THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF INDUSTRIAL AND MANUFACTURING ENGINEERING

SENSOR PLACEMENT FOR ACQUIRING SUBMERGED CONTACTS

MEGAN KATIC
Spring 2011

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Industrial Engineering
with honors in Industrial Engineering

Reviewed and approved* by the following:

Soundar R.T. Kumara
Allen E. Pearce/Allen M. Pearce Professor of Industrial Engineering
Thesis Co-Advisor

Jeffrey J. Weinschenk
Research Associate, Head of Operations Research Department
Applied Research Lab
Thesis Co-Advisor

Paul Griffin
Professor and Head of Industrial Engineering
Honors Advisor

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

Distributed Sensor Networks (DSN) are commonly deployed for undersea surveillance and can be composed of multiple heterogeneous sensors. Many algorithms and methodologies have been developed for the placement of these sensors to monitor a rectangular area of interest. These algorithms have been implemented in various programs, mostly through Monte Carlo simulation. The Geometric Approach presented in this thesis provides a high-level view of the sensor placement problem in order to compare the performance of DSN. The sensors are modeled as disks (circular regions), which represent their acquisition footprint. Any submerged contact that enters this region will be acquired by the sensor. Submerged contacts can only enter along one side of the rectangular area, known as the border. The contacts are described in terms of their offset from a thread axis and the angle they create with the border (also referred to as the heading). Probability density functions (pdfs) are used to describe these offsets and headings, and their combination results in a probability mass function (pmf), referred to as the probability of occurrence matrix. Geometric equations are used to determine which of these offsets and headings are 'covered' by a sensor located at a particular point in the area. A coverage function is developed to summarize these covered combinations. The goal of the approach is to maximize the probability of acquisition, which is calculated as the Frobenius inner product of the probability of acquisition matrix. This acquisition matrix is developed by the Hadamard product of the probability of occurrence matrix and the coverage function. The probability of acquisition of a DSN is plotted against the cost of the DSN and a Pareto analysis is performed to determine which DSN should be implemented for the particular area and pmf. The views expressed in this work is not necessarily the views of the Office of Naval Research (ONR), which is the sponsoring agency.

TABLE OF CONTENTS

TABLE OF CONTENTS ..... **II**

LIST OF TABLES ..... **VI**

LIST OF FIGURES ..... **VII**

LIST OF SYMBOLS ..... **X**

ACKNOWLEDGEMENTS ..... **XIII**

CHAPTER 1: INTRODUCTION **1**

1.1 Undersea Surveillance 1

1.2 Motivation for a High-Level View 1

1.3 Research Objectives 2

1.4 Organization of Thesis 2

CHAPTER 2: BACKGROUND LITERATURE SURVEY **4**

2.1 Anti-Submarine Warfare 4

2.2 Distributed Sensor Networks Overview 5

2.3 Types of Sensors 6

2.3.1 Active Sonar 7

2.3.2 Passive Sonar 9

2.3.3 Electromagnetic Detection Systems 10

2.3.4 Hydrodynamic Detection Systems 11

2.4 Comparison of Stationary and Mobile Platforms 13

2.5 Technological Problems Associated with DSN 14

2.6 Approaches for Solving the Sensor Positioning Problem 16

2.6.1 Track Coverage Problem 16

2.6.2 Set Covering Problem 17

2.6.3 Occupancy Grid Approach 19

2.6.4 Geometric Optimization Approach 20

CHAPTER 3: DSN PERFORMANCE PROBLEM WITH PARETO ANALYSIS..... 23
3.1 Problem Definition and Discussion..... 23
3.2 Assumptions..... 25
3.2.1 Assumptions on the Area of Interest..... 25
3.2.2 Assumptions on the Submerged Contacts..... 26
3.2.3 Assumptions on the Sensors..... 26

CHAPTER 4: METHODOLOGY FOR A SINGLE SENSOR..... 27
4.1 Definition of Area of Interest..... 27
4.2 Parameter Vectors..... 28
4.2.1 Medium Parameter Vectors..... 28
4.2.2 Submerged Contact Parameter Vector..... 29
4.2.3 Sensor Parameter Vector..... 30
4.3 Evaluation Metrics..... 31
4.3.1 Probability of Acquisition..... 31
4.3.2 Lifecycle cost..... 33
4.4 Submerged Contact Tracks..... 33
4.4.1 Discrete Case..... 33
4.4.2 Continuous Case..... 35
4.5 Sensor's Acquisition Radius..... 35
4.5.1 Active Sonar Acquisition Radius..... 36
4.5.2 Passive Sonar Acquisition Radius..... 38
4.5.3 Electromagnetic and Hydrodynamic Sensor Acquisition Radius..... 38
4.6 Sensor's Coverage Function (Discrete Case)..... 39
4.6.1 Range of $\boldsymbol{\delta k}$ Values that Can Result in a Coverage Function Value of One..... 39
4.6.2 Calculations of the Ranges of $\boldsymbol{\theta j}$ When $\boldsymbol{Z \geq D}$..... 40
4.6.3 Calculations of the Ranges of $\boldsymbol{\theta j}$ When $\boldsymbol{Z < D}$..... 45
4.6.4 Coverage Function Calculations..... 47
4.7 The Geometric Approach (Discrete Case)..... 48
4.8 The Geometric Approach (Continuous Case)..... 51

CHAPTER 5: METHODOLOGY FOR MULTIPLE SENSORS 53
5.1 Context and Assumptions 53
5.2 The Geometric Approach for Two Sensors 53
5.2 Geometric Approach for N sensors 57
5.3 Explanation of Pareto Analysis 58

CHAPTER 6: EXPERIMENTATION 60
6.1 MATLAB Program 60
6.2 Results from Experiments of a Single Sensor 60
6.2.1 Complete Results of Run One with Plots 61
6.2.2 Summary of Results 64
6.3 Results from Experiments of Multiple Sensors 69
6.3.1 Complete Results of DSN #1 with Plots 70
6.3.2 Summary of Results 73
6.4 Discussion on Lifecycle Cost 75
6.5 A Pareto analysis of Two Sensor DSN 76

CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH 78
7.1 Contributions 78
7.2 Future Research 78

REFERENCES 79

APPENDIX A: PROOFS FOR EQUATIONS 85
A.1: Proof of Equation (4.14) 85
A.2: Proof of Statement 1 (Congruent Triangles) 86
A.3: Proof of Equation (4.16) 87
A.4: Proof of Equation (4.17) 88
A.5: Proof of Equation (4.18) 88
A.6: Proof of Equation (4.19) 89
A.7: Proof of Equation (4.20) 90
A.8: Proof of Equation (4.21) 91

A.9: Proof of Equation (4.24) 91
A.10: Proof of Equation (4.25) 91

APPENDIX B: RESULTS FROM EXPERIMENTS 92
B.1 Results from Single Sensor Experiments 92
B.1.1 DOE Design 92
B.1.2 Raw Data Results 93
B.1.3 Minitab Results for $\boldsymbol{Q}*$ 94
B.1.4 Minitab Results for $\boldsymbol{x}*$ 96
B.1.5 Minitab Results for $\boldsymbol{y}*$ 98
B.1.6 Minitab Results for Computer Processing Time 100

APPENDIX C: MATLAB CODE 102

LIST OF TABLES

Table 2.1: Summary of the Influences of the Approaches on the Geometric Approach.....22

Table 6.1: Summary of the Factor Levels ……55
Table 6.2: Coding Schemes for Qualitative Four-level Factors……55
Table 6.3: Two-sensor DSN Under Consideration for Example Problem……63
Table 6.4: Results for all DSN from the MATLAB Program…….……..64
Table 6.5: Lifecycle Costs for Sensors…….....65

Table B.1: Experimental Design for a $4^2 2^{5-4}$, or otherwise known as a 2^{9-4}, Design……...74
Table B.2: Results from the $4^2 2^{5-4}$ Experiment…….75

LIST OF FIGURES

Figure 1.1: Thesis Organization……3

Figure 2.1: SNR at two threshold settings T1 and T2……..7
Figure 2.2: Active Sonar…….....8
Figure 2.3: Passive Sonar……9
Figure 2.4: Water Turbulence……..11
Figure 2.5: Coverage cones for track coverage……...16
Figure 2.6: Arc of S_i's Perimeter Covered by S_j……18

Figure 3.1: Example of an Area of Interest……..24

Figure 4.1: Area of Interest……..28
Figure 4.2: Example of the Triangular Distribution for the Offset……..34
Figure 4.3: Nomogram for Computing Transmission Loss…….37
Figure 4.4: Maximum δ_k value at which a sensor at C_i can have a CF value of one…….…….40
Figure 4.5: Coverage cone created by sensor i and a δ_k…….…….41
Figure 4.6: F_i overlapped with a semi-circle centered at a $\delta_k$with a radius of D when $Z \geq D$…41
Figure 4.7: Coverage cone when $Z \geq D$…….…….42
Figure 4.8: Geometry of $\Delta\delta_k CH$ when $Z \geq D$…….……42
Figure 4.9: Geometry for ($\delta_k < x_u$)…….……...43
Figure 4.10: Geometry for ($\delta_k > x_u$)…….……44
Figure 4.11: Footprint F_i overlapped with the semi-circle centered at $\delta_k$with a radius of D when $Z < D$…….……..45
Figure 4.12: Coverage cone when $Z < D$……46
Figure 4.13: Geometry of $\Delta\delta_k C_i H$ when $Z \geq D$…….….46
Figure 4.14: Computational Framework…….49

Figure 5.1: 3D Matrices for N sensors……57
Figure 5.2: Metric space for DSN Analysis……59

Figure 6.1: MATLAB 3D and Pseudocolor Plots for the Probability of Occurrence Matrix….56
Figure 6.2: MATLAB 3D and pseudocolor plots for the CF of a sensor whose R_i = 15nm and position is (51, 10)……57
Figure 6.3: MATLAB 3D and Pseudocolor Plots for the $\boldsymbol{Q_i}$ Matrix……...57
Figure 6.4: Q_iValues For every (x_u, y_v) Combination……58
Figure 6.5: Normal Probability Plot for Q_1^*……59
Figure 6.6: Estimated Effects and Coefficients for Q_1^*……59
Figure 6.7: Normal Probability Plot for x_1^*……60
Figure 6.8: Estimated Effects and Coefficients for x_1^*……60
Figure 6.9: Normal Probability Plot for y_1^*……61
Figure 6.10: Estimated Effects and Coefficients for y_1^*……61
Figure 6.11: Normal Probability Plot for Processing Time……62
Figure 6.12: Estimated Effects and Coefficients for Processing Time……...62
Figure 6.13: MATLAB 3D and Pseudocolor Plots for the Probability of Occurrence Matrix for DSN #1…… 70
Figure 6.14: MATLAB 3D and pseudocolor plots for the CF of a sensor 1positioned at (49, 5)……71
Figure 6.15: MATLAB 3D and pseudocolor plots for the CF of a sensor 2 positioned at (79, 5)……71
Figure 6.16: MATLAB 3D and pseudocolor plots for the TCF for DSN #1……72
Figure 6.17: MATLAB 3D and Pseudocolor Plots for the $\boldsymbol{Q}$ Matrix……..72
Figure 6.18: MATLAB 3D and pseudocolor plots for the CF of a sensor 2 positioned at (77, 4)……74
Figure 6.19: MATLAB 3D and pseudocolor plots for the TCF for DSN #2b……74
Figure 6.20: Q_m^* vs. LC Graph for Pareto Analysis……77

Figure A.1: Geometry for Proof of Equation 4.14 ……67
Figure A.2: Geometry for Proof of Statement 1……68
Figure A.3: Geometry for Proofs of Equations 4.16 and 4.17……69
Figure A.4: Geometry for Proofs of Equations 4.18 and 4.19……70
Figure A.5: Geometry for Proof of Equation 4.20……..72

Figure A.6: Geometry for Proofs of Equations 4.24 and 4.25……73

Figure B.1: Normal Probability Plot for all factors for Q_1^* (with no replicates)……..….77
Figure B.2: Residual Plots for Q_1^* (once non-significant factors are removed)……..……..77
Figure B.3: Normal Probability Plot for all factors for x_1^* (with no replicates)……..…….79
Figure B.4: Residual Plots for x_1^* (once non-significant factors are removed)……..……..79
Figure B.5: Normal Probability Plot for all factors for y_1^* (with no replicates)……..…….81
Figure B.6: Residual Plots for y_1^* (once non-significant factors are removed)……..…..81
Figure B.7: Normal Probability Plot for all factors for processing time (with no replicates)….83
Figure B.8: Residual Plots for processing time (once non-significant factors are removed)…..83

LIST OF SYMBOLS

N	Number of unique sensors under consideration, i = 1,2,…N, p. 23
M	Number of DSN under consideration, p. 24
S	Area of interest, p. 29
W	Width of area of interest, p. 29
L	Length of area of interest, p. 29
A	Left boundary of width located at coordinates (-1/2W, 0), p. 29
B	Right boundary of width located at coordinates (1/2W, 0), p. 29
Δx	Increment (in nm) of the change in x values, p. 29
$\Delta\delta$	Increment (in nm) of the change in the offset values, p. 29
Δy	Increment (in nm) of the change in y values, p. 29
U	Set of x values; Number of possible x_u values, U=W/Δ_k, p. 29
V	Set of y values; Number possible y_v values, V=L/Δ_l, p. 29
x_u	x coordinate of the center of Sensor, where u = 1,…,U, p. 29
y_v	y coordinate of the center of Sensor, where v = 1,…,V ,p. 29
TL	Transmission Loss, p. 30
ANL	Ambient Noise Level, p. 30
RNL	Reverberation Noise Level, p. 30
MD	Depth of medium, $D \in \{0,1\}$, p. 30
v	Velocity of the submerged contact, p.31
δ_k	Element of the offset matrix **δ**; Potential offset value, k = 1,…,K p. 31
θ_j	Element of the heading matrix **θ**; Potential heading value, j = 1,…,J p. 31
RN	Radiated Noise by the submerged contact for passive sonar, p. 31
TS	Target Strength of the submerged contact for active sonar, p. 31
DT_i	Detection Threshold of Sensor i, p. 32
AG_i	Array Gain of Sensor i, p. 32
SL_i	Source Level of Sensor i p. 32
FN_i	Flow Noise of Sensor i, p. 32
SN_i	Self Noise of Sensor i, p. 32

v_{maxi}	Maximum velocity of Sensor i, p. 32
BC_i	Base Cost of Sensor i, p. 32
SC_i	Sensing Cost of Sensor i, p. 32
Q_i	Probability of Acquisition for a sensor, p. 33
T	Patrol Time for the DSN, p. 33
LC	Lifecycle Cost Metric, p. 34
$\boldsymbol{O}$	Probability matrix containing the probability of occurrence, p. 34
o_{jk}	Probability of occurrence for θ_j and δ_k, $o_{jk} \in \boldsymbol{O}$, $o_{jk} = P(\theta_j \cap \delta_k)$, p. 34
$\boldsymbol{CF_i}(x_u, y_v)$	Coverage Function of a sensor located at (x_u, y_v), p. 34
$cf_{ijk}(x_u, y_v)$	Coverage Function value for θ_j and δ_k, $cf_{ijk} \in \boldsymbol{CF_i}(x_u, y_v)$, p. 34
$\boldsymbol{Q_i}(x_u, y_v)$	Probability of acquisition matrix for a sensor at (x_u, y_v), p.34
δ	Set of the Submerged contact's offset from the thread axis, $A \leq \delta_k \leq B$, **δ** = $[\delta_k]$, p. 35
θ	Set of the Submerged contact's heading with respect to the border $\overline{AB}$, $0 \leq \theta_j \leq \pi$, **θ** = $[\theta_j]$, p. 35
$\Delta\theta$	Heading Value Increment (in radians), p. 35
K	Set of offset values; Number of offset values, K=W/Δ_k, p. 35
J	Set of heading values; Number of heading values, J=π/Δ_j, p. 35
$C_i(x_u, y_v)$	Center of Sensor located at (x_u, y_v), p. 36
R_i	Acquisition radius of Sensor i, p, 36
F_i	Acquisition Footprint of Sensor i, represented as a circle with center $C_i(x_u, y_v)$, p. 36
γ	Acquisition coefficient, p. 38
$\alpha_{iuv}(\delta_k)$	Lower bound for range of headings of a sensor for offset δ_k, p. 40
$\beta_{iuv}(\delta_k)$	Upper bound for range of headings of a sensor for offset δ_k, p. 40
$\overline{\delta_k C_i}$	Line segment connecting sensor's center to offset δ_k, p. 40
Z	Length of the line segment $\overline{\delta_k C_i}$, $Z = \sqrt{y_v^2 + (x_u - \delta_k)^2}$, p. 40
D	Maximum distance that a submerged contact can travel during the patrol time T, $D = v * T$, p. 40

G	Point on the circumference of F_i that intersects the line of length *D* that creates the angle α_{uv} with the border $\overline{AB}$, p. 41
H	Point on the circumference of F_i that intersects the line of length *D* that creates the angle $\beta_{uv}$with the border $\overline{AB}$, p. 41
$\overline{\delta_k G}$	Line segment of length *D* that creates the angle α_{uv} with the border $\overline{AB}$, p. 41
$\overline{\delta_k H}$	Line segment of length *D* that creates the angle β_{uv} with the border $\overline{AB}$, p. 41
$\Delta\delta_k C_i H$	Triangle created by line the segments $\overline{\delta_k C_\iota}$, $\overline{\delta_k H}$, and R_i, p. 41
$\Delta\delta_k C_i G$	Triangle created by line the segments $\overline{\delta_k C_\iota}$, $\overline{\delta_k G}$, and R_i, p. 41
r	Angle created by the line segments $\overline{\delta_k C_\iota}$ and $\overline{\delta_k H}$ or $\overline{\delta_k C_\iota}$ and $\overline{\delta_k G}$, p. 42
z	Angle created by the line segments R_i and $\overline{\delta_k H}$ or R_i and $\overline{\delta_k G}$, p. 42
d	Angle created by the line segments R_i and $\overline{\delta_k C_\iota}$, p. 42
E	Length of $\overline{\delta_k G}$ and $\overline{\delta_k H}$ when Z < D, p. 46
x_i^*	x coordinate for optimal performance of a sensor, p. 50
y_i^*	y coordinate for optimal performance of a sensor, p. 50
$Q_i^*(x_i^*, y_i^*)$	Optimal probability of acquisition for a sensor, p. 50
$\boldsymbol{CF_i}(x_i^*, y_i^*)$	Coverage function associated with (x_i^*, y_i^*), p. 50
$\boldsymbol{TCF}$	Total coverage function for a DSN, p. 55
tcf_{jk}	Total coverage function value , $cf_{ijk} \in \boldsymbol{CF_i}(x_u, y_v)$, p. 34
Q_m^*	Optimal probability of acquisition for DSN *m*, p. 56

ACKNOWLEDGEMENTS

I would like to thank professor Dr. Soundar Kumara for all his words of wisdom and guidance throughout the development and writing of this thesis. Without his focus, expertise, perseverance, and commitment to the problem, this thesis would not have taken shape and successfully completed. I want to also acknowledge my IE honors advisors over the past three years, Dr. Paul Griffin and Dr. Richard Koubek for pointing me in the direction of the Integrated Undergraduate/Graduate Program, which enabled me to gain more from the vast body of knowledge at our IE department. I also would like to thank all the professors who have taught me during my five years at Penn State, both at the Behrend College and at University Park.

For providing me insight into the plethora of research topics concerning undersea surveillance sensor networks, I would like to thank Dr. Jeffrey Wienschenk who was my co-advisor on this work. He helped me to understand the impact of my work on the field of undersea surveillance and strive to bring new ideas to the research area. Also, I want to thank those at the Applied Research Lab (ARL) of the Penn State University who met with me to discuss my work and for the sponsorship of the Office of Naval Research (ONR) Code 32, grant #N00014-05-G-0106, under the Persistent Littoral Undersea Surveillance Network (PLUSNet) program.

I also would like to thank my fellow graduate students at the Laboratory for Intelligent Systems and Quality (LISQ) at Penn State for their stimulating discussions on complex networks and for sharing their research experiences with me. I wish you all luck with your future endeavors.

Most importantly, I would like to thank my family for all their love and support throughout my entire academic career. Without them, I would not be where I am today. They have provided me encouragement and strength when I couldn’t find hope and celebrated in all my accomplishment, whether they be large or small. They have listened to everything I had to say, even if they didn’t fully understand any of it, and I appreciated it immensely. Thank you for believing in me!

Chapter 1: Introduction

This chapter introduces the applications of Distributed Sensor Networks (DSN) in undersea surveillance and the need for a high-level view of these systems. The objectives for the research are defined and the organization of the thesis is provided.

1.1 Undersea Surveillance

Since the early twentieth century, undersea surveillance has been used by the United States Navy in anti-submarine warfare (ASW) to protect its interest and support its allies [1]. Sensor networks for ASW began with the introduction of the DSN program by the Defense Advanced Research Projects Agency (DARPA) around 1980 [2]. While the United States has been advancing its knowledge in ASW, other countries have been enhancing their ASW technology and expanding their submarine fleets. The People's Republic of China currently has 60 attack submarines and it is expected that the fleet will expand to include a total of 78 submarines by the year 2025; and India, Australia, Indonesia, and South Korea are all expected to increase the number of attack submarines in their fleets [3]. However, the United States plans to decrease its submarine fleet from 53 attack submarines in 2009 to 41 submarines by 2028 [3]. This declining submarine fleet does not aid in the Navy's ASW efforts. As a result, other ASW platforms, besides submarines, must be enhanced and their usage increased in order to maintain U.S. control of the oceans.

1.2 Motivation for a High-Level View

Most of the research concerning undersea surveillance with sensor networks has consisted of simulation programs and testing of very specific systems, such as the PLUSNet system in [4] and [5]. These programs tend to determine the location and motion of sensors as submerged contacts are sent into the simulated area in a Monte Carlo fashion [6, 7]. Although these approaches may be appropriate for individual systems and very specific submerged or surface targets, they do not provide information concerning the ideal sensors for a given environment. These programs also depend on the order and location at which the simulated submerged contacts enter the area of interest.

We propose a high-level view of undersea surveillance which provides several benefits over the traditional Monte Carlo approach. First, the new approach allows users to investigate many DSN in a relatively short amount of time. Secondly, submerged contacts are explained by a probability mass function (pmf) that takes into account both the headings (angles) and offsets (positions) of the point of entry of submerged contacts into the area of interest. This pmf can be changed by the probability density functions (pdf) describing the headings and/or offsets to reflect what the user believes may be the most likely pattern of entrance. This allows a variety of entrance configuration to be tested, and the sensors can be placed in response to the pmf instead of reacting to individual submerged contacts as in Monte Carlo simulations. The approach also allows users to investigate how the patrol time can affect the performance of the DSN. Within the methodology of our approach, it is assumed that submerged contacts travel at a set velocity, and in combination with the defined patrol time of the DSN, we can calculate the maximum distance that a submerged contact can travel during that time. This can affect the positioning of the sensors and thus their performance.

1.3 Research Objectives

The main objectives of this research are: (1) Develop a high-level approach in both a discrete and continuous case to determine the optimal DSN to detect, classify, localize, and track (DCLT) submerged contacts in a defined two-dimensional area of either shallow or deep sea water; and (2) Build a computer program to test DSN at a high level using the methodology developed..

1.4 Organization of Thesis

The organization of the thesis is shown in figure 1.1. Chapter 2 provides background literature focusing on ASW, various types of sensors, technological challenges of DSN, and the various optimization problems used in the placement of sensors. The formal definition of the problem along with a list of assumptions is explored in chapter 3. The methodology for a single sensor is presented in chapter 4 along with the equations and pseudo code shown in detail. Chapter 5 extends the methodology to multiple sensors, while chapter 6 includes an explanation of the MATLAB program developed from the methodology as well as the results from several experiments. The conclusion with contributions and future research is in chapter 7. Appendix A

includes the proofs of equations from chapter 4 and appendix B includes all the results from the experiments that were conducted. The MATLAB code is included in appendix C.

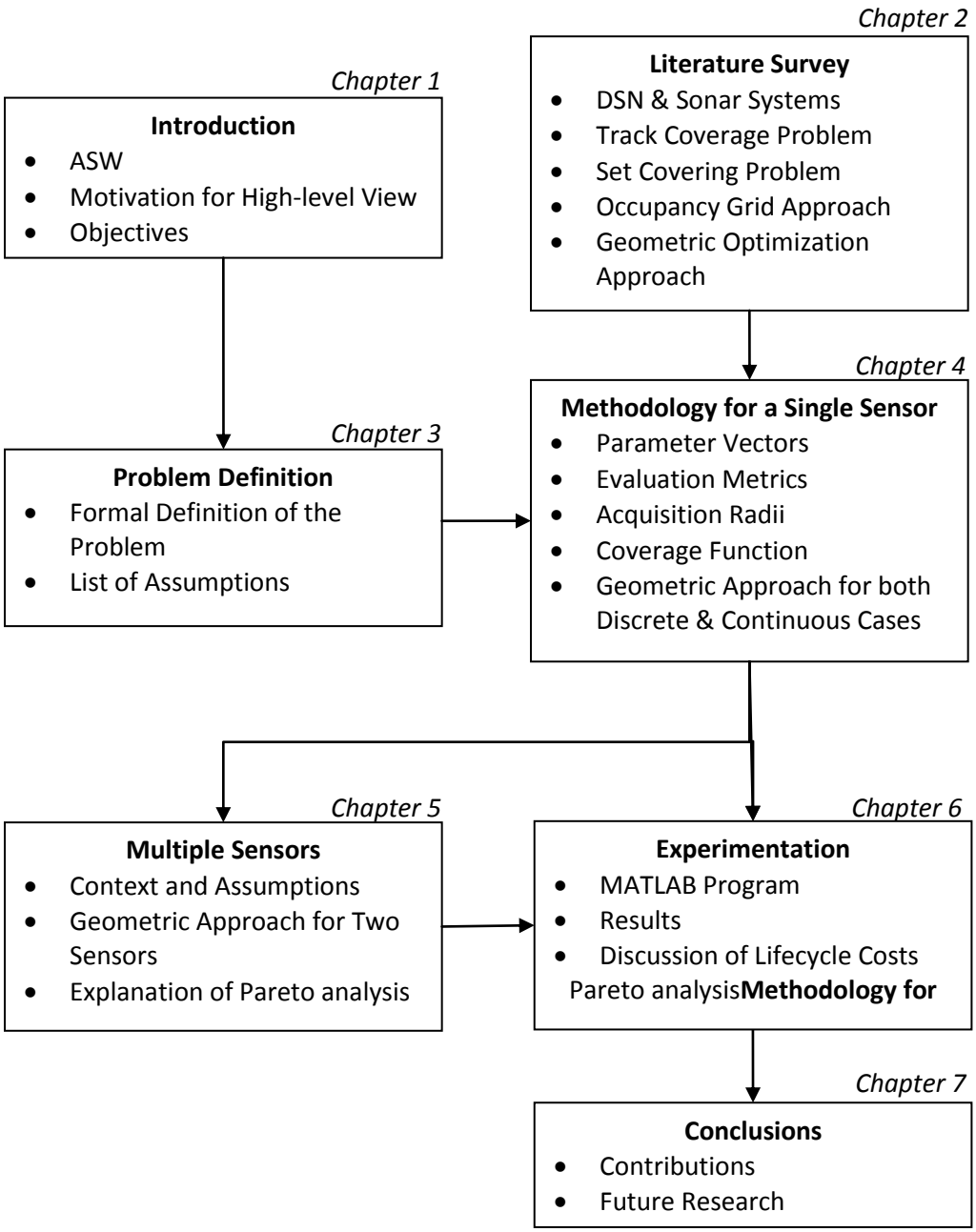

Figure 1.1: Thesis Organization

Chapter 2: Background Literature Survey

This chapter provides background for the problem that is introduced in chapter 3. A brief summary of the history of ASW and DSN is included. Second, details on the various types of sensors will be presented and a comparison of stationary and mobile sensor will be discussed. The technological challenges associated with DSN are briefly discussed in order to explain the acquisition radii in §4.5. The track coverage and set coverage problems along with the occupancy grid and geometric optimization approaches will be discussed in terms of their applications and their influence on the methodology presented in chapter 4.

