelengths unable to measure close distances. For instance, marine RADAR typically has a minimum sensing distance of 20 m, much larger than the required minimum required by the LPG system [28]. Marine radar was particularly explored because it has the smallest traditional radar packages. Even so, the size of the marine radar sets were just slightly too large for the application. Traditional radar is not a good technology for this application due to its inability to meet the minimum sensing distance, resolution and size requirements.

### **MM Wave Radar**

Millimeter wave radar is a promising modification of conventional radar for sensing distances in a range appropriate for a tank inspection localization system, but it is still too young of a technology to apply to this project. MM wave radar operates using the same principles as conventional radar, but it uses millimeter length wavelengths. It does not suffer from the same limitations of traditional radar systems, but has a shorter maximum sensing distance. Its smaller wavelength reduces the minimum sensing distance to 0.6 m and increases the resolution to 5 mm [35]. This is at the expense of the maximum sensing range, now 500 m, which is still over the maximum range the localization system would need. Due to its relatively new nature, it is not widely in use and not readily available. The majority of users create their own system, and the commercially available versions are expensive.

### **Lidar**

The final potential technology considered was LIDAR, light detection and ranging. The wavelength used for LIDAR is in the visible or infrared region of the electromagnetic spectrum. The short wavelength limits the maximum sensing distance to lower than 300 m, but it has a shorter minimum sensing distance of 0.2 m, and a better resolution of 1 mm [36]. LIDAR is commonly used as a distance measuring technology in many fields such as robotics, manufacturing, mapping, and construction so there

are many commercially available laser rangefinders.

Commercial LIDAR systems measure distance using one of two methods: phase shift or time-of-flight. Time-of-flight, TOF, uses the same process described in the RADAR section except in this case the relationship is related to the speed of light instead of the speed of sound. The equation governing the relationship between the time of flight,  $t_{flight}$ , and the speed of light  $c$  are shown in Equ. (2.1) [37].

$$dist = c * t_{flight} \quad (2.1)$$

In phase shift measurements, the sensor emits a wave of known frequency which bounces off the object to be measured. The phase of the returned wave is measured by the sensor. The difference in phase, the phase shift, is related to the time of flight in Equ. (2.2) [37].

$$\theta = 2\pi (t_{flight}) (f) \quad (2.2)$$

where  $f$  is the frequency of the wave. The distance point to the of interest can be found by solving Equ. (2.2) for  $t_{flight}$  and using that value in Equ. (2.1). Aliasing occurs if the distance to be measured is longer than the wavelength of the signal. Multiple waves of varying frequency can be used to remove the aliasing effect [37].

Both TOF and phase shift techniques can be used to achieve the requirements listed above so other considerations were used to determine which technology is the better option. TOF uses laser pulses instead of continuous laser beams to measure distance. Although the pulses are powerful enough to measure large distances, they do not accumulate significant energy due to their short duration. This is important to the LPG device because there is a risk of igniting the fumes remaining in the tank if too much energy accumulates. For example, the Acuity AR3000 is a 193 nJ laser pulsed at 2 khz for 6 ns when using time-of-flight for measuring distances [36]. This is orders of magnitude smaller than the minimum ignition energy of propane in air of 0.25-0.48 mJ [27]. TOF sensors are also better able to measure wet surfaces, which may be present in the tank, and TOF is more capable of measuring off oblique surfaces thus allowing for more freedom in sensor placement [37]. Time-of-flight has multiple

advantages over phase shift sensing making it a better option for this application.

### 2.2.5 Sensor Technology Conclusion

Of the three technologies available for distance measuring, LIDAR is the best option for this application. Vision does not work because of the lack of light and bandwidth needs. GPS is not acceptable because of the Faraday cage effect of the tank and its large resolution. Sonar does not work because there is no commercially available sensor that can achieve the required resolution over the required sensing range. The three EM sensing technologies are compared with each other with regards to each design requirement in the un-weighted Pugh chart, Table 2.2. Conventional radar is eliminated because it does not meet the needed sensing range and does not fit the size requirement. Millimeter wave radar meets the technical requirements, but it is not readily commercially available within an acceptable price range. Table 2.2 clearly shows LIDAR as the best option for this application.

Table 2.2: Pugh Chart of Sensor Technology Options

Technology	Sensing Range	Operating Temp	Size	Energy Released	Commercial Availability	Cost	Total
Acoustic	-1	-1	0	0	-1	-1	-4
Radar	0	-1	-1	-1	-1	1	-3
Lidar	1	-1	0	-1	0	0	-1

LIDAR meets all the sensing technology requirements except for the operating temperature requirement of -100 to 50C. None of the sensors were able to meet this requirement. Electronic component operating temperatures are only rated down to -40 C so no sensor manufacturer will quote a lower operating temperature for any sensing technology. The temperature requirement needs to be removed due to the inability of any sensor to meet it. The temperature problem will have to be accounted for in the design of the sensing platform with some form of heating and insulation.

## 2.3 Sensor Layout

Several factors need to be considered when determining the optimal way to layout the sensor or sensors. The first two considerations are the cost and accuracy. Multiple or expensive sensors increase the cost of the system, but may increase the accuracy. The trade-off between these two factors needs to be considered for each design. The third consideration is the complexity of the design in regards to communication and fabrication. A good design will minimize the complexity because it will reduce construction costs and often is more robust. The last consideration is the flexibility of the design. It would be preferable if the system that locates the position of the robot in the tank can also map the environment in order to make a map of the tank extents and the defect locations.

LIDAR sensors can be bought in two forms: single point sensors and scanning sensors. A single point sensor measures the distance to one point in the direction it is facing. The advantage of this type of sensor is that the operating computer will know the direction the laser is pointing at all times. The disadvantage is that to take measurements in more than one direction, a platform will have to be built with actuation in all directions of interest. A scanning sensor already has one degree of rotation actuated. It is usually made from a single point sensor reflected off of a rotating angled mirror. The advantage of this type of sensor is that it is already actuated about one axis so to take measurements in space a platform will only need to be actuated about one more axis. The first disadvantage is that scanning sensors usually cannot rotate a full 360 degrees so multiple sensors would be needed to be used. The second disadvantage is that it may be difficult to back out and control the instantaneous direction from the sensor. The scanning sensors are meant to create a horizontal slice of the environment, not point at a single entity. This would limit the flexibility of the sensor platform as it would not be able to track the robot.

There are eight concepts discussed in the following sections. They can be broken into two categories: those with the main sensor(s) on the robot and those with the main sensor(s) located elsewhere. The location, number and type of sensors depend

on the concept. These details and a discussion of the advantages and disadvantages of each design are discussed in the following sections. At the end of the chapter the concepts are compared and the final design is chosen.

### **2.3.1 On Robot**

The advantages of a design where the main set of sensors are on the robot include a decreased chance of communication problems and less of an issue with noise. A physical communication connection between the sensors and the robots can be used when the sensors are located on the robot. The physical connection has a lower probability of encountering communication problems than the wireless system that will need to be used by sensors not located on the robot. The on robot sensors will be located in the same location as the robot, so there is no chance of losing the robot in the measurement noise; the noise just decreases the accuracy of the measurements.

The disadvantages of an on robot sensor system include complexity of localizing with no unique environment features and the placement of the sensors within the hazardous environment. A sensing system contained entirely on the robot will need a reference feature to complete its orientation of the robot. The lack of unique environment features will require the placement of an external reference item in addition to the sensors on the robot. The second disadvantage is that all the sensing components will be thoroughly inside the corrosive and thermally challenging environment. Additional advantages and disadvantages depend on the specific setup and are discussed in the concept description.

#### **Concept 1: On Robot Idea #1**

The first on robot concept consists of several scanning LIDAR sensors placed on the robot with a pylon dropped in from the manhole as shown in Figure 2-1. At least two scanning sensors will be needed to see a full 360 degrees around the robot. These sensors will map a horizontal plane of the tank walls and pylon in relation to the robot. The radial position of the robot can be found from the distances to the walls

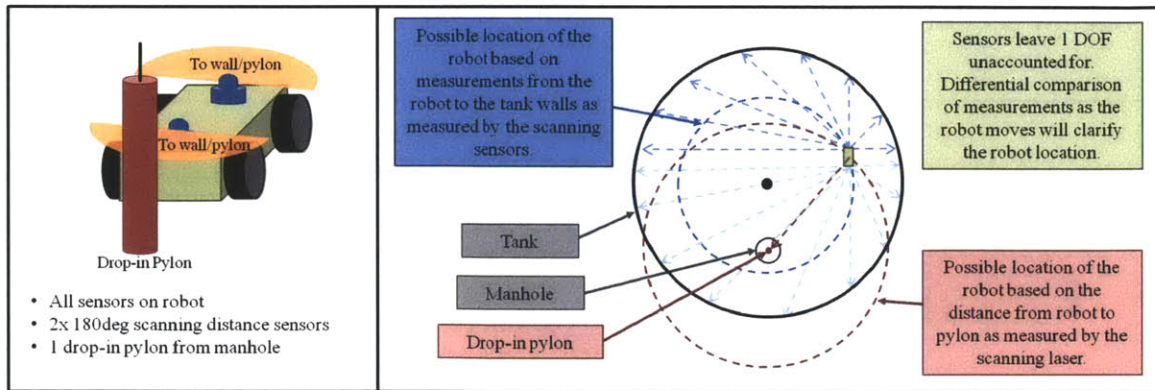


Figure 2-1: Concept 1: A set of scanning distance sensors on the robot in a plane parallel to floor measure to the walls and a drop in locating feature. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed arrows represent the measurements taken by the sensors and the dashed line circles show the possible locations of the robot based on the measurements. The circle intersections are the possible locations of the robot. These possible locations are narrowed down to the actual location based on the integration of previous data.

and the orientation is based on the relative direction of the pylon. The pylon is located below the manhole so its location is known.

In addition to the advantages listed above, the lack of an actuated platform is a specific advantage of Concept 1. The two scanning sensors can see 360 degrees around the robot without any additional actuation. Fabrication of a non actuated platform is easy when compared to the actuated platforms needed in some of the following concepts.

The disadvantages specific to Concept 1 are that the accurate placement of the pylon may be difficult and that the system must continually collect multiple points of data to locate the robot. The placement of the pylon is a significant disadvantage of Concept 1. The placement must be accurate in order to achieve an accurate location of the robot. Dropping the pylon 60m down into the tank will be difficult to achieve with accuracy. The second specific disadvantage is that several measurements to the walls and a measurement to the pylon will be needed each time the position of the robot is updated. The sensor must continue to take multiple points of data increasing the bandwidth needed to handle the constant stream of localization information.



## Concept 2: On Robot Idea #2

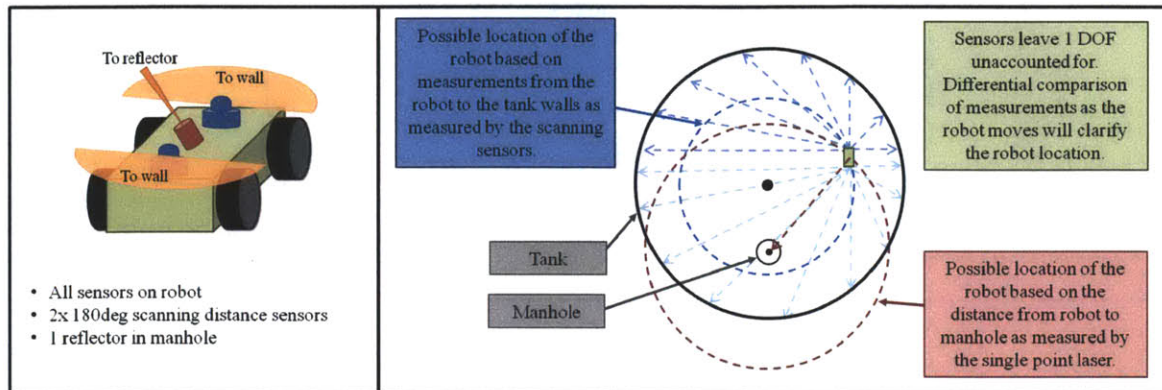


Figure 2-2: On Robot Concept 2: A set of scanning distance sensors on the robot in a plane parallel to floor with 1 single point distance sensor measuring to a reflector in the manhole. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed arrows represent the measurements taken by the sensors and the dashed line circles show the possible locations of the robot based on the measurements. The circle intersections are the possible locations of the robot. The location is finalized by the measurement to the reflector.

The second on robot concept improves on Concept 1 by replacing the orientation pylon with a reflector in the manhole and a single point sensor on the robot to measure the distance to the reflector. This concept is shown in Figure 2-2. The scanning sensors on the robot still measure the walls 360 degrees around the robot. The single point sensor is actuated in order to remain locked on the reflector in the manhole. The orientation of the sensor is measured about two axes which can be directly related to the orientation of the robot.

The specific advantage of Concept 2 is that an accurate reflector placement is easier to achieve than the pylon placement from Concept 1. What Concept 2 does that improves on Concept 1 is the placement of the localization feature in a location that is easily accessible, the manhole. The ease of installation will increase the accuracy of the localization system.

The specific disadvantage of Concept 2 is the potential complications with the tracking mechanism. The single point sensor must be actuated and measured about two axes in order to track the reflector in the manhole. The additional actuators



and sensors increase the complexity of the design, thus increasing the risk. They will also be located in the cold, explosive environment so heating and shielding will be necessary.

### Concept 3: On Robot Idea #3

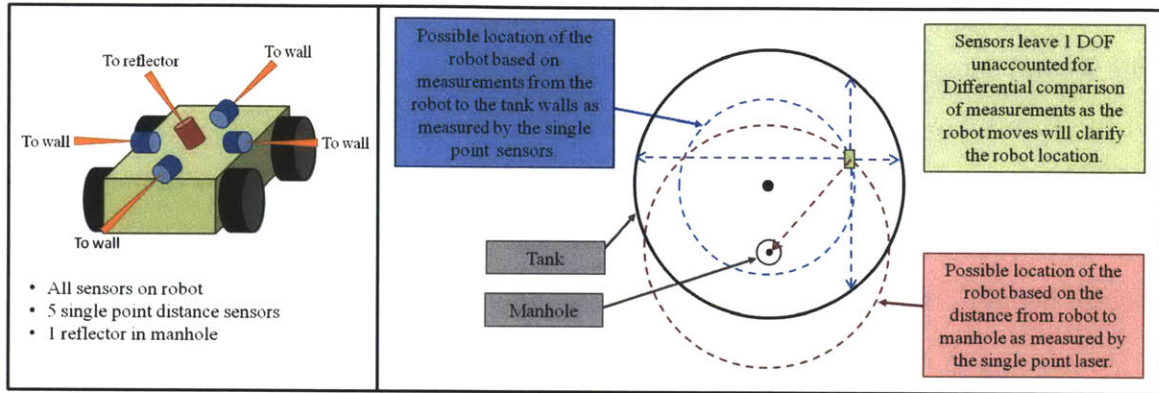


Figure 2-3: Concept 3: A set of single point distance sensors on the robot in a plane parallel to floor with 1 single point distance sensor measuring to a reflector in the manhole. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed arrows represent the measurements taken by the sensors and the dashed line circles show the possible locations of the robot based on the measurements. The circle intersections are the possible locations of the robot. The location is finalized by the measurement to the reflector.

The third on robot concept replaces the scanning sensors in Concept 2 with several single point distance measuring lasers and keeps the sensor measuring to a reflector in the manhole. The multiple single point sensors provide the same function as the scanning sensors in Concept 2. Without the need to measure to a specific feature in the tank, only a few points to the walls are necessary.

The advantages and disadvantages of the Concept 3 are the same as those of Concept 2 with one change. The exchange of scanning for single point sensors reduces the cost of the system at the expense of information about the environment. The single point sensors are less expensive so replacing the scanning sensors with single points will decrease the overall cost of the system. The downside of the replacement is that less information is gathered about the environment so it will take more time to

create a map of the environment with which the robot location and defect locations can be mapped. An a priori map of the tank may be necessary because there is no guarantee enough information will be collected to create the tank map before it is needed.

### 2.3.2 Off Robot

The second category of sensor layout concepts contains those concepts where the main set of sensors are located elsewhere in the tank. One advantage of this type of layout is that the sensor system can be larger because it is not restricted to fitting on the robot. The sensor can be placed anywhere within reach of the robot or operator so its size is only limited by the manhole. The main disadvantage of this type of system is the communication between the sensor(s), the robot and the controlling computer. There are now three entities that must communicate with each other increasing the communication complexity of the system and increasing the potential problems.

#### Concept 4: Off Robot Idea #1

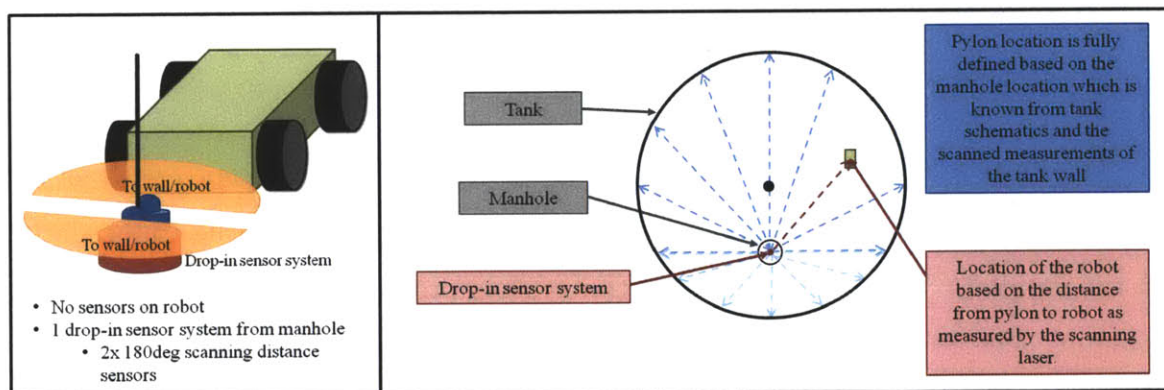


Figure 2-4: Concept 4: A separate sensor module consisting of a set of scanning distance sensors dropped into the tank measuring the distance to walls and robot. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot, while the red dot represents the sensor platform. The dashed arrows represent the measurements taken by the sensors.

The first off robot sensing concept, Concept 4, is to place a separate sensing module

containing two scanning laser rangefinders into the tank to measure the distance to the walls and the robot as shown in Figure 2-4. The location of the sensor module is known to be just below the manhole. The measurements to the walls create a map of the environment while the measurement that is shorter than expected will be the one locating the robot.

The specific advantages of Concept 4 are that it gathers information about the environment and that it acts as both the sensing platform and the reference feature. As the sensor scans to locate the robot, a map is created of the environment. Each sweep can improve the quality of the map while also updating the position of the robot. The sensing module's location is known due to its placement just below the manhole so it does not need a secondary locating feature to orient by.

The specific disadvantages of Concept 4 are that placing the module accurately and repeatedly is difficult, that the sensors are still inside the hazardous environment, that some method of inspecting underneath the module will need to be devised and that there is the possibility of losing the robot in the noise of the measurements. It is difficult to repeatedly place an object down 60m with accuracy as expressed in Concept 1. Concept 4 also does not improve on the on robot concepts in terms of removing the need for heating and insulating the sensors due to the environment. The third disadvantage that is specific to Concept 4 is that inspecting underneath the sensing module would require the removal of the module and loss of the localization method. The last disadvantage is that there is the possibility of losing the robot in the measurement noise. Concept 4 relies on there being a significant disturbance in the distance measurements to locate the robot. The robot is small in comparison to the distances in the tank so it may be possible to lose the robot in the measurement noise when it is located next to the wall at a significant distance from the sensor platform.

### **Concept 5: Off Robot Idea #2**

The second off robot idea, Concept 5, is to attach a physical connection to the robot from the manhole and monitor the movement and length of the tether to determine



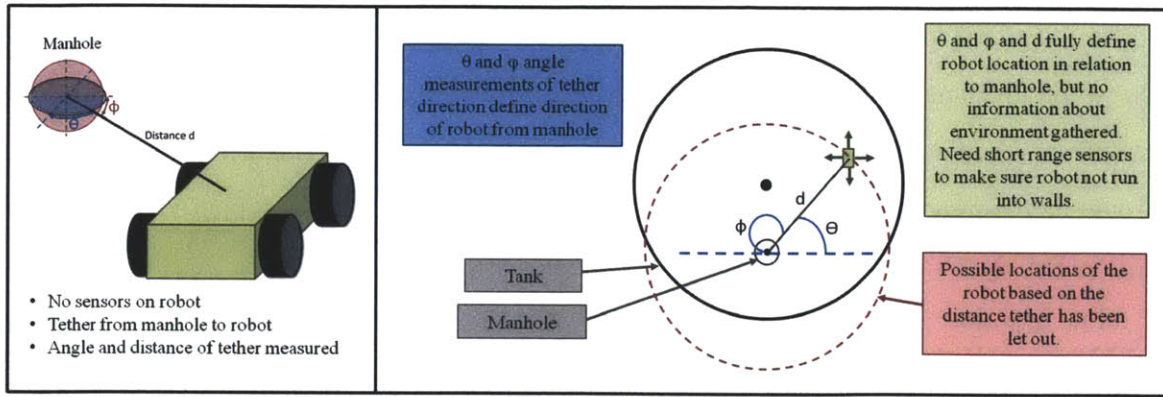


Figure 2-5: Concept 5: A physical tether to the robot from the manhole. The orientation and the length of the tether determine position. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The dashed line circle shows the possible locations of the robot based on the length of the tether. The two angle measurements determine where the robot is on the circle defined by the tether length.

the robot position in relation to the manhole. In this concept a physical tether is connected to the robot. The orientation of the tether is measured in two axes. The length of the tether corresponds to the distance from the manhole. The two axes rotations and the tether length define the position of the robot relative to the manhole.

The specific advantages of Concept 5 are that it will never lose contact with the robot and that the tether can also function as a communication and an ingress and egress method. The robot will always be in contact with the controlling computer through the physical connection. Of all concepts, this is the only one that has no risk of losing the robot when it is in the tank. The other advantage is that the tether can act as the communication method between the robot and the controlling computer instead of a wireless system. It can also act as the ingress and egress method through a winch system if it is reinforced.

The specific disadvantages of Concept 5 are the management of the tether and the lack of information about the environment. Managing the tether will be difficult because it will hinder the robot movement. The tether needs to maintain tension to accurately measure the distance from the sensor, but it needs to be loose enough that

it does not hinder the robot movement. The tether also needs to be made of a material that retains flexibility in the cryogenic temperature characteristic of the tank, or it will need a heating method that will not ignite the fumes. The second disadvantage of the physical tether is that it provides no information about the environment. There is no way to determine the extents of the environment using the tether alone, so another sensing system will need to be integrated or a map of the tank will need to be provided.

### Concept 6: Off Robot Idea #3

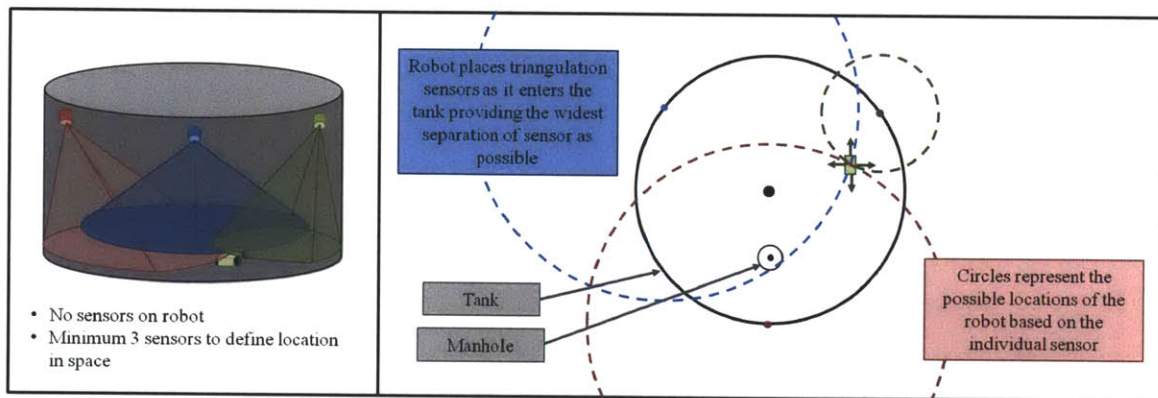


Figure 2-6: Concept 6: A local GPS-like system consisting of distance measuring sensors placed on walls by robot as it heads into tank. The large black circle represents the tank walls. The small black circle with a central dot is the manhole. The larger black dot is the center of the tank. The green rectangle represents the robot. The dashed line circles show the possible locations of the robot based on the distance measurements from each sensor. The circles' intersection is the location of the robot.

Placing triangulation sensors on the walls by the robot as it heads into the tank was also considered as an option for the sensor setup. This local GPS-like system, Concept 6, is shown in Figure 2-6. This concept is dependent upon having a robot enter the tank by climbing down the walls. As it maneuvers its way down into the tank the robot places several sensors that scan the environment. The signals are picked up and returned by receivers and transducers located on the robot. The distance from each sensor to the robot defines a sphere of the possible locations of the robot. The intersection of all these spheres will define the position of the robot.



The main advantage of Concept 6 is that triangulation methods have been thoroughly studied and are often used in localization. The mathematics describing triangulation are well developed so it would be easy to put together a system if a set of appropriate sensors could be fabricated.