2.1 Anti-Submarine Warfare

ASW is the combination of "ships, aircraft, submarines, and other platforms to detect, track, and destroy enemy submarines" [3]. Each platform has its own strengths and weaknesses and typically more than one platform is deployed in a particular system. Surface ships can carry many ASW technologies, such as multiple types of sonar and communication equipment, but are vulnerable to attacks from enemy submarines. Aircrafts can monitor a larger area and can quickly make maneuvers and are the least vulnerable of the platforms, but are extremely expensive and tend to be used for shorter patrol times. Submarines are considered the best ASW platforms since they operate in the same environment and have similar strengths and vulnerabilities as the submerged contacts of the enemy. Other platforms, such as gliders, sonobuoys, and autonomous underwater vehicles (AUVs) or unmanned underwater vehicles (UUVs), provide unmanned options to monitor large areas of the ocean for short and long periods of time. However, these other platforms can be expensive, are designed for single use only, and are usually deployed in hostile areas [1, 3].

One example of an ASW system is the Persistent Littoral Undersea Surveillance Network (PLUSNet), which is an autonomous system that is composed of multiple sensors on multiple platforms. The system uses stationary sensors placed on the ocean floor, UUVs with towed arrays, buoys, and gliders to DCLT diesel submarines in littoral areas. Each platform contains different types of acoustic sonar depending on the mission of the platform. Gateway buoys, which are considered stationary but may be vulnerable to ocean currents, are used to relay

information about the system to the control station located on shore. Seagliders are used as the mobile version of the gateway buoys but with the additional capability of diving to relay information back to the submerged sensors. The main purpose of the fixed bottom sensors and other mobile AUVs are to monitor the area of interest and detect and track the submarines [4, 5]. The PLUSNet system will be revisited in §2.5 in regards to its technological challenges.

2.2 Distributed Sensor Networks Overview

DSN for undersea surveillance are primarily used for the DCLT of multiple targets as they move within a specified area of interest [8]. The PLUSNet is an excellent example of a modern DSN, but sensor networks have been used by the U.S. military for several decades. The Sound Surveillance System (SOSUS), which is a system of acoustic sensors located on the ocean floor, was deployed during the Cold War to detect and track Soviet submarines. Currently, SOSUS is used to monitor seismic and marine life among other oceanic events. Beginning in 1980, the DSN program at DARPA has researched the technological components and challenges of DSN. A DSN typically has four main components: (1) sensors; (2) communication protocols; (3) processing techniques with specific algorithms; and (4) distributed software [2]. The technological challenges associated with some of these DSN components are presented in §2.5.

Sensor networks have helped the military transition from platform-centric warfare, in which sensors are controlled by independently operating platforms, to a more network-centric warfare, which allows sensors to collaborate via a communication network and work cooperatively. These networks improve detection and tracking performance since the sensors can provide multiple observations along with diverse methods of detection. Also, sensors working cooperatively provide a larger detection range and faster response times to events [2].

These DSN can be a homogenous or heterogeneous collection of a large number of smaller sensors or consist of only a few larger sensors. The sensors can be passive or active, stationary or mobile, sparsely distributed or densely packed, have centralized or distributed communication infrastructure, and have short or long lifecycles. The next section discusses the most common types of sensors- active and passive sonar, hydrodynamic, and electromagnetic. Other types, such

as seismic, are used in DSN, but are more common with the environmental monitoring of the oceans instead of the DCLT of hostile forces.

2.3 Types of Sensors

Sonar (SOund Navigation And Ranging) was used pre-World War I for detecting underwater objects by echo ranging. During WWI, it was used mainly for submarine-to-submarine communication. It was not until World War II that sonar became an integral part of ASW. Sonar can fall into two broad categories: Passive and Active. The difference between the two categories focuses on how the sonar receives sound. The acoustic field contains both the desired sound, referred to as the signal, and undesired sound, which is referred to as background noise. A detection threshold (DT) is set as the signal-to-noise ratio (SNR) and refers to the sound level measured in decibel units. Sounds recorded above the level indicate that a target is present and sounds below the level are disregarded as noise and that the target is absent [1]. Each sonar type is characterized by its own advantages and disadvantages in regards to correctly detecting the signals while correctly rejecting the background noise. Figure 2.1, which is taken from [1], describes the SNR. Three signals are shown in (a) and the addition of background noise (b) results in the image shown in (c). Two SNR thresholds are indicate in (c) as T_1 and T_2. All three signals are detected with T_2 but there are also many false alarms. Conversely, only the third signal is detected with T_1 and the other two signals are missed. Low SNR thresholds can result in more false alarms and unnecessary actions to investigate these false alarms, while high SNR thresholds can miss target signals. Therefore setting the correct DT is a delicate balancing act.

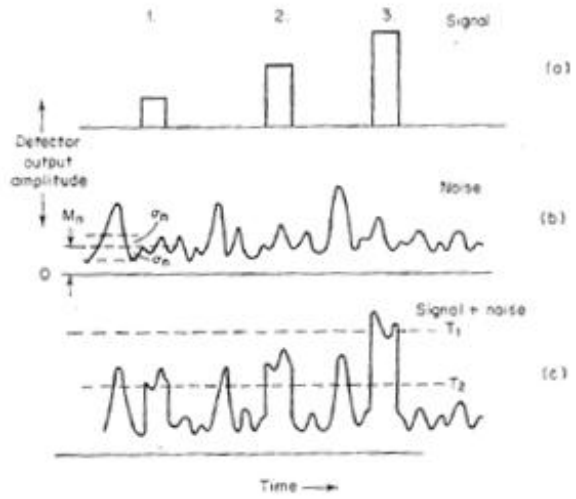

Figure 2.1: SNR at two threshold settings T1 and T2; (a) shows the target signals, (b) shows the background noise, and (c) shows when the noise and signals are combined [9]

Two other types of detection systems, known as hydrodynamic and electromagnetic systems, are emerging in the ASW field. Hydrodynamic systems measure the cavitations in the ocean as a response to a passing object [24]. Electromagnetic systems, also commonly referred to as Magnetic Anomaly Detection (MAD) systems, search a given area for any anomaly in the Earth's magnetic field that may be produced by the target [10]. These two system types are relatively new and there is limited information regarding their performances due to the classified nature of the research concerning such systems. However, they are included in the set of possible sensors since the technology may become more prominent in the ASW field in the near future.

2.3.1 Active Sonar

Active sonar devices radiate a sound into the ocean via a projector. The sound travels through the water until it reaches an object. The sound bounces off of the object and returns to the receiver (also known as a hydrophone) that is associated with the projector; that is the objects create an echo of the sound radiated by the active sonar. Typically, there is more than one hydrophone on a device in the form of an array. An array includes a number of spaced elements (hydrophones) arranged in usually a plane, cylindrical, or line configuration. The number and type of hydrophones in an array affects that array gain (AG), which is the improvement of the SNR [1]. Much research has been done on array geometry and it has been found that cylindrical arrays are

typically very beneficial due to their omni-directional characteristic [11]. Figure 2.2 shows how active sonar works.

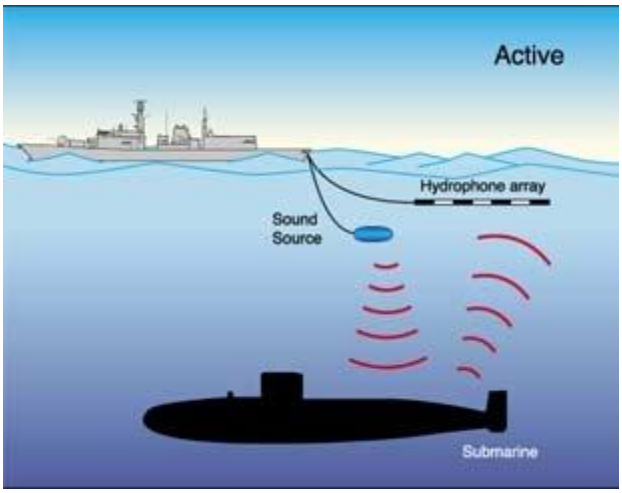

Figure 2.2: Active Sonar [10]

The DT must be set to disregard the source level (SL) of the radiated sounds from the projector and the reverberation noise level (RNL) from the sound that is returned by the ocean floor and the surface of the ocean. The effectiveness of the active sonar also depends on the ability of the targets to return the radiated sound. An object has a specific target strength (TS) depending on the object's shape, material, and the type of sound radiated by the projector [1].

An active sonar device can be monostatic, bistatic, or multistatic. Monostatic active sonar means that the projector and receiver are collocated on the same device. Bistatic and multistatic active sonar are characterized by a projector and receiver that are not collocated [1, 13]. Also, the device can be either low frequency or high frequency, depending on the needs of the human observer.

Also, the signals generated are generally either continuous wave (CW) or frequency modulated (FM) pulses. CW pulses provide good Doppler frequency resolution, but poor range resolution and suppression of reverberation. FM pulses are the complete opposite with poor Doppler frequency resolution, but high range resolution and suppression of reverberation [13].

Some advantages of active sonar include the following: (1) it does not depend on the noise generated by the submerged object; and (2) typically it can locate an object better than passive sonar since it is listening for the specific sound that it emitted [14]. The disadvantages of active sonar include: (1) reverberated sound is captured by the receiver and sometimes can mask the true signal that a target is present; (2) the sonar device emits a sound that can be detected by hostile sonar devices; and (3) it depends on the target strength of the target.

2.3.2 Passive Sonar

Passive sonar is simpler than active sonar since it does not radiate any sound, but rather uses sound that is radiated by a submerged object. A passive sonar platform typically consists of a set of hydrophones listening for sounds. This type of sonar detects ambient noise level (ANL) that is associated with the background noise of the ocean, along with the radiated noise (RN) of the target. Figure 2.3 shows how passive sonar works.

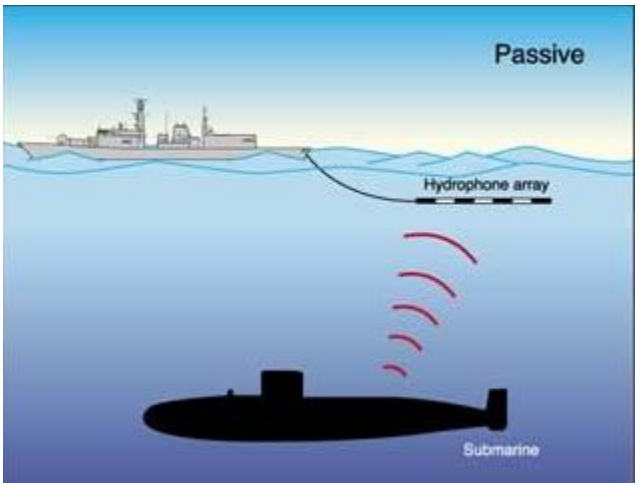

Figure 2.3: Passive Sonar [15]

The advantages of passive sonar include: (1) it cannot be as easily detected by hostile sonar devices as active sonar can since it does not radiate any sound [14]; (2) power consumption is lower since it does not radiate any sound [16]; and (3) there are no reverberation sounds that could cover the signals from the targets. The main disadvantage of passive sonar is that the performance of the sonar depends on the level of RN from the target.

There are similar limitations in acoustic sonar that can affect the range of sensor. Both active and passive sonar can be operated at either high or low frequency. High frequency results in shorter ranges than low frequency. The depth at which the sensor operates affects the speed of sound and therefore the range. Also, there can be many sources of interference from the self noise (SN) of the device, flow noise (FN) caused by the movement of the sensor in the water, and other background noise [17].

2.3.3 Electromagnetic Detection Systems

Electromagnetic (EM) sensors work in a similar fashion to acoustic sonar. An electromagnetic field is transmitted from the sensor and a receiving antenna picks up the transmitted field and signals caused by background fields, such as the Earth, and objects. Although the conventional approach for detecting with electromagnetic sensors is to move the sensors to make additional measurements, the field can greatly change due to time, temperature and shock as the sensor moves. To negate these effects, research is being done to improve the measurement systems as to collect more data while in one location [16]. Also, the aquatic environment can limit the detection range of the EM sensor as well as the frequency range at which they may work since water is a conducting medium [19].

A common detection technique known as Magnetic Anomaly Detection (MAD) is used to locate hidden ferromagnetic objects. The concept of MAD is to detect anomalies in the Earth's magnetic field due to the presence of an object, and can be either a search or alarm system. For a search system, magnetic sensors are placed on a moving platform that follows a predefined path and survey the area for hidden ferromagnetic targets. A spatial magnetic anomaly indicates that a target is present. Alarm systems follow the same concept except the magnetic sensors are placed on a stationary platform and an alarm is triggered when a ferromagnetic target passes by the sensors. It is assumed that the distance between the target and the sensors is relatively large compared to the dimensions of the target so that the target magnetic field can be described by a dipole model [10]. As with any detection system, MAD must use signal processing methods to enhance the SNR.

MAD systems are more commonly found on ASW aircraft instead of submerged platforms. CAE, a manufacturer or ASW systems, has designed a MAD system that uses a highly sensitive sensor to monitor the changes in the Earth's magnetic field caused by a metallic object in the vicinity. The system uses an amplifier computer to process data and to provide digital outputs, while a magnetometer assembly monitors the vectors on the Earth's magnetic field with respect to the aircraft's position and orientation to account for aircraft maneuvers. The CAE MAD system can detect anomalies as far away as 1,200 meters and is used by militaries worldwide [20].

The main advantage that EM has over acoustic sonar is that EM waves are faster than sound waves and can operate at higher frequencies. However, there is high attenuation associated with EM waves propagating in the water and large antennas are needed for higher frequencies [17].

2.3.4 Hydrodynamic Detection Systems

Hydrodynamic detection systems use the cavitations and wake turbulence created by passing objects to detect targets. The hydrodynamic wake is a pattern of vortices in the water created by a passing object [21]. Both surface ships and submerged objects create wakes, but underwater wake turbulence is more difficult to measure. Figure 2.4 shows a schematic of several hydrodynamic mechanisms found in the wake of a submarine.

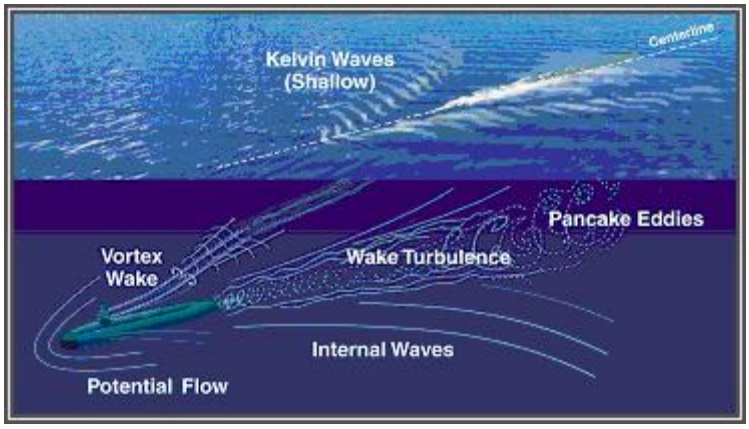

Figure 2.4: Water Turbulence [22]

Since the early 1960s research has been conducted concerning hydrodynamic detection systems. Probes were mounted on a submarine operating in the Pacific Ocean in the early 1960s, and

horizontal measurements of oceanic turbulence were taken and results showed that the probes were able to find sharp interfaces between the turbulent and quiet layers in the water [23]. Research conducted in the 1970s extended to vertical profiles of temperature and velocity fluctuations. Experiments were conducted on the *USS Dolphin* submarine in 1980 and 1982 using probes and sensors, and the system was able to successfully measure turbulent dissipation and temperature fluctuations [23]. In 1984, the researchers at the Naval Research Laboratory developed a model to explain the hydrodynamic wake of a surface ship [24]. More experiments were conducted on the *USS Dolphin* in 1988 to investigate the roles of breaking waves, bubbles, and turbulence in the upper layer of the water column [25]. These experiments helped further researcher's understanding of hydrodynamic turbulence and how to measure the phenomenon.

In 2006, a team of researchers worked on developing an artificial lateral line system that mimics the lateral line, which is a "distributed linear array of flow sensing organs [used] for underwater hydrodynamic imaging and information extraction," found in fish. The artificial lateral line is designed with "a monolithically integrated array of micro fabricated flow sensors" to locate a moving dipole source by a signal-processing algorithm. The dipole near-field flow is created when an object moves in an oscillating fashion, such as a fish moving its tail. The artificial lateral line can also identify the signature of a wake and the general direction of that object. The signature is determined by recording the "spatial distribution of local velocity fluctuations" [26].

The primary disadvantage of a hydrodynamic sensor is that the detection can only occur once the target passes the sensor and only if the sensor array passes across the wake. The sensors can only detect targets in the near distance. Therefore these types of sensors are more reactive than proactive devices. However, there are several advantages to hydrodynamic detection systems if they are fully developed with advanced material, signal processing, controls, and robotics. These systems would be preferred over sonar and vision when operating in dark or murky environments, in sonar blind-zones, and when passive sonar is used for stealth [26].

These four sensor types can result in different acquisition radii, which can affect the performance of a DSN. Their costs also cover a wide range of values, and the sensors have various patrol times.

2.4 Comparison of Stationary and Mobile Platforms

As previously mentioned, sensors can be either stationary or mobile. Each level of mobility has advantages and disadvantages. The category of stationary sensors encompasses both sensors that are placed on the ocean floor and sonobuoys. Sonobuoys are considered stationary sensors since typically they are not equipped with motion control devices; however, the sonobuoys are subject to ocean currents and can move due to theses currents. Gliders, AUVs, UUVs, and arrays located on submarines are considered mobile sensors and can follow a predetermined path or be manually controlled to pursue submerged contacts.

The main advantage of mobile sensors is that they can change speed and direction in response to detections made, thus actively pursuing the submerged contact. Many DSN use stationary sensors to first detect the submerged contact and then deploy mobile sensors to track the contact. Mobility also allows the sensors to position themselves for possibly better detection of the contacts. Conversely, stationary sensors cannot change their locations in order to improve detection. Also, stationary sensors are subject to ocean currents and their initial placement must take into account drifting, which is discussed in [27].

Although mobility may provide means for improved detection and tracking, it comes at a cost. Mobile sensors tend to be larger and must carry more equipment for motion control and energy for movements. For example, in [28] the authors explain the need for a cooperative control methodology known as virtual body and artificial potential (VBAP) that provides adaptable formations of gliders in response to a virtual leader position. The gliders are provided motion plans every time they return to the water's surface, but they do not all receive their instructions simultaneously. Therefore, a method is needed to constrain glider trajectories during the re-planning process so that gliders do not discard their current plans and generate new plans based on the latest information from the leader. Also, gliders must be within a specified distance of each other to effectively communicate [28]. The authors in [29] have developed a software suite known as the Glider Coordinated Control System (GCCS). As each glider resurfaces, the GCCS receives data on the gliders previous positions and depth-average flow estimates and then provides feedback in the form of planned trajectories for each glider. The additional technology for cooperative motion control is costly in terms of both time and money.

Mobile sensors may also have a shorter lifecycle than stationary sensors due to their higher energy consumption, even though many mobile sensors can be placed in sleep mode, during which the energy consumption rate is lower than in active mode, or completely turned off. Stationary sensors are less expensive and have lower energy consumption. Also, stationary sensors tend to be easier and safer to deploy. Sonobuoys are commonly dropped from an aircraft, which is safer than deploying a UUV via a ship while in hostile waters [30]. Whether mobile or stationary sensors are better for undersea surveillance is still a topic of interest in the field of ASW research. The application of the methodology presented in this thesis to compare mobile and stationary sensors is a topic of future research and is addressed in §6.2.

2.5 Technological Problems Associated with DSN

Technological problems that are associated with DSN can belong in three broad categories: (1) data processing, (2) communication, and (3) sensor management [2]. Underwater wireless networks tend to have more difficulty in all three categories due to their working environment. This thesis concentrates mainly on the last category, specifically focusing on sensor placement in order to properly acquire the submerged contacts. A brief discussion on data processing and communication follows since it must be understood that these aspects of DSN must work effectively for acquisition to occur. It is assumed that the acquisition radii used in our methodology take into account that the proper infrastructure does exist in the DSN to process data and communicate within the system.

Data processing includes the collection of sensor readings of the environment, signal processing (detection), execution of motion control plans, and the classification, localizing, and tracking of submerged contacts. A variety of algorithms are used for a wide range of purposes within the data processing category. A genetic algorithm is used in [10] to localize targets. A Two-Tier Data Dissemination routing protocol is used for controlling data packets within stationary networks, while a vector-based forwarding protocol performs the same actions for mobile sensors [17]. Sensors classify a target by comparing the signals received with known data via classifier algorithms, such as k-nearest neighbor and the support vector machine [8]. The authors in [31] use a steepest ascent algorithm with an intelligent step-size selection scheme to determine

the location of the sources and receivers in a multistatic system so that targets cannot pass through a bounded area, such as a seaport.

Communication problems exist when multiple sensors and platforms cooperate within a DSN. The design of the network communication is essential for optimal system performance. Decisions, such as the degree of information shared amongst sensors and the capacities of each sensor, are critical for the timely dissemination of information [2]. In terms of AUVs, traditionally the concept was for them to operate on large platforms either individually or in a small group. In [19], it is suggested that many smaller, low-cost AUVs should be incorporated into a single system to simplify the recovery of data and minimize the cost of lost information when a sensor fails. However, the overall control technology increases with the number of sensors within a system. The PLUSNet system, which is comprised of multiple clusters of sensors, uses intra and inter-cluster communication to improve detecting and tracking capabilities [4].

There is also a wide range of methods for data communication. For example, EM signals can perform better than acoustic signals due to higher throughputs in the same raw channel bit rate, which results in more data communicated per transmission. However, EM signals have a shorter range than acoustic signals and perform best when the farthest distance between any two sensors in the system is equal to the range of the EM signal [19]. Whichever communication medium is employed, it must be able to endure the demands of time, bandwidth, and energy created by the DSN in dynamic environments [2].

The overall management of the DSN is a wide topic in the research community [32]. The primary problem within the management category is the initial placement of sensors and the controlling of sensor positions over time while avoiding obstacles. Ad hoc networks provide a challenge for real-time updating of the network topology as sensors fail and new ones are deployed as well as the drifting of stationary sensors and movements of mobile sensors [2]. A genetic algorithm is used in [7] to assign multiple tasks to UAVs, and medium access control (MAC) protocols are used to coordinate the communication across sensors. In [33], a Greedy Randomized Adaptive Search (GRASP) is used to solve a complex version of the Vehicle

Routing Problem (VRP) referred to as the Course of Action problem. Resource Management is used along with the GRASP to determine the placement of resources as well as assigning information gathering actions to each resource.

The PLUSNet system mentioned in [4] and [5] must control the various types of sensors with respect to each sensor's purpose. The gateway buoys and gliders must relay signal data from submerged sensors to the control station on shore. The gliders also retrieve motion plans for themselves every time they surface. The next section discusses the sensor positioning problem in more depth. This thesis concentrates on the initial placement of sensors in a specific environment.

2.6 Approaches for Solving the Sensor Positioning Problem

Many approaches have been developed over the decades to determine the optimal location for stationary and mobile sensors within a DSN. The following sections discuss the most common approaches and their influence on the geometric approach presented in this thesis.

2.6.1 Track Coverage Problem

The author in [34] proposes a coverage cone to explain the range of straight tracks that can be covered by a sensor. The sensors are modeled as disks, and the coverage cone has a vertex located at a point on the perimeter of the rectangular area of interest. The sides of the coverage cone are tangent to the sensor's disk, as shown in figure 2.5. A k-coverage cone is formed by the intersection of k separate coverage cones that share the same vertex.

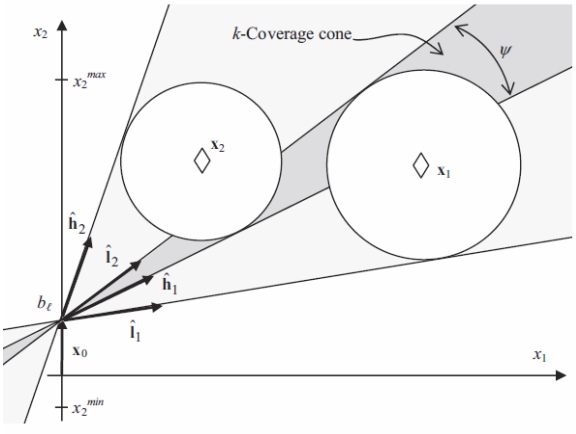

Figure 2.5: Coverage cones for track coverage [34]

The work [27] presents a sensor placement problem to maximize the track coverage of an area using static sensor that move with the ocean current. It is assumed that disks can represent the sensors, which are omnidirectional, and the targets move at a constant heading and speed. The methodology determines the optimal initial placement of sensors so that their trajectories over a defined time period maximize the track coverage of the DSN.

Some of the concepts that are presented in [34] contributed to the geometric approach discussed in chapter 4. In a sense, we are determining the tracks that are included in a coverage cone created by the sensor and an offset value. The tracks are defined by the heading and the offset values rather than the unit vectors as in [34]. Our approach determines which combinations of headings and offsets result in tracks that are included in the coverage cones associated with a particular sensor.

2.6.2 Set Covering Problem

Another approach to sensor placement is the set covering problem. There are several formulations of this particular problem. One is the Art Gallery Problem in which the objective is to determine the minimum number of sensors necessary to cover a specific area so that every point in the area is covered by at least one sensor. The work [35] takes a slightly different approach to the set covering problem by determining if a set of N sensors can provide k-coverage, meaning that every point in the area is covered by at least k sensors. The authors model the sensing range of each sensor as a disk with radius r, and these disks can be homogenous or heterogeneous. They state that a point in the area is covered by a sensor if it is within the sensor's sensing range. Therefore, the algorithm focused on analyzing the interaction of the perimeters of the disks. If the distance between the centers of any homogenous sensors was greater than twice the radius, neither sensor would contribute any coverage to the other sensor's perimeter. However, if the distance was less than twice the radius, one sensor would cover an arc of the other sensor's perimeter. This arc was calculated as falling between two angles that are formed between the line connecting the two sensors' centers and the point of intersection of the two disks' perimeters (see fig. 2.6).

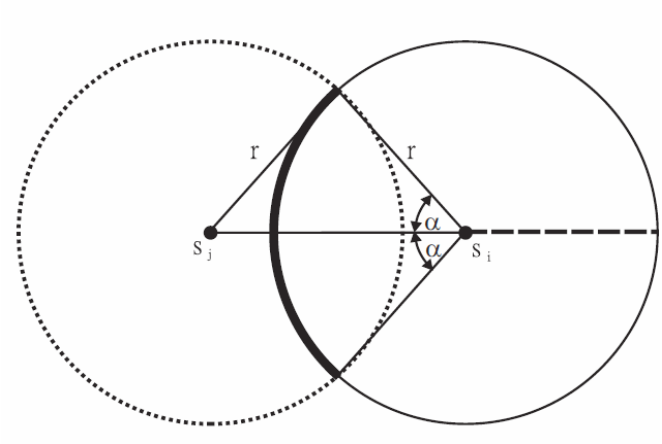

Figure 2.6: Arc of S_i's Perimeter Covered by S_j [35]

Voronoi diagrams are also used frequently in algorithms for the coverage problem. The work [36] uses the Voronoi diagram to create a discrete graph problem and enables search techniques in the graph representation. Both binary-search and breadth-first search algorithms are used in conjunction with the Voronoi diagrams to track a moving target along a path via stationary sensors. The authors in [37] implement the Voronoi diagrams in Lloyd algorithms to coordinate mobile sensors.

Some of the techniques presented in [35] are extended in the coverage function discussed in chapter 4 of this thesis. The sensors are modeled as disks and can be homogenous or heterogeneous, and it is assumed that a submerged contact that is located within the disk can be acquired. Similarly, we analyze the interactions of disks, but in a slightly different light. Instead of analyzing the interactions simply between two sensors, we analyze the interaction of a sensor and an imaginary disk that is formed with the offset value as its center. This imaginary disk has a radius denoted as D, which represents the maximum distance that a submerged contact could travel during the patrol time if it started at the offset at time t=0. The minimum and maximum angles (referred to as headings) are calculated as the angles formed between the line connecting the sensor's center and the offset and a point connecting the offset to a point on the sensor's perimeter that intersects with the imaginary disk. If a submerged contact originates at a particular offset and has a heading value that is between the minimum and maximum angles associated with that offset, it can be acquired by the sensor. However, rather than determining if the area is

k-covered, as in [34], we are concerned with the locating of a set of sensors to maximize the probability of acquisition and we need not cover all the points within an area.

2.6.3 Occupancy Grid Approach

The occupancy grid concept includes separating an area into small cells that can be either occupied or empty. Each cell stores a probability of being occupied or empty in a discrete lattice process. These probabilities are determined via sensor readings. When a sensor indicates that a signal has been observed, a Bayesian estimation procedure is then used to decide the probability of the state of the cell and a local sensor map is developed [38]. These maps are integrated for all the sensors in a network to develop the complete map of an area. The author in [38] suggests that the occupancy grid approach can be used to derive deterministic higher-level geometric representations of the DSN and area. However, the author also warns that this approach can lead to a high sensing-to-computation ratio and needs improved sensor models to be beneficial in the field.

In [39], the authors introduce the concept of a three dimensional occupancy grid framework to interpret stationary objects by an active sensing AUV for the purpose of obstacle avoidance and path planning via environmental mapping. An AUV's close environment can be shown as a grid of the 3D sonar range image, and the occupied cells are associated with objects that the AUV is trying to avoid while it traverses the area. The grid can be spherical or rectangular in shape, and each form has its own pros and cons. The spherical shape can reflect the sonar properties of beam orientation and range readings more accurately than a rectangular coordinate frame. However, the cells in a spherical coordinate frame are not all of the same size, but a rectangular geometry implies homogenous cells. The sizes of the cells are also of concern. Larger cells may not accurately reflect the sonar range readings since an object could occupy only one cell, which may be viewed as a false alarm instead of a true hit. Smaller cells allow an object to occupy multiple cells at one time indicating that there is indeed an object at the cells coordinates. The works [40] and [41] also use occupancy grid maps to plan the paths of mobile robots.

The concept that cells can be either 'occupied' or 'empty is applied to the geometric approach of this thesis in the form of the coverage function (CF). The area of interest is modeled as a 2D

coordinate system with sensors located at (x,y) coordinates. The sensors can cover a range of offsets and headings. The CF is a matrix of boolean values that correspond to the combinations of the offsets and headings that can be covered by the sensor. A 'cell' of the CF matrix is 'occupied' if the heading and offset associated with that cell would result in placing the submerged contact inside the sensor's disk. The increment values for the change in offsets and heading determine the number and size of the 'cells' in the CF matrix and affects the performance of the MATLAB program presented in chapter 6.

2.6.4 Geometric Optimization Approach

The authors in [42] use cell decomposition to divide a space into void and observation cells so that sensors can track moving targets while avoiding obstacles. The objective is to determine the set of policies for all sensors which maximize the total sensing reward and minimizes the total time required to capture the targets. The sensing reward refers to the probability of detection. The methodology incorporates a game that terminates when all the targets have been captured. At the start of the game, all the sensors are placed simultaneously into the square area in detection mode. A new round in the game occurs when either a new partially observed or fully observed track is achieved. To determine the policies of each sensor at the start of each new round, the authors take into account the positions and field-of-views of all the sensors in the cooperative network in order to compute the sensor's path based on cooperative sensing or pursuit objectives. The policies are determined by modifying the cell decomposition approach used in classical motion planning. The sensors' configuration space that is free of obstacles is denoted as C_{free} and is decomposed into two types of cells. "A cell is defined as a closed and bounded subset of C_{free} within which a robotic sensor path can be easily generated" [42].

There are two types of cells, each of which is a convex polygon. A void cell k has the property that for every configuration q_i that is included in that cell k the sensor i has a zero probability of detecting a partially observed target. An observation cell $\underline{k}$ has the property that for every configuration q_i that is included in that cell $\underline{k}$ the sensor i probability of detecting a partially observed target is nonzero. Two cells are adjacent if they share a common boundary. The sensor can move between the adjacent cells without colliding with any obstacles. These cells are represented in a connectivity graph G as nodes. Two nodes in G are connected by an arc if and

only if the corresponding cells are adjacent. The sensing reward is expressed as a reward function that represents the improvement of the probability of detection by moving a sensor from configuration q_i in one cell to a configuration q_i in an adjacent cell.

When a sensor is deployed in detection mode, its trajectory is computed from a sequence of cells referred to as a channel so that the total sensing reward is maximized. The trajectory starts at the cell in which the sensor's current configuration is located. The last cell in the trajectory is the observation cell, or node, in G that has the highest cumulative probability. Every arc in G has the reward function attached to it in order to compute the optimal channel efficiently. An A* graph searching algorithm is then used to compute the optimal channel from G.

When a sensor is deployed in pursuit mode, its trajectory is determined by finding the obstacle-free shortest path from the sensor's current position to an interception point, which refers to the point in S where the target's track and the sensors new position would intersect. At this point, the target is termed "captured" and is removed from the set of targets. The game ends when all the targets are captured.

This geometric optimization approach influenced the objective of our work. We combine the concepts of detecting and capturing a target, which were part of the goals of [41], into one objective, which is referred to as the probability of acquisition. However, we are not concerned with obstacle avoidance, and this approach assumes that the submerged contact's starting location on the perimeter of the area is known. Our approach will instead use probabilities to describe where a contact may enter the area via one side of the area. Table 2.1 summarizes the parts various approaches discussed in this section that influenced the geometric approach presented in chapters 4 and 5.

Table 2.1: Summary of the Influences of the Approaches on the Geometric Approach

Approach	**Part of the Geometric Approach that was Influenced**
Track Coverage Problem	Coverage Cones and Definitions of Submerged Contact Tracks
Set Covering Problem	Coverage Function and Boundaries for Headings
Occupancy Grid Approach	Structure of the Area of Interest and Coverage Function
Geometric Optimization Approach	Acquisition Footprint, PMF for Headings and Offsets

Chapter 3: DSN Performance Problem with Pareto analysis

This chapter provides a formal definition of the problem as well as a list of all assumptions for the area of interest, submerged contacts, and sensors.

3.1 Problem Definition and Discussion

A DSN is comprised of a variety of sensors that may be stationary or mobile and can be equipped with passive or active sonar, electromagnetic sensing, or hydrodynamic sensing capabilities. At a high level, each of these sensors can be modeled by two-dimensional footprints that describe their range of acquisition, which includes the ability to DCLT the submerged contact within that range. It is assumed that these sensors are omnidirectional and isotropic such that their footprints can be modeled as disks.

The following two research problems are addressed in this thesis:

1. Develop a high-level approach to determine the sensor locations within a DSN in order to achieve the best performance that is possible for the given environment (*DNS Performance Problem*).
2. Perform a Pareto analysis to determine the optimal DSN for the given environment (*Pareto analysis*).

Problem Definition 3.1 (DSN Performance Problem). *Consider a two-dimensional regular area of interest of either shallow water or deep sea. For this area, define a thread-axis and a border at which submerged contacts enter the area. Define pdfs for the heading and offset of the submerged contacts. Provided a set of N omnidirectional heterogeneous sensors each with a specified footprint, the problem is to determine the location of these sensors so that the total probability of acquiring the submerged contacts is maximized during a defined patrol time.*

We use a geometric approach implemented in a program developed in MATLAB to perform the track coverage optimization in the DSN Performance Problem. The length and width of the area are in terms of nautical miles (nm) and are defined by the user. The thread axis separates the area's width into two equal halves. It is assumed that submerged contacts can only enter the area

through the border. Figure 3.1 illustrates the area of interest along with examples of submerged contact tracks and sensors. The submerged contacts travel across the border according to a pmf of the heading and offset. The patrol time T is defined as the amount of time from which the sensor is deployed to the end of its life cycle, and its value is determined by the user.

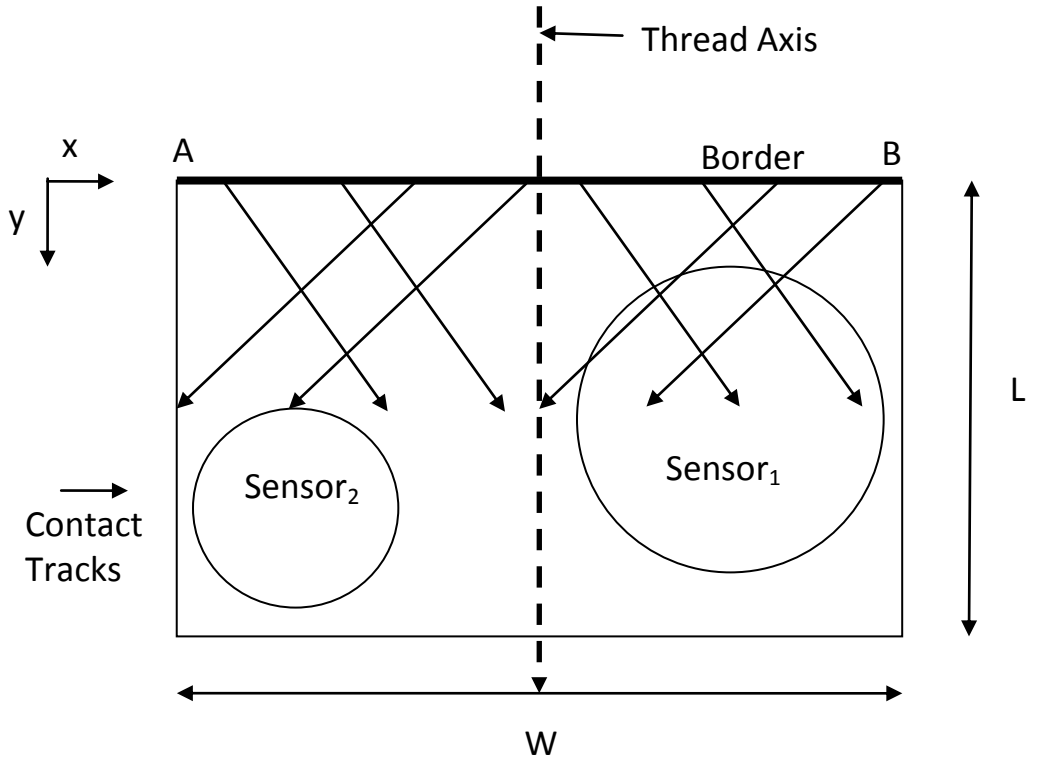

Figure 3.1: Example of an Area of Interest

Given the area of interest and the pmf describing the submerged contacts' tracks, the MATLAB program determines the placement of sensors for each DSN. We use the results of the program to perform a Pareto analysis to determine the optimal DSN for the given area of interest and the submerged contacts' pmf.

Problem Definition 3.2 (Pareto analysis) *Suppose that for each DSN tested its probability of acquisition and the lifecycle cost are recorded. Perform a Pareto analysis using these two metrics to determine the optimal DSN out of M possible systems for the given area of interest and pmfs.*

The geometric approach is fully developed in this thesis for a single sensor and a design of experiments is performed to analyze several variables that may affect performance. The concept is then extended to two sensors, and a Pareto analysis concerning two sensor DSN is performed as an example.

3.2 Assumptions

The assumptions for the area of interest, the submerged contacts, and the sensors are listed in this section. Justification is provided for each assumption.

3.2.1 Assumptions on the Area of Interest

The assumptions for the area of interest are concerned with the way in which the area is defined.

1. The area is two-dimensional. Depth is included in the medium parameter vector and its effects on sensor footprints are included in the calculations of the radii of the footprints.
2. The depth of the sea is constant throughout the defined area.
3. There exists a 'border' through which submerged contacts enter the area. Contacts can only enter the area at this border since either:
    a. The area is between two land masses that prevent submerged contacts from entering the area from the side.
    b. The area covers a large enough amount of water so that submerged contacts can only possibly enter at the border.
4. The transmission loss, ambient noise level, and radiated noise levels of the sea are constant in the defined area.
5. Ocean currents within the area of interest are not strong enough to have an effect on sensors and submerged contacts.
6. The area can be represented by a Cartesian frame and the sensors can only be placed at x, y coordinates.

3.2.2 Assumptions on the Submerged Contacts

The assumptions concerning the submerged contacts focus on their tracks and their influence on a sensor's ability to detect the contact.

1. The radiated noise and target strength are kept constant for all targets that are entering the area. The sensors must acquire the submerged contacts that are the most difficult to DCLT. Therefore, the values of radiated noise and target strength that are presented in the parameter vector are associated with the most difficult submerged contact. It is assumed that any submerged contact whose radiated noise and/or target strength is greater than that presented in the parameter vector can be more easily acquired by the sensors, so these submerged contacts are not considered in the program.
2. The offset is defined as the distance between the thread axis and the location at which the submerged contact enters the area.
3. A submerged contact's heading distribution is independent of its offset.
4. Submerged contacts have straight-line trajectories with the velocity and heading kept constant.
5. Contacts will move only forward in space, never backwards.

3.2.3 Assumptions on the Sensors

The assumptions for the sensors focus on the radius of the footprint, which is termed the acquisition radius.

1. The sensors are omnidirectional and isotropic, and thus can be represented by disks.
2. The acquisition footprint of a sensor is equal to or smaller than that of the footprint associated with only detecting the submerged contact. This is due to the fact that even though a sensor may be able to detect a contact, it may not be close enough to classify, localize or track the contact.
3. A sensor is able to DCLT a submerged contact if that contact is within the acquisition footprint.
4. Stationary sensors remain fixed in one position.
5. There is enough energy in the sensors to operate for the entire length of the patrol time. Therefore, there is no refueling of the sensors.

Chapter 4: Methodology for a Single Sensor

This chapter provides a detailed explanation of the methodology used to develop the MATLAB program. First, the area of interest is defined in $\mathbb{R}^2$. Second, the parameter vectors are defined for the medium, submerged contact, and sensors. These parameters determine the acquisition radii for all of the sensors by use of the sonar equation and graphs found in [1]. Third, the two key metrics used to evaluate the DSN are defined. The chapter continues with the definitions of the submerged contact tracks and the sensor acquisition radius, the development of a sensor's coverage function, and concludes with an explanation of the geometric approach for both the discrete and continuous cases. The geometric approach is fully defined for a single sensor and the case of two sensors is investigated.

4.1 Definition of Area of Interest

The area of interest, denoted as S, is a two-dimensional (L x W) area of either deep sea or shallow waters. Submerged contacts enter at a border ($\overline{AB}$), which is one side of the rectangular area and has length W. Figure 4.1 shows the area of interest with the thread axis and the border .The length (L) and width (W) of the area are in terms of nautical miles (nm) and are defined by the user. The Cartesian Frame has its origin at point A, with the positive x direction pointed to the right and the positive y direction pointed down. Point B is located at the coordinates (W,0).

The number of x and y values depends on the increments set, which are denoted Δx and Δy respectively. There are a total of U possible values of x, where U=W/Δx and Δx is the increment value of the x values. There are also a total of V possible values of y, where V = L/Δy and Δy is the increment value of the y values. A sensor can be positioned at any point in the area, and these points are defined by the coordinates (x_u, y_v).

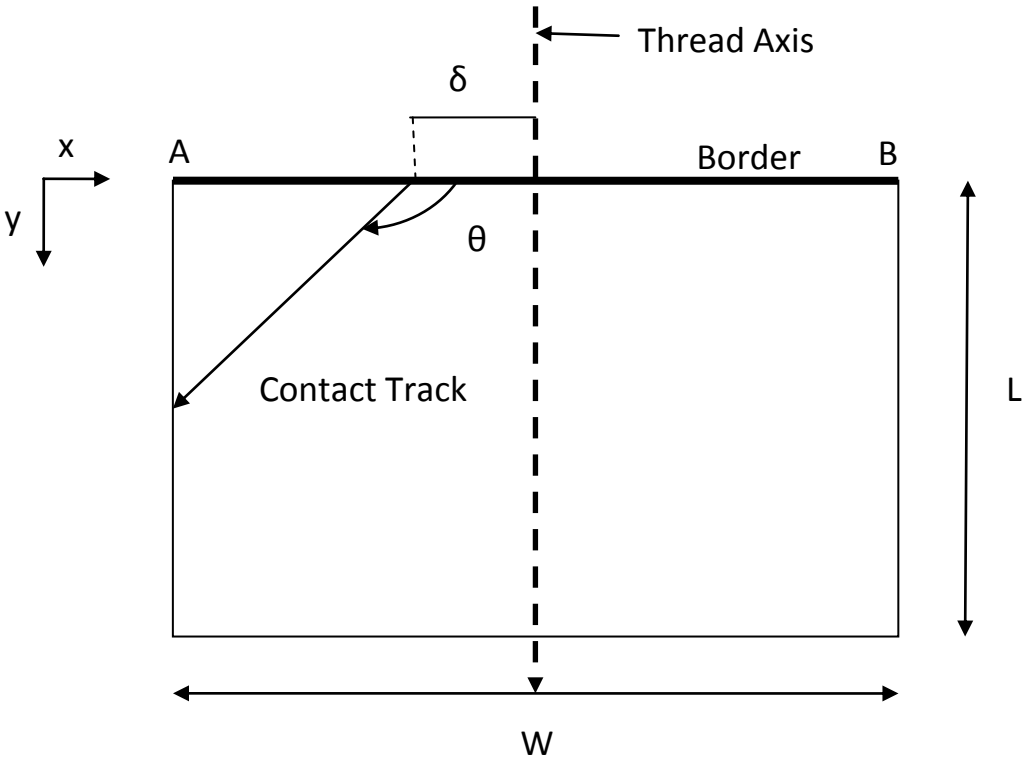

Figure 4.1: Area of Interest

4.2 Parameter Vectors

The following three parameter vectors are input or chosen by the user. These parameters are used to accurately calculate the acquisition radii for the sensors. The parameters are presented here with the calculations for the acquisition radius presented in §4.5.

4.2.1 Medium Parameter Vectors

The medium parameter vector contains four parameters: transmission loss (TL), ambient noise level (ANL), reverberation noise level (RNL), and a "deep vs. shallow" indicator (MD). These parameters represent the effect that the medium has on sound propagation and sonar detection. There is one medium parameter vector per experiment.

$$\begin{bmatrix} Transmission\ Loss \\ Ambient\ Noise\ Level \\ Reverberation\ Noise\ Level \\ Deep\ vs\ Shallow \end{bmatrix} = \begin{bmatrix} TL \\ ANL \\ RNL \\ MD \end{bmatrix}$$

The TL is the weakening of sound between the source and some point at a distance in the sea. The ANL of the medium refers to the background noise of the ocean, while the RNL is the result of the reverberation of an active sensor's signal bouncing off of the water surface and the sea floor. Detection thresholds must take into account these noise levels. The last parameter for the medium is a binary variable to indicate whether the area of interest is located in the deep sea or in shallow waters. A deep sea area will infer spherical spreading, while shallow waters will constitute cylindrical spreading.

Other parameters that were considered for the medium parameter vector include the following: the sound velocity profile of the area, ocean currents, and interferers. Each parameter was not selected for a variety of reasons. The sound velocity profile is reflected in the TL parameter. The ocean currents and interferers are not modeled in order to control the size of the model so that an optimal solution can be achieved. Also, interferers were not included in order to keep the research unclassified.

4.2.2 Submerged Contact Parameter Vector

The submerged contact parameter vector contains five parameters: velocity (v), offset (δ_k), heading (θ_j), radiated noise (RN), and target strength (TS). The first three parameters define the submerged contact's movements, while the remaining two parameters are used in determining the sensor acquisition radius. There is one submerged contact parameter vector per experiment since the RN and the TS is associated with the most difficult contact to acquire. It is assumed that any contact whose RN and/or TS result in a better acquisition radius for a sensor will be acquired by the sensor.

$$\begin{bmatrix} Velocity \\ Offset \\ Heading \\ Radiated\ Noise \\ Target\ Strength \end{bmatrix} = \begin{bmatrix} v \\ \delta_k \\ \theta_j \\ RN \\ TS \end{bmatrix}$$

Throughout the program, the velocity of a submerged contact will remain constant. A submerged contact will enter the area of interest at a particular δ_k and θ_j. The offset refers to the distance

between the thread-axis and the location at which the submerged contact enters the area. The heading refers to the angle the submerged contact creates with the border. These parameters are defined by separate pdfs. Figure 4.1 includes a target track with the heading and offset, labeled.

The parameters of RN and TS will be used in conjunction with the sonar equation to determine the acquisition radius for a sensor. RN refers to the noise that the submerged contact makes by itself, and it is dependent on the velocity of the submerged contact. The RN will be determined from a distribution and is fixed for each experiment. A faster submerged contact will generate louder noise. TS refers to the echo returned by the submerged contact and is aspect dependent for active sonar. It will also remain fixed for each experiment.

4.2.3 Sensor Parameter Vector

There are a total of N unique sensors. The sensor parameter vector for a sensor i contains five parameters: detection threshold (DT_i), array gain (AG_i), source level (SL_i), flow noise (FN_i), self noise (SN_i), maximum velocity (v_{maxi}), base cost (BC_i), and sensing cost (SC_i). The first five parameters are associated with determining the acquisition radius of the sensor. The remaining three parameters define the mobility capability and costs associated with a senor.

$$\begin{bmatrix} Detection\ Threshold \\ Array\ Gain \\ Source\ Level \\ Flow\ Noise \\ Self\ Noise \\ Maximum\ Velocity \\ Base\ Cost \\ Sensing\ Cost \end{bmatrix} = \begin{bmatrix} DT_i \\ AG_i \\ SL_i \\ FN_i \\ SN_i \\ v_{maxi} \\ BC_i \\ SC_i \end{bmatrix}$$

The DT_i is determined for each sensor based on the inputs for the medium and the submerged contact as well as several other sensor parameters. The DT_i depends on the signal to noise ratios (SNR) and will be taken from the Receiver Operating Curves (ROC). The ROC takes into account the probability of detection and the probability of false alarms. The AG_i refers to the ability of a sensor to detect a submerged contact by improving its SNR. A sensor's SL_i must also be taken into account when modeling active sonar. The FN_i and SN_i of a sensor need to be

included since the noise affects the SNR and DT_i. FN_i is created by the hydrodynamic flow around the sensor, while the SN_i is created by the sensor itself.

The v_{maxi} refers to the maximum velocity that the sensor can reach and shows the mobility capability of the sensor; a stationary sensor will have a $v_{maxi} = 0$. In order to ensure that the optimal solutions are tractable, cost must be included in the model. The BC_i is according to the type of sensor (stationary vs. mobile), the sensing capabilities (e.g. passive sonar vs. hydrodynamic), and the wfive of the sensor. The SC_i refers to the cost of deployment, energy consumption, maintenance, and retrieval.

4.3 Evaluation Metrics

There are two metrics for which each DSN is evaluated. The probability of acquisition metric is maximized in the optimization model. The lifecycle cost is recorded for each DSN and consists of base and sensing costs. A discussion on the lifecycle cost is included in §5.4. These two metrics will be plotted on a graph to perform a Pareto analysis and determine the optimal DSN.

4.3.1 Probability of Acquisition

This evaluation metric describes the ability of the DSN to DCLT a submerged contact. A minimum level is set, and any DSN that does not meet this value is discarded from the Pareto analysis. The total probability of acquisition (Q_i) for a DSN is the intersection of two events.

Event A: A submerged contact will enter the area S at δ_k, θ_j with velocity ν

Event B: A sensor i will acquire a submerged contact before patrol time T

Therefore, P(A) corresponds to the 'probability of occurrence' of a submerged contact, and P(B) is the coverage function of a sensor. The probability of occurrence for a combination of heading θ_j and offset δ_k is denoted as o_{jk}, and occurrence matrix $\boldsymbol{O}$ is the collection of these combinations, that is $\boldsymbol{O} = [o_{jk}]$. The development of $\boldsymbol{O}$ is explained in detail in §4.4.1.

As a basic example, suppose that there are only two heading and offset values and all combinations are equally likely to occur. That is,

$$\boldsymbol{O} = \begin{matrix} \\ \theta_1 \\ \theta_2 \end{matrix} \begin{matrix} \begin{matrix} \delta_1 & \delta_2 \end{matrix} \\ \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix} \end{matrix} \tag{4.1}$$

The coverage function (CF) for a sensor located at position (x_u, y_v) is denoted as $\boldsymbol{CF_i(x_u, y_v)}$ and has elements $cf_{ijk}(x_u, y_v)$, that is $\boldsymbol{CF_i(x_u, y_v)} = [\, cf_{ijk}(x_u, y_v)]$. The elements of the CF can only have a value of zero or one, with one indicating that a submerged contact whose track is defined by θ_j and δ_k can be acquired by the sensor positioned at (x_u, y_v). Suppose that the sensor is positioned at an (x_u, y_v) that results in the following matrix:

$$\boldsymbol{CF_i(x_u, y_v)} = \begin{matrix} \\ \theta_1 \\ \theta_2 \end{matrix} \begin{matrix} \begin{matrix} \delta_1 & \delta_2 \end{matrix} \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix} \tag{4.2}$$

Since the CF of a sensor is independent of the probability of occurrence, the two probabilities can simply be multiplied, as shown in equation 4.3. This results in the elements of the probability of acquisition matrix $\boldsymbol{Q_i(x_u, y_v)} = [q_i(x_u, y_v)]$. The matrix $\boldsymbol{Q_i(x_u, y_v)}$ is the Hadamard product of $\boldsymbol{O}$ and $\boldsymbol{CF_i(x_u, y_v)}$**.**

$$q_i(x_u, y_v) = P(A \cap \mathrm{B}) = \mathrm{P(A)} * \mathrm{P(B)} = o_{jk} * cf_{ijk}(x_u, y_v) \;\; \forall u \in U, \forall v \in V \tag{4.3}$$

From the example, matrix $\boldsymbol{Q_i(x_u, y_v)}$ is calculated as follows:

$$\boldsymbol{Q_i(x_u, y_v)} = \boldsymbol{O \; o \; CF_i(x_u, y_v)} = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix} \boldsymbol{o} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0 \end{bmatrix} \tag{4.4}$$

Then the $Q_i(x_u, y_v)$ for a single sensor is the Frobenius inner product of the $\boldsymbol{O}$ and $\boldsymbol{CF_i(x_u, y_v)}$ matrices, which is equivalent to the sum of the elements of $\boldsymbol{Q_i(x_u, y_v)}$.

$$Q_i(x_u, y_v) = \sum_j^J \sum_k^K \big(q_i(x_u, y_v)\big) = \sum_j^J \sum_k^K \Big(o_{jk} * cf_{ijk}(x_u, y_v)\Big) \tag{4.5}$$

Therefore, the $Q_i(x_u, y_v)$ for our example is 0.75. For an entire DSN, the Q_i is the sum of the individual sensor's $Q_i(x_u, y_v)$ and must be less than one.

4.3.2 Lifecycle cost

The lifecycle cost (LC) of a DSN is comprised of both the base and sensing costs of the sensors. The lifecycle cost is therefore a sum of the base cost and the sensing cost of all the sensors in a DSN:

$$LC = \sum_{i=1}^{n} [BC_i + SC_i] \tag{4.6}$$

4.4 Submerged Contact Tracks

The tracks for submerged contacts may be considered straight, which entails constant velocity and heading. A submerged contact's ν is used to determine if the submerged contact will be acquired by a sensor before *T*. Tracks are dependent on the initial heading and offset of the submerged contact as it enters the area of interest at the border. The heading and offset of the submerged contact are each described by pdf, and their joint probability is described by a pmf referred to as the probability of occurrence matrix. The pdfs that describe the heading and offset can be continuous or discrete. The discrete case will be completely developed in this thesis, while the theory of the continuous case will be explored and a basic example will be provided.

4.4.1 Discrete Case

In this case, it is assumed that there exists a discrete number of heading and offset values and that the probability of each value occurring is provided. The heading values can range from 0 to π, that is ($0 \leq \theta_j \leq \pi$); the offset values can range from 0 to W, that is ($0 \leq \delta_k \leq W$). The total number of heading values is J= $\pi/\Delta\theta$+1, where $\Delta\theta$ is the increment between heading values. It is suggested to use an increment value of π/100 to separate the heading values into 100 equal segments. The total number of offset values is K = W/$\Delta\delta$+1, where $\Delta\delta$ is the increment between offset values. It is suggested that this value be adjusted depending on the dimensions of the area of interest. An increment should be chosen that results in creating at least 100 segments for both the width and length. The vector **θ** = $[\theta_j]$ is the set of all the possible values of the heading, while

the vector $\boldsymbol{\delta} = [\delta_k]$ is the set of all the possible values of the offset. The pdf values are calculated at each possible heading value in **θ** and **δ.**

To investigate a wide range of distributions in the discrete sense, the triangular distribution was used as an estimate of continuous functions (see fig. 4.2). The MATLAB program prompts the user to input the offset or heading index for the C value in the triangular distribution. By setting the area under the triangular distribution curve to one, we insure that the sum of the probabilities for a single pdf is always equal to one. The overestimation of the probabilities (OE_k) and the underestimation of the probabilities (UE_k) sum to zero since the values of P_1 and P_K are approximately equal (see eq. 4.7).