The disadvantages of Concept 6 are more numerous than the advantages. The location of each sensor is important, but would be difficult to achieve with accuracy and repeatability. Another significant disadvantage is that there are no commercially available laser sensors that are set up to work in a triangulation system. In addition to these unique disadvantages, Concept 6 still locates the sensors within the tank. The sensors sending signals will be located on the walls and the receivers will be on the robot so they are all still exposed to the hazardous environment.

#### Concept 7: Off Robot Idea #4

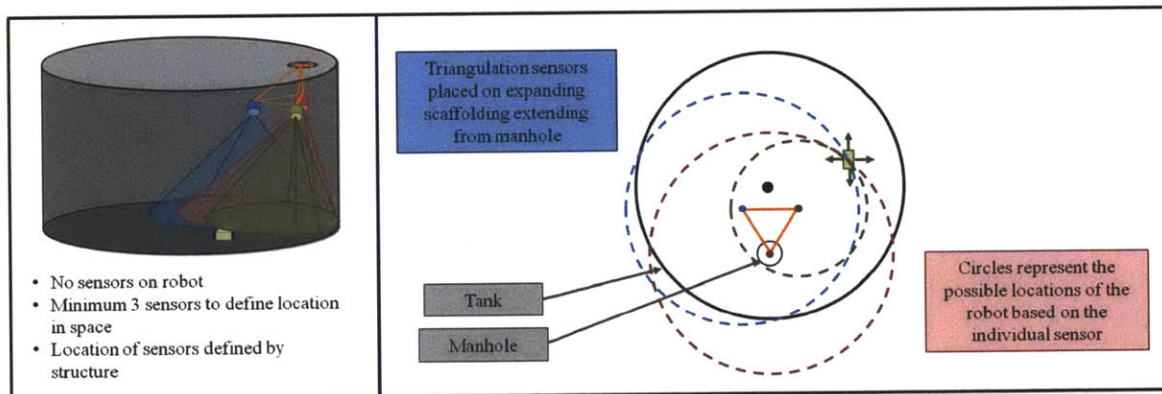


Figure 2-7: Concept 7: A local GPS-like system consisting of distance measuring sensors on a expanding scaffolding hanging from the manhole. The large black circle represents the tank walls. The small black circle with a central dot is the manhole. The larger black dot is the center of the tank. The green rectangle represents the robot. The orange triangle represents the expanding network with sensors on each corner. The dashed line circles show the possible locations of the robot based on the distance measurements from each sensor. The circles' intersection is the location of the robot.

Concept 7 is a slight improvement on Concept 6. It consists of triangulation sensors being used to locate the robot, but the difference is that the sensors would be attached to a network that expands from the manhole as shown in Figure 2-7.

This helps remedy the accurate and repeatable placement issue of Concept 6 but it adds the complexity of fabricating and securing an expanding network. The distance between the sensors needs to be large to achieve accurate localization, adding to the complexity of the expanding network. The other advantages and disadvantages of Concept 6 apply to Concept 7 as well.

### Concept 8: Off Robot Idea #5

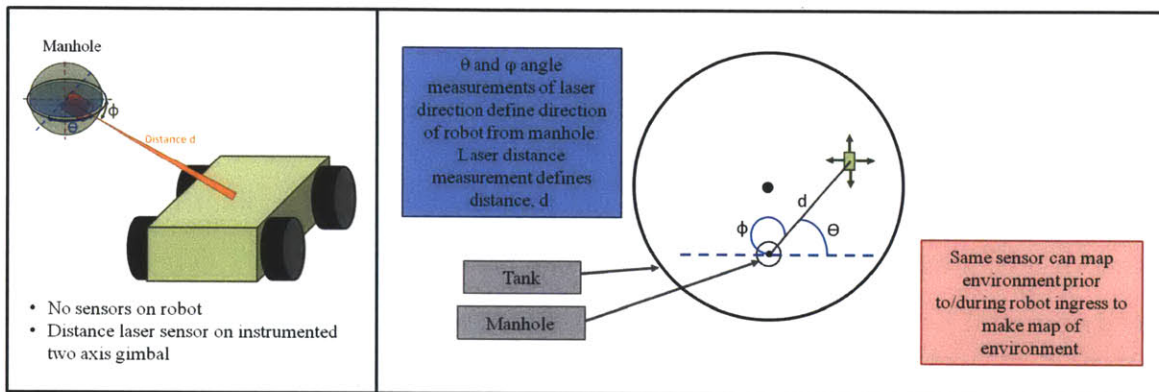


Figure 2-8: Concept 8: A sensor module consisting of a single point distance sensor and two axis angle measurement devices placed in the manhole. The large black circle represents the tank walls. The small black circle is the manhole. The black dot is the center of the tank. The green rectangle represents the robot. The measured distance and the angle of the measurement fully define the location of the robot. Measurements can also be taken to the walls to collect data for a map of the tank.

The final concept is to place a single point distance sensor on a two-axis actuated gimbaled platform in the manhole. Concept 8 is shown in Figure 2-8. Each axis position is measured using angular sensors. The two angle measurements and the distance to the point of interest fully define the position in space. The method of localization is the same as that of Concept 5, but a laser rangefinder is used instead of a physical tether. The platform can scan the distance sensor around tank to locate the robot and track its location because the computer has complete control of the position.

The specific advantages of Concept 8 include the ability to place the sensor in a repeatable manner, the ability to map the environment in addition to locating

the robot in space, and that the sensors are not completely submerged in the hazardous environment. The gimbal location is known since it is placed in the manhole, which is easily accessible by the technician. This allows for repeatable and accurate installation. In addition to a known location, the direction of the measurement is controlled by the external computer so scanning the environment to build a map can be completed in addition to tracking the robot location. The sensor placement in the manhole places it in the boundary between the tank and external environments which will be the warmest and least hazardous position in the tank.

The specific disadvantages of Concept 8 are that it may need some sort of receiver on the robot to complete the tracking control loop and that it will have some dead spots. In order to complete feedback control and track the robot, there may need to be a receiver placed on the robot to indicate that the laser is pointing at it. The other disadvantage of the system is that it may have some dead spots. The one known dead spot will be directly behind the sensor platform. The dead spot issue is not a significant problem because the sensor platform is placed in the manhole so the dead spot is located out the back of the manhole. The places of interest for the sensor system will be the floor and a short distance up the wall, which will not be in the dead spot.

### **2.3.3 Concept Comparisons**

The eight concepts listed in the sections above were evaluated based on five criteria: cost, robot localization accuracy, system and fabrication complexity, communication complexity and flexibility. The cost is the expected cost of the components, which is heavily dominated by the laser sensors. The robot localization accuracy refers to the accuracy of the measurement given the sensors used, the layout and any other influencing factor, such as the accuracy of the external feature placements. System and fabrication complexity is separate from the cost because the complexity is also related to how often problems may arise and how much development work is needed. Communication complexity is similar to system and fabrication complexity in that it is concerned with potential problems during setup and operation, but it refers to all



Table 2.3: Pew Chart of Sensor Placement Concepts

Concept	Cost	Robot Localization Accuracy	System/Fabrication Complexity	Communication Complexity	Flexibility	Total
Concept 1: Scanning sensors on robot w/ dropped in tank locating features	0	-1	1	0	-1	-1
Concept 2: Scanning sensors on robot w/ single point sensor on robot to reflector in manhole	-1	0	0	0	1	0
Concept 3: Single point sensors on robot in XY plane with single point sensor on robot to reflector in manhole	0	0	0	0	0	0
Concept 4: Separate sensing module dropped in tank	-1	-1	1	0	1	0
Concept 5: Physical connection to robot from manhole with sensor monitoring distance and length	1	1	-1	1	-1	1
Concept 6: Triangulation sensors placed on walls by robot as heads into tank (local GPS)	0	-1	-1	0	-1	-3
Concept 7: Triangulation sensor network expanding from manhole	0	-1	-1	0	-1	-3
Concept 8: Sensor platform in manhole with single sensor and angle measurement	1	0	0	0	1	2

communication needed between the robot, the sensors and the controlling computer. Flexibility refers to the overall system flexibility in performing secondary operations, namely, being able to measure the environment to make a map of the tank. Table 2.3 shows a comparison of the concepts in the five criteria. Concept 3 was chosen as the baseline because it is the closest concept to what was proposed in the initial project description. A negative value indicates worse performance in that category while a positive value indicates better performance.

### 2.3.4 Concept Selection

Concept 8 was determined to be the best fit for the robotic inspection system localization sensor platform. The cost is less than any of the other concepts with the exception of Concept 5 which uses no laser sensor. It is less difficult to accurately place the platform than any of the other in tank modules because it is placed in the manhole which is easily accessible by the operator. Concept 8 is about average in terms of system and fabrication complexity due to it being a single module with two rotating axes. The communication complexity is no worse than any of the other concepts that must also communicate with the controlling computer via wireless methods. It has a slight advantage of being able to connect via cable to the controlling computer so only the robot is being wirelessly controlled. Finally, it is one of the few concepts that allows for mapping of the tank in addition to locating the robot.



Figure 2-9: Two surveying total robotic systems. Left: Trimble S8 surveying total station [9]. Right: Leica TPS1200+ surveying total station[10].

Concept 8 is similar to the total stations used in surveying. Two surveying total stations are the Leica TPS1200+ and the Trimble S8, shown in Figure 2-9. These platforms are used to create 3D images of buildings by scanning. They can also take

measurements by tracking a target. The reason these are not considered as options for this thesis is mainly the prohibitive cost. They are too expensive for use in prototype and lab work, but they may be an option when making the full production system. Another problem in some of the systems is the limited range of motion in the head. The Leica system appears to have a large blind spot which would interfere with the measurements needed to maintain contact with the robot. A third difficulty with these systems is the larger than desirable minimum range. Most of the systems have a minimum sensing distance of approximately a meter. The Leica TPS1200+ system has a minimum sensing distance of 1.5m while the Trimble minimum sensing distance is 0.2m [10][9]. The final difficulty in using the system is obtaining access to the data. In a laboratory setting it would be impractical to purchase an expensive piece of equipment only to try to take it apart. In a final production system it may be possible to partner with the company that manufactures these sensor platforms to modify it such that it performs the desired functions.



# Chapter 3

## Mapping Algorithm

One contribution of the work described in this thesis is an algorithm that fits a capped cylinder to data collected by the hardware design from chapter 2. The algorithm's main feature is a robust and efficient method of segmenting and fitting data to a right capped cylinder that is faster and more robust to noise than the current methods which were outlined in the introduction's prior art section. The improved performance comes from a unique combination of object shape knowledge, the Gauss image and 3D histogram techniques which achieve accurate segmentation without iteration. In addition to the segmentation and fitting methods, the algorithm detailed in this chapter includes an optimized method for collecting data.

This chapter discusses each aspect of the algorithm. It starts with a discussion of the algorithm requirements which revolve around the intended application. Next is a coordinate frame and variable overview, followed by a discussion of the optimal data collection process. Then each step of the segmentation and fitting algorithm will be discussed in detail. In conclusion, the algorithm's advantages and limitations are discussed.

### 3.1 Algorithm Requirements

The algorithm requirements were all derived from the intended application of robotic inspection of LPG tanks. The algorithm will be used in conjunction with the sensor

platform from Chapter 2 to create an environment map. Figure 3-1 shows how the six algorithm requirements are derived from three application needs. The first two requirements for the algorithm are derived from financial motivations. Since time is directly related to cost, the actual requirements are time based. The third, fourth and fifth requirements are derived from the inability to guarantee optimal operation by technicians. The last requirement is driven by the suboptimal operating conditions in the tank.

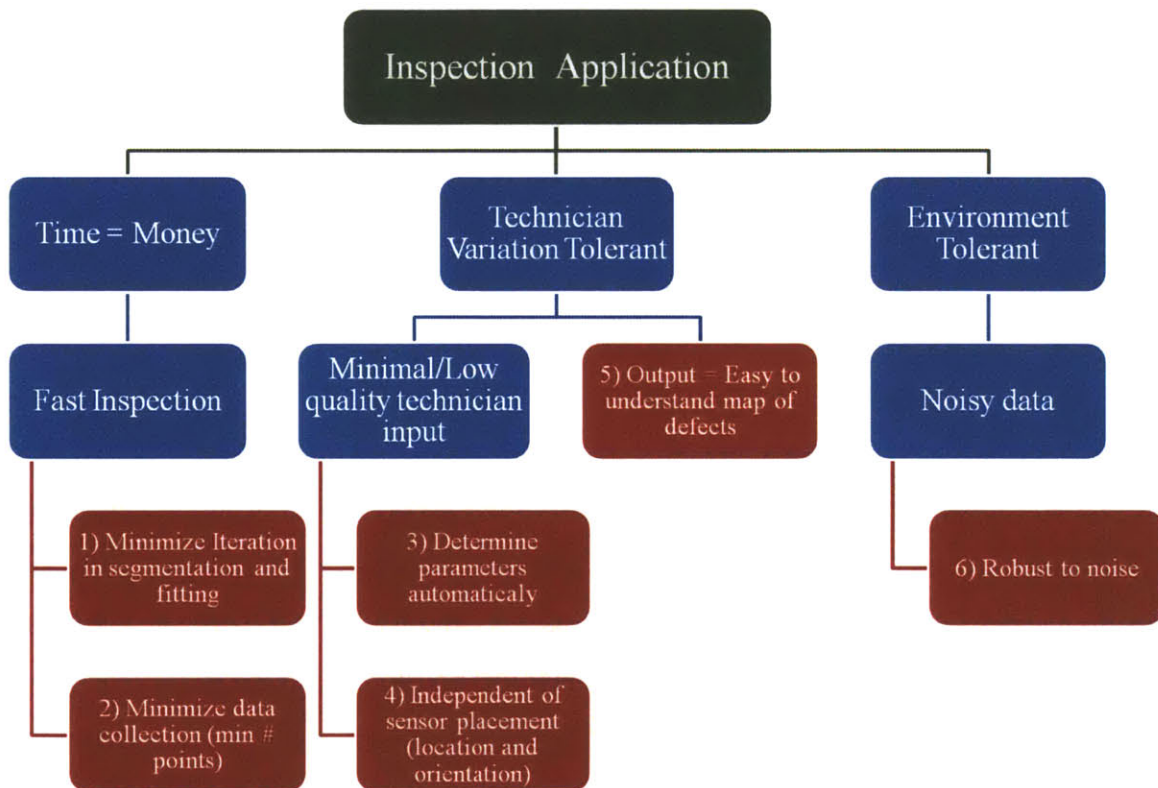


Figure 3-1: How the tank inspection application drives the algorithm requirements. The three application needs of minimizing time, being tolerant of technician variation and error and being tolerant of noise induced by the environment drive the six algorithm requirements of minimizing iteration, minimizing the amount of data, determining the parameters automatically, being sensor location and orientation independent, outputting an easy to understand tank map and being robust to noise.

The fitting algorithm’s first requirement is to minimize iteration. Iteration significantly increases processing time, especially with large amounts of data. Processing time is proportional to the number of data points raised the to number of iterations. Time is directly correlated with cost due to the tank downtime expense. One project

goal was to reduce the inspection cost, thus an algorithm requirement is to rapidly complete processing. To speed up data processing, a method needed to be developed that could fit the data without iteration. The unique combination of the Gauss image, object shape knowledge and a 3D histogram removes the need for iteration.

The second requirement is to minimize the number of data points collected. Even without iteration, the processing speed is directly proportional to the number of data points collected. Using the requirements set by the operator, the algorithm should determine the optimal number of data points. This process is discussed in the data collection process section below.

The third algorithm requirement is to determine the tank parameters automatically. Accurate a priori information about the tank cannot be guaranteed so all parameters to need to be determined from the data. Ideally, there would be no input required by the technician, but to make the algorithm as flexible as possible, some inputs are necessary. These must be tolerant of significant error as the technician may have to estimate when specifications are not available. After the initial data collection, parameters for fitting will be determined from the data to remove the reliance on user inputs.

The fourth algorithm requirement is that segmentation and fitting must function given any sensor placement in regards to both position and starting orientation. Each tank is different and the best installation location may vary between tanks. The goal is to not depend on precise placement by the technician and instead be able to handle being placed in the tank at any location and in any orientation. By not relying on the technician placement, repeatable measurement is more likely to be achieved.

The fifth requirement is that the output must be in an easy to understand form. The end goal of the project is to have a record of defects and their locations. The best output for this is a tank map with defect locations labeled. The algorithm presented here can process data into a parametric map which can be used as the foundation for the defect output map.

The last requirement is that the algorithm must be robust enough to handle noise and measurement error. The environment will pose challenges to any sensing

technology. Noise is to be expected. One lidar specific issue that may occur is that the incidence angle may impact the measurement. Despite these challenges the algorithm must locate the sensing platform and robot inside the tank repeatedly and accurately.

## 3.2 Algorithm Overview

The algorithm presented here combines object shape knowledge, the Gauss image and 3D histogram techniques to meet the requirements listed above when fitting arbitrarily rotated right capped cylinders. Object shape knowledge is knowing the general shape of the object of interest, but not knowing all the information that defines the shape. In the case specifically addressed by the algorithm presented here, the shape is known to be a capped right cylinder, but the orientation in relation to the sensor and the extents are unknown. By combining the limited information about the shape with the Gauss image and 3d histogram, the algorithm can fit the data without iteration and with strong robustness to noise. The knowledge of the shape allows for removing results that would be inconsistent with that knowledge, thus improving fit robustness. Using a histogram also helps the algorithm be robust to noise as the variations due to noise will be averaged out to a common central value.

The algorithm presented here is unique in that it fits and arbitrarily rotated capped cylinders with planar caps to data. Fitting open, or "infinite" cylinders is more prevalent in the literature. The caps, or ends, either do not exist as in the case of fitting to pipes in a building [38], or they have been processed separately [21]. The roof and floor add extra complexity to the fitting problem. The algorithm in this thesis tackles the problem of fitting capped cylinders because that is the most common shape for large LPG storage tanks. Arbitrary rotation is handled because the sensor orientation is unknown.

### Coordinate Frames

Consider a cylindrical tank and laser range sensor located on its wall as depicted in Figure 3-2. The cylinder has a right handed coordinate frame designated by  $X$ ,  $Y$



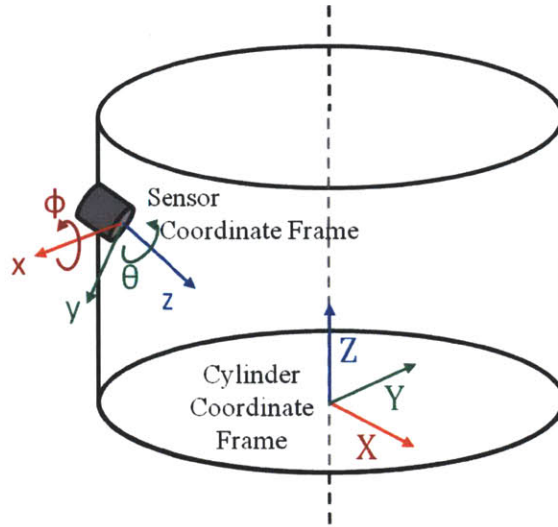


Figure 3-2: Right capped cylinder with tank and sensor coordinate frames. The cylinder coordinate frame is designated by  $XYZ$  where  $Z$  is aligned with the center axis of the cylinder. The sensor has two coordinate frames,  $\theta\phi z$  and  $xyz$ , where  $z$  is aligned with the starting direction of the sensor for both.

and  $Z$ . The  $Z$  axis is aligned with the cylinder's center axis. The sensor has its own coordinate frame designated by  $\theta$ ,  $\phi$  and  $z$  where the  $z$  axis is aligned with sensor lens at startup. The  $\theta\phi z$  coordinate frame is the algorithm's frame of reference. The algorithm does not initially know the orientation of  $XYZ$  in relation to  $\theta\phi z$ . The sensor orientation upon startup will result in data that appears to be a rotated cylinder. Various sensor startup orientations in a variety of tanks is shown in Figure 3-3. The amount the tanks appear to be rotated is dependent upon the startup orientation of the sensor. During the segmentation process, the data is rotated into a "normal" orientation such that the tank  $Z$  axis is vertical, thereby removing effects of the sensor startup orientation.

The coordinate frames detailed in Figure 3-2 will be used throughout this thesis. The sensor orientation as it moves during data collection is measured by the rotation amount in  $\theta$  and  $\phi$ . The distance from the sensor to the surface along the  $z$  axis will be referred to as  $d$ . Some calculations are easier to accomplish in a rectangular coordinate frame. In these instances the sensor rectangular coordinate frame, designated by

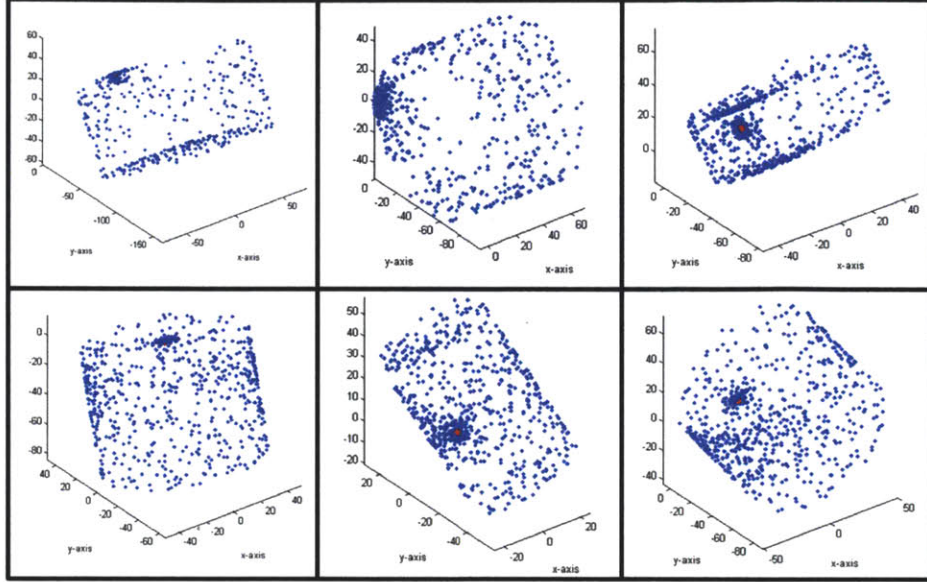


Figure 3-3: Resulting data from different sensor platform orientations upon data collection startup. Tank rotation is dependent upon sensor starting orientation.

$x, y, z$ , can be derived from  $\theta, \phi$  and  $z$ .  $xyz$  and  $\theta\phi z$  are related by:

$$\begin{aligned}
 x &= d \sin \phi \cos \theta \\
 y &= d \sin \phi \sin \theta \\
 z &= -d \cos \phi
 \end{aligned}
 \tag{3.1}$$

### Least Squares Plane Fitting

All fitting is done using standard batch matrix least squares optimization [39]. The equation for a plane is defined in Equ. (3.2).

$$Ax + By + Cz = 1 \tag{3.2}$$

The plane normal coordinates are the coefficients,  $A, B$  and  $C$ , thus the desired optimization output,  $\Theta$ , is defined as:

$$\Theta = \begin{bmatrix} A & B & C \end{bmatrix} \tag{3.3}$$

The inputs to the system are the coordinates collected for the neighborhood. These

are placed into a matrix as shown in Equ. (3.4).

$$\Phi = \begin{bmatrix} x(1) & x(2) & \dots & x(n_n + 1) \\ y(1) & y(2) & \dots & y(n_n + 1) \\ z(1) & z(2) & \dots & z(n_n + 1) \end{bmatrix} \quad (3.4)$$

The output is 1 for all inputs, thus the output matrix,  $w$ , is defined as:

$$w = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \quad (3.5)$$

Theta can then be found from:

$$\Theta = [\Phi\Phi^T]^{-1}\Phi w^T \quad (3.6)$$

## Algorithm Layout

The algorithm is divided into three phases: data collection, segmentation and fitting. The process is outlined in Figure 3-4. The data collection portion contains details about the optimal method of collecting surface data given a sensor platform described in Chapter 2. The normals calculation is included in the data collection phase because doing so increases the algorithm speed. If data collection is done differently than described here, the normals calculation must be done in the segmentation step. This ambiguity is why it is straddling the data collection and segmentation portions in Figure 3-4. Segmentation consists of creating the Gaussian Image, putting the results into a histogram to find the most common normal, calculating a rotation matrix using that normal, rotating the data and finally separating the data into regions based on location. The last step in the algorithm is to fit geometric primitives, such as planes and circles, to each region. The intersection of the fits define the tank limits, thereby creating a parametric tank map. Each phase is discussed in detail in the following sections.