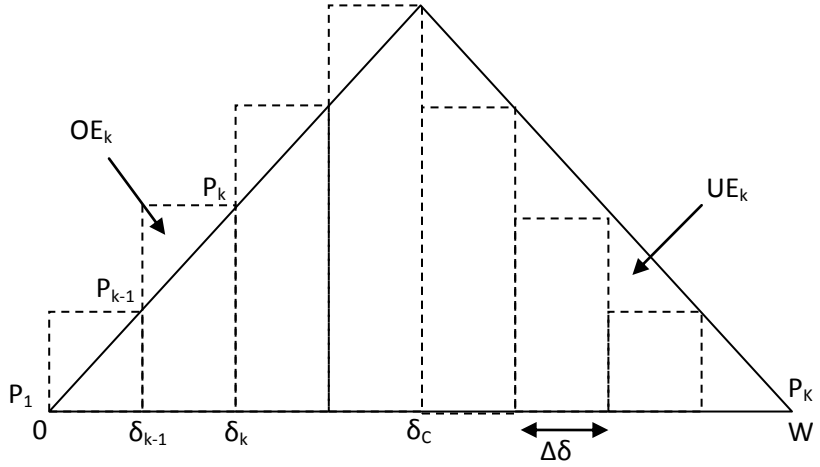

Figure 4.2: Example of the Triangular Distribution for the Offset

$$\sum_{k=1}^{C} OE_k + \sum_{k=C+1}^{K} UE_k = \sum_{k=1}^{C} {}^1\!/_2\,(P_k - P_{k-1})\Delta\delta + \sum_{k=C+1}^{K} {}^1\!/_2\,(P_k - P_{k-1})\Delta\delta$$

$$= {}^1\!/_2\,(P_K - P_1)\Delta\delta = 0 \qquad (4.7)$$

The pmf of all the heading and offset combinations is defined by the occurrence matrix $\boldsymbol{O}$**.**

It is assumed that the discrete pdfs of the heading and offset are independent, so the joint probability of the heading and the offset is the multiplication of the two discrete pdf values for a specific combination:

$$o_{jk} = P(\theta_j \cap \delta_k) = \mathrm{P}(\theta_j) * \mathrm{P}(\delta_k) \qquad \forall j \in J, k \in K \tag{4.8}$$

A $\boldsymbol{O}$ matrix is then calculated as the tensor product (also known as the outer product) of the heading and offset vectors:

$$\boldsymbol{O} = \boldsymbol{\theta} \otimes \boldsymbol{\delta} = [\theta_{ij}] \otimes [\delta_k] = [\theta_j]^{\boldsymbol{T}} * [\delta_k] = \begin{bmatrix} o_{11} & \cdots & o_{1K} \\ \vdots & \ddots & \vdots \\ o_{J1} & \cdots & o_{JK} \end{bmatrix} \tag{4.9}$$

This matrix can only be determined when the values of the heading and offset are discretized and the probability of each value is given. Otherwise, a more complex form of the probability of occurrence is required, which results in more complicated calculations for the probability of acquisition.

4.4.2 Continuous Case

When a continuous pdf is used to describe the probability of the heading and/or offset values occurring, the probability of occurrence matrix becomes more complex. Instead of a single value for the heading or offset, there are ranges. The probabilities are then reported as the probabilities that a particular range of heading values and range of offset values will occur. As a basic example, the heading and offset distributions will be uniform over the possible ranges of the heading and offset. That is, $\theta \sim Uniform\,(0, \pi)$ and $\delta \sim Uniform\,(0, W)$. This results in the same mass value for all combinations of headings and offsets. This example will continue in §4.8 with a discussion on the geometric approach for the continuous case.

4.5 Sensor's Acquisition Radius

The three parameter vectors defined in §4.2 are used to determine the acquisition radius for each sensor in the model. This section describes the models and graphs used in the calculations for stationary sensors. All of the equations presented in this section are taken from [1]. The resulting R_i values are then entered into the MATLAB program for analysis.

For stationary sensors, the footprint F_i is modeled as a disk centered at point C located at (x_u, y_v) with radius R_i. This radius is determined via the sonar equation and several of the parameters from the medium, submerged contact, and sensor parameter vectors. The disk corresponds to a CF that is projected onto the border. This CF is dependent on the acquisition radius and the distance between the border and point C.

4.5.1 Active Sonar Acquisition Radius

For active sonar, there are two different sonar equations categorized by the type of background noise. Ambient noise is always present in the environment, whether the sensor is operating in littoral waters or in the deep sea. Reverberation noise is also prevalent in both environments, but being more prevalent in littoral environments. Therefore the ambient background sonar equation (eq. 4.10) is used for deep sea environments and the reverberation background sonar equation (eq. 4.11), which includes an additional term for the reverberation, is used for the analysis of DSN in littoral environments. Equation 4.10 is used when $\mathrm{MD} = 1$ (indicator for deep sea), and equation 4.11 is used when $\mathrm{MD} = 0$ (indicator for littoral waters).

$$SL - 2TL + TS = NL - AG + DT \tag{4.10}$$

$$SL - 2TL + TS = RL + DT \tag{4.11}$$

The NL in 4.10 is the sum of the ANL, the SN of the sensor, and the FN around the sensor. The reverberation level (RL) in 4.12 replaces the NL – AG term in 4.10, and is equivalent to the NL plus the RNL term. The TL term is solved for in either equation along with the temperature of the water and the frequency of the sonar are used to determine the range r in kilo yards from the nomogram found in [1] (see fig. 4.3).

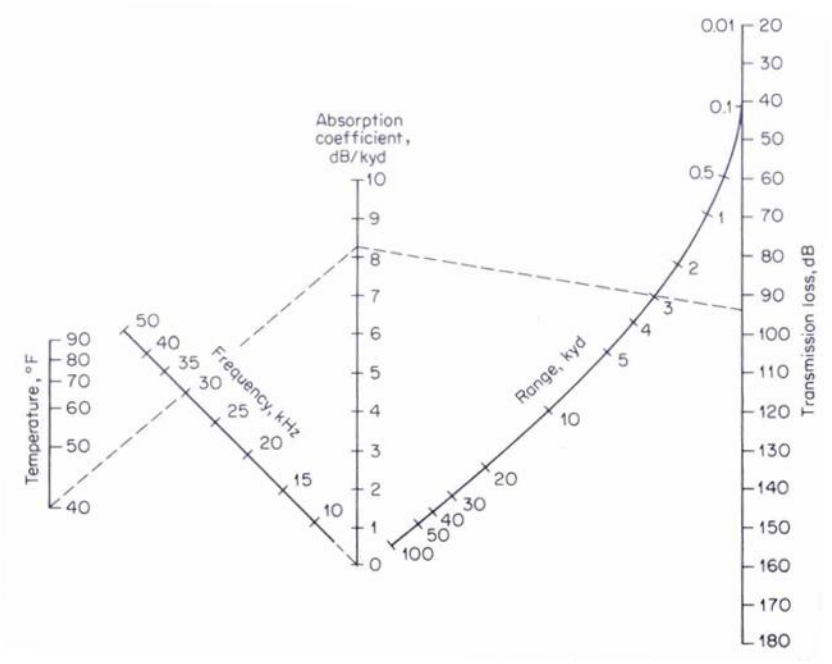

Figure 4.3: Nomogram for Computing Transmission Loss [1]

Once the detection range r is determined, an acquisition coefficient, γ, is applied to r. This coefficient changes the range r from that of a detection range to an acquisition range, and its value is between zero and one. For active sonar, we assign a value of 0.8 to this coefficient.

As an example, consider active sonar that is installed on a sonobuoy deployed that has a power output of 1,000 Watts at a frequency of 10 kHz. The device is deployed in the deep sea, which has a sea state of 2 and a temperature of 75°F. Detection probability must be 75% with the probability of a false alarm at 0.1%. Equation 4.10 is used and the following values for all terms, except TL, were obtained by using the information provided in the example along with graphs found in [1]. From the graphs, SL = 230 dB, TS = 25dB, NL = 70 dB, AG = 30dB, and the DT=20dB. The resulting TL equals 97.5dB. From figure 4.3, a temperature of 75°F and a frequency of 10kHz results in an absorption coefficient of 0.5dB/kyd. A TL of 97.5dB along with the absorption coefficient indicate that this TL occurs at 30kyd, which is equivalent to approximately 15nm. This means that a submerged contact could be detected 75% of the time when it passes within 15nm of the sensor. The acquisition radius is then equal to 80% of the detection radius; that is R_i = 0.80*15nm = 12nm,

4.5.2 Passive Sonar Acquisition Radius

For passive sonar, the approach for finding the acquisition radius is the same as that for active sonar, but the passive sonar equation is used in lieu of equations 4.10 and 4.11. The passive sonar equation is:

$$RN - TL = NL - AG + DT \tag{4.12}$$

There are two main differences between the active and passive sonar equations. First, there is no TS term in 4.12 since a passive sonar sensor does not propagate a signal that is returned by the submerged contact. The second difference is the meaning of the SL and RN terms. In the active sonar equation, SL refers to the source level of the active sonar itself. For passive sonar, RN replaces this term and refers to the radiated noise of the submerged contact. The TL term is solved for and the nomogram is referenced to determine the range. The acquisition coefficient is assigned a value of 0.9.

Consider the same deep sea environment and detection requirements from the active sonar example. However, the device is now equipped with passive sonar. The SL and TS terms are now replaced by RN = 160dB for a submarine target traveling at 15 knots. All other terms maintain the same values, which results in a TL value of 100dB. In junction with the temperature of 75°F and absorption coefficient of 0.5dB/kyd, the TL occurs at 35 kyds, which is equivalent to approximately 17nm. The acquisition radius is then 0.90*17nm = 15nm.

4.5.3 Electromagnetic and Hydrodynamic Sensor Acquisition Radius

There is a limited amount of literature on how to calculate the range of detection for electromagnetic and hydrodynamic sensor, and much of it is classified. Therefore, no explanation of the calculations is provided here. It is assumed that the user knows the detection radius of the electromagnetic and hydrodynamic sensors, and the acquisition radii are calculated with $\gamma = 0.9$ for both sensor types.

4.6 Sensor's Coverage Function (Discrete Case)

The coverage function of sensor i (CF_i) is the probability that that sensor will acquire a contact traveling at a specific velocity with a certain heading and offset combination. When the tracks have constant velocity and heading, the coverage function is based on simple geometry.

A sensor i is located at the point $C_i(x_u, y_v)$ and has an acquisition radius R_i. For stationary sensors, the coverage function depends on the acquisition radius, R_i, as well as the distance between the border and point C_i.To define the coverage function for a stationary sensor, ranges are needed for both the heading and offset values. The lower and upper bounds for the range of offset values are points A and B located on the border. Then, for each offset value, a range of heading values that will position the submerged contact inside the sensor's footprint is determined base on the offset's relation to the x coordinate of the sensor. The resulting coverage function has a value of 1 for the heading and offset values that result in an acquisition when the sensor is positioned at C_i and a value of 0 for all other combinations. It results in a two-dimensional area that exists in the same space as that of the $\boldsymbol{O}$ matrix.

4.6.1 Range of δ_k Values that Can Result in a Coverage Function Value of One

Any submerged contacts whose offset, δ_k, is between A and B are eligible to be acquired by a sensor i. However, not all offset values will be able to be acquired by the sensor since there is a predefined control time during which sensors can attempt to acquire contacts. A submerged contact can travel at v during T. Therefore, the maximum distance that it can travel, denoted as D, is defined by its velocity and the patrol time.

$$D = v * T \tag{4.13}$$

The line segment $\overline{\delta_k C}$ in figure 4.2 has a length of Z, where

$$Z = \sqrt{y_v^2 + (x_u - \delta_k)^2} \tag{4.14}$$

Figure 4.4 shows that the farthest offset from the sensor that can result in a submerged contact being acquired is located at the distance

$$Z = R_i + D \tag{4.15}$$

Therefore any $Z < R_i + D$ will have several heading values that will result in a contact being acquired. If $Z = R_i + D$, then there is only one heading value, and any value of $Z > R_i + D$ cannot possibly be acquired by the sensor before the patrol time expires. If $Z < R_i$, then the sensor is touching the border, which will result in a coverage function value of one for all the heading values for the $\delta_k$associated with the *Z* value.

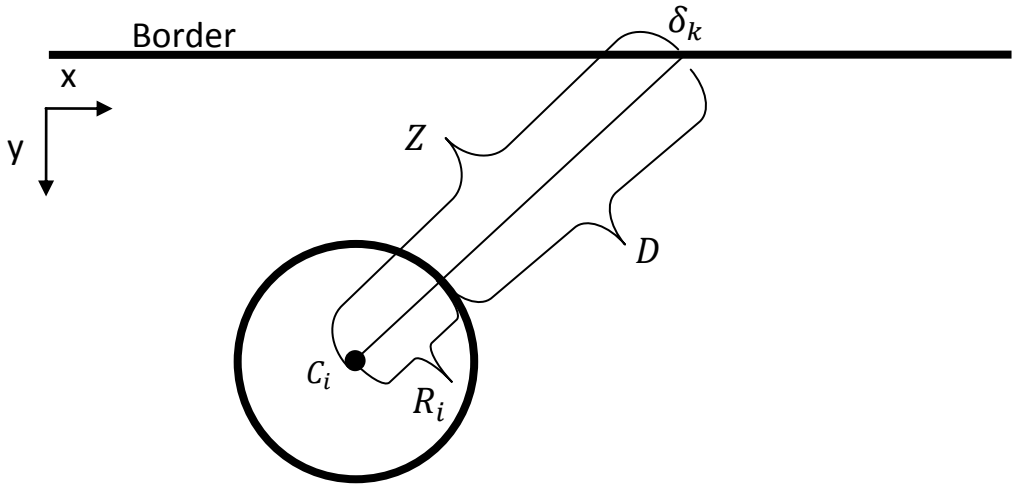

Figure 4.4: Maximum δ_k value at which a sensor at C_i can have a CF value of one

4.6.2 Calculations of the Ranges of θ_j When $Z \geq D$

For values of δ_k that result in $Z \leq R_i + D$, the lower and upper bounds, which are denoted as α_{iuv} and β_{iuv} respectively, can be calculated. Figure 4.5 shows an example of a coverage cone (shaded in gray) created by the sensor i and the offset value δ_k. The lower and upper bounds for the range of heading values are determined by geometry. Two lines, each originating at $\delta_k$intersect two points on the circumference of F_i, denoted as *G* and *H*. The line segment $\overline{\delta_k G}$ creates the angle $\alpha_{iuv}$with the border, and the line segment $\overline{\delta_k H}$ creates the angle β_{iuv} with the border. The coverage cone created by the sensor's acquisition radius and the offset δ_k is bisected by the line segment $\overline{\delta_k C_i}$. Two congruent triangles are created by the bisection: $\Delta\delta_k C_i H$

and $\Delta\delta_k C_i G$ (Statement 1; see proof in appendix A). The length of $\overline{\delta_k G}$ and $\overline{\delta_k H}$ are equal due to congruency, but the length of these lines depends on the relationship between the value *Z* and *D*.

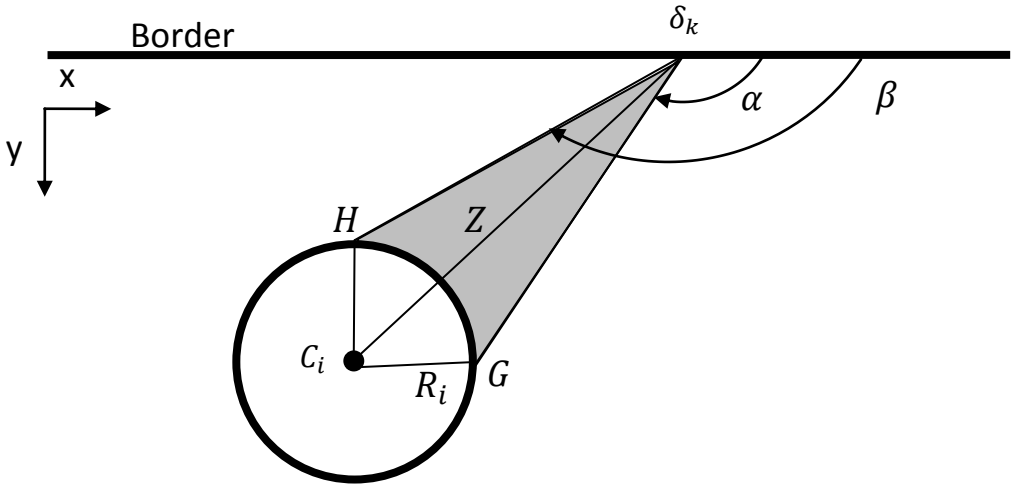

Figure 4.5: Coverage cone created by sensor i and a δ_k

Suppose that $Z \geq D$, then the values of α_{iuv} and β_{iuv} depend on the distance *D*. In figure 4.6, the distance *D* acts as the radius of a semi-circle originating at δ_k. This semi-circle represents the area in which the submerged contact can traverse within the defined patrol time provided straight trajectories with constant speed. Since $Z \geq D$, the points *G* and *H* lie along the circumference of the semi-circle and therefore the length of $\overline{\delta_k G}$ and $\overline{\delta_k H}$ is *D* (see fig. 4.7). Figure 4.8 shows the geometry of the $\Delta\delta_k C_i H$ for when $Z \geq D$. The $\Delta\delta_k C_i G$ would have the same geometry.

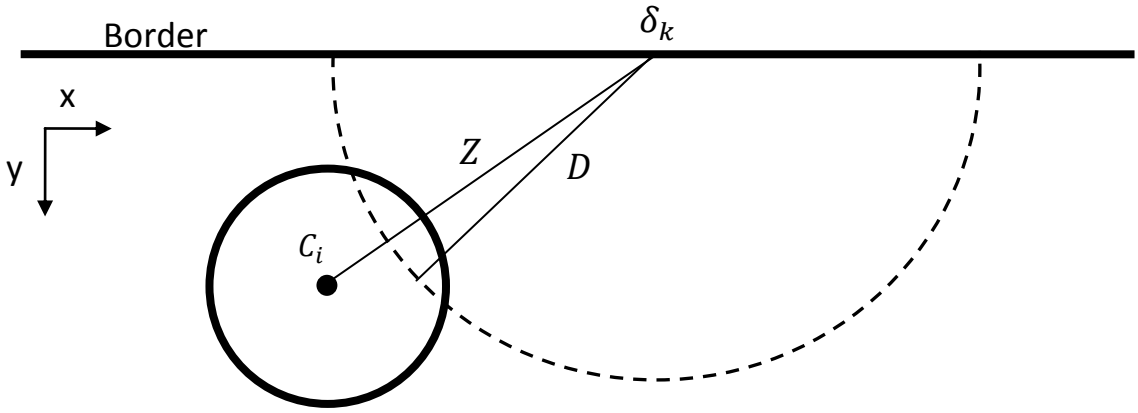

Figure 4.6: F_i overlapped with a semi-circle centered at a $\delta_k$with a radius of *D* when $Z \geq D$

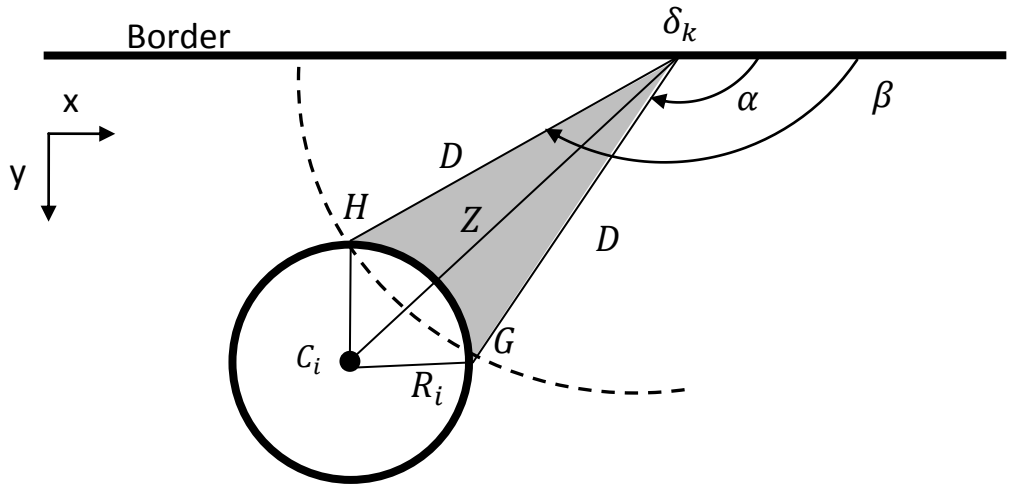

Figure 4.7: Coverage cone when $Z \geq D$

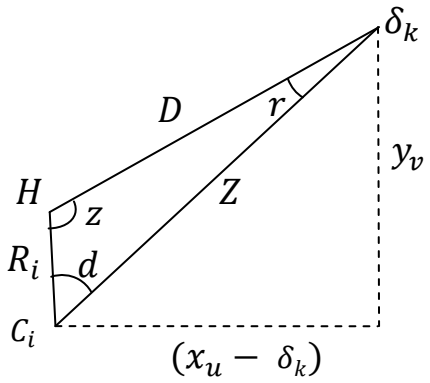

Figure 4.8: Geometry of $\Delta\delta_k CH$ when $Z \geq D$

The angle r will be used in the calculations of the angles α_{iuv} and β_{iuv}. Angle z is determined first using equation 4.16 and then used in equation 4.17 to calculate the angle r.

$$z = \cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_i D}\right] \tag{4.16}$$

$$r = \sin^{-1}\left[\frac{R_i \sin(z)}{Z}\right] \tag{4.17}$$

Figure 4.9 shows the geometry for offset values that are less than the x-coordinate of the sensor ($\delta_k < x_u$). Equations 4.18 and 4.19 are used to calculate α_{uv} and β_{iuv}, respectively. Equation 4.19 will always hold true regardless of the relationship between δ_k and x_u.

$$\alpha_{iuv} = \tan^{-1}\left(\frac{y_v}{x_u - \delta_k}\right) - r \tag{4.18}$$

$$\beta_{iuv} = \alpha_{iuv} + 2r \tag{4.19}$$

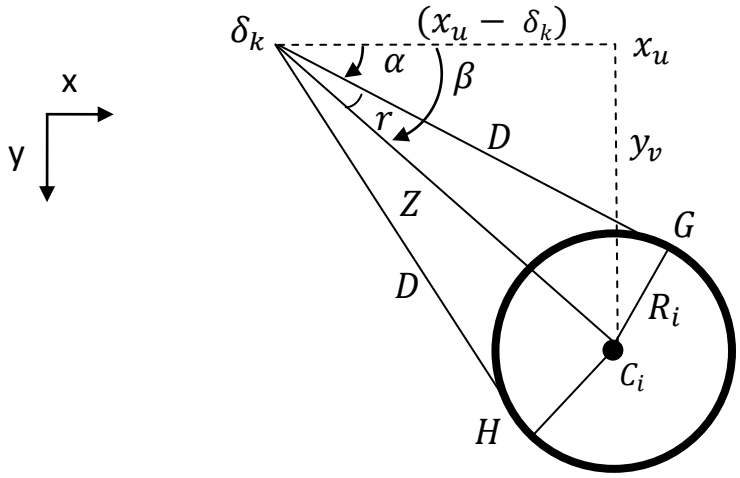

Figure 4.9: Geometry for ($\delta_k < x_u$)

Figure 4.10 shows the geometry for offset values that are greater than the x-coordinate of the sensor ($\delta_k > x_u$). Only the calculations for α_{iuv} change as the relationship between δ_k and x_u changes.

$$\alpha_{iuv} = \pi - \tan^{-1}\left(\frac{y_i}{\delta_k - x_u}\right) - r \tag{4.20}$$

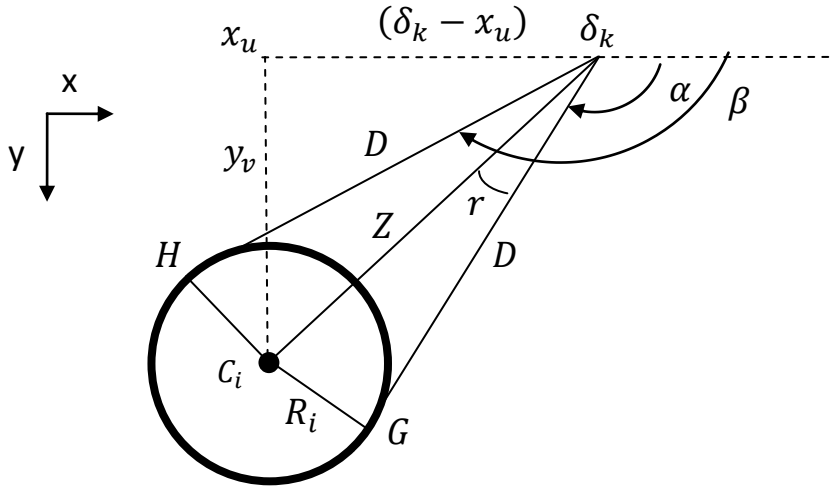

Figure 4.10: Geometry for ($\delta_k > x_u$)

Equation 4.20 reduces to equation 4.21 when $\delta_k = x_u$, which results from taking the arctangent of an undefined value.

$$\alpha_{uv} = {}^{\pi}/_{2} - r \tag{4.21}$$

In summary, the functions of α_{iuv} and β_{iuv} are as follows for when $Z \geq D$:

$$\alpha_{iuv}(\delta_k) = \begin{cases} \tan^{-1}\left(\frac{y_v}{x_u - \delta_k}\right) - \sin^{-1}\left[\frac{R_i \sin\left(\cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_iD}\right]\right)}{Z}\right] & \text{for A} \leq \delta_k < x_u \\ {}^{\pi}/_{2} - \sin^{-1}\left[\frac{R_i \sin\left(\cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_iD}\right]\right)}{Z}\right] & \text{for } \delta_k = x_u \\ \pi - \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) - \sin^{-1}\left[\frac{R_i \sin\left(\cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_iD}\right]\right)}{Z}\right] & \text{for } x_u < \delta_k \leq \text{B} \end{cases} \tag{4.22}$$

$$\beta_{iuv}(\delta_k) = \begin{cases} \tan^{-1}\left(\frac{y_v}{x_u - \delta_k}\right) + \sin^{-1}\left[\frac{R_i \sin\left(\cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_iD}\right]\right)}{Z}\right] & \text{for A} \le \delta_k < x_u \\ \pi/2 + \sin^{-1}\left[\frac{R_i \sin\left(\cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_iD}\right]\right)}{Z}\right] & \text{for } \delta_k = x_u \\ \pi - \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) + \sin^{-1}\left[\frac{R_i \sin\left(\cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_iD}\right]\right)}{Z}\right] & \text{for } x_u < \delta_k \le \text{B} \end{cases} \quad (4.23)$$

4.6.3 Calculations of the Ranges of θ_j When $Z < D$

Suppose that $Z < D$, then the values of α_{iuv} and β_{iuv} depend on a distance that is less than *D*, which is denoted as E. The points *G* and *H* that are associated with the lines that form the angles α_{iuv} and β_{iuv} are no longer located on the circumference of the semi-circle with radius *D* and centered at δ_k. Instead, the points lie on lines of length *D* that are tangent to the F_i. Since the lines are tangent to the circle, the length of $\overline{\delta_k G}$ and $\overline{\delta_k H}$ are less than *D* and the triangles $\Delta\delta_k C_i H$ and $\Delta\delta_k C_i G$ become right triangles. Figure 4.12 shows the coverage cone created when *Z* < *D*.

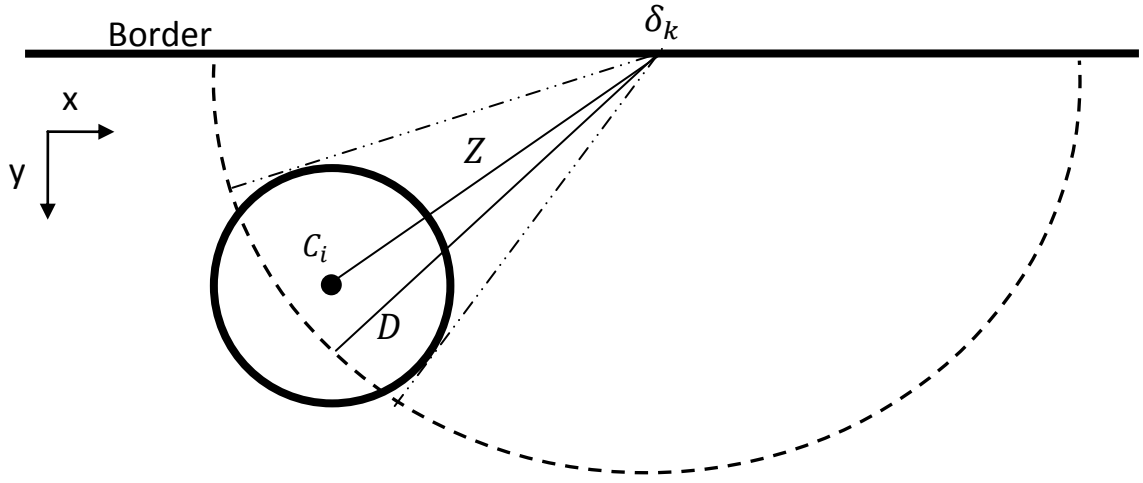

Figure 4.11: Footprint F_i overlapped with the semi-circle centered at $\delta_k$with a radius of *D* when $Z < D$

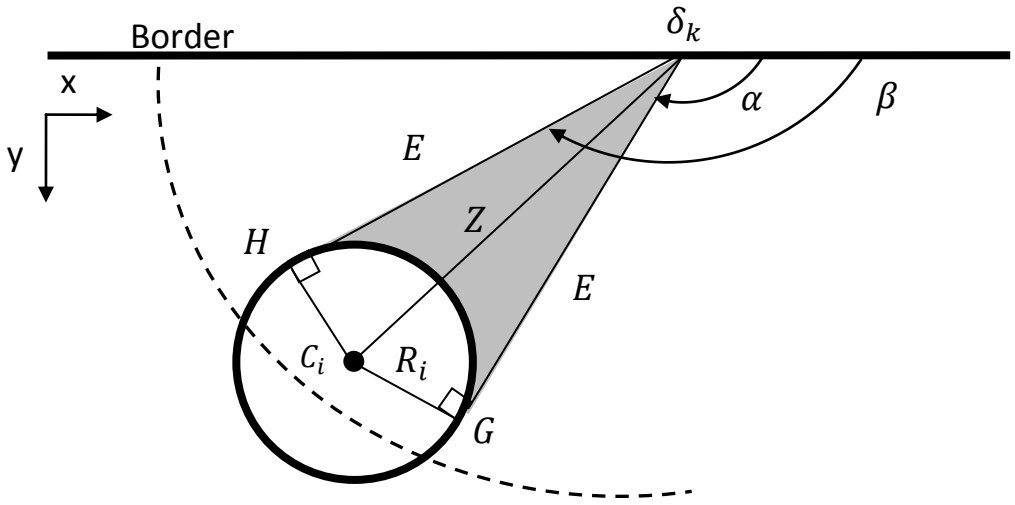

Figure 4.12: Coverage cone when $Z < D$

The length of $\overline{\delta_k G}$ and $\overline{\delta_k H}$ is calculated by equation 4.24.

$$E = \sqrt{Z^2 - R_i^2} \tag{4.24}$$

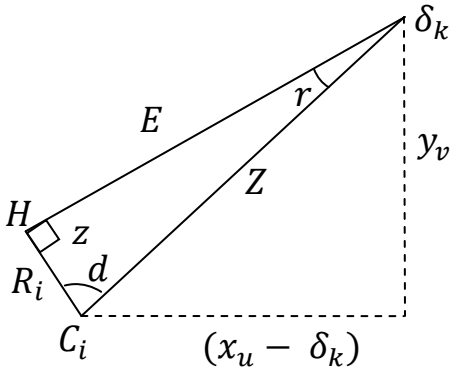

Figure 4.13: Geometry of $\Delta\delta_k C_i H$ when $Z \geq D$

Since the angle z is a right angle, equation 4.17 simplifies to

$$r = \sin^{-1}\left[\frac{R_i}{Z}\right] \tag{4.25}$$

The piecewise functions for the upper and lower bounds of the headings are then simplified to equations 4.26 and 4.27, respectively.

$$\alpha_{iuv}(\delta_k) = \begin{cases} \tan^{-1}\left(\frac{y_v}{x_u - \delta_k}\right) - \sin^{-1}\left[\frac{R_i}{Z}\right] & \text{for A} \le \delta_k < x_u \\ \pi/2 - \sin^{-1}\left[\frac{R_i}{Z}\right] & \text{for } \delta_k = x_u \\ \pi - \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) - \sin^{-1}\left[\frac{R_i}{Z}\right] & \text{for } x_u < \delta_k \le \text{B} \end{cases} \quad (4.26)$$

$$\beta_{iuv}(\delta_k) = \begin{cases} \tan^{-1}\left(\frac{y_v}{x_u - \delta_k}\right) + \sin^{-1}\left[\frac{R_i}{Z}\right] & \text{for A} \le \delta_k < x_u \\ \pi/2 + \sin^{-1}\left[\frac{R_i}{Z}\right] & \text{for } \delta_k = x_u \\ \pi - \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) + \sin^{-1}\left[\frac{R_i}{Z}\right] & \text{for } x_u < \delta_k \le \text{B} \end{cases} \quad (4.27)$$

4.6.4 Coverage Function Calculations

The coverage function, $\boldsymbol{CF_i}(\boldsymbol{x_u}, \boldsymbol{y_v})$ of a stationary sensor located at the coordinate(x_u, y_v) is a J x K matrix that contains Boolean values for a combination of θ_j and δ_k values. That is $\boldsymbol{CF_i}(\boldsymbol{x_u}, \boldsymbol{y_v}) = \left[cf_{ijk}(x_u, y_v)\ \right]$. An element of $\boldsymbol{CF_i}(\boldsymbol{x_u}, \boldsymbol{y_v})$has a value of 1 for a particular δ_k and θ_j if that $\theta_j$is between the α_{iuv} and β_{iuv} associated with that δ_k:

$$cf_{jk}(x_u, y_v) = \begin{cases} 1 & if\ \ \alpha_{iuv} \le \theta_j \le \beta_{iuv} \quad for\ \ \delta_k \\ & 0 \end{cases} \quad (4.28)$$

Algorithm 1 shows the pseudo code for calculating the coverage function of a sensor located at (x_u, y_v). This algorithm is used for every combination of (x_u, y_v) within algorithm 2.

Algorithm 1. Pseudo code for the Coverage Function for a Single Sensor at (x_u, y_v)

For a sensor with acquisition radius R_i
  Set $\delta_k = A$ and $\theta_j = 0$
    While $\delta_k < B$
      Calculate Z
      Calculate $\alpha_{iuv}(\delta_k)$ and $\beta_{iuv}(\delta_{ik})$
      While $\theta_j < \pi$
        If $Z > R_i + D$ **Then**
          $cf_{ijk}(x_u, y_v) = 0$ for θ_j, δ_k
        Else If $Z \leq R_i$ **Then**
          $cf_{ijk}(x_u, y_v) = 1$ for θ_j, δ_k
        Else If $R_i < Z \leq R_i + D$ **Then**
          If $\alpha_{iuv} \leq \theta_j \leq \beta_{iuv}$ **Then**
            $cf_{ijk}(x_u, y_v) = 1$ for θ_j, δ_k
          Else
            $cf_{ijk}(x_u, y_v) = 0$ for θ_j, δ_k
          End If
        End If
        Increase $\theta_j$by $\Delta\theta$
      End While
      Increase $\delta_k$by $\Delta\delta$
    End While
End For

4.7 The Geometric Approach (Discrete Case)

The geometric approach can be used for a single sensor or for a set of N sensors. The computational framework for the geometric approach is illustrated in figure 4.14. The double lined boxes represent the inputs to the model, while the single lined boxes care calculations within the geometric approach.

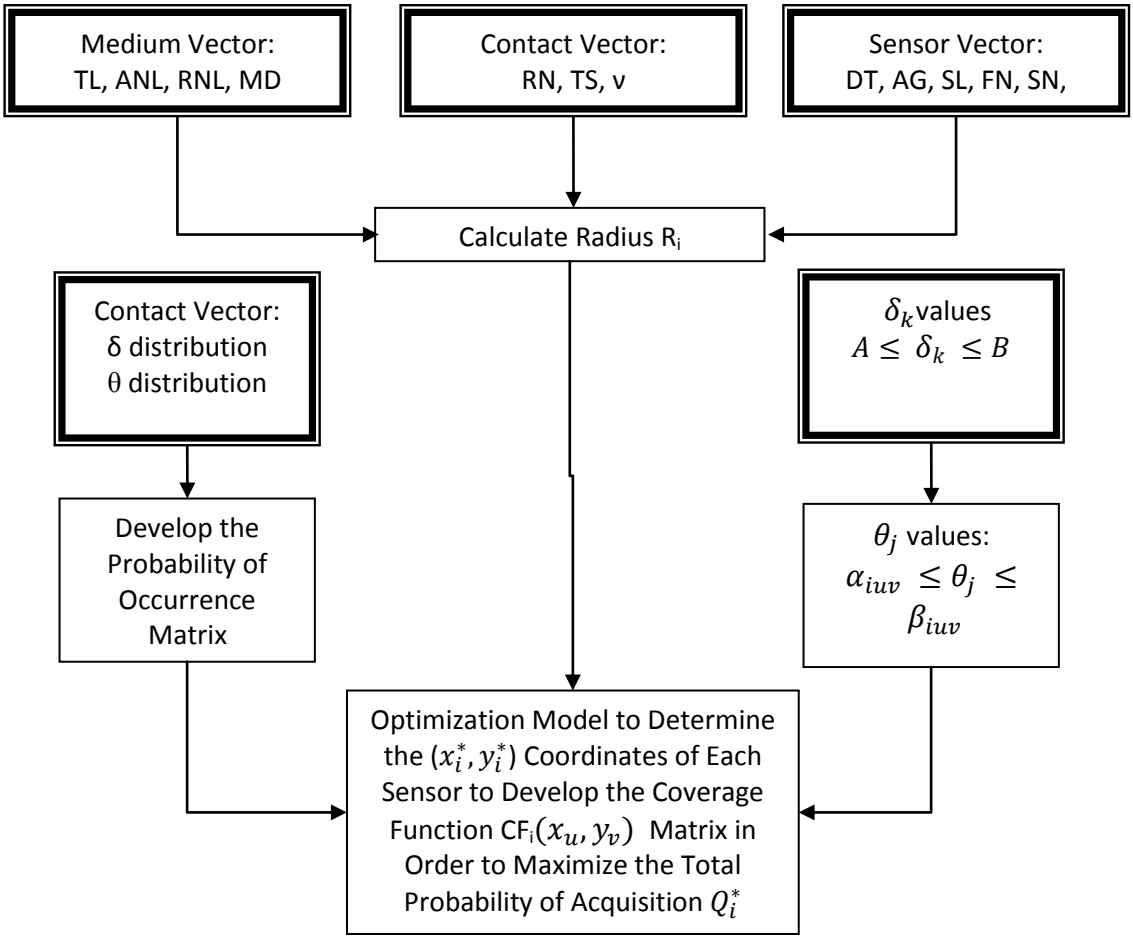

Figure 4.14: Computational Framework

Once the coverage functions are calculated for every (x_u, y_v) combination for all sensors, the placement of the sensor i within the area of interest can be determined by finding the maximum probability of acquisition, Q_i^*. The value Q_i is the Frobenius inner product of the probability of acquisition matrix $\boldsymbol{Q_i(x_u, y_v)}$. This matrix $\boldsymbol{Q_i(x_u, y_v)}$ for sensor i is formed by taking the Hadamard product of the occurrence matrix $\boldsymbol{O}$ and $\boldsymbol{CF_i(x_u, y_v)}$ matrices and has the a size of J x K.

$$\boldsymbol{Q_i(x_u, y_v)} = \boldsymbol{O} \; o \; \boldsymbol{CF_i(x_u, y_v)}$$

$$= \begin{bmatrix} o_{11} & \cdots & o_{1K} \\ \vdots & \ddots & \vdots \\ o_{J1} & \cdots & o_{JK} \end{bmatrix} \begin{bmatrix} cf_{11} & \cdots & cf_{1K} \\ \vdots & \ddots & \vdots \\ cf_{J1} & \cdots & cf_{JK} \end{bmatrix} = \begin{bmatrix} o_{11}cf_{11} & \cdots & o_{1K}cf_{1K} \\ \vdots & \ddots & \vdots \\ o_{J1}cf_{J1} & \cdots & o_{JK}cf_{JK} \end{bmatrix} \qquad (4.29)$$

The probability of acquisition for sensor i located at (x_u, y_v) is the sum of the elements of $\boldsymbol{Q_i(x_u, y_v)}$ (also known as the Frobenius inner product):

$$Q_i(x_u, y_v) = \sum_j^J \sum_k^K \left(o_{jk} * cf_{ijk}(x_u, y_v)\right) \tag{4.30}$$

To determine the optimal position of the sensor, an exhaustive search of $Q_i(x_u, y_v)$ is performed. Let Q_i^* be the optimal Q and let (x_i^*, y_i^*) be the coordinates of the sensor's center associated with Q_i^*. Every time $Q_i(x_u, y_v)$ is calculated, it is compared to $Q_i^*(x_i^*, y_i^*)$, and if $Q_i(x_u, y_v) > Q_i^*(x_i^*, y_i^*)$, then $Q_i(x_u, y_v)$ becomes the new $Q_i^*(x_i^*, y_i^*)$. Once all (x_u, y_v) combinations have been investigated, $Q_i^*(x_i^*, y_i^*)$ is reported. The $\boldsymbol{CF_i^*}(x_i^*, y_i^*)$ is the coverage function associated with the optimal position of the sensor.

Algorithm 2 shows the pseudo code for the geometric approach for a single sensor. The parameter vectors for both the medium and submerged contact are defined the same for all the DSN under consideration. Therefore, the matrix $\boldsymbol{O}$ is the same for all the DSN tested.

As an example, suppose that we are investigating the performance of a sensor with $R_i = 2$nm in a 10x10nm area of water to track a submerged contact travel at 5 knots during a patrol time of 2 hours. Suppose that $\Delta\delta = \Delta x = \Delta \mathrm{y} = 5\mathrm{nm}$ and $\Delta\theta = \pi/2$. These increments result in three discrete heading and offset values. Also, both the heading and offset values are uniformly distributed, resulting in the following $\boldsymbol{O}$ matrix:

$$\boldsymbol{O} = \begin{array}{c} \\ 0 \\ \pi/2 \\ \pi \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 5 & 10 \end{array} \\ \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \end{array} \tag{4.31}$$

Nine combinations of (x_u, y_v) values are investigated to determine the optimal location of the sensor. A coverage function is calculated for each (x_u, y_v) combination using algorithm 1. The probability of acquisition is then calculated for the (x_u, y_v) coordinate. By hand calculations, the coordinates (0, 0), (0, 5), and (0, 10) all have a $Q_i = 5/9$, so the sensor can be placed at any of these coordinates and have the same performance. The MATLAB program computes that the

sensor can be placed at either (0, 0), (0, 5), or (0, 10) which will give a value of $Q_i^* = 0.5556$. This agrees with the hand calculations and demonstrates that the program does work properly.

Algorithm 2. Pseudo code for the Geometric Approach for a Single Sensor

```
Define medium parameter vector
Define submerged contact parameter vector
Define distributions for the heading and offset of the submerged contact
Calculate the O matrix for the submerged contacts
Set Q_i^*(x_i^*, y_i^*) = 0
For a sensor i with acquisition radius R_i
    Set x_u = A and y_v = 0
    While x_u < B
        While y_v < L
            Calculate the CF_i(x_u, y_v) {see Algorithm 1}
            Calculate Q_i for sensor positioned at (x_u, y_v)
            If Q_i(x_u, y_v) > Q_i^*(x_i^*, y_i^*) Then
                Q_i^*(x_i^*, y_i^*) = Q_i(x_u, y_v)
            End If
            Increase y_v by Δy
        End While
        Increase x_u by Δx
    End While
End For
Report Q_i^*(x_i^*, y_i^*) for the sensor i
```

4.8 The Geometric Approach (Continuous Case)

In the continuous case, the geometric approach is slightly different. When the pdfs explaining the heading and offset values are continuous, the coverage function becomes dependent on the distribution. In our example, we have that $\theta \sim Uniform\ (0, \pi)$ and $\delta \sim Uniform\ (0, W)$ and the resulting mass is equivalent for all combinations of θ and δ. The probability of acquisition is dependent on the sensor's CF, which is a function of the (x_u, y_v) coordinates of the sensor's center and the θ and δ values. When the pdf for the heading is uniform, the CF is then only dependent on the (x_u, y_v) coordinates and the δ values.

$$CF = f(x_u, y_v, \delta) = \frac{\beta - \alpha}{\pi} \tag{4.32}$$

From equation 4.19,

$$\beta - \alpha = 2r = 2\sin^{-1}\left[\frac{R_i \sin(z)}{Z}\right] \tag{4.33}$$

Written in terms of x_u, y_v, and δ, equation 4.33 becomes

$$\beta - \alpha = 2r = 2\sin^{-1}\left\{\frac{R_i \sin\left[\cos^{-1}\left(\frac{R_i^2 + D^2 - y_v^2 + (x_u - \delta_k)^2}{2R_i D}\right)\right]}{\sqrt{y_v^2 + (x_u - \delta_k)^2}}\right\} \tag{4.34}$$

And equation 4.32 then becomes

$$CF = f(x_u, y_v, \delta) = \frac{2}{\pi}\sin^{-1}\left\{\frac{R_i \sin\left[\cos^{-1}\left(\frac{R_i^2 + D^2 - y_v^2 + (x_u - \delta_k)^2}{2R_i D}\right)\right]}{\sqrt{y_v^2 + (x_u - \delta_k)^2}}\right\} \tag{4.35}$$

The probability of acquisition dependent on only the (x_u, y_v) coordinates of the sensors location is then

$$g(x_u, y_v) = P(Q|x_u, y_v) = \int_{\delta=0}^{W} f(x_u, y_v, \delta)\, d\delta$$

$$= \int_{\delta=0}^{W} \frac{2}{\pi}\sin^{-1}\left\{\frac{R_i \sin\left[\cos^{-1}\left(\frac{R_i^2 + D^2 - y_v^2 + (x_u - \delta_k)^2}{2R_i D}\right)\right]}{\sqrt{y_v^2 + (x_u - \delta_k)^2}}\right\} d\delta \tag{4.36}$$

The global optimum Q_i^* can then be found using ∇g to search the x_u, y_v, Q space.

Chapter 5: Methodology for Multiple Sensors

This chapter continues the geometric approach from the discrete case of a single sensor and extends it to two sensors. The calculations for the probability of occurrence matrix and the coverage functions of the sensor are the same as in the single sensor case. However, the algorithm developed to place the sensors is slightly different.

5.1 Context and Assumptions

Multiple sensors of various sizes may be included in a DSN. The problem is that of determining where to place these sensors so that they maximize their total probability of acquisition for the DSN. This maximization occurs when the sensors cover more heading and offset combinations for the area of interest. The methodology presented in the next section results in a lower bound in the optimal performance of the DSN due to the heuristic approach. Sensor *n +1* is placed in response to the locations of the *n* sensors that have already been positioned. There exists *N* number of sensors that will completely cover all the heading and offset combinations; at which point additional sensors will increase the cost, but not the probability of acquisition.

There are several assumptions concerning the heuristic approach. First, once a sensor is placed, it does not change its position as more sensors are added. Also, once a particular heading and offset combination is acquired by at least one sensor, it is of less importance than combinations that have yet to be acquired. Some combinations will be covered by multiple sensors, which will provide extra coverage and backup if a sensor stops working. The goal, however, is to maximize the total number of unique combinations that are acquired by the DSN.

5.2 The Geometric Approach for Two Sensors

We first look at the simplest example of a DSN- the two sensor network. In our heuristic approach, the first sensor is placed according to algorithm 2 and then the second sensor is placed in response to the first sensor's position. The first sensor is placed at the (x_u, y_v) for which the sensor covers the largest number of heading and offset combinations that have a higher probability of occurring. Once the optimal location is determined for the first sensor, the

coverage function that is associated with that position influences the positioning of the second sensor. Consider the example from §4.7, whose $\boldsymbol{O}$ matrix is as follows:

$$\boldsymbol{O} = \begin{matrix} \\ 0 \\ \pi/2 \\ \pi \end{matrix} \begin{matrix} \begin{matrix} 0 & 5 & 10 \end{matrix} \\ \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \end{matrix} \tag{5.1}$$

Algorithm 2 places the sensor at (0, 0) with a $Q_i^* = 5/9$, and the associated coverage function is as follows:

$$\boldsymbol{CF_1}(\mathbf{0},\mathbf{0}) = \begin{matrix} \\ 0 \\ \pi/2 \\ \pi \end{matrix} \begin{matrix} \begin{matrix} 0 & 5 & 10 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{matrix} \tag{5.2}$$

When we place the second sensor, we do not want to cover the same heading and offset combinations that have already been covered. In some sense, we want to ignore these combinations and focus on placing the next sensor at a location where it can cover the heading and offset values that the first sensor has missed. Therefore, we transform the $\boldsymbol{O}$ matrix to reflect the combinations that have been covered by the first sensor.

This new $\boldsymbol{O}$ matrix is created by replacing the o_{jk} of the $\boldsymbol{O}$ matrix with zeros for the $cf_{1jk}(x_u, y_v)$ values that equal one, while all other values of o_{jk} remain the same as in the original $\boldsymbol{O}$ matrix (see algorithm 3). With this new $\boldsymbol{O}$ matrix, the second sensor is then placed using algorithm 2. By transforming the $\boldsymbol{O}$ matrix, we force the second sensor to be positioned at a location that covers the combinations that the first sensor has missed. This occurs because the geometric approach algorithm chooses the coverage function that covers the heading and offset combinations that have the highest o_{jk}.

Returning to our example, the new $\boldsymbol{O}$ matrix becomes

$$\boldsymbol{O} = \begin{matrix} \\ 0 \\ \pi/2 \\ \pi \end{matrix} \begin{matrix} \begin{matrix} 0 & 5 & 10 \end{matrix} \\ \begin{bmatrix} 0 & 1/9 & 1/9 \\ 0 & 1/9 & 1/9 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix} \tag{5.3}$$

Notice that the sum of the elements is less than one. We are concerned with maximizing the probability of acquisition for the second sensor over only a select set of heading and offset combinations and not the entire set of combinations. There will be some combinations that are covered by both sensors, but their o_{jk} value is counted only once for the total Q^* of the DSN.

Assuming the two sensors are homogeneous, that is they have the same acquisition radius, we can use the same set of coverage functions that were already calculated for sensor one. When algorithm 2 is run with the new $\boldsymbol{O}$ matrix, the second sensor can be placed at (5, 0) or (10, 0) since both locations result in a Q_i^* = 3/9. The following are the coverage functions for these two locations. Notice that both matrices have three of the four originally missed combinations covered.

$$\boldsymbol{CF_1}(\mathbf{5},\mathbf{0}) = \begin{matrix} \\ 0 \\ \pi/2 \\ \pi \end{matrix} \begin{matrix} \begin{matrix} 0 & 5 & 10 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{matrix} \tag{5.4}$$

$$\boldsymbol{CF_1}(\mathbf{10},\mathbf{0}) = \begin{matrix} \\ 0 \\ \pi/2 \\ \pi \end{matrix} \begin{matrix} \begin{matrix} 0 & 5 & 10 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \tag{5.5}$$

Either one of these locations can be chosen for the second sensor. The total coverage function (***TCF***) for the DSN is then calculated as the union of the two individual coverage functions.

$$\boldsymbol{TCF} = \boldsymbol{CF_1}(x_1^*, y_1^*) \cup \boldsymbol{CF_2}(x_2^*, y_2^*) \tag{5.6}$$

The total Q_m^* of the DSN is then the Frobenius inner product of the Hadamard product ($\boldsymbol{Q}$ matrix) of this ***TCF*** and the original $\boldsymbol{O}$ matrix. The total Q^* of the DSN is then 8/9. The MATLAB program supports these hand simulations.

$$\boldsymbol{Q} = \boldsymbol{O} \; o \; \boldsymbol{TCF} \tag{5.7}$$

$$Q_m^* = \sum_j^J \sum_k^K \left(o_{jk} * tcf_{jk}\right) \tag{5.8}$$

Suppose that a third homogeneous sensor is available to be placed. The $\boldsymbol{O}$ matrix that resulted from the previous transformation is then changed using the coverage function associated with the second sensor's position. The third sensor is then placed by using algorithm 2 with the newest $\boldsymbol{O}$ matrix. For instance, let the second sensor from our example be located at (10, 0). Then the $\boldsymbol{O}$ matrix is update to

$$\boldsymbol{O} = \begin{matrix} \\ 0 \\ \pi/2 \\ \pi \end{matrix} \begin{matrix} \begin{matrix} 0 & 5 & 10 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/9 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix} \tag{5.9}$$

The third sensor is then placed to cover the combination $\theta = {}^{\pi}/_{2}$ and $\delta = 5$. However, the sensor cannot be placed at any position where another sensor is currently occupying. If the sensors are heterogeneous, the coverage functions will need to be calculated for each unique sensor. Sensors are 'unique' when their acquisition radii are not equal.

The process continues for all N sensors in the DSN, and the result is a lower bound for the total Q_m^* of the DSN. It may occur that not all N sensors are needed to cover all the combinations, and the total Q_m^* = 1. Therefore, any additional sensors will provide duplicated coverage. Algorithm 3 shows the logic for N sensors with this heuristic approach.

Algorithm 3. Pseudo code for the Geometric Approach for N Sensors

```
Define medium parameter vector
Define submerged contact parameter vector
Define distributions for the heading and offset of the submerged contact
Calculate the original O matrix for the submerged contacts
While i <= N
        Determine the optimal location (x*_iu, y*_iv) for sensor i
        Transform the O matrix by using the CF_i(x*_i, y*_i)
                if cf_ijk(x*_i, y*_i) = 1, then o_jk = 0 for the new O matrix
                if cf_ijk(x*_i, y*_i) = 0, then o_jk remains the same for the new O matrix
End For
Calculate the TCF
Use the TCF to calculate the total Q*_m for the DSN
Report (x*_i, y*_i) for all the sensors, and Q*_m for the DSN
```

5.2 Geometric Approach for N sensors

Instead of positioning the sensors in response to previously placed sensors, the sensors can be simultaneously placed in the area, which can lead to the global optimum for the problem of N sensors. However, this requires an immense amount of computer memory and computational time. The methodology is presented in this section, but its development into a MATLAB program is part of the future research.