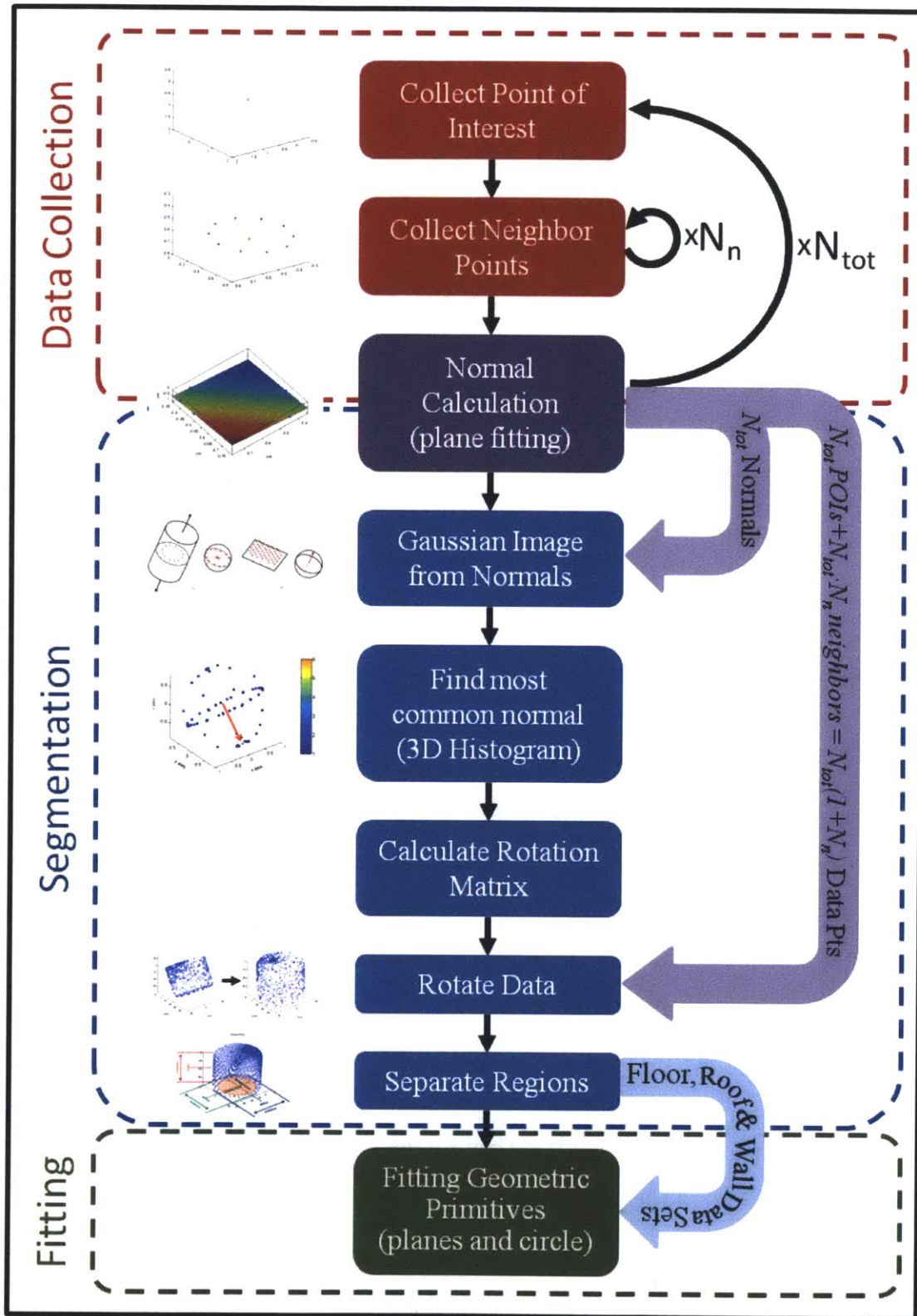


Figure 3-4: Algorithm processing flow. Process movement is shown by black arrows. Note the repeating steps within the data collection process. Data passed from one step to the next is shown by large arrows.

### 3.2.1 Data Collection and Normal Calculation Steps

There are two data sets needed as inputs to the segmentation portion of the algorithm: the measured cylinder surface points' locations in space, and the surface normals. The normals can be calculated by collecting adjacent surface point measurements and fitting a plane to these "neighborhoods". If the surface points are collected separately from the normal calculations, the resulting data cloud can be parsed into neighborhoods using distance from the point of interest, POI, as the segmentation criteria. A faster, more efficient method of collecting data and finding normals is to collect the points in the neighborhood and calculate the normal to the neighborhood before moving on to collect data for the next neighborhood. This reduces the time required to parse the data cloud, removes iteration and can be controlled to minimize the number of data points collected. This section discusses the combined collection and processing method developed to compliment the fitting algorithm.

A modified version of a method developed by Hoppe et al. is used to find the normal at each point of interest. A plane is fit to the point and its  $k$  nearest neighbors using least squares fitting [40]. The normal to this plane is assumed to be the normal of the sampled point. In Hoppe et al., the value for  $k$  is chosen manually based on trial and error [40]. There have been several other attempts at determining the best number of neighbors for normal estimation, because the number of nearest neighbors used in the normal calculation effects the normal estimate quality [38]. Chaperon uses the 10 nearest neighbors. Mitra et al. developed a method to determine the best number of neighbors based on the noise of the point cloud data and the density and distribution of the samples [26]. The Mitra et al. method is very useful when fitting to a large data cloud, but since we have control over the data collection process, the computationally expensive collection and processing of cloud data is unnecessary. A better method for automatically choosing  $k$  was developed to optimize the data collection process without involving the user.

In addition to the number of neighbors, three other properties influence the normal calculation accuracy: the neighborhood shape, the characteristic distance describing



the neighborhood, and the noise in the data points. Two neighborhood shapes were explored, a checkered pattern and a circular pattern. The number of neighbors,  $n_n$ , is either evenly spaced in the checkerboard or evenly distributed around the circle as shown in Figure 3-5 a and b, respectively. The characteristic length,  $L$ , is defined as the circle radius or the diagonal distance from the center to the corner of the checkerboard.

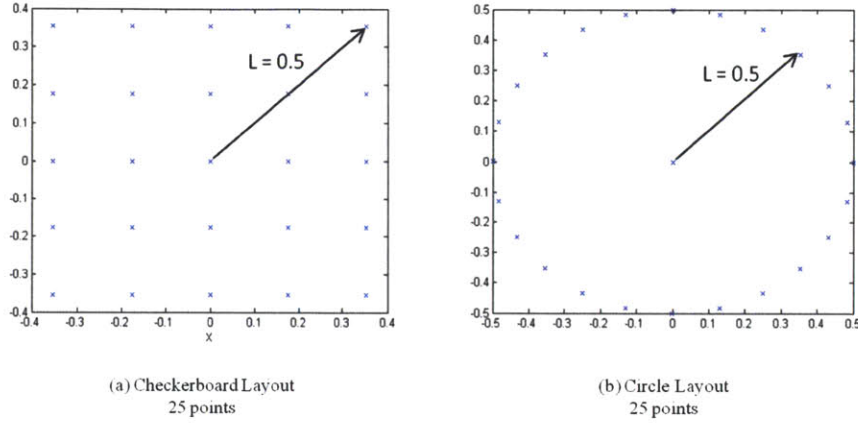


Figure 3-5: Possible layouts for neighborhood points collection. a) a checkerboard layout with points evenly distributed along both axes b) a circular layout with points evenly distributed around the circle perimeter.

Simulations varying  $n_n$ ,  $L$ , and the noise in the points were conducted for both shapes to find the relationship between the parameters and fit accuracy. Gaussian noise with zero mean and a standard deviation of  $\sigma_z$  is added to the  $z$  component of each data point. Accuracy was measured as the standard deviation of the angle between the fitted plane and the  $z$  axis,  $\sigma_{circle}$  and  $\sigma_{checker}$ . Figure 3-6 shows the fit accuracy over a range of parameter values for both the circular and checkerboard layouts.

Figure 3-6 shows that for the same number of points and same characteristic length, the circular layout gives better accuracy. Fitting a plane to the accuracy data results in Equ. (3.7) and Equ. (3.8).

$$\sigma_{circle} = \frac{2.26\sigma_z}{N^{0.56}L^{0.96}} \approx \frac{2.26\sigma_z}{N^{0.56}L} \quad (3.7)$$

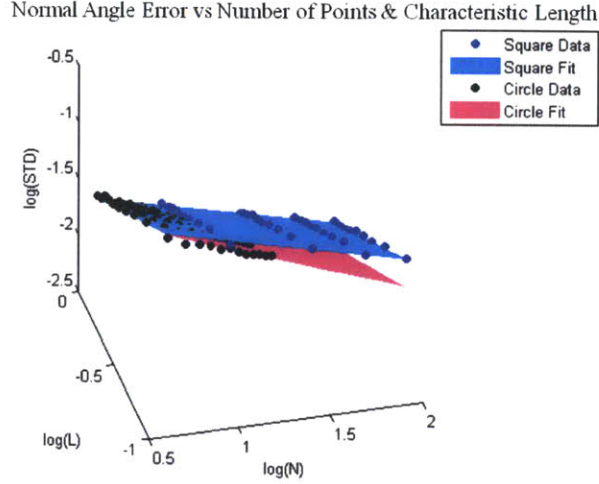


Figure 3-6: Standard Deviation in Normal Angle, STD, vs the number of points (N) and characteristic length (L) of a circular point layout and square checkerboard layout for noise standard deviation of 0.02. The equations for the planar fits are shown in Equ. (3.7) and Equ. (3.8).

and

$$\sigma_{checker} = \frac{1.69\sigma_z}{N^{0.38}L^{0.99}} \approx \frac{1.69\sigma_z}{N^{0.38}L} \quad (3.8)$$

Simple analysis suggests that the error should vary with  $L^{-1}$ , which is approximately observed in the fits above, thus for the later analysis, the exponent of L is assumed to be -1.

From a purely plane fitting perspective the largest number of neighbors and largest characteristic length give the best accuracy, but the largest values are not the best option when the algorithm method and fitting subject are taken into consideration. Too large a neighborhood radius,  $r_n$ , will smooth the resulting normal values. In the case of the algorithm proposed here, it would cause the walls to fit to a limited number of planes instead of the large variety of directions that the histogram method needs to find the floor plane. The number of neighbors,  $n_n$ , is directly related to the algorithm speed, thus the accuracy of having more points must be balanced with the cost in speed to collect and process these points. An optimization analysis was completed to determine the best values for  $r_n$  and  $n_n$ .

The optimal values of  $r_n$  and  $n_n$  are calculated from the desired fit accuracy, the distance sensor noise standard deviation, the encoder resolution, and the tank

Table 3.1: Required User Inputs

Input	Variable	Units	Description
Desired Fit Accuracy	$a$	m	Error that is acceptable in the fit parameters
Distance Sensor Error	$\sigma_s$	m	Manufacturer quoted sensor resolution (Standard Deviation)
Encoder Resolution	$\theta_e$	rad	Encoder step resolution in radians
Rough Tank Height Estimate	$h_r$	m	A rough estimate of the tank height
Rough Tank Diameter Estimate	$d_r$	m	A rough estimate of the tank diameter

height or radius. These required user inputs are shown in Table 3.1. The desired fit accuracy is the acceptable error in the measurement. A user input of 1 m means all the fit values should be accurate to within 1 m. Distance sensor error,  $\sigma_s$ , is also known as the sensor repeatability. The encoder resolution is the smallest size the encoder can read in radians. The user must also input rough estimates of the tank height,  $h_r$ , and diameter,  $d_r$ . These values only need to be rough estimates of the appropriate magnitude since they are only used during the data collection parameter selection. During the fitting process, the data extents replace these estimated values. The advanced user can also choose to bypass the automatic determination of  $r_n$  and  $n_n$  and the number of neighborhoods,  $n$ , by entering them directly. Table 3.2 provides details about the advanced inputs.

Table 3.2: Optional or Derived User Inputs

Input	Variable	Units	Description
Neighborhood Radius	$r_n$	m	Usually calculated as 0.2 times the smaller of the tank size estimates
Number of Neighbors	$n_n$	—	Number of neighbor points in each neighborhood. Usually found from the $a$ , $r_n$ and $e$
Number of Normals/ Neighborhoods	$n$	—	Number of neighborhoods (number or normals). Input as the number of evenly distributed points per side on a cube (default $7 = 218$ normals)

The total system error, used in determining  $r_n$  and  $n_n$ , is a combination of the



error from the encoder on both axes and the distance sensor error. Using RMS, the total error standard deviation is found to be:

$$e = \sqrt{2(\sigma_e^2) + \sigma_s^2} \quad (3.9)$$

where  $\sigma_e$  is the standard deviation of the worst case measurement error caused by the encoder resolution.

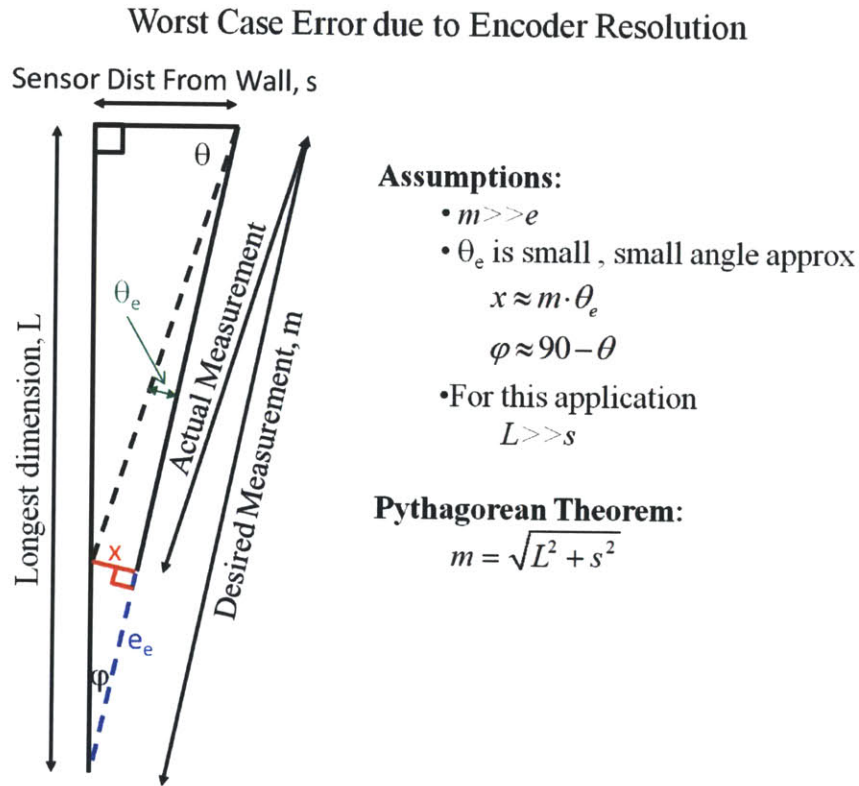


Figure 3-7: Encoder resolution effect on distance measurement error. The actual position of the axis can be anywhere within the step size of the encoder resulting in an angular error and mis-measurement of the distance,  $e_e$

The worst case error cause by the encoder resolution,  $e_e$ , is found from the geometry of the sensing platform and tank, which is detailed in Figure 3-7.  $\theta_e$ , is the worst possible angular error caused by the resolution of the encoder. This should be a very small value, so small angle approximations are valid. The distance of the sensor from the walls,  $s$  is determined by the geometry of the sensor platform and should be inserted into the program during the sensor platform fabrication.  $s$  is assumed to

be much smaller than the longer of  $h_r$  and  $r_r$ , here referred to as  $L$ .  $e_e$  is determined from trigonometry, the Pythagorean theorem and the small angle approximations to be:

$$e_e = \frac{L^2 \cdot \theta_e}{s} \quad (3.10)$$

Dividing  $e_e$  by three, approximates the standard deviation of the error,  $\sigma_e$ . Setting the standard deviation to 1/3 the desired max error value generates a distribution that is within the desired range 99% of the time. Combining Equ. (3.10), Equ. (3.9) and the user inputs, the total error of the system is found to be:

$$e = \sqrt{2 \left( \frac{\max(h_r, d_r)^2 r_e}{3s} \right)^2 + \sigma_s^2} \quad (3.11)$$

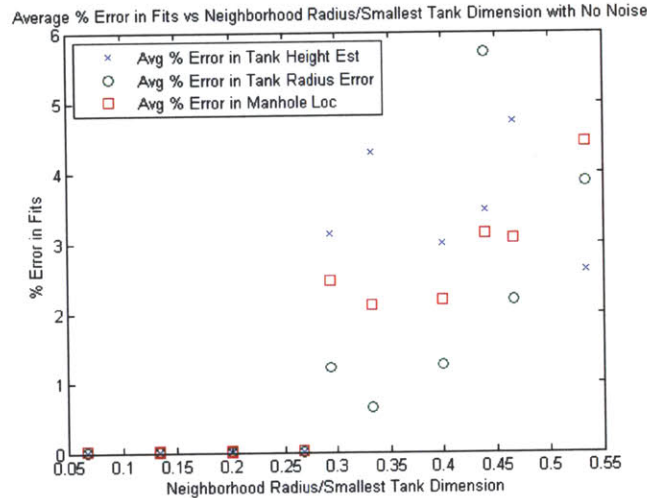


Figure 3-8: Simulated average error in cylinder fits vs the neighborhood radius as a fraction of either the diameter or height, whichever is smallest. This was simulated without noise to isolate the effects due to smoothing. A safe upper limit for neighborhood radius size is approximately 20% of the smaller tank dimension

The limits on the size of  $r_n$  are imposed by  $h_r$ ,  $r_r$  and  $e$ .  $r_n$  needs to be relatively large in comparison to  $e$  to achieve accurate normal fits.  $r_n$  needs to be smaller than the smaller of  $h_r$  and  $r_r$  to prevent smoothing of the wall normals and in the tank corners. Too large a radius will result in enough smoothing that the errors in normal fits will be too large to meet the desired accuracy. Figure 3-8 shows the effect of

varying  $r_n$  on the fit accuracy. The simulations show a clear increase in error as the size of the radius increases in comparison to the tank size. A safe value of 0.2 times the smaller tank dimension is used in the algorithm for  $r_n$ .

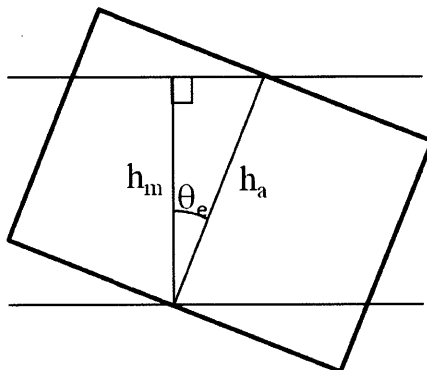


Figure 3-9: Derivation of allowable angular error,  $\theta_e$  from input allowable accuracy,  $a$ . Difference between  $h_m$  and  $h_a$  must be equal to or less than  $a$ .

The optimal  $n_n$  can be found from the desired accuracy,  $a$ , and the error,  $e$ . Three is the absolute minimum value for  $n_n$  because that is the number of points needed to define a plane. More than three neighbors increases the accuracy as the noise is averaged out, but it takes more time to run. The optimal  $n_n$  is the minimum number of neighbors for which the algorithm will achieve the desired accuracy or three, whichever is larger.  $a$  must be converted to the allowable angular error of the estimated tank floor normal and the actual tank floor normal,  $\theta_e$ , to determine the  $n_n$  needed to achieve the desired accuracy, . The geometric relationship is shown in Figure 3-9, where  $h_a$  is the actual height of the tank and  $h_m$  is the measured height of the tank. The difference between  $h_a$  and  $h_m$  must be less than or equal to the desired accuracy,  $a$ .  $\theta_e$  can be calculated using Equ. (3.12).

$$\theta_e = \cos^{-1} \left( 1 - \frac{a}{2} \right) \quad (3.12)$$

$a$  is divided by 2 in Equ. (3.12) to make  $\theta_e$  a standard deviation. Usually this is done by dividing by 3, but 2 was chosen after testing because it speeds up the algorithm and still mimics the standard deviation up to 95% instead of 99%. Plugging  $\theta_e$  into

Equ. (3.7) results in an equation for  $n_n$ .

$$\eta = \left( 2 * \frac{2.26e}{r_n \theta_e} \right)^{1.79}$$

$$n_n = \begin{cases} \eta, & \eta > 3 \\ 3, & \eta \leq 3 \end{cases} \quad (3.13)$$

In addition to  $n_n$  and  $r_n$ , the number of neighborhoods,  $n$  is also important to the accuracy of the fit and speed of the algorithm. More normals will result in better accuracy, but significantly increase the time to find the fit because for each neighborhood (each normal) there are  $n_n + 1$  points. Two factors significantly impact the number of normals needed to obtain a good fit: the ratio of cap area to the wall area and the placement of the sensor.

A large cap area to wall area ratio will need less neighborhoods than a small ratio. The algorithm depends on there being more normals pointing in the direction of the floor than the walls. More neighborhoods need to be found using points from the capped end than on a vertical patch of wall. There is no way to predict the ratio of normals found on the cap to those found on a vertical patch of wall because there is no known information about the orientation of the tank. In general, calculating more neighborhoods will result in more normals along the floor than will repeat from the walls. Approximately 50 normals are needed when the sensor is placed in a decent location in the tank.

The goodness of a sensor location can be defined by how much of its view is obstructed. A good location is one that has the majority of its view unobstructed. A location in the center a large patch of space, such as the center of the roof, will still result in a good fit when using fewer normals. Difficult locations are those in the corners where a large portion of the view is obstructed. More normals are needed to get enough floor normals for the histogram when the sensor is placed in a corner.

Simulations of the resulting data collection when the sensor is placed in various locations within a tank are shown in Figure 3-10. Upwards of 200 normals are needed if the sensor is in the extreme corner of the tank or if the ratio of cap to wall is very small. The algorithm presented here will default to 218 neighborhoods (7 normals per

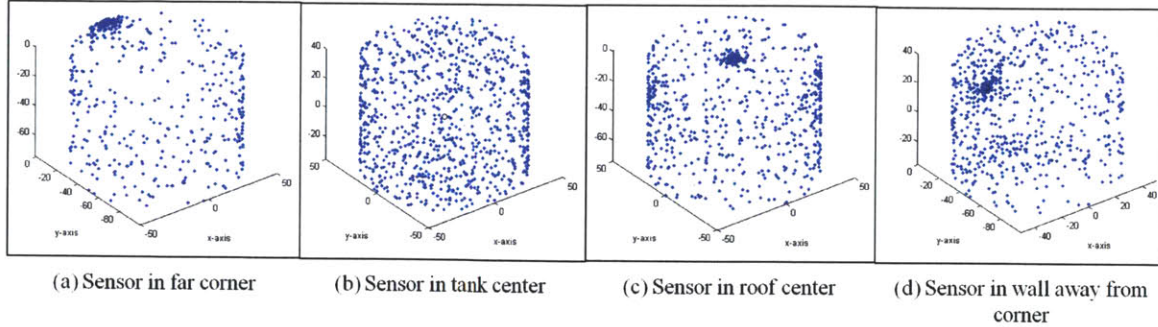


Figure 3-10: Effect of sensor placement on data collection. As shown in a, c and d, data points collect around the sensor when a portion of the sensor view is obstructed. The best data is evenly distributed about the cylinder. The best place for the sensor is to be placed in the center of the tank as shown in b. The worst place is in the corner of the tank as shown in a.

side evenly distributed about a cube centered on the sensor). The user can manually lower  $n$  to decrease the run time if the sensor is known to be placed in a decent place in a tank with a good ratio.

The equations described above were developed to find the best fit despite the location and orientation of the sensors.  $n_n$  and  $r_n$  can be derived using these equations, or they can be modified directly to optimize the algorithm based on the application requirements and features. Just like  $n$ ,  $n_n$  can be decreased if the user knows that the sensor is placed in a decent location. The opposite can also be done; the number of neighbors and neighborhoods can be increased if the user is unhappy with the fit. Despite these variables being optimized for robustness at the expense of speed, the algorithm maintains an edge over other methods because of the non-iterative segmentation described in the next section.