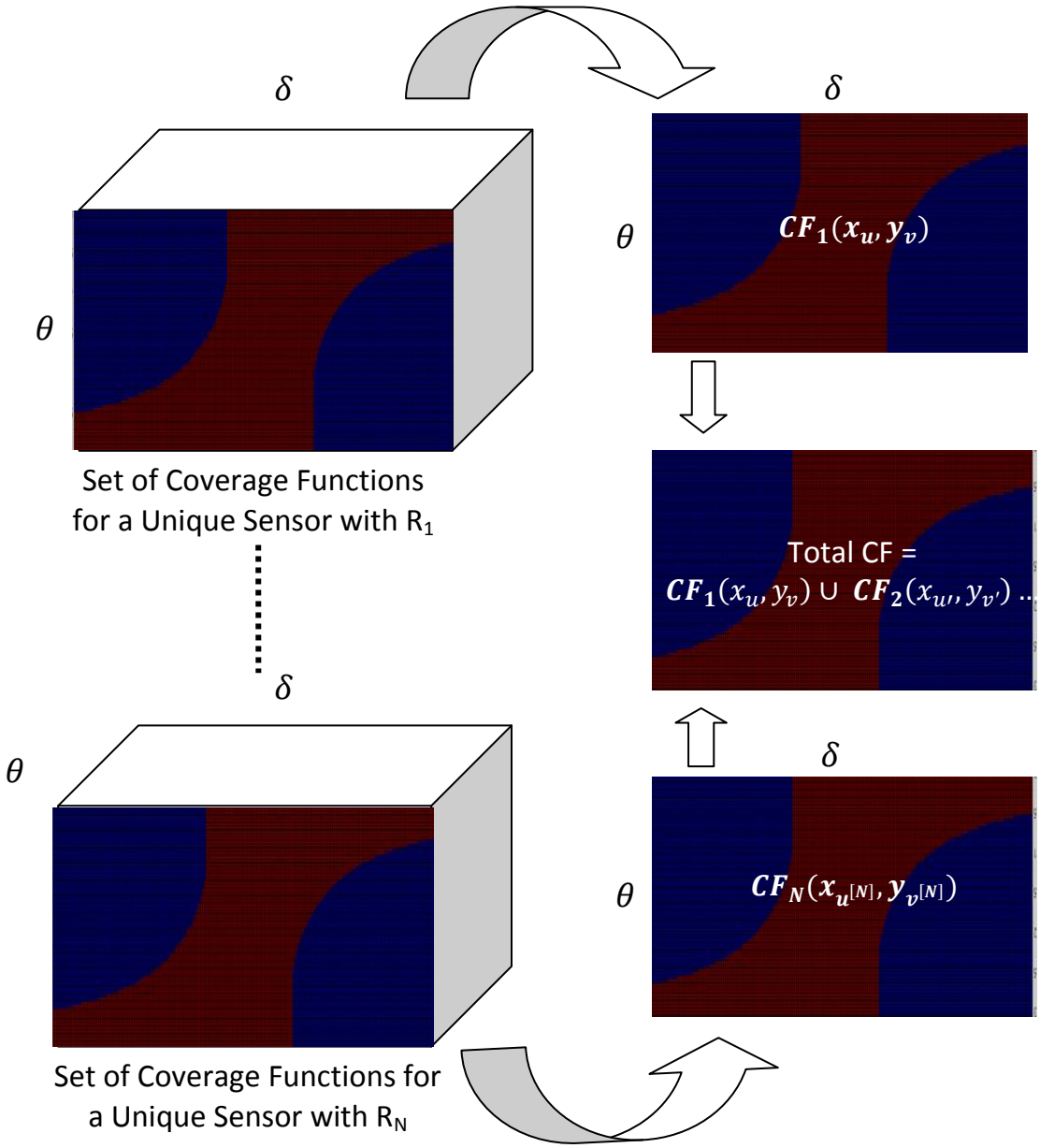

Figure 5.1: 3D Matrices for N sensors and Coverage Functions

For each unique sensor, a set of coverage functions are developed for all the (x_u, y_v) combinations that exist. The collection of the coverage functions can be thought of as a 3D matrix where the pages of the matrix are the coverage functions (see fig. 5.1). To simultaneously place sensors, a coverage function is chosen for each sensor from the associated 3D matrix that satisfies the following objective function:

$$Max\ \sum_j^J \sum_k^K (o_{jk} * tcf_{jk}) \tag{5.10}$$

subject to

$$tcf_{jk} = cf_{ijk}\left(x_{u^{[i]}}, y_{v^{[i]}}\right) \vee cf_{ijk}\left(x_{u^{[i+1]}}, y_{v^{[i+1]}}\right) \vee \dots cf_{Njk}\left(x_{u^{[N]}}, y_{v^{[N]}}\right) \tag{5.11}$$

$$\left(x_{u^{[i]}}, y_{v^{[i]}}\right) \neq \left(x_{u^{[i+1]}}, y_{v^{[i+1]}}\right) \neq \left(x_{u^{[i+2]}}, y_{v^{[i+2]}}\right) \neq \cdots \left(x_{u^{[N]}}, y_{v^{[N]}}\right) \tag{5.12}$$

This optimization problem chooses the coverage functions whose union creates a ***TCF*** whose Hadamard product with the $\boldsymbol{O}$ matrix results in the highest Q_m^* value. To perform this optimization, large 3D matrices must be stored in the computer's memory for all unique sensors under consideration, and then the program must search through all these coverage functions to find the optimal combination of coverage functions that cover the most heading and offset combinations.

5.3 Explanation of Pareto Analysis

Once the total Q_m^* of the DSN is calculated for all DSN under consideration, a Pareto analysis is performed using the two metrics previously discussed in §4.3. The analysis is used to decide which DSN is the best for the given environment and submerged contact pdfs. There exists a feasible region in the Q_m^* vs. LC graph that contains all the DSN that can exist (see fig. 5.2). The Pareto frontier of this region is the region's boundary for which the probability of acquisition is maximized while the lifecycle cost is minimized. For a given set of DSN, their performance and cost are plotted on the graph.

A DSN is a Pareto improvement over another DSN if it has the same or higher Q_m^* without increasing the cost. For instance, DSN A is a Pareto improvement over DSN B in figure 5.2 since

both DSNs perform equally well, but A costs less than B. Also, A is a Pareto improvement over C since the costs are equal, but A performs better than C. Any DSN on the Pareto frontier would be strictly dominate over A, B, and C.

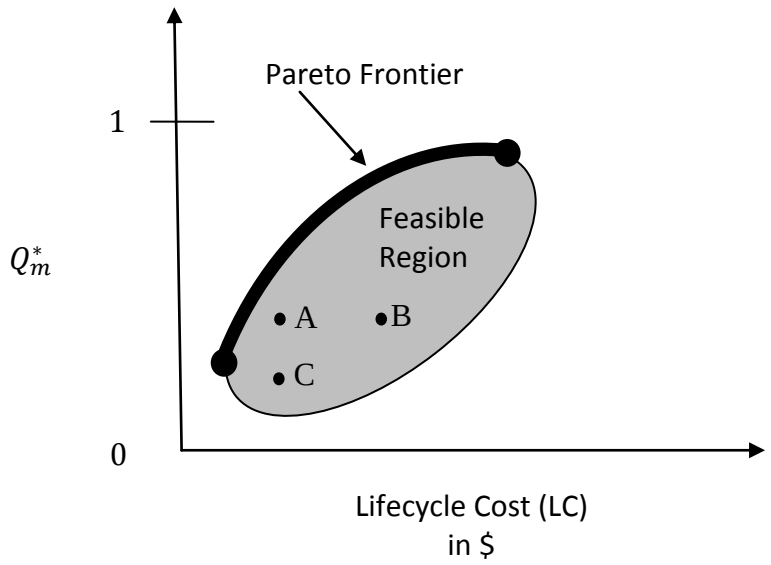

Figure 5.2: Metric space for DSN Analysis

When analyzing the set of *M* DSNs, the best DSN is closest to the Pareto Frontier. This frontier may be unknown, but the *M* DSN can be compared to one another to determine the optimal system. An example of the Pareto analysis for a set of five two-sensor DSN is included in §6.5.

Chapter 6: Experimentation

This chapter describes the MATLAB program that implements the methodology from §4.7. Results from a Design of Experiments (DOE) are summarized and an example of a Pareto assessment of DSNs composed of two sensors is performed with results discussed in length.

6.1 MATLAB Program

A MATLAB program was developed for the discrete case. It takes input from the user to calculate the probability of occurrence matrix, the coverage functions, and the probability of acquisition. Users can choose from built-in pdfs for the headings and offsets or manually enter the probabilities for each heading and/or offset value. The program prompts the user to input the acquisition radius for all sensors along with the sensor's base and sensing costs. For a single sensor, the program uses the radius of the sensor, the submerged contact's velocity, and the patrol time to create the coverage functions for each (x_u, y_v) coordinate in the area of interest. The value of Q_i is then calculated and compared to the current Q_i^*. If the new value of $Q_i > Q_i^*$, Q_i replaces Q_i^*. The program performs an exhaustive search of the (x_u, y_v) combinations and reports the coordinates associated with Q_i^*. The experiments were conducted on a Dell Optiplex 755 desktop with an Intel Core2 Quad processor.

6.2 Results from Experiments of a Single Sensor

A $4^2 2^{5-4}$ mixed-level fractional factorial DOE was performed to determine the effects that some variables may have on the positioning and performance of the sensors. The factors were radius size, increment values for the heading and offsets, patrol times, submerged contact velocities, and the pdfs of headings and offsets. For all experiments, the width and the length of the area of interest are each 100nm. Table 6.1 summarizes the factor levels. There are four levels for the pdfs so that various pdfs may be analyzed, but there are two levels for all other factors to reduce the number of runs in order to still achieve meaningful results. The experimental design contains 32 runs and is shown in appendix B. Four responses were measured: Q_i^*, x_i^*, y_i^*, and the processing time. A coding scheme is used to convert the four-level factor into two two-level factors [43]. Table 6.2 shows the coding schemes for the pdf factors. This results in a 2^{9-4} fractional factorial design. All results are shown in appendix B with Minitab output.

Table 6.1: Summary of the Factor Levels

Factors		Level 1 (-)	Level 2 (+)	Level 3 (coded)	Level 4 (coded)
A, B	**Heading PDF**	Uniform	Triangular (C = K/4)	Triangular (C = K/2)	Triangular (C = 3K/4)
C, D	**Offset PDF**	Uniform	Triangular (C = J/4)	Triangular (C = J/2)	Triangular (C = 3J/4)
E	**Acquisition Radius (nm)**	10	15	NA	NA
F	**Heading Increment (radians)**	0.0031416	0.031416	NA	NA
G	**Offset Increment (nm)**	0.5	1	NA	NA
H	**Patrol Time (hours)**	2	4	NA	NA
J	**Submerged Contact Velocity (nm/h)**	10	15	NA	NA

Table 6.2: Coding Schemes for Qualitative Four-level Factors

Two-Level Factors		Four-Level Factor	Two-Level Factors		Four-Level Factor
A	B	Heading PDF Type	C	D	Offset PDF Type
-	-	1	-	-	1
+	-	2	+	-	2
-	+	3	-	+	3
+	+	4	+	+	4

6.2.1 Complete Results of Run One with Plots

The 32 runs were randomized and the four responses were recorded (see appendix B). The first run considered a single sensor with R_i = 15nm, $\Delta\delta$ = 0.5nm, $\Delta\theta$ = 0.03 radians, T = 4 hours, and ν =15 nm/h (knots). The heading pdfs was a triangular distribution with the index for the C value

set to 50, indicating symmetric functions over $\theta = \pi/2$. The offset pdf followed a uniform distribution. Results from the program report that Q_1^* = 0.38231 when the sensor is located at position (51, 10). As expected, the sensor was placed close to the border near the thread axis. The sensor was positioned near the thread axis in order to acquire contacts that would be coming from most of the offsets. It was also placed close to the border so that it can acquire a majority of the headings, but it was also placed far enough away from the border in order to acquire the contacts at the headings of $\pi/2$. The experiment took 259.2655 seconds to run.

The 3D and the pseudocolor (checkerboard) plots from MATLAB show the probabilities of a specific heading and offset combination occurring (see fig. 6.1). As indicated, the pmf is a triangular distribution for the heading values projected uniformly for all the offset values.

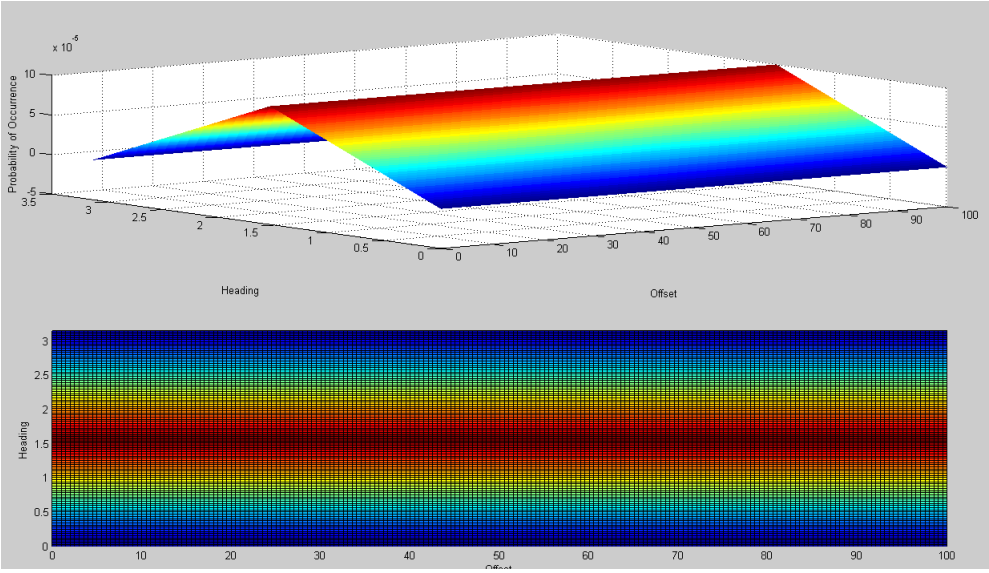

Figure 6.1: MATLAB 3D and Pseudocolor Plots for the Probability of Occurrence Matrix

The program then determines the CF_i for the sensor for every possible (x_u, y_v) coordinate. The CF related to the optimal location is shown in figure 6.2. The red indicates a CF value of 1, while the blue indicates 0. Notice that for the offsets within $\pm$15 units of 50nm have all heading values covered. This is due to the fact that the sensor is positioned close enough to the border as to overlap the border at these x_u values.

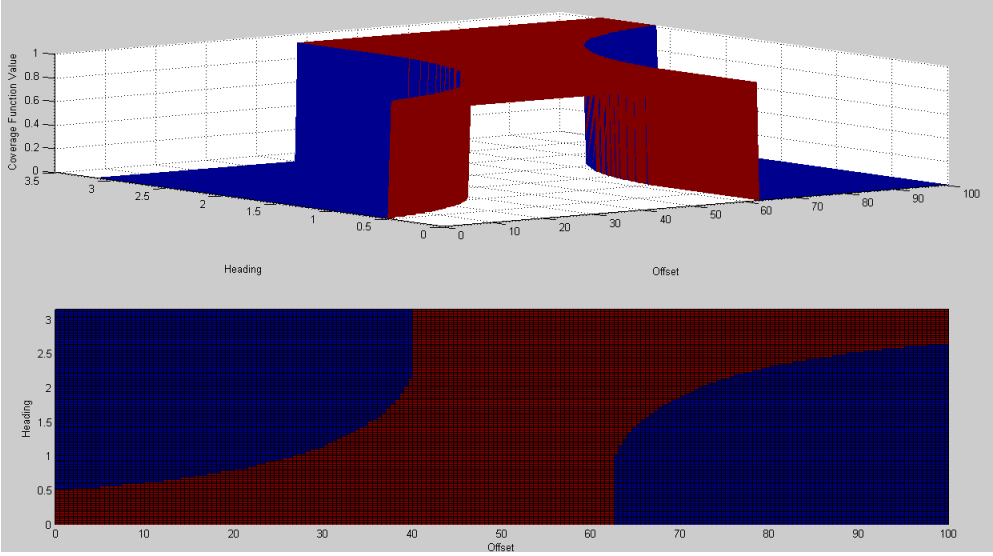

Figure 6.2: MATLAB 3D and pseudocolor plots for the CF of a sensor whose R_i = 15nm and position is (51, 10)

The $\boldsymbol{Q_i}$ matrix that is used to determine Q_i is shown graphically in figure 6.3. Note that the plots are simply the intersection of both the probability of occurrence plots and the coverage function plots. Figure 6.4 shows the Q_i values calculated at each (x_u, y_v) coordinate for the single sensor. The red in figure 6.4 corresponds to a higher Q_i value. Notice that the Q_i values drop off considerably at 60nm in the y direction since this is the maximum distance that the submerged contact could travel within the defined patrol time.

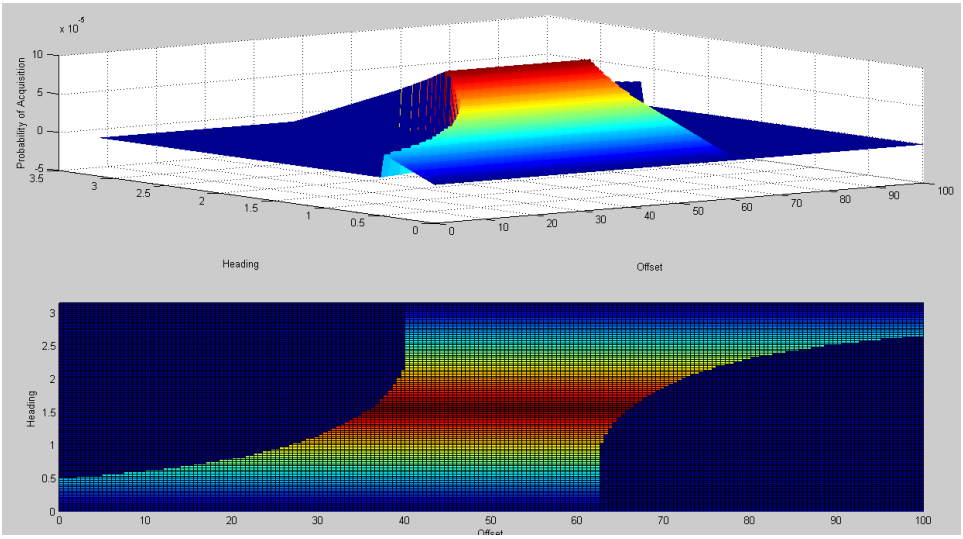

Figure 6.3: MATLAB 3D and Pseudocolor Plots for the $\boldsymbol{Q_i}$ Matrix

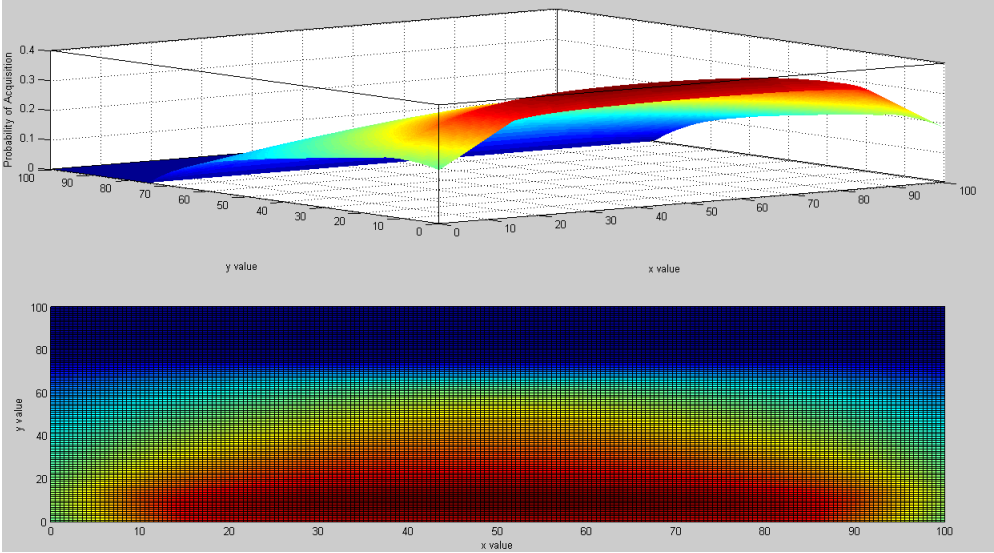

Figure 6.4: Q_iValues For every (x_u, y_v) Combination

6.2.2 Summary of Results

All results from the $4^2 2^{5-4}$ mixed-level fractional factorial DOE are shown in appendix B. Results were analyzed in Minitab to draw conclusions about the effects that various factors had on one or more of the responses. When a run resulted in more than one optimal location to place the sensor, the coordinate with the minimum x value was used in the analysis since the two sensor approach transforms the set of offset values by removing the offset values from 0 to the first sensor's offset value plus the radius of the first and second sensors. Therefore, we take the coordinate with the lowest x value in order to reduce the number of offset values removed. All results from the Minitab analysis are in appendix B, however, the normal probability plots are shown in this section.

For the response of Q_1^*, the normal probability plot shows that the offset pdf (factors C,D,CD), heading pdf (factor A,B, AB), radius R_i (factor E), patrol time T (factor H), and the interaction between the offset pdf and the radius (factor DE) have significant affects on the response (see fig. 6. 5). As expected, both pdfs impact the response since they affect the matrix $\boldsymbol{O}$, which affects the Q_i^*. The acquisition radius and patrol time should impact the performance of the sensor since they both directly affect the sensor's coverage function. Results show that the larger acquisition radius size and longer patrol time correspond to a higher Q_1^* value (see fig. 6.6). The interaction DE is not as significant as the other factors.

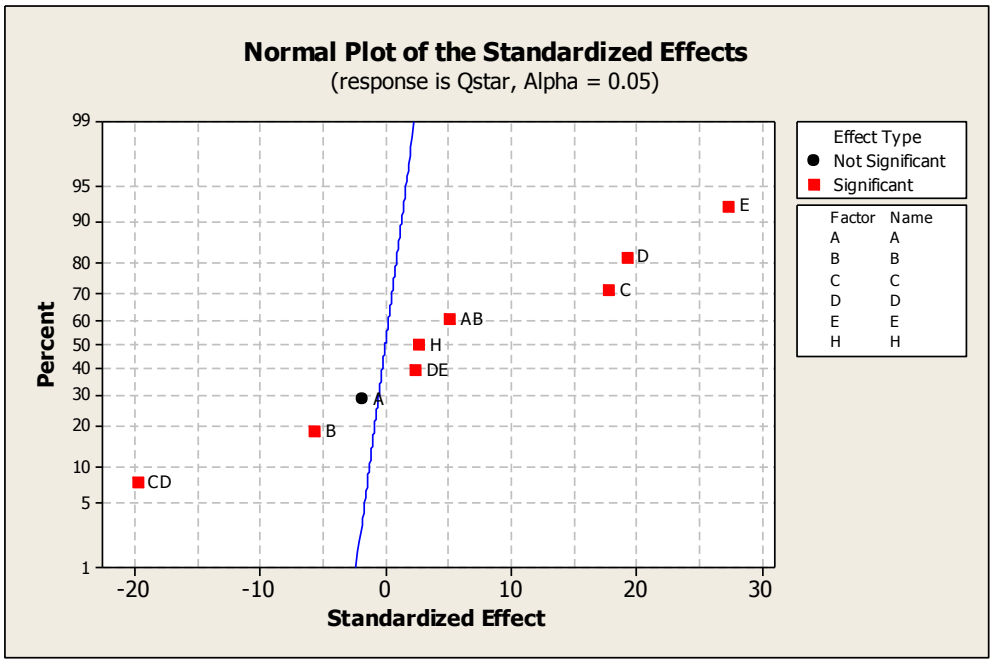

Figure 6.5: Normal Probability Plot for Q_1^*

Term	Effect	Coef	SE Coef	T	P
Constant		0.47967	0.002504	191.58	0.000
A	-0.00887	-0.00443	0.002504	-1.77	0.090
B	-0.02818	-0.01409	0.002504	-5.63	0.000
C	0.08916	0.04458	0.002504	17.81	0.000
D	0.09656	0.04828	0.002504	19.28	0.000
E	0.13735	0.06868	0.002504	27.43	0.000
H	0.01330	0.00665	0.002504	2.66	0.014
A*B	0.02580	0.01290	0.002504	5.15	0.000
C*D	-0.09890	-0.04945	0.002504	-19.75	0.000
D*E	0.01193	0.00596	0.002504	2.38	0.026

Figure 6.6: Estimated Effects and Coefficients for Q_1^*

The x_1^* response is dependent on a few factors, but mostly on the offset pdf (see fig. 6.7). Other significant main factors are T and the submerged contact velocity v. The interaction of the offset pdf and v is also significant (factor CJ). The results support the concept that a sensor will be placed close to the offsets that have a high probability of occurring since it depends on the offset pdf (see fig. 6.8). The two other influential main factors also are logically significant factors since they are concern with the distances that the submerged contact could travel. The interaction CJ is interesting since the velocity affects how far into the area of interest a submerged contact

will travel and the offset determines where that contact would enter the area. Therefore, their interaction would affect the placement of the sensor.

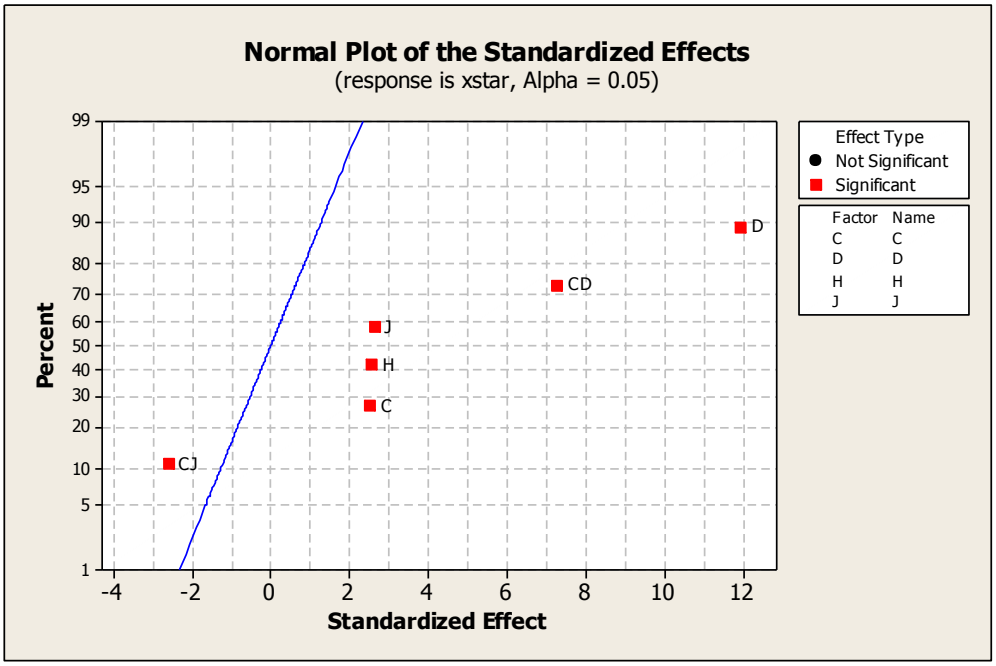

Figure 6.7: Normal Probability Plot for x_1^*

Term	Effect	Coef	SE Coef	T	P
Constant		47.344	0.8886	53.28	0.000
C	4.500	2.250	0.8886	2.53	0.018
D	21.250	10.625	0.8886	11.96	0.000
H	4.562	2.281	0.8886	2.57	0.017
J	4.688	2.344	0.8886	2.64	0.014
C*D	12.937	6.469	0.8886	7.28	0.000
C*J	-4.625	-2.312	0.8886	-2.60	0.015

Figure 6.8: Estimated Effects and Coefficients for x_1^*

All of the same significant factors that affect x_1^*, except the interaction CJ, also affect the y_1^* response. However, R_i is a significant term since a larger radius will tend to push the sensor farther away from the border, resulting in a higher y_1^* value (see fig. 6.10).

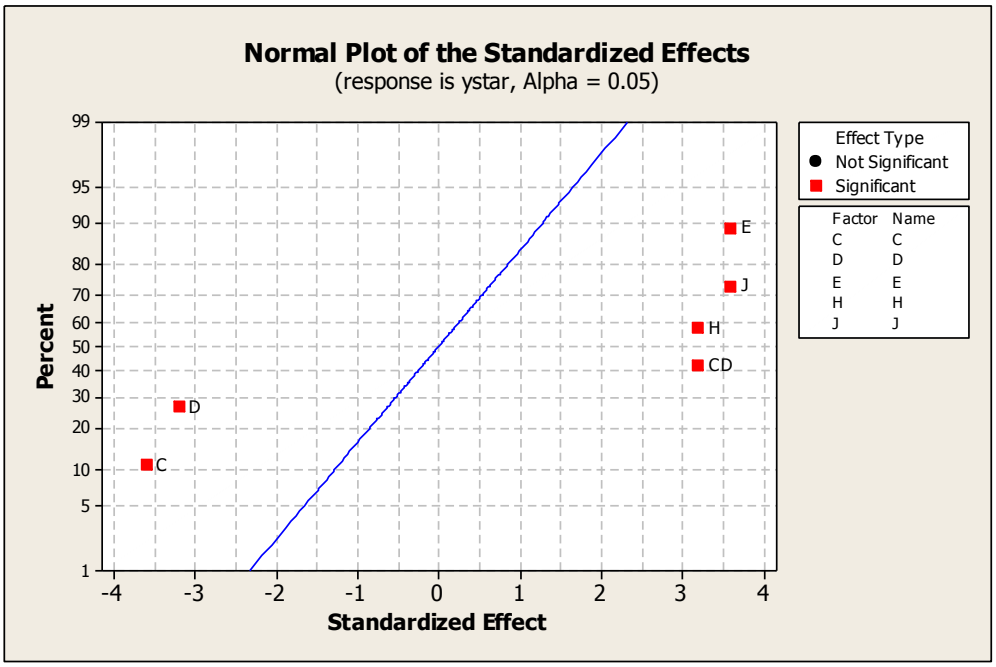

Figure 6.9: Normal Probability Plot for y_1^*

Term	Effect	Coef	SE Coef	T	P
Constant		5.0625	0.1561	32.43	0.000
C	-1.1250	-0.5625	0.1561	-3.60	0.001
D	-1.0000	-0.5000	0.1561	-3.20	0.004
E	1.1250	0.5625	0.1561	3.60	0.001
H	1.0000	0.5000	0.1561	3.20	0.004
J	1.1250	0.5625	0.1561	3.60	0.001
C*D	1.0000	0.5000	0.1561	3.20	0.004

Figure 6.10: Estimated Effects and Coefficients for y_1^*

The computer processing time was recorded to show the relatively quick solutions for a single sensor in the MATLAB program. As indicated in the normal probability plot, all the main factors affect the processing time (see fig. 6.11). The most significant factors are the $\Delta\theta$ and $\Delta\delta$ (factors F and G, respectively). A higher increment value will result in a short processing time (see figure 6.12). This makes sense since smaller increments will results in more iterations of the code for more values of the heading and offsets. There are also many significant two-way interactions. The heading and offset pdfs (factors AD and CD, respectively) that are non uniform increase the processing time since the program requires more calculations for the triangular distributions than the uniform distribution. Although the results indicate that an interaction between the pdfs and the acquisition radius would affect the processing time, logically this interaction should have no

impact on the processing time since the pdfs and the radius do not interact in the program. The results may have shown the interaction is significant due to confounding in the experimental design, and would require more experiments to be conducted to determine if indeed the interaction is significant.

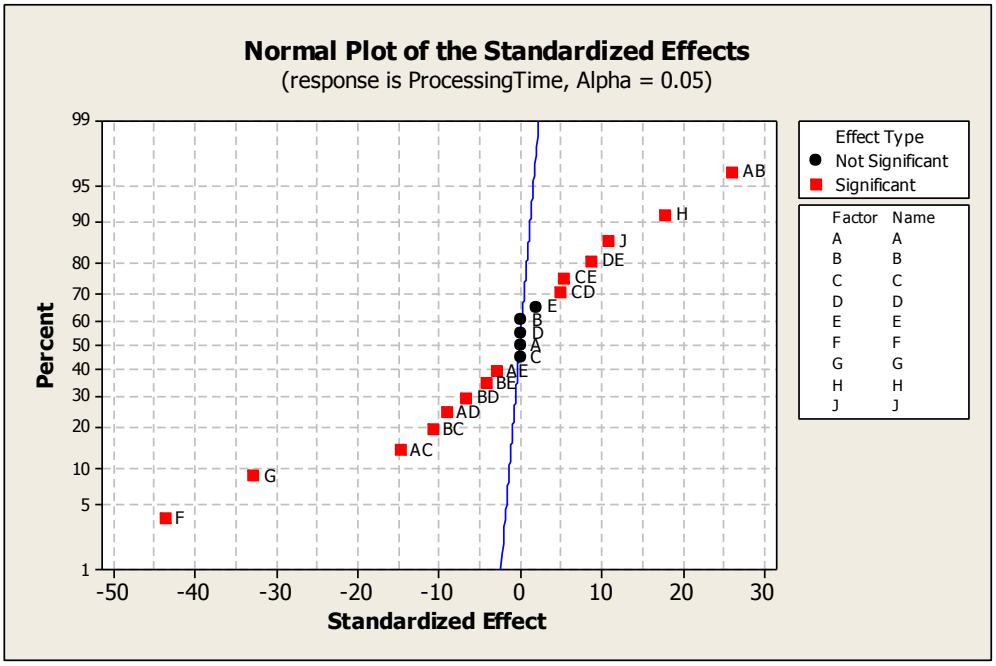

Figure 6.11: Normal Probability Plot for Processing Time

Term	Effect	Coef	SE Coef	T	P
Constant		497.9	9.085	54.80	0.000
A	1.0	0.5	9.085	0.06	0.955
B	3.3	1.6	9.085	0.18	0.861
C	0.6	0.3	9.085	0.03	0.973
D	3.1	1.6	9.085	0.17	0.866
E	38.1	19.1	9.085	2.10	0.058
F	-792.3	-396.2	9.085	-43.61	0.000
G	-596.4	-298.2	9.085	-32.82	0.000
H	326.2	163.1	9.085	17.95	0.000
J	199.6	99.8	9.085	10.98	0.000
A*B	474.8	237.4	9.085	26.13	0.000
A*C	-267.1	-133.5	9.085	-14.70	0.000
A*D	-163.8	-81.9	9.085	-9.01	0.000
A*E	-52.9	-26.4	9.085	-2.91	0.013
B*C	-195.1	-97.5	9.085	-10.74	0.000
B*D	-119.3	-59.6	9.085	-6.56	0.000
B*E	-75.6	-37.8	9.085	-4.16	0.001
C*D	91.0	45.5	9.085	5.01	0.000
C*E	97.9	48.9	9.085	5.39	0.000
D*E	159.4	79.7	9.085	8.77	0.000

Figure 6.12: Estimated Effects and Coefficients for Processing Time

The results from the DOE analysis show how the various factors affect the four responses. The results also support the MATLAB program, showing that the program is running properly.

6.3 Results from Experiments of Multiple Sensors

We consider a set of five distinct two-sensor DSN that include a variety of sensor combinations. Table 6.3 summarizes the DSN under consideration in our example. Homogenous and heterogeneous DSN are analyzed, as well as the order in which the sensors are placed. The acquisition radii for the passive and acoustic sensors are reasonable values according to the calculations from §4.5. Hydrodynamic and electromagnetic sensors were not considered since their footprints are difficult to determine. However, some discussion on the costs of EM sensors are included in §6.4. The MATLAB program was run for each DSN. The Q^* for the DSN are recorded in table 6.4 along with the x_i^*, and y_i^* for each sensor.

For all DSN, the length and width of the area of interest are each 100nm, $\Delta\delta$ = 1nm, $\Delta\theta$ = 0.03 radians, T = 4 hours, and v =15 nm/h (knots). The pdfs for the heading values is uniform, while the offset pdf is a triangular distribution with C = 50, indicating that the probabilities are symmetric over $\delta = 50$.

Table 6.3: Two-sensor DSN Under Consideration for Example Problem

DSN	Sensor 1 Type	Sensor 2 Type
1	Passive ($R_i = 15$)	Passive ($R_i = 15$)
2a	Passive ($R_i = 10$)	Passive ($R_i = 15$)
2b	Passive ($R_i = 15$)	Passive ($R_i = 10$)
3	Active ($R_i = 10$)	Active ($R_i = 10$)
4a	Active ($R_i = 10$)	Active ($R_i = 12$)
4b	Active ($R_i = 12$)	Active ($R_i = 10$)
5a	Active ($R_i = 10$)	Passive ($R_i = 15$)
5b	Passive ($R_i = 15$)	Active ($R_i = 10$)

6.3.1 Complete Results of DSN #1 with Plots

The MATLAB plots for the DSN #1 are shown below. Figure 6.13 shows the probability of occurrence matrix of the pmf formed from the uniform pdf for the headings and the triangular distribution for the offset values.

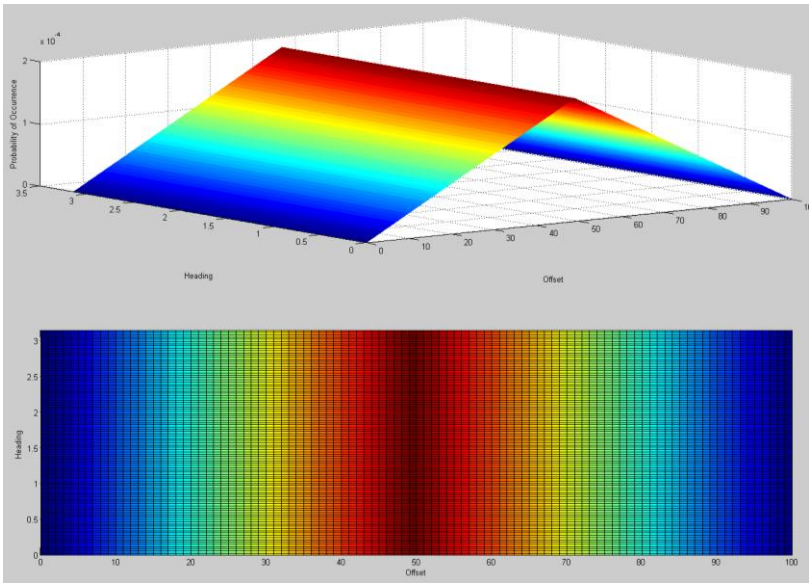

Figure 6.13: MATLAB 3D and Pseudocolor Plots for the Probability of Occurrence Matrix for DSN #1

Using this original $\boldsymbol{O}$ matrix, the first sensor is placed at (49, 5), and has a Q_1^* = 0.63971. Its coverage function is shown in figure 6.14. The $\boldsymbol{O}$ matrix is transformed and the second sensor is placed at (79, 5), and has a Q_2^* = 0.0.18269. Its coverage function is displayed in figure 6.15. Notice that the second sensor's CF covers some of the heading and offset combinations that were not covered by sensor 1.

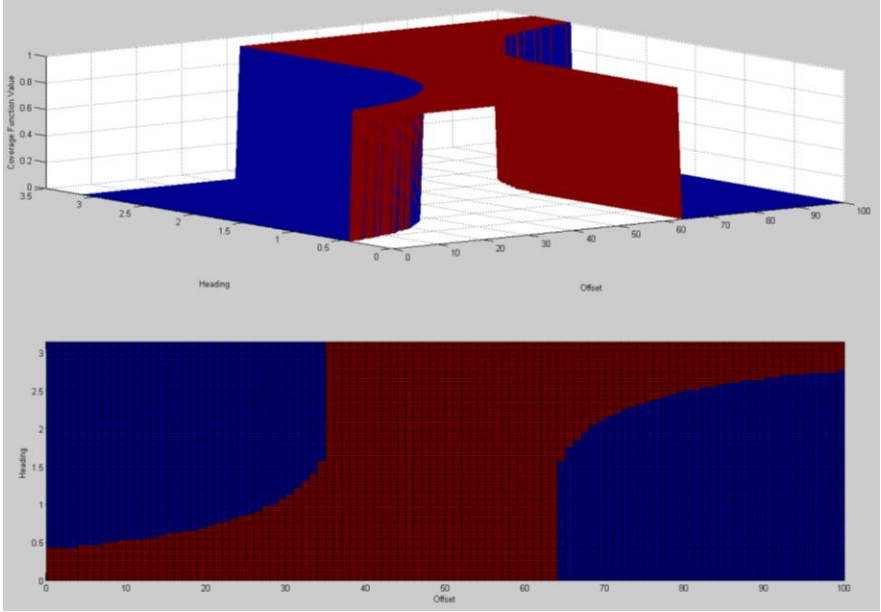

Figure 6.14: MATLAB 3D and pseudocolor plots for the CF of a sensor 1positioned at (49, 5)

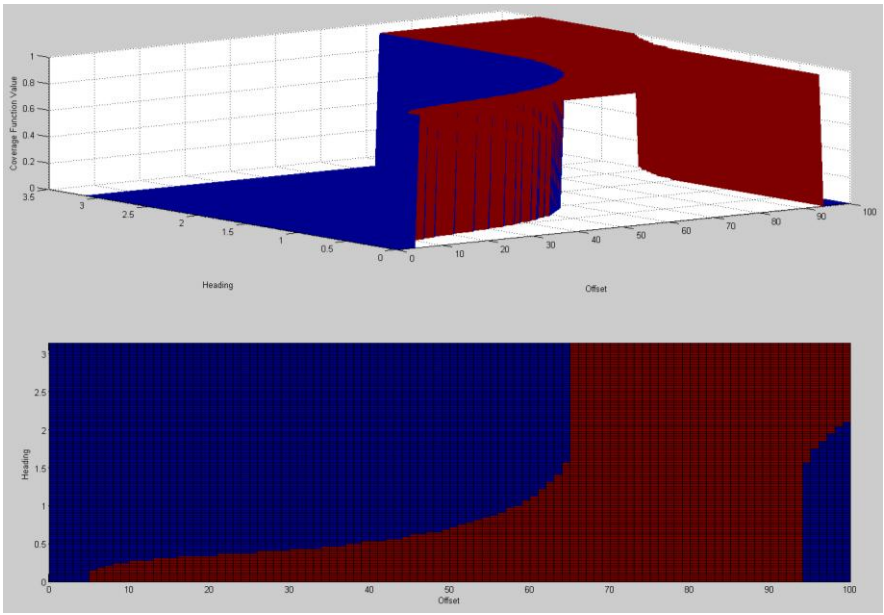

Figure 6.15: MATLAB 3D and pseudocolor plots for the CF of a sensor 2 positioned at (79, 5)

The ***TCF*** of the union between the two coverage functions is shown in figure 6.16. This coverage function is then used to calculate the $\boldsymbol{Q}$ matrix that is used to determine Q_i (see fig. 6.17). The Frobenius inner product of $\boldsymbol{Q}$ is Q_m^* = 0.8224. Notice that $Q_m^* = Q_1^* + Q_2^*$ since Q_2^* is calculated with the transformed $\boldsymbol{O}$ matrix, which has zeros for the heading and offset

combinations that are already covered by sensor 1. Therefore, Q_2^* fundamentally only is calculating the sum of the probabilities of occurrence for combinations that are only covered by sensor 2.

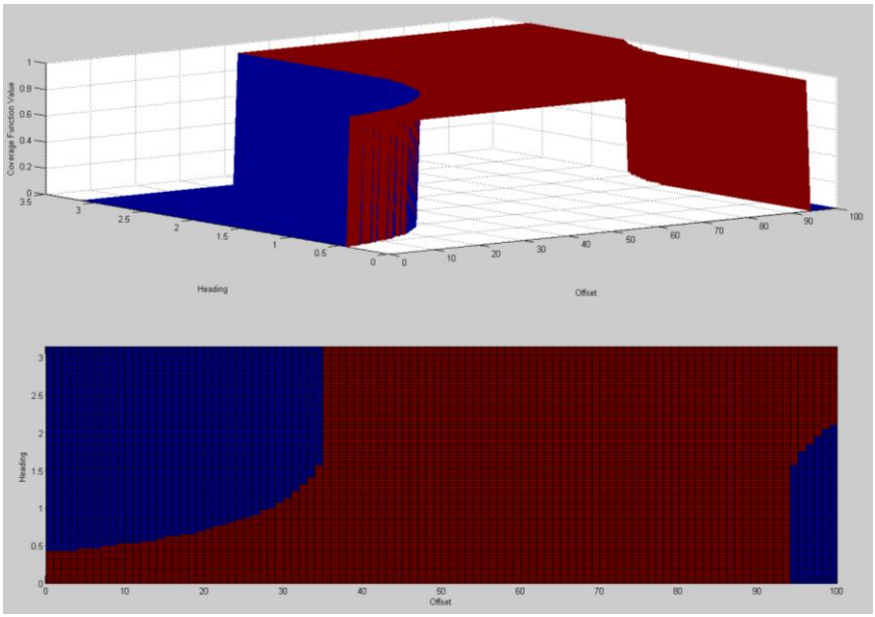

Figure 6.16: MATLAB 3D and pseudocolor plots for the TCF for DSN #1

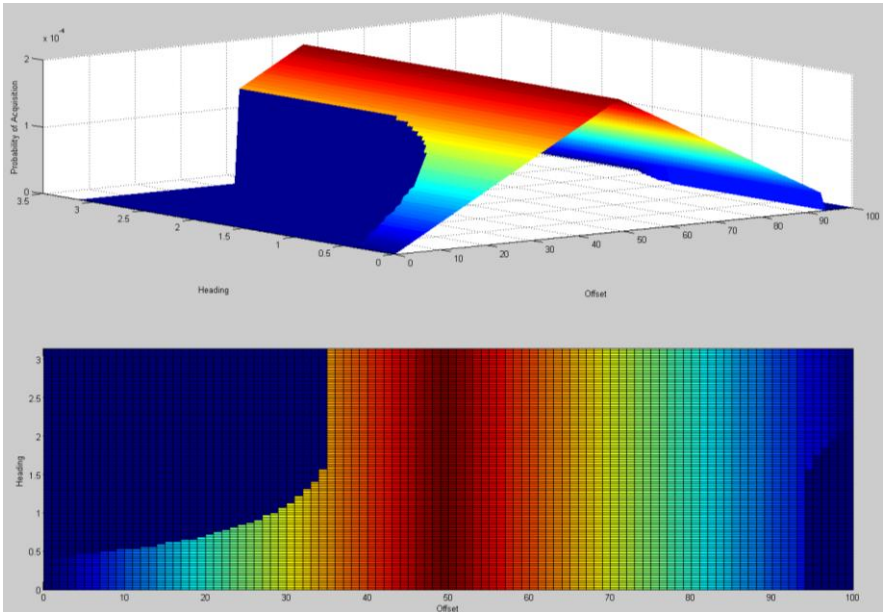

Figure 6.17: MATLAB 3D and Pseudocolor Plots for the $\boldsymbol{Q}$ Matrix

6.3.2 Summary of Results

All five DSN were analyzed using the MATLAB program. Heterogeneous DSN were tested twice to determine if the order in which the sensors are placed had any effect on the performance of the DSN. Results are shown in table 6.4 and the Q^* values are plotted for the Pareto analysis in §6.5. On average, a single run for a DSN took 65 seconds.

Table 6.4: Results for all DSN from the MATLAB Program

DSN (m)	Q_m^*	(x_i^*, y_i^*) for Sensor 1	(x_i^*, y_i^*) for Sensor 2
1	0.8224	(49, 5)	(79, 5)
2a	0.74428	(49, 4)	(75, 5)
2b	0.79638	(49, 5)	(77, 4)
3	0.70407	(49, 4)	(72, 4)
4a	0.712315	(49, 4)	(73, 4)
4b	0.74255	(49, 4)	(74, 4)
5a	0.74428	(49, 4)	(75, 5)
5b	0.79638	(49, 5)	(77, 4)

Some interesting results were found in the experiments. For DSN #2b, the union of the CF_1 (fig. 6.14) and CF_2 (fig. 6.18) results in a small gap of coverage for a few offset values (see fig. 6.19). This gap occurs because the gain of covering the small set of combinations in the gap is less than the cost of losing coverage over other areas. If a different position for sensor 2 was chosen that shifted the coverage function to the left, the gap would be close, but the number of combinations that are not covered to the right of CF_2 is greater than the number of combinations in the gap. Note that the CF_1 for DSN#2b is the same as that for sensor 1 in DSN #1.

The order in which the sensors are placed does affect the performance of the DSN, which indicates that simultaneously placing sensors should result in a higher performance. The better performance occurred when the sensor with the higher acquisition radius was placed first.

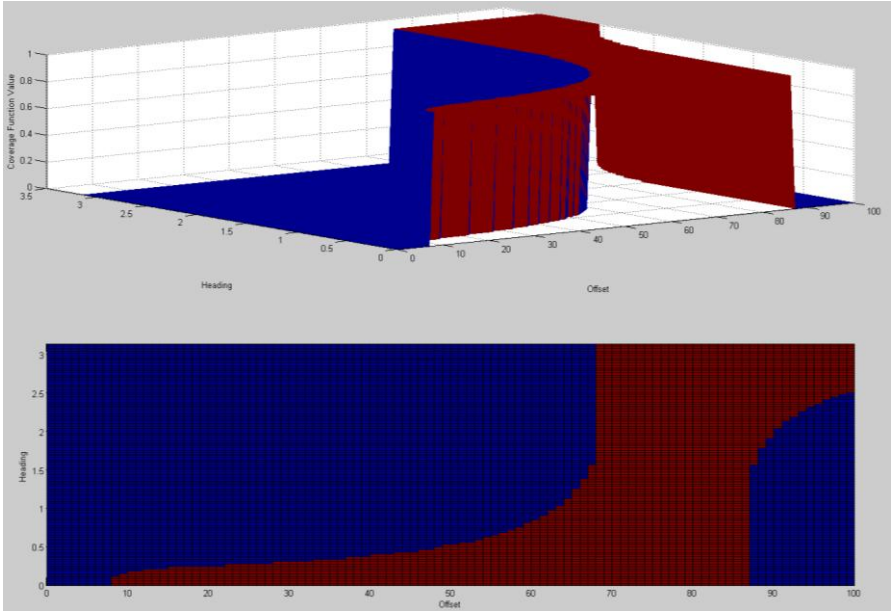

Figure 6.18: MATLAB 3D and pseudocolor plots for the CF of a sensor 2 positioned at (77, 4)

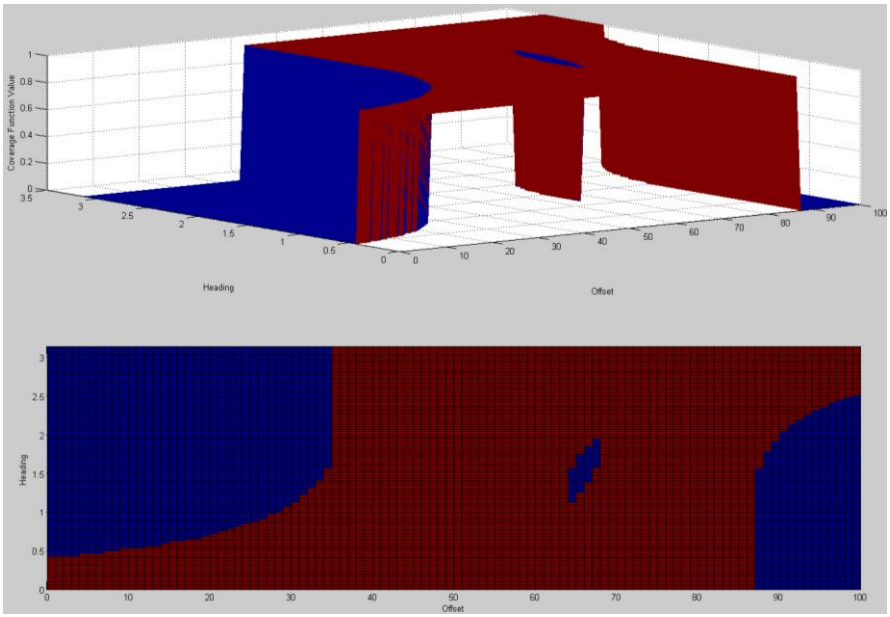

Figure 6.19: MATLAB 3D and pseudocolor plots for the TCF for DSN #2b

Another interesting observation is that all the first sensors were placed in the same x coordinate, since is depended more on the offset pdf than the size of the sensor's acquisition footprint. The first sensors are placed slightly to the left of the midpoint of the area of interest, which results in a coverage function that has a few more heading and offset combinations uncovered on the right

side of the **CF** matrix. Therefore, the program places the second sensor in a position to the right of the first sensor to cover a higher number of uncovered combinations.

The Q_m^* values that are determined by the MATLAB program is then used in conjunction with the lifecycle cost for the DSN to determine the optimal DSN. For the heterogeneous DSN, the larger Q_m^* value from the two experiments for that DSN are used in the Pareto analysis.

6.4 Discussion on Lifecycle Cost

The lifecycle cost of a DSN has been simplified in this thesis to include a base cost and a sensing cost. The base cost is the cost associated with purchasing the sensor and is dependent on the type, size, and mobility of the sensor. The two-sensor DSN under analysis in this thesis example are all stationary sensors, such as sonobuoys, but vary in type and size. A common passive sensor is the AN/SSQ-53 directional frequency analysis and recording (DIFAR) sonobuoy, which cost $357 in 1990, which is equivalent to $580 today. An active sonobuoy, such as the AN/SSQ-62B directional command activated sonobuoy system (DICASS), cost significantly more at $1850 at present ($1100 in 1990) [30, 44].

The sensing cost for the sensors includes the cost of deployment, energy consumption, maintenance, and retrieval. The author in [45] estimates that an underwater sensor can cost over $3000 for manufacturing, deployment, maintenance, and retrieval. Deployment for sonobuoys typically consists of dropping the device from an aircraft. Since all the sensors in the DSN for this thesis example are all sonobuoys, we assume that the deployment cost is the same for all DSN. The energy consumption is dependent on the size and type of sensor. Larger sensors will consume more energy, and since the patrol time is the same for all DSN in our example, the larger sensors require more fuel to last the entirety of the patrol time. Active sonar typically will use more energy than passive sonar since they continuously radiate a sound. Information on the energy consumption of electromagnetic and hydrodynamic sensors are still limited, but are assumed to be higher than that of acoustic sonar. Sonobuoys are one-time use sensors, so there is no maintenance cost for our example. The sonobuoys must be retrieved, but since all of the DSN have the same number of sensors, this cost is the same for all the DSN in our example.

The lifecycle costs for the five DSN in our example are summarized in table 6.5. The values from [44] were used as benchmarks for the base costs of the acoustic sensors.. An energy cost of $5/lb/hr is used for the acoustic sensors. The passive sonobuoy AN/SSQ-53D is 22 lbs, while the active sonobuoy AN/SSQ -62B is 34lb [44]. It is assumed that a sensor with a larger radius increases in wfive by a rate of 2lb/5nm. For some perspective, an EM sensor, such as the QMax EM3, can cost and a value of $125,000 and weighs 500 lbs [46].

The sensors of the same type, but different radii have a slightly higher base cost since the sensor with the larger radius is assumed to be larger in size as well, and the sensing cost is increased slightly due to higher energy consumption associated with the larger sized device.

Table 6.5: Lifecycle Costs for Sensors

DSN	Sensor 1 Base Cost ($)	Sensor 1 Sensing Cost ($)	Sensor 1 Total LC Cost ($)	Sensor 2 Base Cost ($)	Sensor 2 Sensing Cost ($)	Sensor 2 Total LC Cost ($)	Total LC for DSN ($)
1	650	120	770	650	120	770	1,540
2	580	110	690	650	120	770	1,460
3	1,850	170	2,020	1,850	170	2,020	4,040
4	2,200	175	2,375	1,850	170	2,020	4,395
5	1,850	170	2,020	650	120	770	2,790

6.5 A Pareto analysis of Two Sensor DSN

Data from tables 6.4 and 6.5 were plotted on the Q_m^* vs. LC graph (see fig. 6.20) and a Pareto analysis was performed. The Pareto frontier line is just an arbitrary curve. Since we want to maximize Q_m^* and minimize cost, we look towards the top left corner of the graph for the optimal DSN. From figure 6.20, DSN#3,4, and 5 cost more than the DSN #1 but do not provide more coverage than DSN#1 or 2. Therefore, all these DSN are strictly dominated by DSN #1 and #2. Between these two remaining possible DSN, the cost difference is $80, and the added Q_m^* is 0.05302. Since the change in cost is small, it is suggested to select DSN #1 as the best system.