The last step in the normal collection algorithm is converting the  $n$  POIs to locations in space and converting  $n_n$  and  $r_n$  to angle commands for each motor. The  $n$  POIs are evenly distributed across the surface of a unit cube centered about the sensor. This will not result in a perfectly uniform distribution over a cylinder in space; however, it does result in the best achievable distribution given what little information is known about the tank. For each POI a distance is measured from which the neighbor point locations are calculated. Figure 3-11 shows the projection from  $r_n$  onto a plane tangent to the unit sphere surrounding the sensor location and



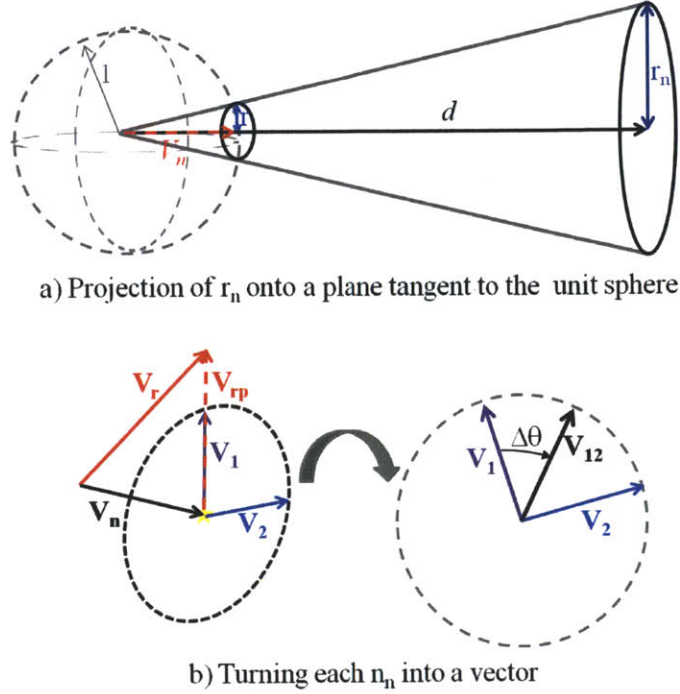


Figure 3-11: Using  $n_n$  and  $r_n$  to find commands for sensor platform. a) The projection of the desired neighborhood circle onto a unit sphere surrounding the sensor. b) The vectors used to calculate the neighbor locations.

the vectors involved in converting each point location to  $\theta$  and  $\phi$ . The process is as follows: First,  $r_n$  is converted to a radius  $r$  on the unit sphere using triangular relationships. The resulting radius is:

$$r = \frac{r_n}{\sqrt{r_n^2 + |\vec{d}|^2}} \quad (3.14)$$

The angle between each point on the circle is calculated from Equ. (3.15).

$$\Delta\theta = \frac{2\pi}{n_n} \quad (3.15)$$

A unit vector in the direction of the center of the neighborhood,  $\vec{V}_n$ , is found by dividing the measured distance to the POI by its length:

$$\vec{V}_n = \frac{\vec{d}}{|\vec{d}|} \quad (3.16)$$

3 random numbers between 0 and 1 are combined into a random vector,  $\vec{V}_r$ .

$$\vec{V}_r = [\text{rand}(0 - 1), \text{rand}(0 - 1), \text{rand}(0 - 1)] \quad (3.17)$$

Next,  $\vec{V}_r$  is projected onto a plane parallel to the plane containing  $r_n$  but tangent to the unit sphere by subtracting the components along  $\vec{V}_n$ . The result is  $\vec{V}_{rp}$  as shown in Equ. (3.18).

$$\vec{V}_{rp} = \vec{V}_r - (\vec{V}_n \bullet \vec{V}_r) \vec{V}_n \quad (3.18)$$

$\vec{V}_{rp}$  is converted to a unit vector,  $\vec{V}_1$ .

$$\vec{V}_1 = \frac{\vec{V}_{rp}}{|\vec{V}_{rp}|} \quad (3.19)$$

A second unit vector in the circle is found by taking the cross product of  $\vec{V}_1$  and  $\vec{V}_n$ .

$$\vec{V}_2 = \vec{V}_1 \times \vec{V}_n \quad (3.20)$$

Now that two perpendicular unit vectors are known, the plane of the circle intersecting the unit sphere is fully defined.  $\vec{V}_1$  defines the first axis corresponding to 0 degrees around the circle. The coordinates of each neighborhood point are found using Equ. (3.21).

$$\begin{aligned} \vec{V}_c &= \vec{V}_n + \vec{V}_1 r \cos \Delta\theta_i + \vec{V}_2 r \sin \Delta\theta; \\ i &= [0 : n_n] \end{aligned} \quad (3.21)$$

These points are converted from the Cartesian coordinate frame to the spherical coordinate frame. Theta and phi can then be sent to their respective axes.

For the work described in this thesis, a circular neighborhood collection process is used with values of  $r_n$ ,  $n_n$  and  $n$  found using the equations described above. A summary of the steps for the optimized data collection process are in red and purple in Figure 3-4 and are as follows:

1. POI locations are calculated

2. Gimbal moves to the POI  $\theta$  and  $\phi$
3. Measure the distance,  $d$ , and actual  $\theta$  and  $\phi$
4. Calculate the neighbor locations for this POI
5. Gimbal moves to the neighbor  $\theta$  and  $\phi$
6. Measure the distance,  $d$ , and actual  $\theta$  and  $\phi$
7. Repeat steps 5 and 6 until all neighbors' data is collected
8. Calculate the normal for the neighborhood
9. Repeat steps 2-8 for each POI

If the user already has a cloud of data, the  $n$  neighborhoods can be found using the same  $r_n$ . The process would be as follows:

1. Randomly select  $n$  data points from the data cloud
2. Find all points within  $r_n$  distance from the POI in the data cloud
3. Calculate the normal for the neighborhood
4. Repeat steps 2 and 3 for each POI

Whether the data collection process described in this section is used or cloud data is processed as described above, the resulting data to be input to the segmentation algorithm includes  $n$  normals to the surfaces and  $n(n_n + 1)$  points on the surface.

### 3.2.2 Segmentation

This section describes the segmentation of the data into groups corresponding to the floor, roof and walls. Segmentation is shown in blue in Figure 3-4. The first step is to find the cylinder  $Z$  axis in the  $\theta\phi z$  coordinate frame by creating a histogram of the Gaussian sphere. The next step is to calculate the rotation matrix that will align the  $Z$  axis with the  $z$  axis. In other words, this step finds the orientation of the sensor



as it relates to the orientation of the tank and then aligns the two. The last step is to separate the data into the floor, roof and walls by using knowledge of the shape, rotated orientation and extents of the data.

The unique ability of this algorithm to fit an arbitrarily rotated capped cylinder without iteration comes from the unique method of segmentation. Conventional segmentation methods do not take into consideration any knowledge about the item they are segmenting. Not taking into consideration any of this information makes them very flexible in what they can fit. The disadvantage is that to complete a fit, they must iterate through many possible solutions. This is computationally expensive and time consuming. The algorithm here is different in that it is limited to one shape, but that it can be done without iteration, thereby greatly reducing computational load and increasing speed.

### Gaussian Image of a Right Cylinder

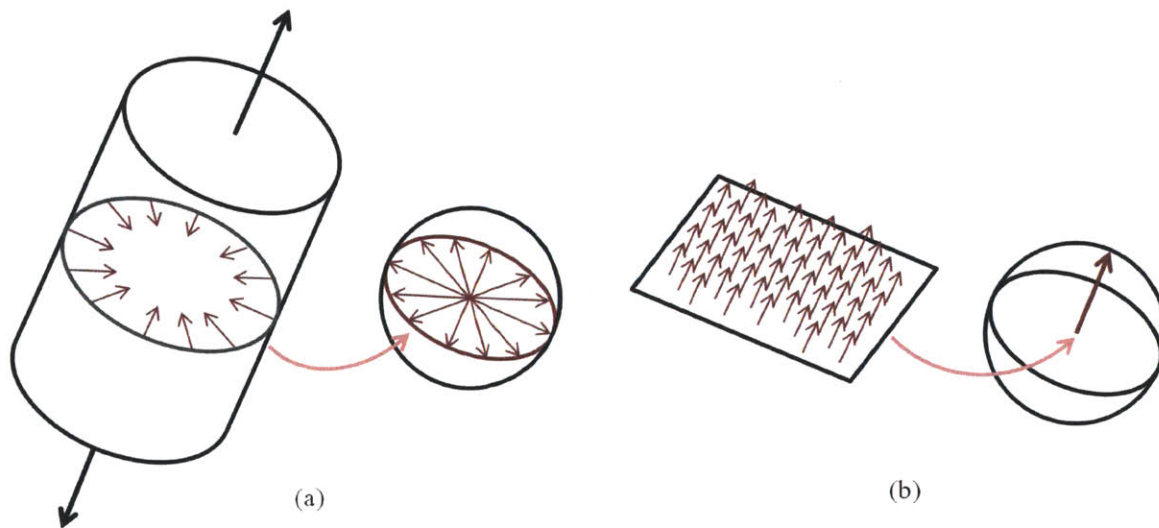


Figure 3-12: Gaussian mapping of an uncapped cylinder (a) and a plane (b) onto the Gaussian sphere.

The Gauss image is formed through the placement of the ends of normals, collected from a surface, onto the origin. These normals then intersect a unit sphere, referred to as the Gaussian sphere, forming a Gauss map [41]. The result of the entire

surface being mapped is referred to as the Gaussian image. The Gaussian image of an uncapped cylinder is a circle on the Gaussian sphere as shown in Figure 3-12a. The mapping of a plane onto the Gaussian sphere is a repeated single point as shown in Figure 3-12b.

A capped cylinder is a combination of two planes and a cylinder. This means one can expect the Gaussian image to be a combination of the two images shown in Figure 3-12; the cylindrical walls will have a wide spread of points while the floor and roof planes will result in many repeated points on the Gaussian sphere. It is easy to identify the normal to the floor plane because it is the normal corresponding to the most common repeated point on the Gaussian sphere.

### 3D Histogram to Find Caps

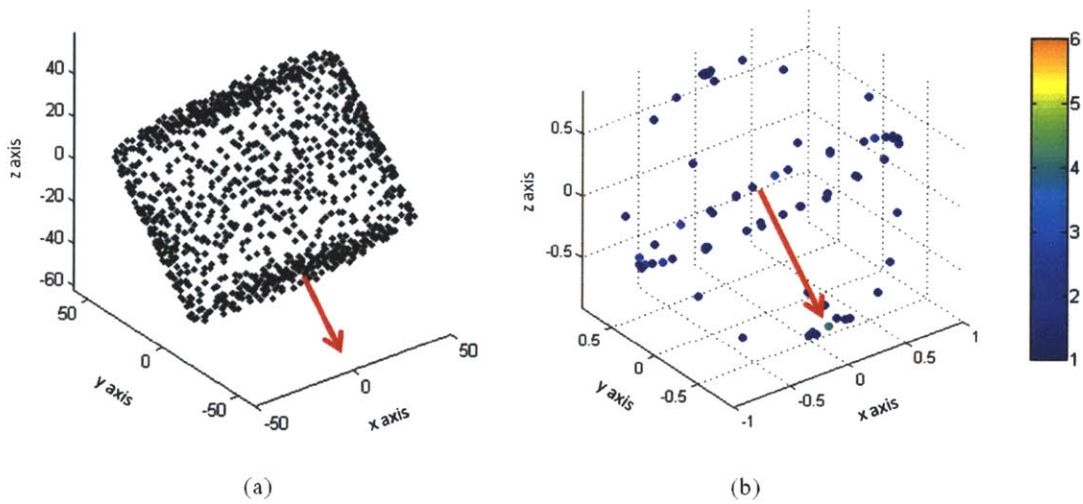


Figure 3-13: (a) Simulated raw data with arrow indicating floor normal. (b) 3D Histogram of the Gaussian mapping of a capped cylinder. Note the one point (at the end of the arrow) where the color difference indicates many normals is in line with the expected floor normal

The coordinates of the points on the Gaussian sphere are collected and sorted into a 3D histogram to identify the  $Z$  axis. The histogram is composed of bins corresponding to the coordinates of the points on the Gaussian sphere. The histogram bin containing the most votes will correspond to the normal of the cap. Figure

3-13 shows simulated raw data of a tank and a histogram of the normals on the Gaussian sphere. The bright blue point indicates multiple normals pointing in that direction, which corresponds to the normal to the tank floor as expected. The ring corresponds to the normals drawn on the circular walls of the tank. The noise-like points distributed over the unit sphere are drawn from plane fits in the corners of the tank. The most common point corresponds to the normal of the tank cap. This normal,  $n_c$ , lies along the  $Z$  axis of the cylinder coordinate frame.  $n_c$  is composed of the three coordinates,  $x_i$ ,  $y_i$  and  $z_i$ , corresponding to the bin with the most votes, the  $i$ th bin.

$$n_c = [x_i, y_i, z_i] \tag{3.22}$$

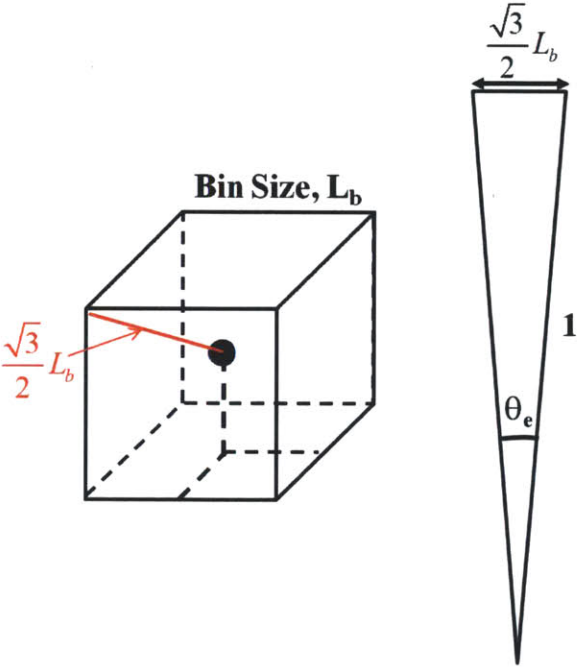


Figure 3-14: Deriving bin size from  $\theta_e$ . Worst case error is from the center of the bin to its corner. Bins are arranged at a unit distance from the center of the sphere, thus the arm length is 1.

The histogram bin size is derived from the desired accuracy. The worst resulting offset needs to be smaller than the allowable angular error from Equ. (3.12). The worst possible offset for a bin is from the center to the corner of the bin as shown in Figure 3-14. The histogram bins are going to occupy space around the center

of Gaussian sphere where all the vectors are unit vectors. The arm of the triangle shown in Figure 3-14 corresponds to that distance. A small angle approximation is assumed for  $\theta_e$ . Using trigonometry and the listed assumptions, the bin length  $L_b$  can be calculated to be:

$$L_b = \frac{2\theta_e}{\sqrt{3}} \quad (3.23)$$

### Rotation Matrix Calculation

The sensor  $z$  axis is aligned with  $Z$  through the use of rotation matrix. The axis of rotation is found from:

$$axis_{rot} = [0, 0, -1] \times n_c \quad (3.24)$$

and the angle about which to rotate is found from:

$$angle_{rot} = [0, 0, -1] \bullet n_c \quad (3.25)$$

A rotation matrix formed from  $axis_{rot}$  and  $angle_{rot}$  orients the cylinder allowing for the non-iterative segmentation. Note:  $n_c$  is compared to the -1 direction because from the perspective of the sensor the distance to the floor is in a negative direction. This means the -1 axis must be used to keep the orientation correct.

### Region Segmentation

The data needs to be broken down into sections that can be used in the fitting process. Figure 3-15 shows the three sections corresponding to the floor, roof and walls. To obtain an accurate fit, these sections need to be clear of any points corresponding to the other sections. This is accomplished by defining a value above which the data will correspond to the roof  $h_{rc}$ , a value below which the data will correspond to the floor  $h_{fc}$ , and a radius creating a boundary between wall and cap data  $r_c$ . The segmentation is done according to the Equ. (3.26). The roof contains all points whose  $z$  components are greater than  $h_{rc}$  and whose radial distance from the center of the data is less than  $r_c$ . The floor contains all points whose  $z$  components are less than  $h_{fc}$  and whose radial distance from the center of the data is less than  $r_c$ . The



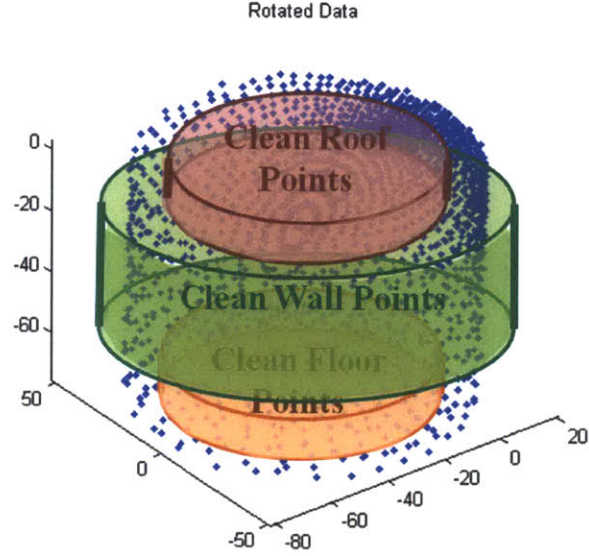


Figure 3-15: Isolation of the floor, roof and wall data sets. The only points excluded are those from the corners that are ambiguous as to which group they belong

wall contains all points whose  $z$  components lie between  $h_{fc}$  and  $h_{rc}$  and whose radial distance from the center of the data is greater than  $r_c$ .

$$\begin{aligned}
 roof &= (p_z > h_{rc}) \& \left( \sqrt{(p_x - mid(x))^2 + (p_y - mid(y))^2} < r_c \right) \\
 floor &= (p_z < h_{fc}) \& \left( \sqrt{(p_x - mid(x))^2 + (p_y - mid(y))^2} < r_c \right) \\
 wall &= (h_{fc} < p_z < h_{rc}) \& \left( \sqrt{(p_x - mid(x))^2 + (p_y - mid(y))^2} > r_c \right)
 \end{aligned} \tag{3.26}$$

The clean heights,  $h_{rc}$  and  $h_{fc}$ , and clean radius,  $r_c$ , are calculated by modifying the 5th and 95th percentiles by encoder and sensor error values. The clean values will consist of the length of the tank in the dimension of interest plus a  $\delta$  value. The  $\delta$  is derived from the amount of tip in the tank allowed by the allowable angular error from Equ. (3-9) as shown in Figure 3-16. The values of the radial delta,  $\delta r$ , and the height delta,  $\delta z$ , are calculated using Equ. (3.27) and Equ. (3.28).

$$\begin{aligned}
 \Delta d &= \delta r + \delta z \cot \theta_e \\
 \Delta z &= \delta z + \delta r \cot \theta_e
 \end{aligned} \tag{3.27}$$

$$\begin{bmatrix} \delta r \\ \delta z \end{bmatrix} = \begin{bmatrix} 1 & \cot \theta_e \\ \cot \theta_e & 1 \end{bmatrix}^{-1} \begin{bmatrix} \Delta d \\ \Delta z \end{bmatrix} \quad (3.28)$$

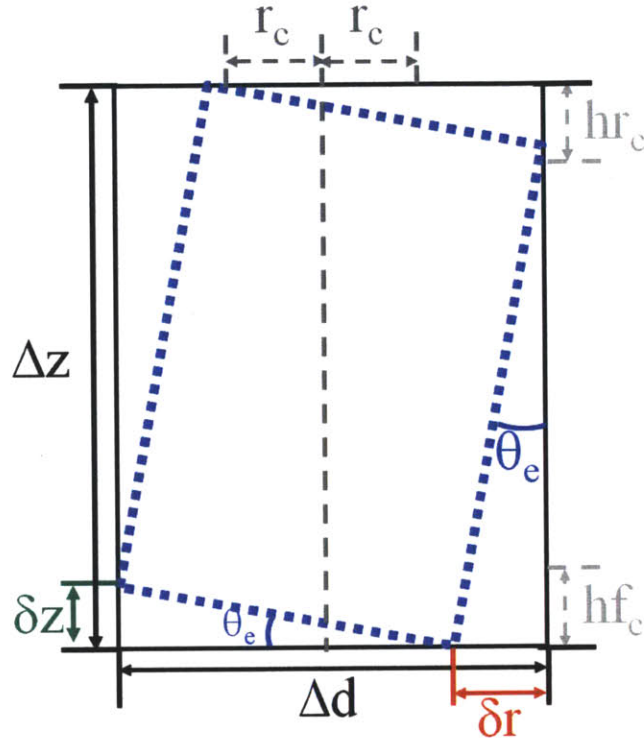


Figure 3-16: Derivation of  $\delta z$  and  $\delta r$ . The tank is tipped by  $\theta_e$ .  $\Delta d$  and  $\Delta z$  are the extents of the data in the corresponding direction.

An additional modification is based on the error of the sensing platform. 10 times the error derived in Equ. (3.11) is used as the modifying value. The value of 10 noise standard deviations is drawn from Gaussian statistics. Less than one point in three million will lie beyond  $\pm 5$  standard deviations, ensuring that for a standard tank measurement set of one to ten thousand points, wall points will be retained in the floor data set less than 1% of the time.

The equation for the final clean values used in Equ. (3.26) are shown in Equ. (3.29). For the roof clean,  $\delta z$  and  $10e$  are subtracted from the 95th percentile height. For the floor these values are added to the 5th percentile height. For the radial clean value, the larger of the 5th to 95th percentile ranges in the x and y directions is used

as the reference diameter,  $d_d$ , as shown in Equ. (3.30).  $\delta r$  and  $10e$  are subtracted from half  $d_d$ .

$$\begin{aligned} r_c &= \frac{d_d}{2} - \delta r - 10e \\ hr_c &= \text{quantile}(z, 0.95) - \delta z - 10e \\ hf_c &= \text{quantile}(z, 0.05) + \delta z + 10e \end{aligned} \tag{3.29}$$

$$\begin{aligned} d_x &= \text{quantile}(x, 0.95) - \text{quantile}(x, 0.05) \\ d_y &= \text{quantile}(y, 0.95) - \text{quantile}(y, 0.05) \\ d_d &= \max(r_x, r_y) \end{aligned} \tag{3.30}$$

The segmentation portion of the algorithm is complete when the data is segmented into the roof, floor and wall sections. Not all points are contained within a segmented set. The points around the intersection between the wall and the caps are not included because the section they should belong to is ambiguous. Despite these points' exclusion, there will be enough points in each set to accurately fit a plane or circle.

### 3.2.3 Fitting

Each data set is fit to its corresponding shape using batch least squares. The floor and roof data sets are fit to planes using the same process described in Equ. (3.2) through Equ. (3.6). A simplification of the wall data set is used to estimate the radius of the walls. By neglecting the z component of the data points identified as wall data, a simple batch least squares fitting of a circle can estimate the parameters of the wall. The equation for a circle is

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \tag{3.31}$$

This form is non-linear. The equation must be rewritten in a linear form to use batch least squares. The linear rearrangement is shown in Equ. (3.32).

$$\underbrace{\frac{1}{r^2 - x_0^2 - y_0^2}}_A (x^2 + y^2) + \underbrace{\frac{-2x_0}{r^2 - x_0^2 - y_0^2}}_B x + \underbrace{\frac{-2y_0}{r^2 - x_0^2 - y_0^2}}_C y = 1 \quad (3.32)$$

The  $A, B$  and  $C$  that make up  $\Theta$  are indicated below their respective expression. The new  $\Phi$  derived from Equ. (3.32) can be written as

$$\Phi = \begin{bmatrix} (x^2(1) + y^2(1)) & (x^2(2) + y^2(2)) & \dots & (x^2(n_n + 1) + y^2(n_n + 1)) \\ x(1) & x(2) & \dots & x(n_n + 1) \\ y(1) & y(2) & \dots & y(n_n + 1) \end{bmatrix} \quad (3.33)$$

Using the  $w$  from Equ. (3.5),  $\Theta$  from Equ. (3.3) and the updated  $\Phi$ , the batch least squares process defined in Equ. (3.6) can be completed. Once  $A, B$  and  $C$  are known the following expressions are used to find the properties of the circle fit:

$$\begin{aligned} x_0 &= \frac{-B}{2A}, \\ y_0 &= \frac{-C}{2A}, \\ r &= \frac{\sqrt{4A+B^2+C^2}}{2A} \end{aligned} \quad (3.34)$$

Fitting the planes and circle completely defines the tank extents. The floor and roof planes are allowed to be tipped to find the best fit to the data. The difference between the floor and roof planes at the  $xy$  location of the sensor, as found in Equ. (3.34), corresponds to the height of the tank. This is the place where the measurement will be vertical thereby removing and angle incidence bias in the measurements. Equ. (3.34) gives the radius of the tank. The location of the sensor is defined by the position of the center of the circle and the distance to the floor plane. The height, radius and location of the center of the tank mathematically define the cylinder. All three section fits and the final cylinder fit are shown in Figure 3-17.



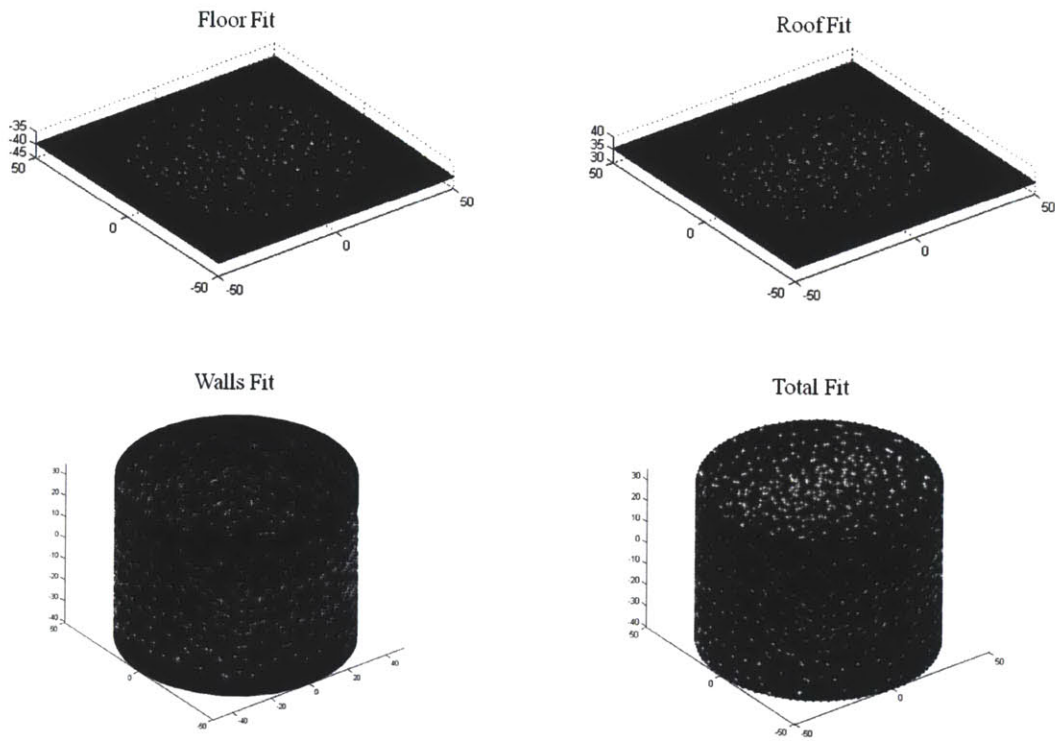


Figure 3-17: Cylinder Fitting Steps and Final Completed Fitting. The white stars are the data points, while the black is the fitted plane/cylinder. In the top row are the floor and roof fits from left to right. The bottom left is the cylinder fit. The bottom right is the trimmed and combined fit.

### 3.3 Algorithm Advantages and Limitations

The algorithm presented here is a robust, fast segmentation and fitting method. The combination of the object shape knowledge and the histogram make the algorithm very robust to noise. Many possible erroneous fits are not even considered as options because the general shape is known. The histogram also averages out noise, increasing the robustness of the fit. The use of object shape knowledge also allows for segmentation without iteration. The lack of iteration is unique to this algorithm and is what allows for its quick and efficient processing compared to other methods found in literature.

The one major limitation of the algorithm is that it is limited to data for right angled capped cylinders in its current configuration. To find a common normal for the floor the algorithm relies on the data consisting of flat caps and circular walls. The segmentation and fittings steps are also inflexible in the need for the planes and circle. The limitation is caused by all steps relying on object shape knowledge, the same thing that makes the algorithm capable of being robust and quick.

# Chapter 4

## Results

### 4.1 Evaluation Through Simulation

Initial testing of the algorithm was done using simulation. The distance measurements are mathematically simulated for given  $\theta$ s and  $\phi$ s. The sensor can be simulated in any starting orientation or location. Noise can be added to the distance sensor measurement and to the encoders. Simulations were done varying the sensor orientation, tank size, sensor location, and sensor noise values. Figure 4-1 show several simulated runs using tank dimensions similar to an LPG tank. The results of simulated testing show an algorithm that is robust to noise and fits a cylinder without iteration. The algorithm is able to handle the sensor starting in any orientation and a wide range of heights and radii.

The algorithm displayed some difficulty when the radius is significantly smaller than the height, similar to test run 3 in Figure 4-1. There are a significant number of normals around the sensor location that will fall within the same bin. The radius is so small that less normals are measured on the cap than are measured next to the sensor. This leads to a histogram peak for the bin containing the normals on the wall which is larger than bin containing the normal from the caps. The rotation is not completed correctly because the wrong normal is chosen. The algorithm is unable to fit the data to a cylinder because the rotation was not completed correctly. A filter can be implemented that removes points within a certain distance from the sensor

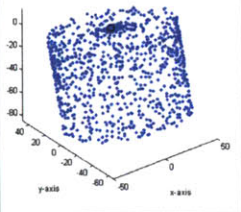
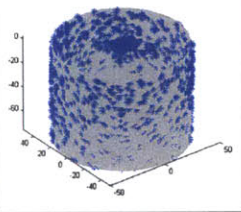
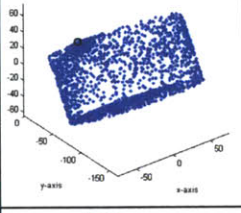
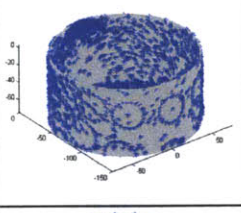
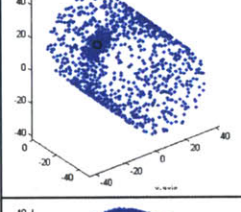
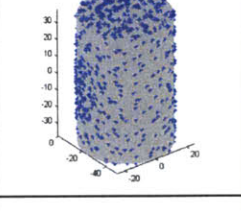
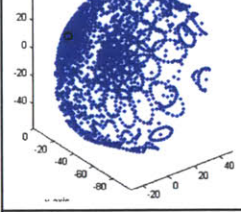
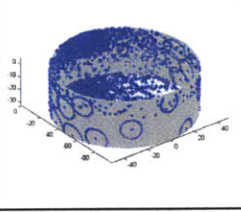
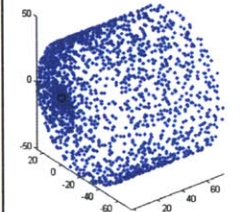
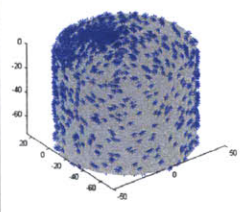
Raw Data	Fit	Real Dimensions	Fit Dimensions
		Height: 75m Radius: 50m Sensor Radius: 0 Sensor Height: 74 Des Acc: 0.05 Rotation: 15 deg	Fit Height 75.0071 Fit Radius: 49.995 Fit Sensor Radius: 0.3327 Fit Sensor Height: 74.0032
		Height: 75m Radius: 75m Sensor Radius: 74 Sensor Height: 74 Des Acc: 0.05 Rotation: 30 deg	Fit Height: 75.0055 Fit Radius: 75.0339 Fit Sensor Radius: 74.3520 Fit Sensor Height: 73.0976
		Height: 75 Radius: 25 Sensor Radius: 24 Sensor Height: 37 Des Acc: 0.05 Rotation 45 deg	Fit Height: 74.9983 Fit Radius: 25.0012 Fit Sensor Radius: 24.0009 Fit Sensor Height: 37.0005
		Height: 35 Radius: 50 Sensor Radius: 49 Sensor Height: 30 Des Acc: 0.05 Rotation: 60	Fit Height: 35.0131 Fit Radius: 50.0194 Fit Sensor Radius: 48.7260 Fit Sensor Height: 30.9139
		Height: 75 Radius: 50 Sensor Radius: 24 Sensor Height: 74 Des Acc: 0.05 Rotation: 90	Fit Height: 75.0152 Fit Radius: 50.0291 Fit Sensor Radius: 24.3413 Fit Sensor Height: 73.7099

Figure 4-1: Raw data, fits and fit accuracies of varying simulated tests. The simulated sensor was rotated by 15, 30, 45, 60 and 70 deg with varying tank heights, diameters and manhole locations. The manhole locations are indicated by the black circle in the raw data image.

which would fix the problem by removing the duplicates around the sensor that are influencing the normal selection.

A second problem shown in simulation is that the sensor location is often not within tolerance. The sensor radial location is determined by the walls circle fit. There is some allowable angular error based on the desired accuracy that allows the tank to remain slightly tilted. The slight angularity of the tank results in the data projection forming an ellipse of data as shown in Figure 4-2. The cylinder is slightly rotated about the line which passes through the points where the expected and estimated circles intersect. The algorithm fits a circle to the center of the skew, resulting in an offset of the center location. The sensor location is found within tolerance when the tank is rotated into a perfectly vertical orientation and no skew is observed. The slight tip and misplacement of the sensor horizontal location may also be the cause of the sensor height not meeting the desired accuracy as the height measurement is dependent on the location of the sensor.

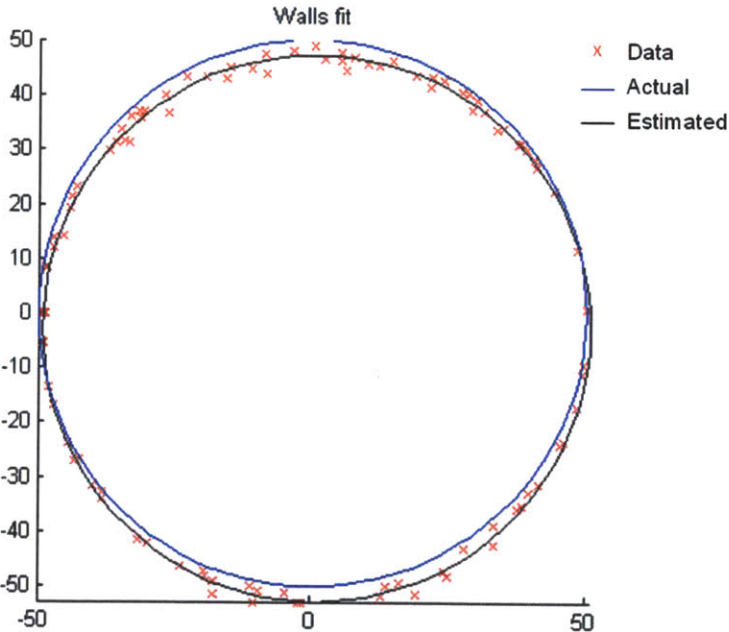


Figure 4-2: Skewed circle data. The blue line indicates the expected circle, while the black shows the resulting circle fit from the algorithm. The line passing through the points where the two circle intersect is the axis the cylinder is rotated.



There are two modifications to the code that would remove the skew effects from the sensor location values: fitting a 3D cylinder instead of a circle, or completing a second rotation based on the floor plane normal. A non iterative cylinder fit, like the one described by David Eberly, can be completed because the cylinder central axis is the same as the normal of the floor [42]. Another option would be to complete a second rotation where the rotation axis and rotation angle are calculated using the floor fit's normal. Using the plane normal would result in a more accurate orientation of the tank because there is no loss of resolution that occurs with the histogram normal selection process. The skew will be removed from the data when the cylinder is more vertically aligned and the location of the sensor will be within tolerance.

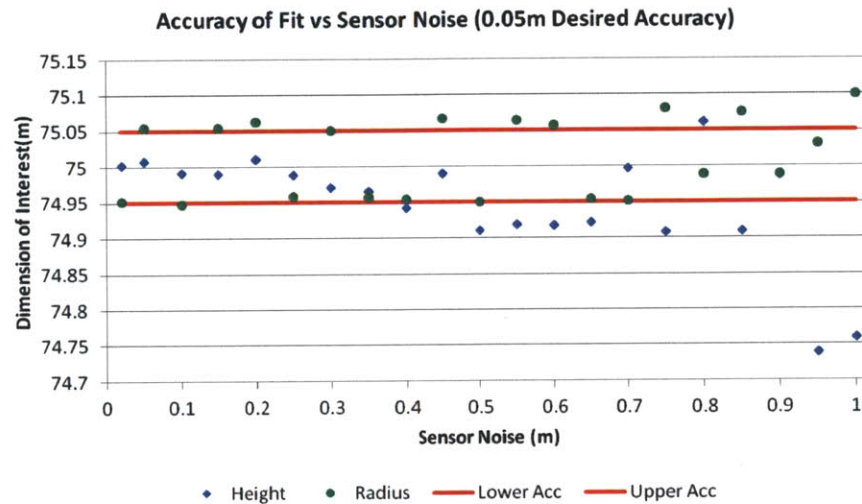


Figure 4-3: Accuracy versus sensor noise. The desired accuracy for the tests that make up the chart was 0.05m. The accuracy is within tolerance for noise levels up to the desired accuracy and sometimes above.

The simulated tests show that the algorithm is robust to noise. The algorithm uses the majority of the floor and roof data to make the respective plane fits, resulting in height measurements that are robust to measurement noise. The robustness is due to the averaging effect of fitting large data sets. The circle fit also uses the majority of the available data points, thus the radius fit is also robust to noise. Figure 4-3 shows that the noise can be as large as the desired accuracy, and occasionally even larger, and still result in a fit that is within the desired accuracy bounds. The estimated

radius gets worse much faster than the estimated height. This may be because of the skewed circular data as discussed above.

Speed was one of the main driving motivations for the algorithm requirements as described in Chapter 3. Removing iteration was the main method used for speeding up the processing. Figure 4-4 shows that the processing speed grows exponentially as the number of points increase. The segmentation and fitting only takes a few seconds to process upwards of 10000 points despite the exponential growth. A few seconds is significantly faster than time it will take to collect the data points, thus segmentation and fitting will not be the limiting factor in a mapping system.

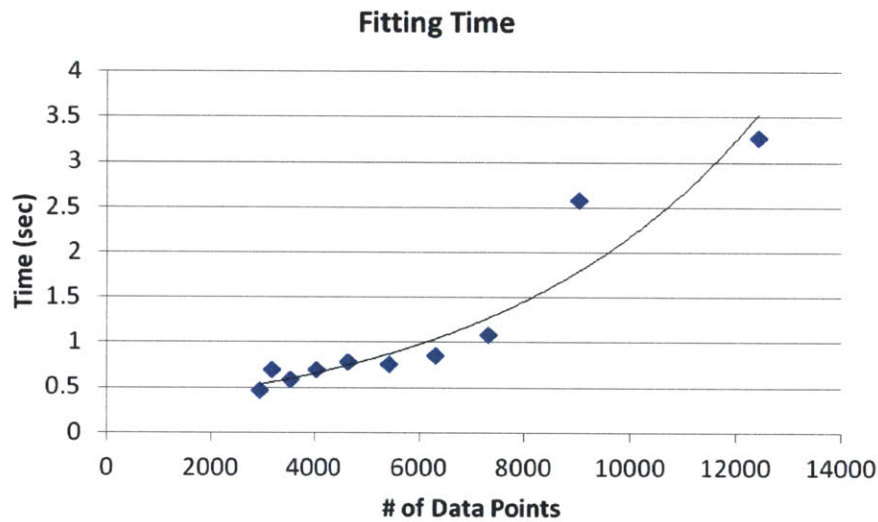


Figure 4-4: Algorithm segmentation and fitting time as a function of the number of points. It shows an exponential increase. Even above 10000 points, the time to process the data points is only a few seconds

## 4.2 Prototype Hardware

A prototype sensor platform was fabricated to test the algorithm in the laboratory. The platform is shown in Figure 4-5. It consists of a nested U structure. The outer U rotates about the center of its bottom, the  $\theta$  axis. It is driven by a Maxon RE 50 motor [43]. The inner U rotates about an axis intersecting the two arms. This axis corresponds to  $\phi$  and is driven by a Maxon EC-Max 40 motor [44]. Both axes

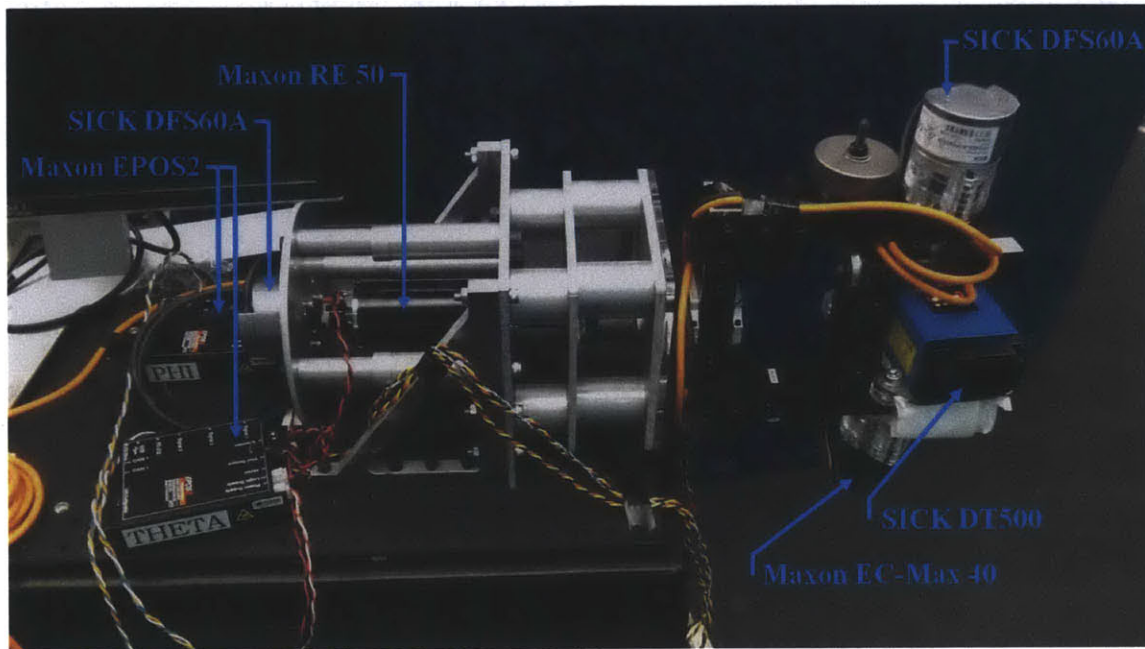


Figure 4-5: Prototype Gimbal. Consists of a nested U shaped structure driven by a Maxon EC-Max 40 and RE 50 motor, controlled by Maxon EPOS2 controllers. The rotational sensors are SICK DFS60A 65536 count encoders. The distance sensor is a 17m range SICK DT500.

rotational positions are measured with a SICK DFS60A, a 65536 line encoder [45]. Quadrature encoding using this encoder creates steps of  $24 \mu\text{rad}$ . Both motors are controlled by Maxon EPOS2 positioning controllers [46]. The distance measuring sensor used in the prototype platform is a SICK DT500 [47]. It has a distance sensing range of 0.2 to 15 m with a resolution of 0.02 m.

### 4.2.1 Prototype Tank

The prototype tank is a PVC pipe frame covered with stretched fabric. The bottom of the tank is the black rubber floor of the laboratory. The tank diameter is approximately  $2.025\text{m} \pm 0.025\text{m}$ . The height of the tank is approximately  $1.55\text{m} \pm 0.025\text{m}$ , as measured using a tape measure. The sensor platform is placed with its end hanging off a cart. The gimbal is placed into a gap in the fabric as shown in Figure 4-6. The cart also holds the controlling computer.



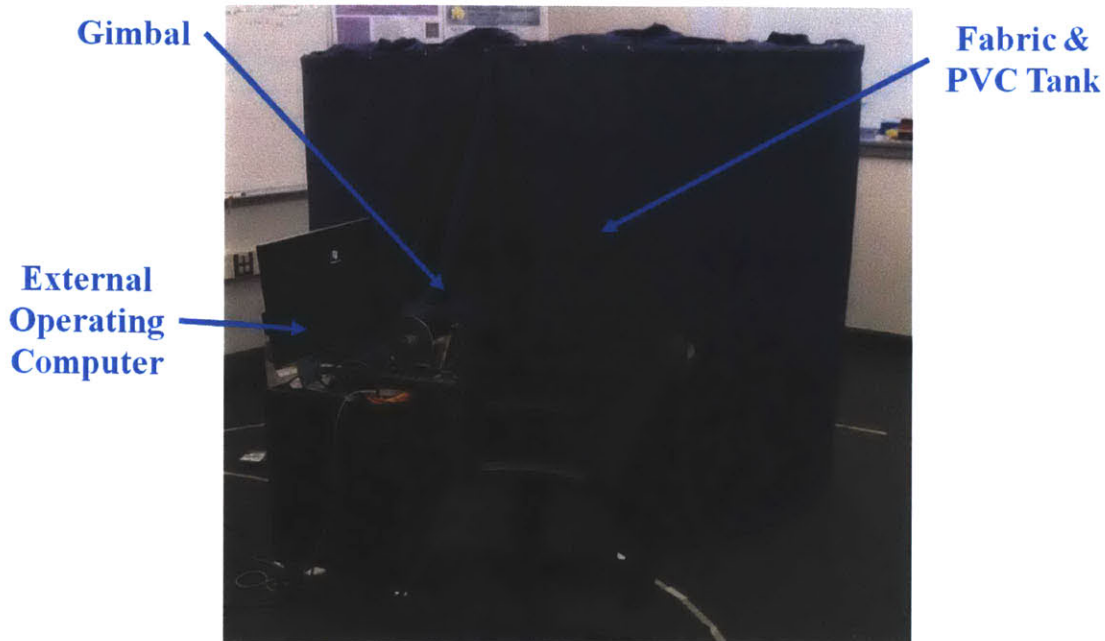


Figure 4-6: Prototype Tank. Made of a PVC frame covered by stretched fabric. The tank was placed on a rubber floor.

### 4.2.2 Prototype Operating Code

The algorithm was developed and tested using two programming languages: Matlab and Labview. The initial development and simulations were completed using Matlab. The sensing platform is run using Labview. Labview calculates the positions, commands the positioning controllers to the desired values, reads in the data and compiles it into a data file. After the tank data is collected, Matlab reads in and processes the data file and outputs the estimated parameters and fit images. Appendix A contains the Matlab files and Appendix B contains the Labview files.

## 4.3 Evaluation Through Hardware Testing

The data for one of the test runs is shown in Figure 4-7. There are significant outliers due to the sensor reading points through the hole in the tank where it enters the tank. The process for removing the outliers is detailed in Appendix C. There is also significant measurement error in the roof as the distance from the sensor gets larger,

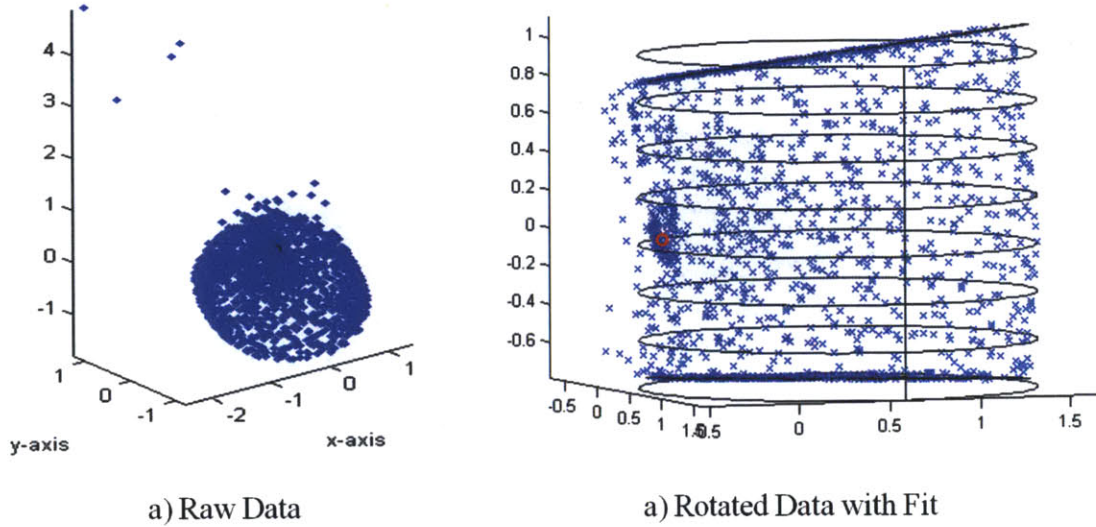


Figure 4-7: Test Run Data. a) The raw test run data using the tank and sensor platform described above. b) The same data rotated with the outliers removed and its fit. The roof data is very angled, most likely due to reflection problems between the laser and stretched fabric.

resulting in a tipped roof when it should be flat. The error is most likely due to incidence angle and reflection issues between the laser and the fabric. This effect does not show up with the floor data, which is made of a different material. The difference in the data from the two surfaces indicates that the sensor is compatible with only certain materials. A sensor would need to be chosen that is compatible with the material of the environment to achieve accurate results.

Table 4.1: Estimated Parameters Using Various Numbers of Neighborhoods

Number of Neighborhoods	Height	Radius	Sensor Height	Sensor Radius
218	1.7042	1.0401	0.7389	0.7941
152	1.7014	1.0467	0.7164	0.7908
98	1.701	1.0407	0.7358	0.7908
56	1.6983	1.0503	0.7171	0.7869
26	1.9917	1.0483	1.7716	0.198
8	1.9403	0.8572	1.1655	0.4755

Tests were run varying numbers of neighborhoods using a constant desired accuracy of 0.02 m. The resulting estimated parameters are shown in Table 4.1. The height of the tank, as measured by a tape measure is approximately 1.55 m. The

fit's estimated height is outside the desired accuracy of 0.02 m but is consistent for test runs down to 56 neighborhoods. The radius was measured at 1.013 m with the tape measure and estimated by the algorithm to be 1.04 m. The measured radius is also outside the desired accuracy but shows a consistent measurement down to only 26 neighborhoods. The radial location and height of the sensor are also measured consistently for as few as 56 neighborhoods. The error in the estimated parameters is due to the measurement error of the laser sensor on the fabric walls and the skew of the circular data as discussed above. The fits are consistent despite the values being incorrect. This shows that the algorithm functions properly despite noise and other issues of the sensor and environment.

Table 4.2: Estimated Parameters Using Various Desired Accuracies

Desired Accuracy	Height	Radius	Sensor Height	Sensor Radius
0.01	1.704	1.038	0.609	0.738
0.02	1.701	1.041	0.736	0.791
0.1	1.9897	0.878	1.226	0.288

Tests were also run varying the desired accuracy of the fit. The resulting parameters are shown in Table 4.2. There is a large discrepancy in the parameter values when the desired accuracy was 10% of the height of the tank. The estimates are likely bad at large acceptable errors because the final rotation of the tank can be significantly rotated resulting a significant skewing of the data. At smaller desired accuracies the skew does not cause significant problems because the tank is close to vertical.

The hardware tests show that it is possible to get good tank estimates despite significant noise. The tests also show that accuracies at or smaller than the sensor error should be used to achieve good estimates and that runs with 56 or more neighborhoods result in good tank parameter estimates for a sensor located approximately halfway up the tank wall.



# Chapter 5

## Conclusion

The purpose of this work was to design and prototype a localization and mapping platform for an autonomous mobile LPG tank inspection robot. The design of the localization hardware and development of the environment mapping software are the first steps in the process of replacing human inspectors with an inspection robot. Robotic inspection has the potential to improve the speed of testing, save millions of dollars and reduce the environmental impact of inspection.

The selection of a compatible set of technologies and their layouts were completed for the localization hardware. LIDAR was determined to be the best technology for the inspection application because of its distance sensing range, resolution, commercial availability, low energy release and ability to withstand the tough environmental demands. The best sensor architecture was then determined to be a gimbaled platform with two actuated and measured axes and a single point distance sensor placed in the manhole. It was the best option because of its lower cost, acceptable accuracy, lower fabrication complexity, average communication complexity and its flexibility of function. The sensor platform can scan or track which allows it to map the environment and track the inspection robot. A first run prototype of the gimbal was created and run in a laboratory setting. The prototype acted as expected, proving the concept is valid.

The second large contribution of the work described in this thesis was the development and testing of customized data collection, segmentation and fitting algorithms.

The collection algorithm is an optimized method for collecting data of an unknown environment from a gimbaled platform based on the expected extents of the data, the resolution of the sensors, and the desired accuracy of the cylinder fit. The segmentation and fitting algorithm combines the Gauss image, 3D histogram techniques and object shape knowledge to non-iteratively fit a right angled cylinder with planar capped ends without iteration. The algorithm is robust to noise and faster than current segmentation and fitting methods due to its unique lack of iteration. The algorithms were tested using simulation and laboratory tests involving the prototype hardware platform described above. In both cases the algorithms were able to fit a capped cylinders to the data despite significant error.

## **5.1 Future Work**

At the time of this writing, the prototype hardware system and the first version software are functioning. The end goal of the project is to have a completely independent robust integrated software and hardware module ready to place into a robotic inspection system. There are some modifications to both the hardware and software that would make the integration into a complete system easier and more robust.

### **5.1.1 Hardware Future Work**

A full design process should be completed on the gimbal hardware. The prototype presented in this thesis was fabricated in a laboratory setting with sensors that would work in the lab environment. The gimbal is fairly large and a redesign could significantly reduce the size and bulkiness of the system. The technology to make the system more compact is available, but it is expensive or difficult to manage in a laboratory setting. The DT500 laser rangefinder has the correct range for a laboratory, but would need to be replaced in the actual device with a longer range sensor, such as the Acuity AR3000. The final modification that should be made to the hardware is a better cable management system. Currently the cables are wrapped around the theta axis rod. The amount of rotation in the theta axis does not exceed a full rota-

tion using the current version of the software, therefore the wrapped method works acceptably. A better solution would use a hollow bearing with the cables through the center such that they do not wrap around any axis.

### 5.1.2 Software Future Work

A few small changes to the collection, segmentation and fitting algorithms could be completed to make the fit more robust and accurate. The overall structure and methods of the algorithms are complete and running. The current version of the software is able to segment and fit the majority of cases, but a few changes to the code would diminish the number of cases where the software fails.

The first software change is to add a filter that removes all data points within a certain distance around the gimbal. There are instances when the most common normal is found to be around the sensor instead of on the floor. These instances occur when a large portion of the sensor view is obstructed and the floor area is significantly smaller than the wall area. Only a few normals are calculated on the floor and many normals are calculated around the sensor. The bias of the sensor location could be removed by removing all normals calculated within a small distance around the sensor.

The second software change is to modify the method of removing outliers before segmentation. The current outlier removal method is to use modifications to the 5th and 95th percentiles of the data. When there are outliers the clipping ranges are large and do not remove valid data. The problem occurs when there are no outliers and the data is heavily biased to one side of the tank, such as when the sensor platform is obstructed and many points are collected around the sensor. Valid data is removed in the biased, no outlier case. A better method of handling outliers needs to be added to the code.

The third software change is to add a method to remove the skew of the cylinder data shown in Figure 4-2. As mentioned in Chapter 4, the data skew causes the location of the sensor and the estimated tank parameters to be incorrect. A second rotation using the sectioned floor data normal will result in a more vertically aligned

tank. Another method that could resolve the problem is fitting a rotated 3D cylinder to the wall data instead of a circle to the xy data. The 3D cylinder fit could be completed without iteration because the cap data is removed and the axis of the cylinder is known; it is the normal of the floor plane located at the center of the floor plane data. Both of the methods mentioned here should be tested for robustness and the best option should be implemented in the final algorithm.

The fourth and final software change is the addition of a reality check on the floor and roof locations. Sometimes the roof may have more normals than the floor. In this case the tank is rotated upside down. The tank is still rotated into a right angled position and segmentation occurs as expected. The only problem is that the map will be upside down. A reality check should be completed on the data to make sure the tank is rotated correctly. This can be done by confirming that the measured value of the floor is negative and the roof is positive, but a more robust method would be to make sure the robot is located on the bottom of the fit. If the fit is upside down, a simple rotation by 180 degrees can be completed on all the data to rotate the tank into the correct orientation.

### **5.1.3 Inspection System Future Work**

The next step in the development of the inspection application is the creation of an algorithm and the associated hardware to locate and track the robot as it inspects the tank floor. The work described in this thesis covers the beginning development of a robot localization system. Once the environment is mapped, the hardware can be used to locate and track the robot in the same environment. The next step is to locate and track the robot. At the time of this writing, the tracking algorithm and associated hardware is being developed in the Mechatronics Research Laboratory at MIT.



# Appendix A

## Matlab Files

This appendix contains the Matlab code that is described in Chapter 3. The code can be configured to run using simulation or to process data from a file. The first five functions in the appendix are used for either situation. The Normal Collection Simulaton, Distance Sensor Simulaiton and Tip Tilt functions are used only when simulating the data. The description, inputs and outputs of each function are detailed in the comments at the top of each file.

## A.1 Main Tank Mapping File

```
1 %% TANK MAPPING
2 % Main program for the LPG/LNG Tank Mapping
3 % 1)Runs a function that simulates the collection of data from a
4 %     laser sensor mounted on a pan/tilt platform or processes
5 %     data from a file.
6 % OR
7 % 1)Reads in data file containing columns of phi,theta,d data and
8 % calculates the normals
9 % 2)Runs fitting function to fit tank to data.
10 % 3)Runs plotting function to plot data and fits
11
12 clear all
13 close all
14 clc
15
16 %% User Inputs
17     desacc = 0.02; % Desired accuracy in m
18     noise_sensor = 0.02; % Standard Dev of Sensor Error, sigma_sensor
19     res_enc = 2*pi/(65536*4); % Resolution of encoder steps in deg
20     rough_height = 1;
21     rough_diam = 1*2;
22     pointsperside = 5; %Number of normals collecting on each side ...
23         of the cube
24         %Ntot = pointsperside^3-(pointsperside-2)^3
25         %Corresponds to only the outside of the cube
26
27 %% Constants
28
29 % Neighborhood Radius (r_n)
30     dim = min(rough_height,rough_diam);
31     neighborhood_radius = 0.2*dim;
32
33 % Accuracy
34     acc = desacc/dim;
```

```

33     accuracy = acos(1-acc/3); %Acc converted to normal angle
34                                     %accuracy (3 stdevs) with 75%
35                                     %of to make sure get it
36
37 % Gimbal Noise Standard Deviation, noise_gimbal
38     space=.2;
39     error = ...
        sqrt(2*((max(rough_height,rough_diam)^2)/(3*space)*res_enc)^2 ...
        +...
40     noise_sensor^2);
41
42 % Number of points in neighborhood (num = Nn)
43     num = (2*2.26*error/(neighborhood_radius*accuracy))^1.79;
44     if num > 3
45         numpoints = ceil(num);
46     else
47         numpoints = 3;
48     end
49
50 %% 1) Data Collection Simulation
51 % Can either simulate data collection and normal processing or
52 % can read in a text file containing the theta, phi and d in
53 % columns seperated by tabs
54
55 %% Uncomment the following to mathematically simulates data ...
        collection
56 %     [normals_x, normals_y, normals_z,x,y,z] =...
57 %     normalcollection(pointsperside, numpoints, ...
        neighborhood_radius, noise_sensor, res_enc);
58
59 %% Uncomment the following to read in file, process it and ...
        calculate_normals
60 % % If have outliers need to uncomment clipping from Cylinder_Fit ...
        file_as_well
61 tpd = importdata('120511Run3','\t',0);
62

```

```

63 theta = tpd(:,1);
64 phi = tpd(:,2);
65 d = tpd(:,3);
66
67 %Calculate the number of points corresponding to each neighborhood
68 p = length(d)/(pointsperside^3-(pointsperside-2)^3);
69
70 k = 1;
71 n = length(d)/p;
72 for j = 1:n
73     bad = 0;
74     for i = 1:p
75         %Collecting the neighborhood points
76         thetaint(i) = theta(12*(j-1)+i);
77         phiint(i) = phi(12*(j-1)+i);
78         dint(i) = d(12*(j-1)+i);
79         %If sensor reads above 50, sensor didn't read correctly
80         if dint(i) > 50 | dint(i)==0;
81             bad = 1;
82         end
83     end
84     if bad==0 %If there are no bad points (meaning the sensor ...
85         read properly)
86         [xint,yint,zint] = sph2cart(thetaint,phiint-pi/2,dint);
87         [a,b,c] = planeFitLS(xint,yint,zint); %Calculate normal
88         l_pn = sqrt(a^2+b^2+c^2); %Ensure it is a unit normal
89         a1(k) = a/l_pn;
90         b1(k) = b/l_pn;
91         c1(k) = c/l_pn;
92     else %If there were bad points in the normal
93         a1(k) = 0;
94         b1(k) = 0;
95         c1(k) = 0;
96     end
97     k = k+1;
98 end

```

```

98
99 a1 = a1';
100 b1 = b1';
101 c1 = c1';
102
103 norms = [a1,b1,c1];
104
105 [row col] = find(norms==0);
106 norms(row,:) = [];
107
108 [row2 col2] = find(tpd>50); %Remove the points that are reading ...
    above 50m
109 tpd(row2,:) = [];
110
111 [row3 col3] = find(tpd==0); %Remove the points that read 0
112 tpd(row3,:) = [];
113
114 theta = tpd(:,1);
115 phi = tpd(:,2);
116 d = tpd(:,3);
117
118 normals_x = norms(:,1);
119 normals_y = norms(:,2);
120 normals_z = norms(:,3);
121
122 [x,y,z] = sph2cart(theta,phi-pi/2,d);
123
124 %% Fitting
125 tstartfit = tic;
126
127     [Tank_Height,Tank_Radius,xcent,ycent,floor_height,rotmat,data]...
128         = CylinderFit_OneHist_LS(normals_x,normals_y,normals_z,...
129             x,y,z,accuracy,error);
130
131 timefit = toc(tstartfit)
132

```

```
133 %Print out the fit parameters
134     Tank.Height
135     Tank.Radius
136     xcent
137     ycent
138     floor_height
139     radial = sqrt(xcent^2+ycent^2)
140
141 %% Plotting
142     CylinderPlots(x,y,z,data,Tank.Height,Tank.Radius,xcent,ycent)
```

## A.2 Cylinder Fitting Function

```
1 function ...
   [Tank_Height, Tank_Radius, xcent, ycent, floor_height, rotmat, data]...
2       = CylinderFit_OneHist_LS(normals_x, normals_y, normals_z...
3       , x, y, z, accuracy, error)
4 %% CYLINDER FITTING
5 % This function takes in theta, phi, d point data and normals, corrects
6 % for error in the tip/tilt (assuming the tank should always have
7 % its center axis vertical) and fits a bounded cylinder
8 % (including roof, if data is present and floor) using least
9 % squares fitting.
10 %
11 % Inputs: theta and phi from the pan/scan sensors
12 %         distance, d, from the distance measuring device
13 %         x, y, z coords of plane normals for points of interest
14 %         desired accuracy of the fit
15 %         sensor error estimate
16 %
17 % Outputs: estimated tank height
18 %          estimated tank radius
19 %          estimated center coordinates (xcent, ycent)
20 %          estimated sensor height (same as -floor_height)
21 %          rotation matrix
22 %          collected data for plotting
23 %
24 % Steps:
25 % 1) Creating 3D histogram from the normals data
26 % 2) Use most common normal to calculate rotation matrix
27 % 3) Rotate data using rotation matrix
28 % 3a) Remove outliers if necessary
29 % 4) Separate the data into floor, roof and walls
30 % 5) Fit planes and circle to separated data
31 % 6) Calculate tank fit parameters
32 % 7) Collect data into structure for exporting to plotting function
```

```

33
34 %% Finding most common normal using 3D histogram
35
36 %Creating bins
37 bins = sqrt(3)/accuracy;
38 xi = linspace(-1,1,bins);
39 yi = linspace(-1,1,bins);
40 zi = linspace(-1,1,bins);
41
42 %Adding to nearest bin for each point
43 xr = interp1(xi,0.5:numel(xi)-0.5,normal.s.x','nearest');
44 yr = interp1(yi,0.5:numel(yi)-0.5,normal.s.y','nearest');
45 zr = interp1(zi,0.5:numel(zi)-0.5,normal.s.z','nearest');
46
47 W = accumarray([xr yr zr]+0.5, 1, [bins bins bins]);
48
49 %Finding the max bin
50 [num idx] = max(W(:));
51 [a b c] = ind2sub(size(W),idx);
52
53 %Coordinates of most common normal
54 A = xi(a);
55 B = yi(b);
56 C = zi(c);
57
58 %% FINDING ROTATION MATRIX
59 zaxis = [0 0 -1]; %z-axis unit normal vector
60
61 l_pn = sqrt(A^2+B^2+C^2); %length of floor plane normal vector
62 planenorm = [(A/l_pn) (B/l_pn) (C/l_pn)]; %floor plane unit ...
        normal vector
63 rotaxis = cross(zaxis,planenorm);
64 rotangle = acos(dot(zaxis,planenorm));
65 Rot = [rotaxis rotangle];
66 rotmat = vrrotvec2mat(Rot);
67

```



```

68 %% ROTATE DATA
69 %Preallocating vectors to speed program
70 rotxyz = zeros(length(x),3);
71
72 % Rotating
73 for k=1:length(x)
74     rotxyz(k,:) = [x(k) y(k) z(k)]*rotmat;
75 end
76
77 rot_x = rotxyz(:,1);
78 rot_y = rotxyz(:,2);
79 rot_z = rotxyz(:,3);
80
81 %% Uncomment if need to remove outliers
82
83 rotated = [rot_x, rot_y, rot_z];
84
85 x100 = quantile(rot_x,0.95)+(quantile(rot_x,0.95)-...
86     quantile(rot_x,0.05))*0.10/0.90+3*error;
87 x0 = quantile(rot_x,0.05)-(quantile(rot_x,0.95)-...
88     quantile(rot_x,0.05))*0.10/0.90-3*error;
89 y100 = quantile(rot_y,0.95)+(quantile(rot_y,0.95)-...
90     quantile(rot_y,0.05))*0.10/0.90+3*error;
91 y0 = quantile(rot_y,0.05)-(quantile(rot_y,0.95)-...
92     quantile(rot_y,0.05))*0.10/0.90-3*error;
93 z100 = quantile(rot_z,0.95)+(quantile(rot_z,0.95)-...
94     quantile(rot_z,0.05))*0.10/0.90+3*error;
95 z0 = quantile(rot_z,0.05)-(quantile(rot_z,0.95)-...
96     quantile(rot_z,0.05))*0.10/0.90-3*error;
97
98 xrad = (x100-x0)/2;
99 yrad = (y100-y0)/2;
100 rad = max(xrad,yrad);
101
102 xc = xrad+x0;
103 yc = yrad+y0;

```

```

104
105 [rowxy]=find(sqrt((rot_x-xc).^2+(rot_y-yc).^2) > rad);
106 [rowz] = find(rot_z > z100 | rot_z < z0);
107
108 rows = [rowxy;rowz];
109
110 rows = unique(rows);
111
112 rotated(rows,:) = [];
113
114 rot_x = rotated(:,1);
115 rot_y = rotated(:,2);
116 rot_z = rotated(:,3);
117
118 %% SEPARATE DATA INTO FLOOR/ROOF/WALL
119 % Finding middle of data
120 midx = mean([quantile(rot_x,0.95) quantile(rot_x,0.05)]);
121 midy = mean([quantile(rot_y,0.95) quantile(rot_y,0.05)]);
122 midz = mean([quantile(rot_z,0.95) quantile(rot_z,0.05)]);
123
124 %Find range of data
125 rangex = quantile(rot_x,0.95)-quantile(rot_x,0.05);
126 rangey = quantile(rot_y,0.95)-quantile(rot_y,0.05);
127 ranged = min([rangex rangey]);
128 rangez = quantile(rot_z,0.95)-quantile(rot_z,0.05);
129
130 % corrections for estimates based on worse case error from bin
131 temp = ([1 cot(accuracy);cot(accuracy) 1])\[ranged; rangez];
132 delr = temp(1);
133 delz = temp(2);
134
135 % correction based on sensor error (wall data that might
136 % accidentally be inside the cleaned up radius because of error
137 account4error = 10*error;
138
139 % Safe estimates to be used to separate data

```

```

140 cleanradius = ranged/2-delr-account4error;
141 cleanroof = quantile(rot_z,0.95)-delz-account4error;
142 cleanfloor = quantile(rot_z,0.05)+delz+account4error;
143
144 o = 1;
145 p = 1;
146 s = 1;
147
148 for q=1:length(rot_z)
149     %If inside clean radius
150     if ((rot_x(q)-midx)^2+(rot_y(q)-midy)^2 ≤ cleanradius^2)
151         %If below middle of z height
152         if (rot_z(q) < midz)
153             %Then is floor
154             intfloor_z(o) = rot_z(q);
155             intfloor_y(o) = rot_y(q);
156             intfloor_x(o) = rot_x(q);
157             o = o+1;
158         %Else if above middle of z height
159         elseif (rot_z(q) > midz)
160             %Then is roof
161             introof_z(p) = rot_z(q);
162             introof_y(p) = rot_y(q);
163             introof_x(p) = rot_x(q);
164             p = p+1;
165             %%end
166         end
167     %If below clean roof and above clean floor
168     elseif ( (rot_z(q)≤cleanroof) && (rot_z(q)≥cleanfloor) )
169         %Then is walls
170         intwall_z(s) = rot_z(q);
171         intwall_y(s) = rot_y(q);
172         intwall_x(s) = rot_x(q);
173         s = s+1;
174     end
175 end

```

```

176
177 %Rotating the data to make columns instead of rows
178 floor_x = intfloor_x.';
179 floor_y = intfloor_y.';
180 floor_z = intfloor_z.';
181
182 roof_x = introof_x.';
183 roof_y = introof_y.';
184 roof_z = introof_z.';
185
186 wall_x = intwall_x.';
187 wall_y = intwall_y.';
188 wall_z = intwall_z.';
189
190 %% CIRCLE FIT TO WALLS X,Y DATA
191 if (isempty(wall_x))
192     fprintf('No wall data')
193 else
194     [xcent,ycent,R] = circlefitLS(wall_x,wall_y);
195 end
196
197 %% FITTING PLANES TO FLOOR AND ROOF DATA
198 if (isempty(wall_x))
199     fprintf('No floor data')
200 elseif (isempty(roof_x))
201     fprintf('No roof data')
202 else
203     [fA,fB,fC] = planefitLS(floor_x,floor_y,floor_z);
204     [rA,rB,rC] = planefitLS(roof_x,roof_y,roof_z);
205
206 %Finding fit floor and roof
207     num = 50;
208     u = 1;
209     for beta = 0:2*pi/num:2*pi;
210         for D = 0:R
211             intfit_x(u) = xcent+cos(beta)*D;

```

```

212         intfit_y(u) = ycent+sin(beta)*D;
213         u = u+1;
214     end
215 end
216 fit_floor_z = -(fA/fC)*intfit_x.'-(fB/fC)*intfit_y.'+1/fC;
217 fit_roof_z = -(rA/rC)*intfit_x.'-(rB/rC)*intfit_y.'+1/rC;
218
219 floor_height = -(fA/fC)*xcent-(fB/fC)*ycent+1/fC
220 roof_height = -(rA/rC)*xcent-(rB/rC)*ycent+1/rC;
221 end
222
223 %% FINAL TANK SIZE ESTIMATES
224
225 Tank_Radius = R;
226
227 %Height is the distance between the floor and roof
228 Tank_Height = roof_height-floor_height;
229
230 %% Collecting data to be sent to plotting function
231 data.rot_x = rot_x;
232 data.rot_y = rot_y;
233 data.rot_z = rot_z;
234 data.floor_x = floor_x;
235 data.floor_y = floor_y;
236 data.floor_z = floor_z;
237 data.roof_x = roof_x;
238 data.roof_y = roof_y;
239 data.roof_z = roof_z;
240 data.wall_x = wall_x;
241 data.wall_y = wall_y;
242 data.wall_z = wall_z;
243 data.intfit_x = intfit_x;
244 data.intfit_y = intfit_y;
245 data.fit_floor_z = fit_floor_z;
246 data.fit_roof_z = fit_roof_z;
247 data.bins_x = xi;

```

```
248 data.bins_y = yi;  
249 data.bins_z = zi;  
250 data.bins_W = W;
```

## A.3 Plotting Function

```
1 function ...
    CylinderPlots(x,y,z,data,Tank_Height,Tank_Radius,xcent,ycent)
2 %% Plotting Function
3 % This is the main plotting function. It takes the data output by the
4 % fitting algorithm and makes plots to show the data, fits, etc.
5 %
6 % Inputs: x,y,z coordinates of the raw data
7 %         data structure output from fitting function
8 %         Estimated tank height and radius (Tank_Height and ...
           Tank_Radius)
9 %         Estimated tank center (xcent,ycent)
10 %
11 % Outputs: Raw data plot
12 %          3D Histogram
13 %          Rotated and trimmed data plots
14 %          Fit plots
15 %
16 % Steps:
17 % 1) Pull data out of structure sent from fitting function
18 % 2) Create array of data to be used to plot the wall fit
19 % 3) Create plot of raw data
20 % 4) Create plot of rotated data
21 % 5) Create plots of each fit (floor, roof, walls) and the total fit
22
23 %% Pulling Pieces out of data
24 rot_x = data.rot_x;
25 rot_y = data.rot_y;
26 rot_z = data.rot_z;
27 floor_x = data.floor_x;
28 floor_y = data.floor_y;
29 floor_z = data.floor_z;
30 roof_x = data.roof_x;
31 roof_y = data.roof_y;
```

```

32 roof_z = data.roof_z;
33 wall_x = data.wall_x;
34 wall_y = data.wall_y ;
35 wall_z = data.wall_z;
36 fit_x = data.intfit_x;
37 fit_y = data.intfit_y;
38 fit_floor_z = data.fit_floor_z;
39 fit_roof_z = data.fit_roof_z;
40 bins_x = data.bins_x;
41 bins_y = data.bins_y;
42 bins_z = data.bins_z;
43 bins_W = data.bins_W;
44
45 %% CALCULATE FITTED WALLS DATA
46 t = 1;
47 zmin = min(rot_z);
48 zmax = max(rot_z);
49 num = 100;
50
51 for z1=zmin:.25:zmax;
52     for alpha = 0:2*pi/num:2*pi;
53         fit_wall_x(t) = xcent+cos(alpha)*Tank.Radius;
54         fit_wall_y(t) = ycent+sin(alpha)*Tank.Radius;
55         fit_wall_z(t) = z1;
56         t = t+1;
57     end
58 end
59
60 %% CALCULATE FITTED WALLS DATA Centered at zero
61 % tc = 1;
62 % zmin = min(rot_z);
63 % zmax = max(rot_z);
64 % num = 100;
65 %
66 % for z1c=zmin:.25:zmax;
67 %     for alpha = 0:2*pi/num:2*pi;

```



```

68 %         fit_wall_xc(tc) = cos(alpha)*Tank.Radius;
69 %         fit_wall_yc(tc) = sin(alpha)*Tank.Radius;
70 %         fit_wall_zc(tc) = z1;
71 %         tc = tc+1;
72 %     end
73 % end
74
75
76 %% Plot raw data
77 figure(1);
78 plot3(0,0,0,'ko'); %Manhole
79 xlabel('x-axis');
80 ylabel('y-axis');
81 hold on;
82 plot3(x,y,z,'.','markersize',6); %Raw Data
83 title('Raw Data')
84 axis square;
85 axis equal;
86 hold off;
87
88 %% Plot histogram
89 figure(2);
90 [x1,y1,z1] = meshgrid(bins_y,bins_x,bins_z);
91 test = [x1(:) y1(:) z1(:) bins_W(:)];
92 [row] = find(test(:,4)==0);
93 test(row,:) = [];
94 scatter3(test(:,2),test(:,1),test(:,3),30,test(:,4),'filled');
95 xlabel('x')
96 ylabel('y')
97 zlabel('z')
98 axis square
99 axis equal
100
101
102
103 %% Plot rotated data

```

```

104 figure(3);
105 plot3(0,0,0,'ko'); %Manhole
106 xlabel('x-axis');
107 ylabel('y-axis');
108 hold on;
109 plot3(rot_x,rot_y,rot_z, '.', 'markersize',2); %Rotated Data
110 title('Rotated Data')
111 axis square;
112 axis equal;
113 hold off;
114
115 %% Plot the individual fits and final fit in 4x4 grid
116 figure(4);
117
118 % Plot the floor fit
119 set(gcf, 'position', [20,60,1280,800]);
120 subplot(2,2,1)
121 plot3(fit_x,fit_y,fit_floor_z, 'k')
122 axis square;
123 axis equal;
124 title('Floor Fit')
125 hold on;
126 plot3(floor_x,floor_y,floor_z, 'r*'); %Rotated Floor Data
127
128 % Plot the roof fit
129 subplot(2,2,2);
130 plot3(fit_x,fit_y,fit_roof_z, 'k')
131 axis square;
132 axis equal;
133 title('Roof fit')
134 hold on;
135 plot3(roof_x,roof_y,roof_z, 'r*'); %Rotated Roof Data
136
137 %Plot the walls fit
138 subplot(2,2,3);
139 plot3(fit_wall_x,fit_wall_y,fit_wall_z, 'k'); %Wall Fit

```

```

140 % hold on
141 % plot3(fit.wall_xc,fit.wall_yc,fit.wall_zc,'b'); %Wall Fit
142 axis square;
143 axis equal;
144 title('Walls fit')
145 hold on;
146 plot3(wall_x,wall_y,wall_z,'rx'); %Rotated Walls Data
147
148 %Plot all fit
149 % grey = [0.8,0.8,0.8];
150 subplot(2,2,4);
151 plot3(rot_x,rot_y,rot_z,'bx'); %Rotated Data
152 axis square;
153 axis equal;
154 hold on;
155 plot3(fit_x,fit_y,fit_floor_z,'k'); %'color',grey); %Floor Fit
156 plot3(fit_x,fit_y,fit_roof_z,'k'); %'color',grey); %Roof Fit
157 plot3(fit_wall_x,fit_wall_y,fit_wall_z,'k'); %'color',grey); ...
    %Wall Fit
158 title('Total fit')

```

## A.4 Plane Fitting Function

```
1 function [A,B,C] = planefitLS(x,y,z)
2 %% Least Squares Plane Fitting
3 % Calculates the best fit plane using least squares matrix methods
4 %
5 % Inputs: x,y,z data for plane to be fit
6 %
7 % Outputs: Plane normal coordinates A,B,C
8
9 for i = 1:length(x)
10     phi(1,i) = x(i);
11     phi(2,i) = y(i);
12     phi(3,i) = z(i);
13 end
14 w = ones(1,length(z));
15 P = inv(phi*phi. ');
16 B = phi*w. ';
17 planecoeff = P*B;
18
19 A = planecoeff(1,1);
20 B = planecoeff(2,1);
21 C = planecoeff(3,1);
```

## A.5 Circle Fitting Function

```
1 function [X0,Y0,R] = circlefitLS(x,y)
2 %% Least Squares Circle Fitting
3 % Calculates the best fit circle using least squares matrix methods
4 %
5 % Inputs: x and y data for circle to be fit
6 %
7 % Outputs: Origin location X0 and Y0 and circle Radius
8
9 for i = 1:length(x)
10     phic(1,i) = x(i)^2+y(i)^2;
11     phic(2,i) = x(i);
12     phic(3,i) = y(i);
13 end
14 wc = ones(1,length(x));
15 Pc = inv(phic*phic. ');
16 Bc = phic*wc. ';
17 coeffc = Pc*Bc;
18
19 Ac = coeffc(1,1);
20 Bc = coeffc(2,1);
21 Cc = coeffc(3,1);
22
23 R = sqrt(4*Ac+Bc^2+Cc^2)/(2*Ac);
24 X0 = -Bc/(2*Ac);
25 Y0 = -Cc/(2*Ac);
```

## A.6 Normal Collection Simulation Function

```
1 function [normals_x, normals_y, ...
           normals_z, xreturn, yreturn, zreturn]...
2         = normalcollection(pointsperside, numpoints, ...
           neighborhood_radius, sensor_noise, res_enc)
3 %% DATA COLLECTION
4 % This function does the math for the data collection and then
5 % calls the distance simulator to return data to the fitting
6 % algorithm
7 %
8 % Inputs: Number of points per side of a unit cube for points of
9 %         interests
10 %        Number of points of interest
11 %        Radius of neighborhood
12 %        Standard deviaion of the sensor noise
13 %        Resolution of the encoder in degrees
14 %
15 % Outputs: x,y,z components of surface normals at points of interest
16 %          x,y,z components of each point where distance data was
17 %          collected
18 %
19 % Steps:
20 % 1) Calculates the directions for the points for which normals
21 %    will be calculated.
22 % 2) Collects the data for these points of interest
23 % 3) Calculates the directions for each point in the neighborhood
24 % 4) Collects the data for the neighborhood points
25 % 5) Fits planes and finds normal for each neighborhood
26
27 %% Constants
28 R = neighborhood_radius;
29 xreturn=[];
30 yreturn=[];
31 zreturn=[];
```

```

32
33 %% Calculating Normal Collection Angles from Unit Grid
34 % This gives a more even distribution than evenly spreading
35 % around a unit sphere, but the end result is still not
36 % evenly distributed
37
38 %Making the grid of points
39 n = pointsperside;
40 l=linspace(0,1,n+2);
41 a=l(2:n+1); b=a;
42 [A1, B1] = meshgrid(a, b);
43 A = reshape(A1, [], 1);
44 B = reshape(B1, [], 1);
45 C = [min(A)*ones(length(A),1); max(A)*ones(length(A),1)];
46 A = [A; A];
47 B = [B; B];
48
49 %Applying the grid to all sides of the cube
50 X1 = A;
51 Y1 = B;
52 Z1 = C;
53
54 X2 = B;
55 Y2 = C;
56 Z2 = A;
57
58 X3 = C;
59 Y3 = A;
60 Z3 = B;
61
62 %Centering the cube
63 X = [A;B;C]-0.5;
64 Y = [B;C;A]-0.5;
65 Z = [C;A;B]-0.5;
66
67 points = [X,Y,Z];

```



```

68
69 %Getting rid of duplicate points (edges of the cube)
70 P = unique(points,'rows');
71
72 %Final evenly spaced points about the curve
73 Xfinal = P(:,1);
74 Yfinal = P(:,2);
75 Zfinal = P(:,3);
76 %%
77 %Converting the XYZ to Theta Phi Z so can send commands to motor axes
78 [theta_normals phi_normals dist] = cart2sph(Xfinal, Yfinal, Zfinal);
79 phi_normals=phi_normals+pi/2; %I defined my phi, theta, z
80     %different than Matlab does for cart2sph so this converts
81     %to my orientation
82
83 %% Simulating the data collection
84
85 %Preallocating
86 normals_x = zeros(length(phi_normals),1);
87 normals_y = zeros(length(phi_normals),1);
88 normals_z = zeros(length(phi_normals),1);
89
90 xint = zeros(length(phi_normals)*numpoints,1);
91 yint = zeros(length(phi_normals)*numpoints,1);
92 zint = zeros(length(phi_normals)*numpoints,1);
93
94 % Collects distance for point of interest (POI) in the direction
95 % calculated above, then calculates the direction to face for
96 % each point in the neighborhood around the POI, then collects
97 % distance for each neighborhood point
98 i = 1;
99
100 for normal= 1:length(phi_normals)
101
102     % Isolating POI
103     phi_n = phi_normals(normal);

```

```

104     theta_n = theta_normals(normal);
105
106     % Collecting distance for POI
107     [theta_n, phi_n, d] = simdist(theta_n, phi_n, sensor_noise, ...
108         res_enc);
109
110     % Calculating the neighborhood point locations
111     [xint(i) yint(i) zint(i)] = sph2cart(theta_n, phi_n-pi/2, 1);
112     Vn = [xint(i) yint(i) zint(i)];
113
114     Vrp = 0;
115     while Vrp == 0
116         %generate random vector
117         Vr = [rand, rand, rand];
118         %Find component of random vector in circle plane
119         Vrp = Vr-dot(Vn,Vr)*Vn;
120     end
121     %Making first orthogonal vector a unit vector
122     V1 = Vrp/sqrt(Vrp(1)^2+Vrp(2)^2+Vrp(3)^2);
123
124     %Second orthonormal vector
125     V2 = cross(V1,Vn);
126     %Making 2nd orthogonal vector a unit vector
127     V2 = V2/sqrt(V2(1)^2+V2(2)^2+V2(3)^2);
128
129     r=R/sqrt(R^2+d^2);
130     Δ_t = 2*pi/numpoints;
131
132     for t=0:Δ_t:2*pi
133         i=i+1;
134         Vc = Vn+cos(t)*r*V1+sin(t)*r*V2;
135         xint(i) = Vc(1);
136         yint(i) = Vc(2);
137         zint(i) = Vc(3);
138     end

```

```

139 %Collecting distance for each neighborhood point
140 for n= 1:length(xint)
141     [theta_int(n) phi_int1(n) dist(n)] = cart2sph(xint(n), ...
142         yint(n), zint(n));
143     phi_int(n)=phi_int1(n)+pi/2;
144     [theta(n), phi(n), d(n)] = simdist(theta_int(n), ...
145         phi_int(n), sensor_noise, res_enc);
146     [x(n) y(n) z(n)]= sph2cart(theta(n), phi(n)-pi/2, d(n));
147 end
148
149 %Fitting plane to the neighborhood
150 [A,B,C] = planefitLS(x,y,z);
151
152 %length of plane normal vector
153 l_pn = sqrt(A^2+B^2+C^2);
154
155 %plane unit normal vector
156 normals_x(normal) = (A/l_pn);
157 normals_y(normal) = (B/l_pn);
158 normals_z(normal) = (C/l_pn);
159
160 %Isolating the data to be returned to the fitting algorithm
161 %Both the normals calculated above and all the data
162 %collected will be returned to the fitting algorithm
163 xreturn = [xreturn; x'];
164 yreturn = [yreturn; y'];
165 zreturn = [zreturn; z'];
166
167 % Clearing Variables
168 i = 1;
169 xint = 0;
170 yint = 0;
171 zint = 0;
172 theta_int = 0;
173 phi_int = 0;
174 theta = 0;

```

```
173     phi = 0;  
174     x = 0;  
175     y = 0;  
176     z = 0;  
177 end
```

## A.7 Distance Sensor Simulation Function

```
1 function [theta, phi, d] = simdist(theta, phi, sense_noise, res_enc)
2 %% Distance Sensor Simulation Function
3 % This function simulates the data results one would get from
4 % using a laser distance measuring device on a pan/scan mount
5 %
6 % Inputs: theta and phi you want to send the laser to measure out
7 %         standard deviation of the sensor noise
8 %         resolution of the encoder
9 %
10 % Outputs: theta and phi "measured" from pan/scan mount
11 %          distance, d, "measured" from the laser
12 %
13 % Steps:
14 % 1) Input tank and sensor information
15 % 2) Add in noise due to encoder resolution
16 % 2) Run the tip tilt function to simulate a tipped sensor
17 % 3) Calculate the simulated distance via intersection of line
18 %    and cylinder
19 % 4) Add noise to the data
20 %% Constants
21
22 % Error in sensor tip/tilt start location
23 phix = 0*pi/180;
24 phiy = 45*pi/180;
25
26 x_man = 0; % Manhole x location (center bottom of tank is 0,0,0)
27 y_man = 74; % Manhole y location
28 z_man = 70; % Manhole z location
29 height = 75; %Tank height 35
30 radius = 75; %Tank Radius 92/2
31
32 %% Add gimbal noise
33 phi_noisy = phi+res_enc*1/3*randn;
```

```

34 theta_noisy = theta+res_enc*1/3*randn;
35
36 %% Run tip tilt program to simulate the sensor being slightly off ...
    from perpendicular/parallel to tank
37 [phi_eff, theta_eff] = tiptilt(phi_noisy, theta_noisy, phix, phiy);
38
39 %% Getting "measured" distance values, d
40
41 % If the sensor is facing straight down the data is just the
42 % height of the sensor. This is necessary because there is a
43 % discontinuity with the trigonometry when phi_eff=0
44 if (phi_eff == 0)
45     d = z_man;
46
47 % Using equation for a 3D line (initial position + d*direction
48 %   vector = position of laser point in space
49 % Direction vector is equations for converting spherical to
50 %   cartesian without radius (want unit vector)
51 % Using equation for cylinder  $x^2+y^2=r^2$ 
52 % Find intersection of 3D line with cylinder by substitution
53 %   and solving for d, which is the distance to the measured
54 %   point in space.
55 else
56     % intermediate variables to make equations easier to type
57     a = ((sin(phi_eff))^2)*((cos(theta_eff))^2)+((sin(phi_eff))^2)...
58         *((sin(theta_eff))^2);
59     b = ...
60         (2*x_man*sin(phi_eff)*cos(theta_eff))+(2*y_man*sin(phi_eff)...
61         *sin(theta_eff));
62
63     c = x_man^2+y_man^2-radius^2;
64
65     % using quadratic formula to solve for d
66     d1 = (-b+sqrt(b^2-4*a*c))/(2*a);
67     d2 = (-b-sqrt(b^2-4*a*c))/(2*a);
68
69     %identify positive root

```

```

68     if (d2 ≥ 0)
69         d_int = d2;
70     elseif (d1 ≥ 0)
71         d_int = d1;
72     else
73         % If there is no positive root return an error
74         sprintf('error')
75     end
76
77     % Find z height of measurement given calculated d
78     z_int = d_int*cos(phi_eff)+z_man;
79
80     % Check that it is within the logical height limits of the tank
81     % If within logical height limits, calculated d is correct
82     if ((0 ≤ z_int) && (z_int ≤ height))
83         d = d_int; % Return calculated d
84     % If z is taller than tank, then laser is hitting roof.
85     elseif (z_int > height)
86         % Recalculate d given that it is hitting the roof
87         d = (-height+z_man)/cos(phi_eff);
88     % If z is shorter than tank, then laser is hitting floor.
89     elseif ((z_int < 0))
90         % Recalculate d given that it is hitting floor.
91         d = z_man/cos(phi_eff);
92     end
93 end
94
95 %% Adding Noise to Data
96 %randn is random data from 0 to 1, by adding and subtracting I
97 %get noise from -1 to 1 around data
98 d = d+sense_noise*randn(length(d),1);

```



## A.8 Tip Tilt Function

```
1 function [phi_eff, theta_eff] = tiptilt(phi, theta, phix, phiy)
2 %% Tip Tilt Function
3 % Calculates the phi and theta that will be measured if the
4 % sensor is not facing straight downward (ie changes phi,
5 % theta to match sensor reference frame instead of cylinder
6 % reference frame)
7 %
8 % Inputs: Phi and Theta as reference off the cylinder coordinate
9 %         frame
10 %        The rotation of the sensor from the cylinder coordinate
11 %        frame about the x and y axes, phix and phiy
12 %
13 % Outputs: The new phi and theta in the sensor coordinate frame
14 %          (tipped sensor)
15 %
16 % Steps:
17 % 1) Convert phi and theta to x,y,z (easier to rotate in xyz
18 %    coord frame)
19 % 2) Rotate about the x-axis by phix amount
20 % 3) Rotate about the y-axis by phiy amount
21 % 4) Convert back to phi, theta
22
23 %% Convert phi,theta to x,y,z unit directions
24 [x0, y0, z0] = sph2cart(theta, phi-pi/2, 1);
25
26 %% Rotate around x-axis, phix
27 x1 = x0;
28 y1 = y0*cos(phix)-z0*sin(phix);
29 z1 = z0*cos(phix)+y0*sin(phix);
30
31 %% Rotate around y-axis, phiy
32 x2 = x1*cos(phiy)+z1*sin(phiy);
33 y2 = y1;
```



```

34 z2 = z1*cos(phiy)-x1*sin(phiy);
35
36 %% Convert from shifted x,y,z to rotated phi,theta
37 r = sqrt(x2^2+y2^2+z2^2);
38
39 phi_eff = acos(-z2/r);
40
41 %Accounting for discontinuity of trig functions
42 if ((phi_eff==0) || (phi_eff==pi))
43     theta_eff = 0;
44     % Just made it zero because I am only using this program
45     % to simulate data collection. The discontinuity of phi
46     % being 0 or pi makes me lose information. Setting to zero
47     % gives a series of data points that are acceptable to a
48     % model but do not necessarily reflect what the actual
49     % data would be. This one discrepancy does not cause a
50     % problem in the model.
51 else
52     theta_eff = atan2(y2,x2);
53 end
54 end

```

# Appendix B

## Labview Code

This appendix contains images of the Labview code used to run the prototype hardware. The hardware is activated and commanded and the data is collected and written to a file using the Labview functions shown here.

## B.1 Main Function

This is the main function used to run the system. Figure B-1 shows the front panel where the inputs to the system can be modified by the user and any errors are output. At the end of the program a file is written containing the  $\theta$ ,  $\phi$ , and  $d$  data. Figure B-2 shows the inner workings of the main program. The main VI uses several subVIs that will be detailed next.

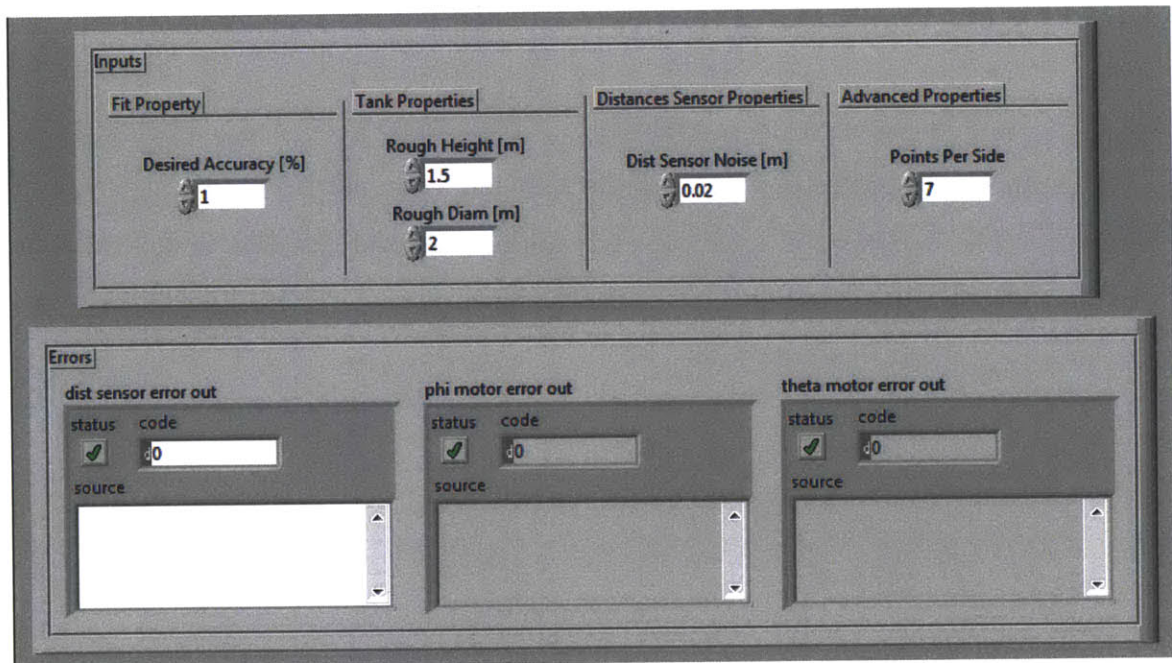


Figure B-1: Main Program Front Panel



## B.2 Point of Interest Calculation subVI

This VI takes in the number of points on the cube edge and calculates the corresponding  $\theta$  and  $\phi$  commands to send to the motor controllers. These POIs are the locations of the normals in space that will be used in the segmentation portion of the algorithm.

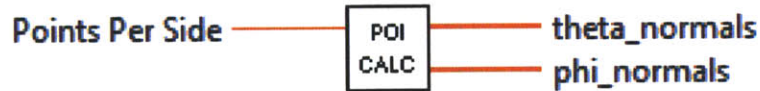


Figure B-3: POI Calculation SubVI Connector

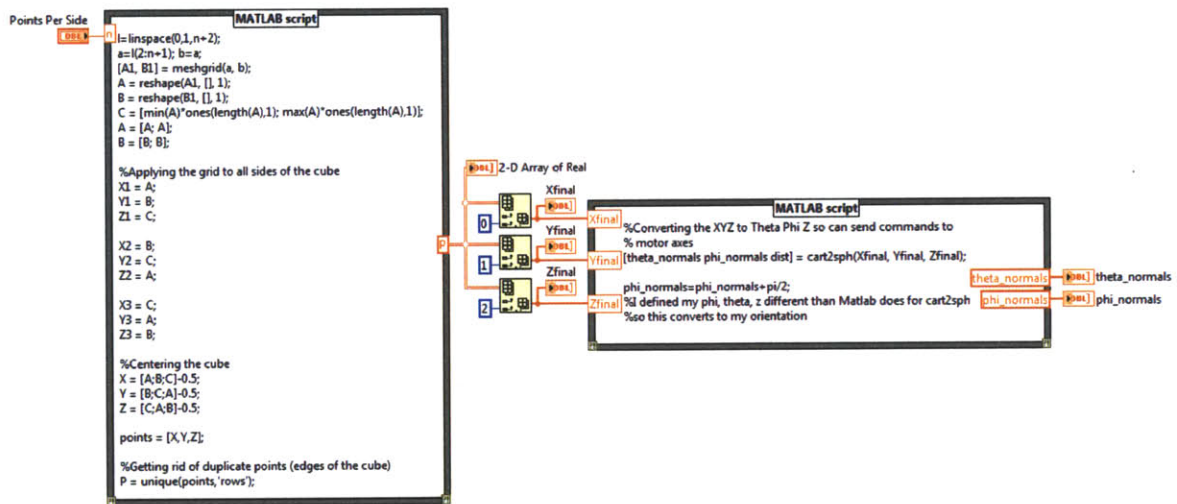


Figure B-4: POI Calculation SubVI Block Diagram

## B.3 Neighborhood Property Calculations subVI

This VI takes in the user inputs and calculates the properties of the neighborhood needed to achieve the desired accuracy.

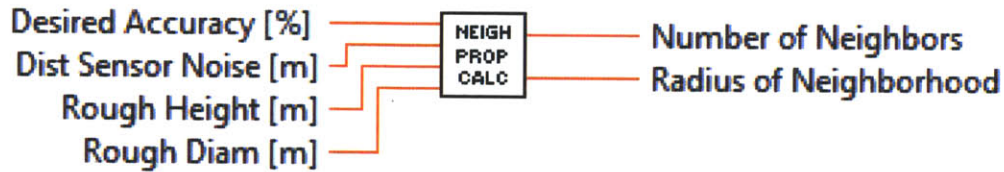


Figure B-5: Neighborhood Property Calculations SubVI Connector

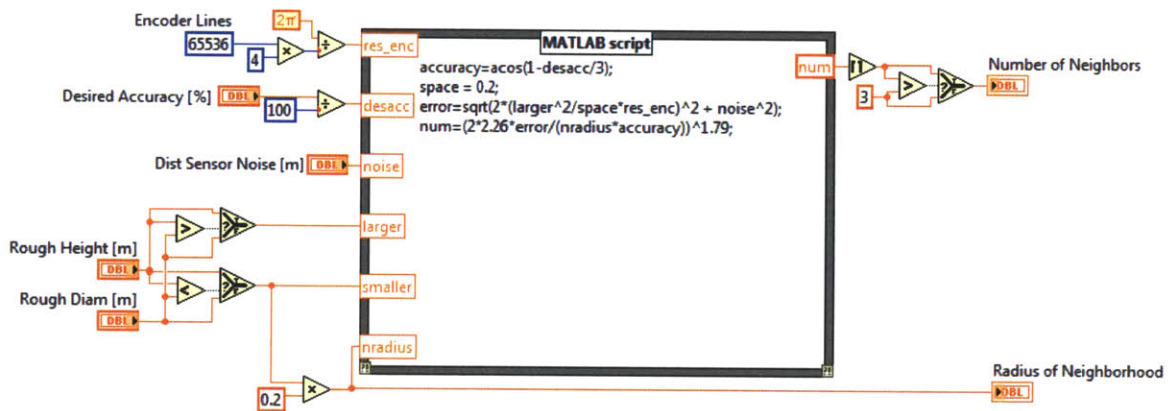


Figure B-6: Neighborhood Property Calculations Block Diagram



## B.4 Neighbor Location Calculations subVI

This VI uses the properties calculated in the Neighborhood Property Calculations subVI and the POI  $\theta\phi d$  information to calculate the position of each of the neighbors and the necessary motor commands to collect data from these neighbors.

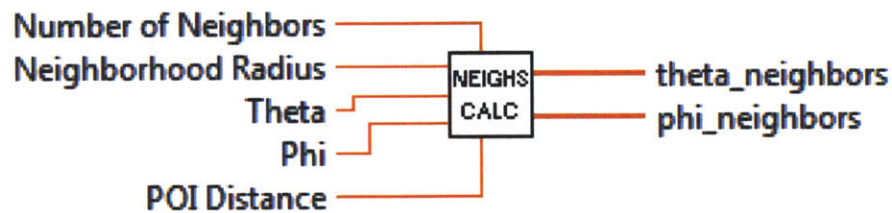


Figure B-7: Neighbor Location Calculations SubVI Connector

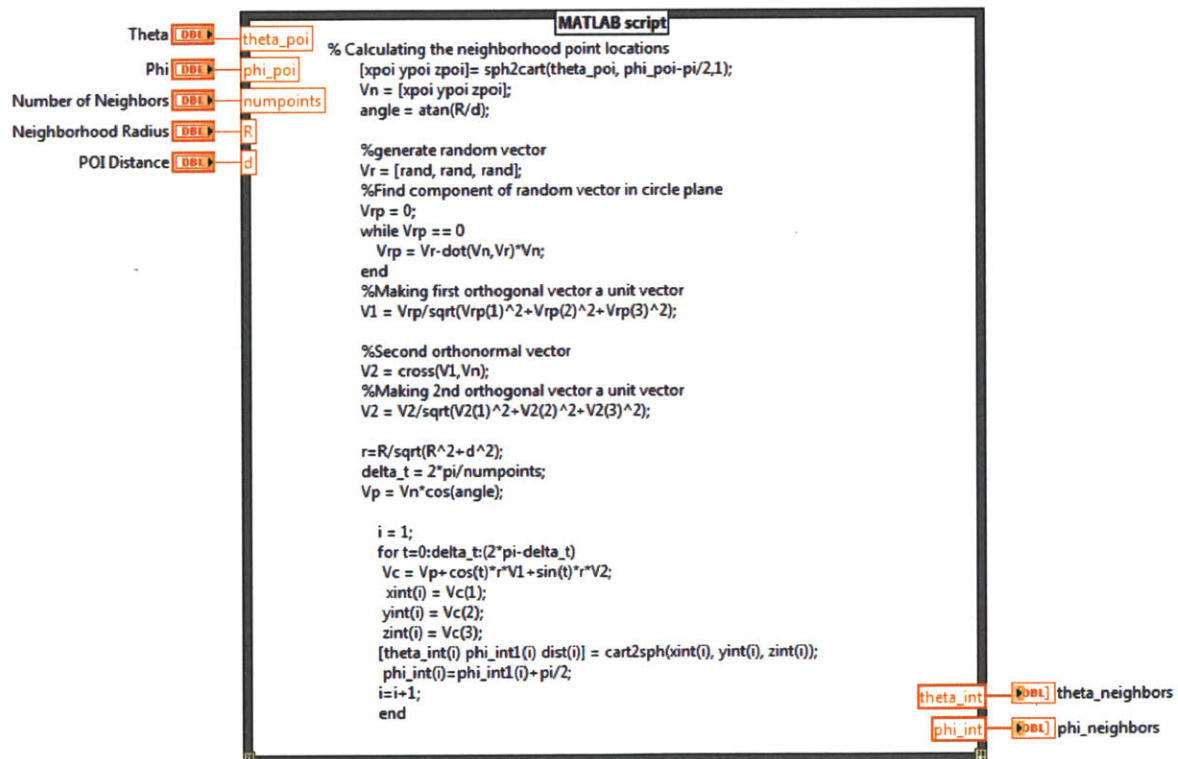


Figure B-8: Neighbor Location Calculations Block Diagram



## B.5 Radian to Quadrature Count and Quadrature Count to Radian subVIs

These VIs convert radians to quadrature counts and quadrature counts to radians. They are needed because the calculations use radians whereas the motor controllers take in and output quadrature counts.

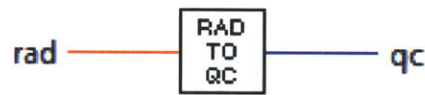


Figure B-9: Radian to Quadrature Count SubVI Connector



Figure B-10: Quadrature Count to Radian SubVI Connector

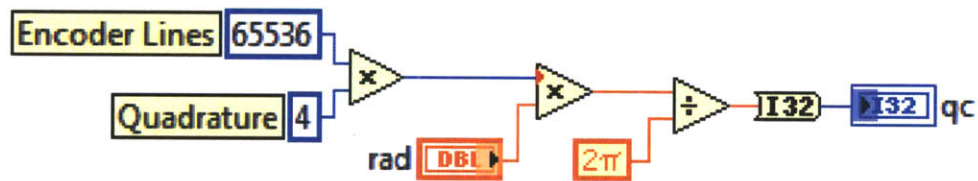


Figure B-11: Radian to Quadrature Count SubVI Block Diagram

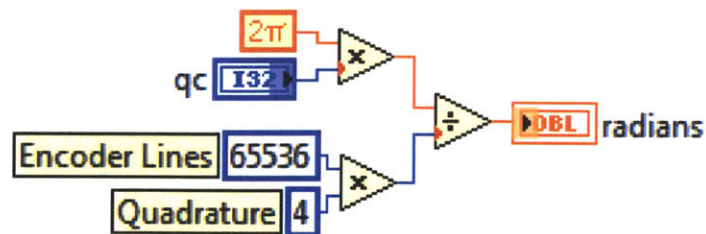


Figure B-12: Quadrature Count to Radian SubVI Block Diagram

## B.6 Motor Movement subVIs

There is one motor movement subVI for each axis. These VIs take in the target axis and the target position. The axis is commanded to move to the target position. When the axis reaches the target position or after a set amount of time passes the subVI is exited and the error in the position is output. Each axis needs its own subVI in order to actuate both axes at the same time. Only the theta axis connector and code is shown in Figures B-13 and B-14 because the code is exactly the same for each axis.

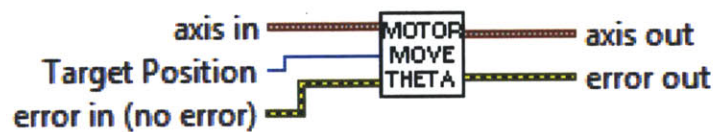


Figure B-13: Motor Movement SubVI Connector

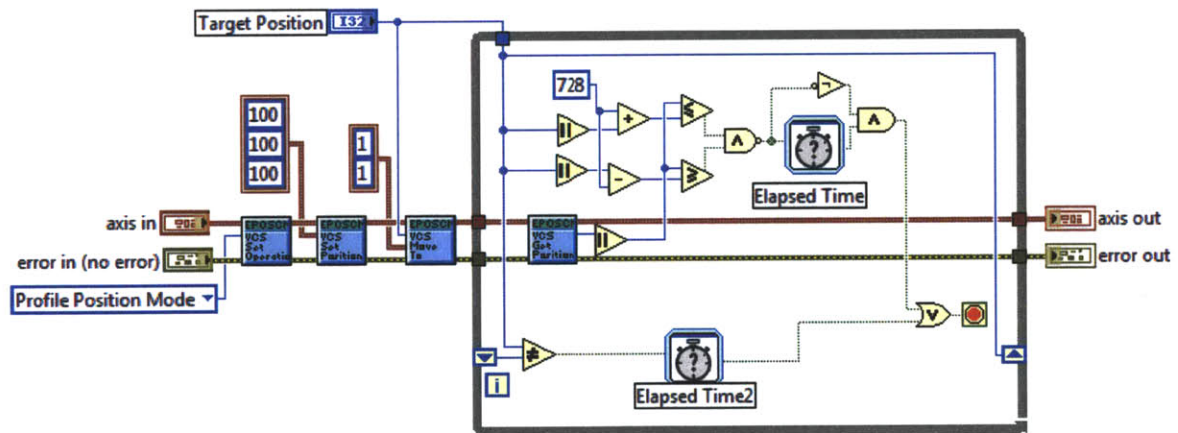


Figure B-14: Motor Movement SubVI Block Diagram

## B.7 Read Distance subVI

The read distance subVI reads in the measurement from the laser rangefinder and converts the signal to a distance in meters.



Figure B-15: Read Distance SubVI Connector

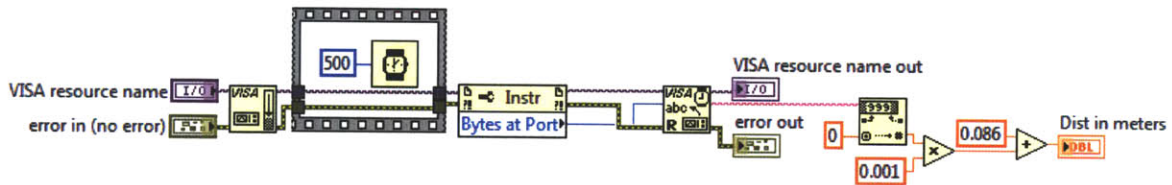


Figure B-16: Read Distance SubVI Block Diagram



# Appendix C

## Outlier Removal Method

The outliers need to be removed before segmentation can occur. Up until segmentation, the algorithm can handle having outliers, but the segmentation and fitting steps are more sensitive. To improve accuracy, extreme outliers are removed from the data. Figure 4-7a shows the original data with the outliers. Figure 4-7b shows the data after the tank is rotated and the outliers are removed.

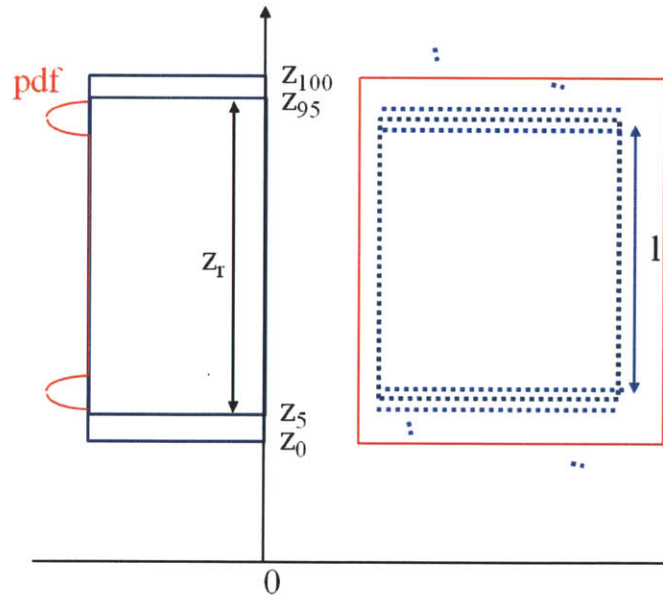


Figure C-1: Clipping process in the  $z$  direction.  $z_5$  is the 5th percentile of the data.  $z_{95}$  is the 95th percentile of the data.  $z_0$  and  $z_{100}$  are the modified extents containing all valid data but excluding outliers.  $l$  is the actual height of the data.

A process of clipping based on quantiles is used to separate the outliers from the valid data. The z axis clipping is shown in Figure C-1. Up to 10% of the valid data will be lost using only the points between the 5th and 95th percentiles. This much loss could bias the circle fit. A method to collect all valid points by adding distance to the 5th and 95th percentiles is outlined below: The range  $z_r$ , is defined as the range between the 5th and 95th percentiles, which is equivalent to 90% the actual length.

$$z_r \equiv z_{95} - z_5 = 0.90l \quad (\text{C.1})$$

The maximum distance within which the valid data will occur is defined as  $z_{100}$ . As shown in Eq C.2,  $z_{100}$  is found by adding another 10% of the length and 3 times the error to  $z_r$ .

$$z_{100} = z_{95} + 0.10l + 3e \quad (\text{C.2})$$

Rearranging Eq C.1 and replacing  $l$  in Eq C.2 results in Eq C.3.

$$z_{100} = z_{95} + \frac{0.10}{0.90}z_r + 3e \quad (\text{C.3})$$

$z_0$  can be found by subtracting the additional distances from  $z_5$  resulting in Eq C.4

$$z_0 = z_5 - \frac{0.10}{0.90}z_r - 3e \quad (\text{C.4})$$

Any data that is outside the bounds of  $z_0$  to  $z_{100}$  is considered an outlier and removed from the data set before segmentation.

A similar process is used for removing outliers in the radial direction as shown in Figure C-2. The equations for the x and y axes' extents are derived using the same logic and are shown in Eq C.5.

$$\begin{aligned} x_0 &= x_5 - \frac{0.10}{0.90}(x_{95} - x_5) - 3e \\ x_{100} &= x_{95} + \frac{0.10}{0.90}(x_{95} - x_5) + 3e \\ y_0 &= y_5 - \frac{0.10}{0.90}(y_{95} - y_5) - 3e \\ y_{100} &= y_{95} + \frac{0.10}{0.90}(y_{95} - y_5) + 3e \end{aligned} \quad (\text{C.5})$$

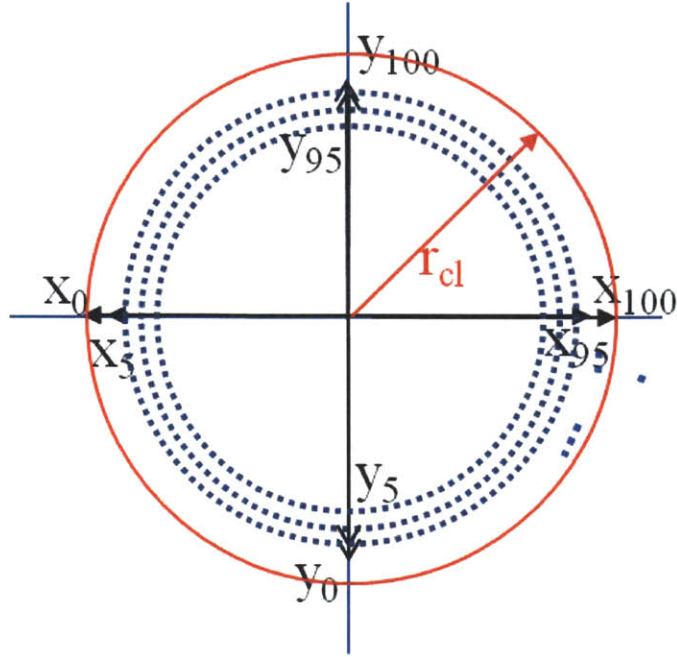


Figure C-2: Clipping process in the radial direction.

The difference with the radial calculation is that instead of clipping a box using the x and y extents, the larger of distance between the extents is considered the bounding radius  $r_{clip}$  as shown in Eq C.6. All points that are outside  $r_{clip}$  are considered outliers.

$$r_{clip} = \frac{\max((x_{100} - x_0), (y_{100} - y_0))}{2} \quad (C.6)$$





# Bibliography

- [1] Photo of Dominion Cove Point LNG facility. Photo Courtesy of Dominion. Lusby, Maryland. [Online]. Available: <http://www.dom.com/business/gas-transmission/cove-point/index.jsp>
- [2] Photo of two LPG storage tanks. Photo Courtesy of Lusas Corporation. [Online]. Available: [http://www.lusas.com/case/civil/images/lng\\_tank\\_0017\\_300.jpg](http://www.lusas.com/case/civil/images/lng_tank_0017_300.jpg)
- [3] Photo of row of LPG storage tanks. Photo Courtesy of Gauging Systems Inc. [Online]. Available: <http://gaugingsystems.net/images/tank10.jpg>
- [4] Photo of external crawler inspection robot. Photo Courtesy of NDT-ED.org. [Online]. Available: <http://www.ndt-ed.org/AboutNDT/SelectedApplications/TankInspection/crawler.jpg>
- [5] Photo of Maverick. Photo Courtesy of NDT-ED.org. [Online]. Available: <http://www.ndt-ed.org/AboutNDT/SelectedApplications/TankInspection/mavericksm.jpg>
- [6] Photo of Neptune. Photo Courtesy of the Field Robotics Center of Carnegie Mellon University. Pittsburgh, Pa. [Online]. Available: [http://www.frc.ri.cmu.edu/~hagen/samplers/figures/NEPTUNE\\_on\\_Rustyplate.gif](http://www.frc.ri.cmu.edu/~hagen/samplers/figures/NEPTUNE_on_Rustyplate.gif)
- [7] Photo of ICM climber with EMAT-NDT. Photo Courtesy of Internation Climbing Machines. [Online]. Available: <http://www.icm.cc/ICMwithEMAT-NDT2.htm>
- [8] Photo of Techcorr cleaning and inspection robot. Photo Courtesy of the Techcorr. [Online]. Available: <http://techcorrintro.com/tank.html>
- [9] Trimble S8: The power to excel. Trimble Navigation Limited. Sunnyvale, California, USA. [Online]. Available: <http://www.trimble.com/trimbles8.shtml>
- [10] Leica TPS1200+ : The total station with the plus. Leica Geosystems. Heerbrugg, Switzerland. [Online]. Available: [http://www.leica-geosystems.com/en/Leica-System-1200-Leica-TPS1200\\_4547.htm](http://www.leica-geosystems.com/en/Leica-System-1200-Leica-TPS1200_4547.htm)
- [11] H. Schempf, B. Chemel, and N. Everett, "Neptune: Above-ground storage tank inspection robot system," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1994, pp. 1403 – 1408.

- [12] V. Masuraha and S. Carr, “Advances aboveground storage tank inspections - in-service robotics tank inspections,” in *NPRA 2004 National Environmental & Safety Conference Sept 27, 2004*, Sept 2004.
- [13] LP gas modern energy anywhere. WLPGA. Paris, France. [Online]. Available: [http://worldlpgas.com/page\\_attachments/0000/0727/what\\_is\\_lp\\_gas.pdf](http://worldlpgas.com/page_attachments/0000/0727/what_is_lp_gas.pdf)
- [14] R. T. Pack, J. L. Christopher, and K. Kawamura, “A rubbertuator-based structure-climbing inspection robot,” in *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, vol. Vol 3, 1997, pp. 1869–1874. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=619060&isnumber=13470>
- [15] W. Shen, J. Gu, and Y. Shen, “Proposed wall climbing robot with permanent magnetic tracks for inspecting oil tanks,” in *2005 IEEE International Conference on Mechatronics and Automation*, vol. 4, 2005, pp. 2072–2077. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1626882&isnumber=34149>
- [16] D. R. Hartsell, “Putting the maverick fuel tank inspection robot to the test,” *IEEE Robotics and Automation Magazine*, vol. Vol 6, Issue 3, 1999.
- [17] T. Somers, “Precise tracking of underwater sensors,” in *Oceans '91. 'Ocean Technologies and Opportunities in the Pacific for the 90's'. Proceedings.*, vol. 1, oct 1991, pp. 274–279.
- [18] S. Atiya and G. Hager, “Real-time vision-based robot localization,” *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 6, pp. 785–800, dec 1993.
- [19] E. Byler, W. Chun, W. Hoff, and D. Layne, “Autonomous hazardous waste drum inspection vehicle,” *Robotics Automation Magazine, IEEE*, vol. 2, no. 1, pp. 6–17, mar 1995.
- [20] P. Gotardo, O. Bellon, and L. Silva, “Range image segmentation by surface extraction using an improved robust estimator,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, june 2003, pp. 33–38.
- [21] G. Lukacs, R. Martin, and D. Marshall, “Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation,” in *Proc. 5th European Cong. Computer Vision*. Springer, 1998, pp. 671–686.
- [22] A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher, “An experimental comparison of range image segmentation algorithms,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 7, pp. 673–689, jul 1996.

- [23] P. Gotardo, K. Boyer, O. Bellon, and L. Silva, "Robust extraction of planar and quadric surfaces from range images," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, aug. 2004, pp. 216–219.
- [24] U. Larsson, J. Forsberg, and A. Wernersson, "On robot navigation using identical landmarks: integrating measurements from a time-of-flight laser," in *Multisensor Fusion and Integration for Intelligent Systems, 1994. IEEE International Conference on MFI '94.*, oct 1994, pp. 17–26.
- [25] J. Forsberg, U. Larsson, and A. Wernersson, "Mobile robot navigation using the range-weighted hough transform," *Robotics Automation Magazine, IEEE*, vol. 2, no. 1, pp. 18–26, mar 1995.
- [26] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," in *Proceedings of the nineteenth annual symposium on Computational geometry*, 2003, pp. 322–328. [Online]. Available: <http://doi.acm.org/10.1145/777792.777840>
- [27] R. Eckhoff, M. Ngo, and W. Olsen, "On the minimum ignition energy (mie) for propane/air," *Journal of Hazardous Materials*, vol. 175, no. 1-3, pp. 293 – 297, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0304389409016409>
- [28] What is GPS? Garmin. Olathe, Kansas. [Online]. Available: <http://www8.garmin.com/aboutGPS/>
- [29] Air Force Space Command. (2010, Sept) Global positioning system. United States Air Force Space Command. Peterson AFB, Colo. [Online]. Available: <http://www.af.mil/information/factsheets/factsheet.asp?id=119>
- [30] J. D. Kraus, *Electromagnetics*, 4th ed. New York: McGraw-Hill, 1992.
- [31] C. Canali, G. De Cicco, B. Morten, M. Prudenziati, and A. Taroni, "A temperature compensated ultrasonic sensor operating in air for distance and proximity measurements," *Industrial Electronics, IEEE Transactions on*, vol. IE-29, no. 4, pp. 336–341, nov. 1982.
- [32] M. Haller and B. Khuri-Yakub, "A surface micromachined electrostatic ultrasonic air transducer," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 43, no. 1, pp. 1–6, jan 1996.
- [33] Acoustic impedance, intensity and power. UNSW Schoole of Physics. Sydney, Australia. [Online]. Available: <http://www.animations.physics.unsw.edu.au/jw/sound-impedance-intensity.htm>
- [34] S. Kingsley and S. Quegan, *Understanding Radar Systems*. Mendham, NJ: SciTech Publishing Inc, 1999.

- [35] 94GHz millimeter wave industrial distance sensor. ELVA-1. Riga, Latvia. [Online]. Available: [http://www.elva-1.com/products/industrial/FMCW\\_Distance\\_Sensor.html](http://www.elva-1.com/products/industrial/FMCW_Distance_Sensor.html)
- [36] *AR3000 Distance Measurement Sensor*, Acuity Laser Measurement, 2010. [Online]. Available: [https://portal.schmitt-ind.com/SelfServe\\_Acuity/Library/Global%20Documents%5CUS%5C/ar3000-data-sheet.pdf](https://portal.schmitt-ind.com/SelfServe_Acuity/Library/Global%20Documents%5CUS%5C/ar3000-data-sheet.pdf)
- [37] R. Hoglund and P. Large, "Direct reflex edm technology for the surveyor and civil engineer," Trimble Integrated Surveying Group, Tech. Rep., 2003. [Online]. Available: [http://www.trimble.com/survey\\_wp\\_totalstations.asp?Nav=Collection-27591](http://www.trimble.com/survey_wp_totalstations.asp?Nav=Collection-27591)
- [38] T. Chaperon and F. Goulette, "Extracting cylinders in full 3d data using a random sampling method and the gaussian image." in *Proceedings of the Vision Modeling and Visualization Conference 2001*, 2001, pp. 35–42.
- [39] H. Asada, "Lecture notes no. 2 part 1: Estimation," Feb 2011.
- [40] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics*, vol. Vol 26(2), pp. 71–78, 1992.
- [41] A. Gray, *Modern Differential Geometry of Curves and Surfaces*. Boca Raton, Florida: CRC Press, Inc., 1993.
- [42] D. Eberly. (2008) Fitting 3D data with a cylinder. Geometric Tools, LLC. [Online]. Available: <http://www.geometrictools.com/Documentation/CylinderFitting.pdf>
- [43] Re 50 datasheet. Maxon Motor. Sachseln, Switzerland. [Online]. Available: [https://downloads.maxonmotor.com/Katalog\\_neu/eshop/Downloads/Katalog\\_PDF/maxon\\_dc\\_motor/RE-programm/new/newpdf\\_11/RE-50-370354\\_11\\_EN\\_083.pdf](https://downloads.maxonmotor.com/Katalog_neu/eshop/Downloads/Katalog_PDF/maxon_dc_motor/RE-programm/new/newpdf_11/RE-50-370354_11_EN_083.pdf)
- [44] EC-max 40 datasheet. Maxon Motor. Sachseln, Switzerland. [Online]. Available: [https://downloads.maxonmotor.com/Katalog\\_neu/eshop/Downloads/Katalog\\_PDF/maxon\\_ec\\_motor/EC-max-programm/new/newpdf\\_11/EC-max-40-283866\\_11\\_EN\\_168.pdf](https://downloads.maxonmotor.com/Katalog_neu/eshop/Downloads/Katalog_PDF/maxon_ec_motor/EC-max-programm/new/newpdf_11/EC-max-40-283866_11_EN_168.pdf)
- [45] Rotary incremental, DFS60. SICK. [Online]. Available: <http://mysick.com/eCat.aspx?go=DataSheet&Cat=Gus&At=Pa&Cult=English&Category=Produktfinder&ProductID=56594>
- [46] Epos2 50/5 catalog page. Maxon Motor. Sachseln, Switzerland. [Online]. Available: [https://downloads.maxonmotor.com/Katalog\\_neu/eshop/Downloads/Katalog\\_PDF/maxon\\_motor\\_control/Positionierung/EPOS/EPOS2\\_50\\_5/new/newpdf\\_11/EPOS2-50-5-347717\\_11\\_EN\\_301-305.pdf](https://downloads.maxonmotor.com/Katalog_neu/eshop/Downloads/Katalog_PDF/maxon_motor_control/Positionierung/EPOS/EPOS2_50_5/new/newpdf_11/EPOS2-50-5-347717_11_EN_301-305.pdf)

- [47] Long range distance sensors Dx500. SICK. [Online]. Available: <http://mysick.com/eCat.aspx?go=DataSheet&Cat=Gus&At=Pa&Cult=English&Category=Produktfinder&ProductID=42971>