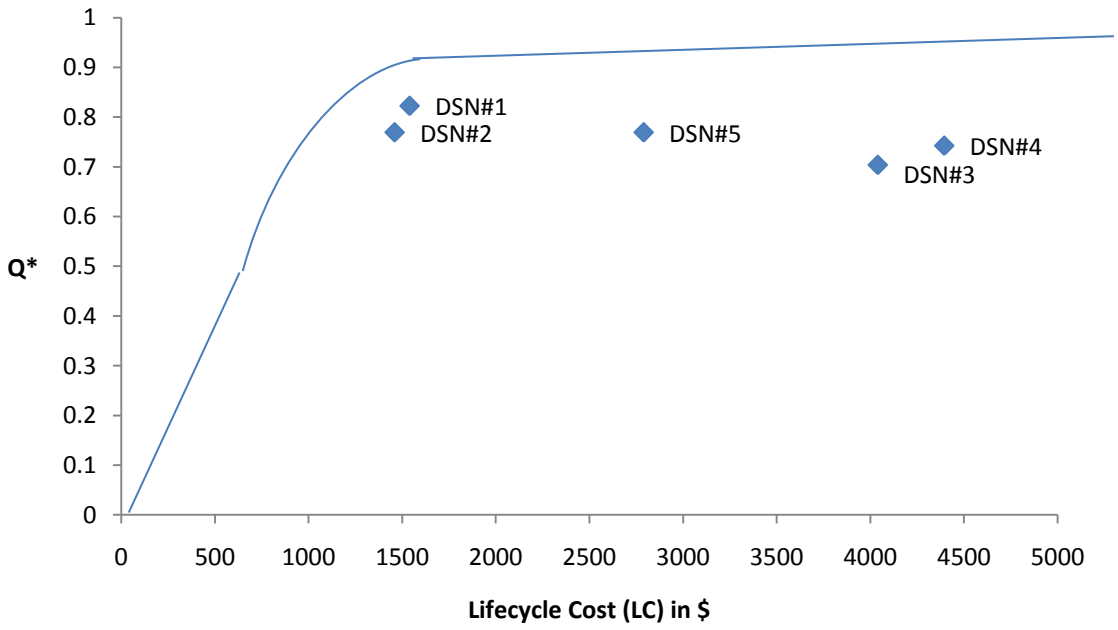

Figure 6.20: Q_m^* vs. LC Graph for Pareto Analysis

Chapter 7: Conclusions and Future Research

This chapter discusses the contributions that the research has in the fields of undersea surveillance and sensor networks. The chapter concludes with a discussion on the future research associated with the work performed in this thesis.

7.1 Contributions

The work presented in this thesis introduces a high-level view of DSN and provides a new methodology to determine sensor location, given a pmf describing the probability that a particular combination of heading and offset values will occur. A MATLAB program was built incorporating the geometric approach for the discrete case to analyze one or two sensors. A basic example was explained in the continuous case.

A DOE was performed to determine the significant factors that affect the probability of acquisition for a single sensor, the positioning of that sensor, and the computer processing time. Results supported concepts discussed in this thesis. A set of two-sensor DSN were analyzed and compared to one another using a Pareto analysis in order to determine the optimal DSN for the given environment and submerged contacts.

7.2 Future Research

In the future, it will be beneficial to extend the geometric approach to N sensors and coding the MATLAB program to handle N sensors using the methodology presented in §5.2. Also, the continuous case can be developed more in depth for other distributions. The coverage functions of mobile sensors need to be developed and take into account their paths over the patrol time. In the future, the submerged contact tracks may be able to employ tactics instead of assuming constant speed and heading.

References

[1]. Urick, Robert J. *Principles of Underwater Sound.* 3rd ed. New York, NY: McGraw-Hill., 1983. Print.

[2] Chong, Chee-yee, and Srikanta P. Kumar. "Sensor Networks: Evolution, Opportunities, and Challenges." *Proceedings of the IEEE* 91.8 (2003): 1247-256. *IEEEXplore*. Web. 2 Mar. 2011.

[3] Eaglen, Mackenzie, and Jon Rodeback. "Submarine Arms Race in the Pacific: The Chinese Challenge to U.S. Undersea Supremacy." *Backgrounder* 2367 (2010). *The Heritage Foundation*. The Heritage Foundation, 2 Feb. 2010. Web. 10 Apr. 2010. <http://www.heritage.org/research/reports/2010/02/submarine-arms-race-in-the-pacific-the-chinese-challenge-to-us-undersea-supremacy>.

[4] Grund, M., Freitag, L., Preisig, J.; Ball, K., "The PLUSNet Underwater Communications System: Acoustic Telemetry for Undersea Surveillance." *OCEANS 2006* (2006): 1-5. *IEEEXplore.*Web. 13 Jan. 2010.

[5] Klamper, Amy. "ONR and Academia Devise PLUSNet, and Autonomous Web of Underwater Robots." *SEA POWER*. Navy League of the United States, Feb. 2007. Web. 25 Jan. 2010. <http://www.seapower-digital.com/seapower/200702/?pg=18#pg18>.

[6] Coraluppi, Stefano. "Multistatic Sonar Localization." *IEEE Journal of Oceanic Engineering*. 31.4 (2006): 964-974. *IEEEXplore.* Web. 1 Sept. 2010.

[7] Shima, T., S. Rasmussen, A. Sparks, and K. Passino. "Multiple Task Assignments for Cooperating Uninhabited Aerial Vehicles Using Genetic Algorithms." *Computers & Operations Research* 33.11 (2006): 3252-269. *SciVerse Science Direct*. Elsevier B.V Web. 22 Nov. 2010.

[8] Li, Dan, K.D. Wong, Yu Hen Hu, and A.M. Sayeed. "Detection, Classification, and Tracking of Targets." *Signal Processing Magazine, IEEE* 19.2 (2002): 17-29. *IEEEXplore*. Web. 12 Mar. 2010.

[9] Urick, Robert J. *Signal and noise at two threshold settings.* Drawing. Urick, Robert J. *Principles of Underwater Sound. 3rd ed.* New York, NY: McGraw-Hill, Inc., 1983, 381.

[10] Ginzburg, B., L. Frumkis, B.Z. Kaplan, A. Sheinker, and N. Salomonski. "Investigation of Advanced Data Processing Technique in Magnetic Anomaly Detection Systems." *International Journal on Smart Sensing and Intelligent Systems* 1.1 (2008): 110-22. Web. 5 Feb. 2010.

[11] Kirkebo, J.E., Austeng, A., and S. Holm. "Layout-optimized cylindrical sonar arrays," *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04* 2 (2004): 598- 602. *IEEEXplore.* Web. 30 Sept. 2010.

[12] *Active Sonar Diagram*. Digital image. *The National Archives*. Ministry of Defence, 10 Oct. 2005. Web. 12 Mar. 2011. <http://webarchive.nationalarchives.gov.uk/+/http://www.mod.uk/dpa/projects/sonarenvir/sonar_types.htm>.

[13] Baldacci, Alberto, and Georgios Haralabus. "Signal Processing for an Active Sonar System Suitable for Advanced Sensor Technology Applications and Environmental Adaptation Schemes." Proc. of European Signal Processing Conference, Florence, Italy. 2006. Web. 23 Feb. 2010. <http://www.nurc.nato.int/publications/pubs/2006/NURC-PR-2006-019.pdf>.

[14] Dawe, Ross L. *Detection Threshold Modelling Explained*. Tech. Australian Government Department of Defense: Defence Science and Technology Organisation. Web. 25 Jan. 2010. <http://hdl.handle.net/1947/4252>.

[15] *Passive Sonar Diagram*. Digital image. *The National Archives*. Ministry of Defence, 10 Oct. 2005. Web. 12 Mar. 2011. <http://webarchive.nationalarchives.gov.uk/+/http://www.mod.uk/dpa/projects/sonarenvir/sonar_types.htm>.

[16] Mattos, Leonardo, and Edward Grant. "Passive Sonar Applications: Target Tracking and Navigation of an Autonomous Robot." *Proceedings of the 2004 IEEE IEEE International Conference on Robotics & Automation, New Orleans, LA, April 2004*. Vol. 5: 4265-270. *IEEE Xplore*. Web. 8 Feb. 2010.

[17] Lanbo, Liu, Shengli, Zhou, and Cui Jun-Hong. "Prospects and Problems of Wireless Communication for Underwater Sensor Networks." *ACM Digital Library*. Web. 16 Mar. 2011. <http://portal.acm.org/citation.cfm?id=1416622>.

[18] Krichenko, Oleg, Sternberg, Ben K. ,and Steven L. Dvorak. "A New High-Sensitivity Subsurface Electromagnetic Sensing System: Part II--Measurement Results." *Journal of Environmental & Engineering Geophysics* 13.3 (2008): 263-75. Web. 14 Mar. 2011.

[19] Frater, Michael R., Michael J. Ryan, and Robin M. Dunbar. "Electromagnetic Communications within Swarms of Autonomous Underwater Vehicles." *WUWNet '06 Proceedings of the 1st ACM International Workshop on Underwater Networks*. Los Angeles, California. ACM. Web. 5 Apr. 2010. <http://portal.acm.org/citation.cfm?id=1161053>.

[20] "Magnetic Anomaly Detection System." *CAE*. CAE. Web. 14 Mar. 2011. <http://www.cae.com/en/military/magnetic.anomaly.detection.asp>.

[21] Kiehn, Robert M. "Hydrodynamic Wakes and Minimal Surfaces with Fractal Boundaries." *CiteSeerX*. The College of Information Sciences and Technology at Penn State. Web. 4 Feb. 2010. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.8138>.

[22] *Schematic of key hydrodynamic mechanisms in the wake of a typical submarine.* Digital image. *Hydrodynamic and Naval Systems.* Continuum Dynamics, Inc. 12 Mar. 2011. <http://www.continuum-dynamics.com/solution-hn-overview.html>.

[23] Osborn, Thomas R. and Rolf G. Lueck. "Turbulence Measurements with a Submarine." *Journal of Physical Oceanography* 15.11 (1985): 1502-520. Web. 14 Sept. 2010.

[24] United States. Naval Research Laboratory. Office of Naval Research. *DTIC: Online Information for the Defense Community*. By Richard A. Skop. DTIC. Web. 4 Feb. 2010. <http://handle.dtic.mil/100.2/ADB084604>.

[25] Osborn, T., D.M. Farmer, S. Vagle, S.A. Thorpe, and M. Cure. "Measurements of Bubble Plumes and Turbulence From a Submarine." *Atmosphere-Ocean* (1992): 419-40. Web. 14 Sept. 2010.

[26] Yang, Yingchen, Jack Chen, Jonathan Engel, Saunvit Pandya, Nannan Chen, Craig Tucker, Sheryl Coombs, Douglas L. Jones, and Chang Liu. "Distant Touch Hydrodynamic Imaging with an Artificial Lateral Line." *Proceedings of the National Academy of Sciences* 103.50 (2006): 18891-8895. Web. 4 Feb. 2010.

[27] Baumgartner, K. and S. Ferrari. "Optimal Placement of a Moving Sensor Network for Track Coverage," *American Control Conference, 2007. ACC '07*. (2007): 4040-4046. *IEEEXplore.* Web. 25 Jan. 2010.

[28] Fiorelli, Edward, Naomi Ehrich Leonard, Pradeep Bhatta, Derek A. Paley, Ralf Bachmayer, and David M. Fratantoni. "Multi-AUV Control and Adaptive Sampling in Monterey Bay." *IEEE Journal of Oceanic Engineering* 31.4 (2006): 935-48. *IEEXplore*. Web. 25 Jan. 2010.

[29] Paley, Derek A., Fumin Zhang, and Naomi Ehrich Leonard. "Cooperative Control for Ocean Sampling: The Glider Coordinated Control System." *IEEE Transactions on Control Systems Technology* 16.4 (2008): 735-44. *IEEEXplore*. Web. 25 Jan. 2010.

[30] Wolf, George W. Jr. "U.S. Navy Sonobuoys--key to Antisubmarine Warfare." *BNET*. CBS Interactive, Nov. 1998. Web. 23 Sept. 2010. <http://findarticles.com/p/articles/mi_qa5367/is_199811/ai_n21430235/pg_3/>.

[31] Erdinc, Ozgur, Peter Willett, and Stefano Coraluppi. "Multistatic Sensor Placement: A Tracking Approach." *Information Fusion, 2006 9th International Conference*: 1-8. *IEEEXplore*. Web. 1 Sept. 2010.

[32] Heidemann, John, Wei Ye, Jack Wills, Affan Syed, and Yuan Li. "Research Challenges and Applications for Underwater Sensor Networking." *Wireless Communication and Networking Conference*. Las Vegas, NV. 2006. 228-35. *IEEEXplore*. Web. 5 Feb. 2010.

[33] Hirsch, Michael J., Rakesh Nagi, and David Sudit. "A Stochastic Optimization Framework for Resource Management and Course of Action Analysis." Proc. of Information Fusion, 2008 11th International Conference on, Cologne, Germany. 2008. 1-7. *IEEEXplore*. Web. 22 Nov. 2010.

[34] Ferrari, S. "Track coverage in sensor networks," *Proceedings of the 2006 American Control Conference.* (2006): 2053-2059. *IEEEXplore.* Web. 25 Jan. 2010.

[35] Huang, Chi-Fu, and Yu-Chee Tseng. "The Coverage Problem in a Wireless Sensor Network." *Mobile Networks and Applications* 10.4 (2005): 519-28. *ACM Digital Library*. Web. 22 Nov. 2010.

[36] Meguerdichian, Seapahn, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. "Coverage Problems in Wireless Ad-hoc Sensor Networks." *IEEE INFROCOM* (2001): 1380-387. *CiteSeerX*. The College of Information Sciences and Technology at Penn State. Web. 23 Nov. 2010.

[37] Cortes, J., Martinez, S., Karatas, T., Bullo, F. "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation.* 20.2 (2004): 1327-1332. *IEEEXplore.* Web. 8 Feb. 2010.

[38] Elfes, A., "Using occupancy grids for mobile robot perception and navigation ," *Computer*. 22.6 (1989): 46-57. *IEEEXplore*. Web. 5 Oct. 2010.

[39] Auren, P.G., and O. Silven. "Underwater Sonar Rangge Sensing and 3D Image Formation." *Control Engineering Practice* 4.3 (1996): 393-400. *SciVerse ScienceDirect*. Elsevier B.V. Web. 28 Mar. 2011.

[40] Thurn, Sebastian. "Learning Occupancy Grid Maps with Forward Sensor Models." *Autonomous Robots* 15.2 (2003): 111-27. *SpringerLink.com*. SpringerLink. Web. 13 Oct. 2010.

[41] Miguel Ribo, Axel Pinz, "A comparison of three uncertainty calculi for building sonar-based occupancy grids." *Robotics and Autonomous Systems.* 35.3-4(2001): 201-209. *SciVerse ScienceDirect*. Elsevier B.V. Web.5 Oct. 2010.

[42] Ferrari, Silvia, Brent Perteet, Chenghui Cai, and Kelli Baumgartner. "A Geometric Optimization Approach to Detecting and Intercepting Dynamic Targets Using a Mobile Sensor Network." *SIAM Journal on Control and Optimization* 48.1 (2009): 292-320. *SIAM*. SIAM-Society for Industrial and Applied Mathematics. Web. 1 Dec. 2010.

[43] Ankenman, B. E. (1999). Design of experiments with two- and four level factors. *Journal of Quality Technology, 31.4*(1999): 363. 10 Apr. 2011.

[44] Friedman, Norman. *The Naval Institute Guide to World Naval Weapon Systems 1997-1998.* Annapolis: U.S. Naval Institute, 1997. Google Book Search. Web 14 Apr. 2011.

[45] Jiang, Zaihan. "Underwater Acoustic Networks - Issues and Solutions." *International Journal of Intelligent Control and Systems* 13.3 (2008): 152-61. Web. 14 Apr. 2011.

[46]Randolph, Ned. "Quasar Rolls Out New Sensor for Undersea Explorations." *San Diego Business Journal*. 22 Jun. 2009. Web 14 Apr. 2011. <http://www.allbusiness.com/science-technology/earth-atmospheric-science/12600586-1.html>.

Appendix A: Proofs for Equations

A.1: Proof of Equation (4.14)

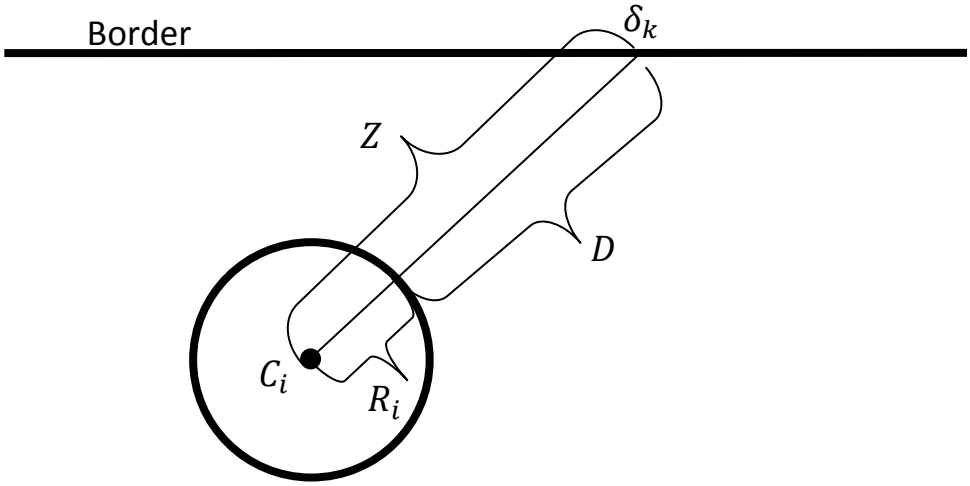

Figure A.1: Geometry for Proof of Equation 4.14

Let point δ_k be located at coordinates $(\delta_k, 0)$ and point C_i be located at coordinates (x_u, y_v). By the Pythagorean Theorem, the length of the line segment $\overline{\delta_k C_i}$, denoted as Z is $\sqrt{y_v^2 + (x_u - \delta_k)^2}$.

A.2: Proof of Statement 1 (Congruent Triangles)

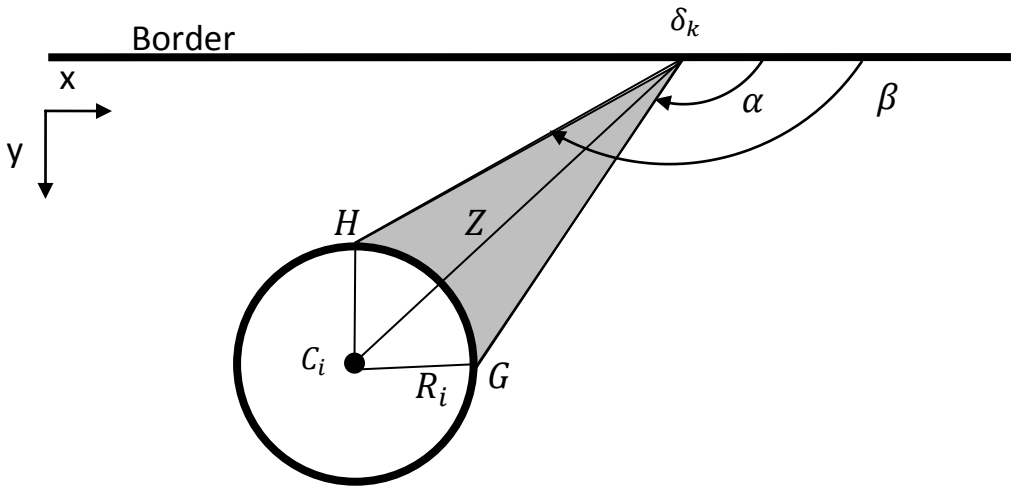

Figure A.2: Geometry for Proof of Statement 1

Let $\Delta\delta_k C_i H$ and $\Delta\delta_k C_i G$ be two triangles formed by the bisection of $\angle H\delta_k G$ (see fig. A.2). Both triangles share the side $\overline{\delta_k C_i}$, and each have a side of length R_i. By the definition of bisection, $\angle H\delta_k C_i$and $\angle H\delta_k C_i$ are equal. Therefore the $\Delta\delta_k C_i H$ and $\Delta\delta_k C_i G$ are congruent triangles.

A.3: Proof of Equation (4.16)

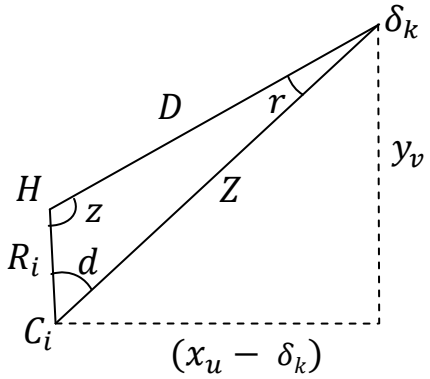

Figure A.3: Geometry for Proofs of Equations 4.16 and 4.17

Let point δ_k be located at coordinates$(\delta_k, 0)$, point C_i be located at coordinates (x_u, y_v), and point H be located at a point a distance of R_i from C_i. Suppose these three points are connected to form a triangle, denoted as $\Delta\delta_k C_i H$.Let the side $\overline{\delta_k C_\iota}$ have length Z, side $\overline{\delta_k H}$ have length D, and side $\overline{HC_\iota}$ have length R_i. Let ∠z be opposite the side $\delta_k C_i$, ∠d be opposite the side $\overline{\delta_k H}$, and ∠r be opposite the side $\overline{HC_\iota}$. The angle z is found by the cosine rule and algebra:

$$Z^2 = R_i^2 + D^2 - 2R_i D * \cos(z) \quad (1)$$

$$2R_i D * \cos(z) = R_i^2 + D^2 - Z^2 \quad (2)$$

$$\cos(z) = \frac{R_i^2 + D^2 - Z^2}{2R_i D} \quad (3)$$

$$z = \cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_i D}\right] \quad (4)$$

A.4: Proof of Equation (4.17)

From the proof in Appendix A.3, $\angle z = \cos^{-1}\left[\frac{R_i^2 + D^2 - Z^2}{2R_i D}\right]$. The angle *r* can then be found using the sine rule and algebra:

$$\frac{R_i}{\sin(r)} = \frac{Z}{\sin(z)} \tag{5}$$

$$\frac{\sin(r)}{R_i} = \frac{\sin(z)}{Z} \tag{6}$$

$$\sin(r) = \frac{R_i \sin(z)}{Z} \tag{7}$$

$$r = \sin^{-1}\left[\frac{R_i \sin(z)}{Z}\right] \tag{8}$$

A.5: Proof of Equation (4.18)

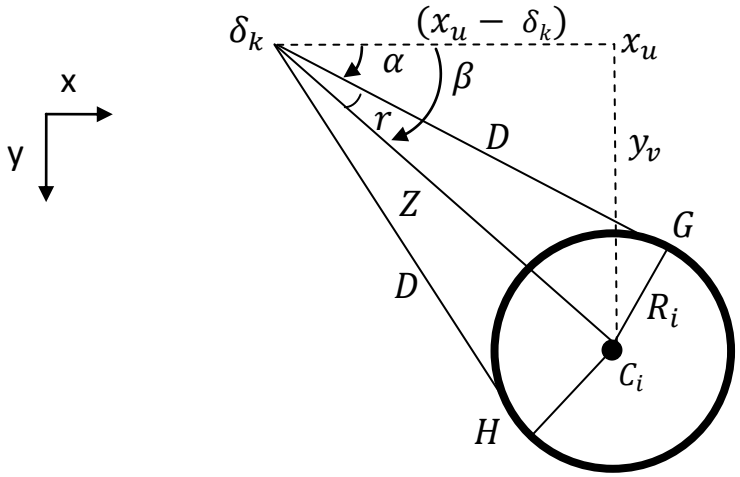

Figure A.4: Geometry for Proofs of Equations 4.18 and 4.19

Let point δ_k be located at coordinates$(\delta_k, 0)$, point C_i be located at coordinates (x_u, y_v), and point x_i be located at $(x_u, 0)$. Suppose these three points are connected to form a triangle (shown by the dotted lines in figure A.4), denoted as $\Delta\delta_k C_i x_u$.Let the side $\overline{\delta_k C_\iota}$ have length Z, side $\overline{\delta_k x_u}$

have length y_v, and side $\overline{x_u C_i}$ have length $(x_u - \delta_k)$. The $\angle C_i \delta_k x_u$ contains the angles α_{iuv} and r; that is $\angle C_i \delta_k x_u. = \alpha_{ik} + r$. By definition, the tangent of an angle is equal to the length of the side opposite the angle divided by the length of the side adjacent to the angle.

Therefore the tangent of the $\angle C_i \delta_k x_u$ is

$$\tan(\angle C_i \delta_k x_u) = \frac{y_v}{(x_u - \delta_k)} \quad (9)$$

By substitution,

$$\tan(\alpha_{iuv} + r) = \frac{y_v}{(x_u - \delta_k)} \quad (10)$$

Then solving for α_{iuv} ,

$$\alpha_{iuv} = \tan^{-1}\left(\frac{y_v}{x_u - \delta_k}\right) - r \quad (11)$$

A.6: Proof of Equation (4.19)

Let $\beta_{iuv} = \angle x_u \delta_k H$ and let $\alpha_{iuv} = \angle x_u \delta_k G$. The $\angle G \delta_k H$ is the opening angle of the coverage cone for the sensor positioned at (x_u, y_v) and the point δ_k. From proof A.2, the coverage cone is bisected into two congruent triangles, $\Delta \delta_k C_i H$ and $\Delta \delta_k C_i G$. Since $\angle G \delta_k C_i$ and $\angle H \delta_k C_i$ are equal, let r denote their measurement.

Therefore, since

$$\angle G \delta_k H = \angle G \delta_k C_i + \angle C_i \delta_k H \quad (12)$$

By substitution,

$$\angle G \delta_k H = r + r = 2r \quad (13)$$

Then, since

$$\angle x_u \delta_k H = \angle x_u \delta_k G + \angle G \delta_k H \quad (14)$$

By substitution,

$$\beta_{iuv} = \alpha_{iuv} + 2r \quad (15)$$

A.7: Proof of Equation (4.20)

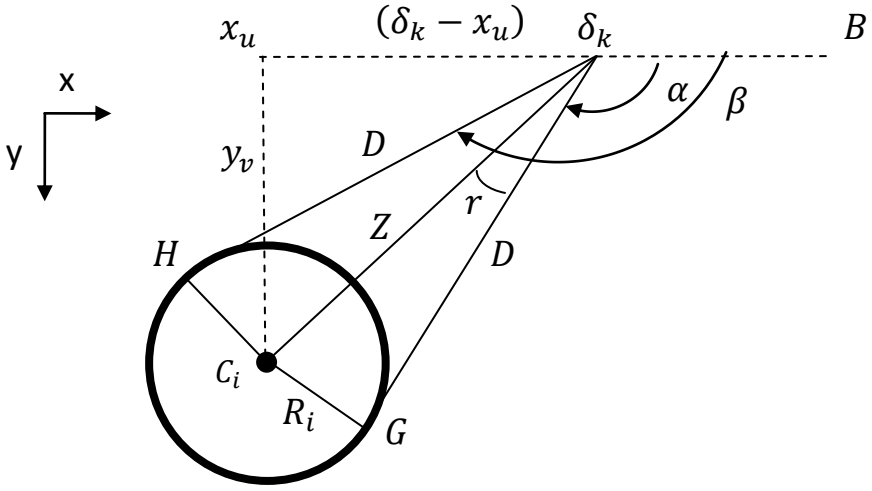

Figure A.5: Geometry for Proof of Equation 4.20

Since $\overline{x_u B}$ is a straight line, $\angle x_u \delta_k B = \pi$. Let $r = \angle C_i \delta_k G$ and $\alpha_{iuv} = \angle G \delta_k B$. By the definition of tangent,

$$\tan(\angle x_u \delta_k C_i) = \frac{y_v}{\delta_k - x_u} \tag{16}$$

$$\angle x_u \delta_k C_i = \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) \tag{17}$$

By the addition of angles,

$$\angle x_u \delta_k B = \angle x_u \delta_k C_i + \angle C_i \delta_k G + \angle G \delta_k B \tag{18}$$

By substitution,

$$\pi = \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) + r + \alpha_{iuv} \tag{19}$$

Solving for α_{iuv},

$$\alpha_{iuv} = \pi - \tan^{-1}\left(\frac{y_v}{\delta_k - x_u}\right) - r \tag{20}$$

A.8: Proof of Equation (4.21)

Let $\delta_k = x_u$. Using equation 20 from proof A.7, we get

$$\alpha_{iuv} = \pi - \tan^{-1}\left(\frac{y_v}{0}\right) - r \tag{21}$$

The arctangent of an undefined value is $\pi/2$. Therefore, by substitution

$$\alpha_{iuv} = \pi - \pi/2 - r = \pi/2 - r \tag{22}$$

A.9: Proof of Equation (4.24)

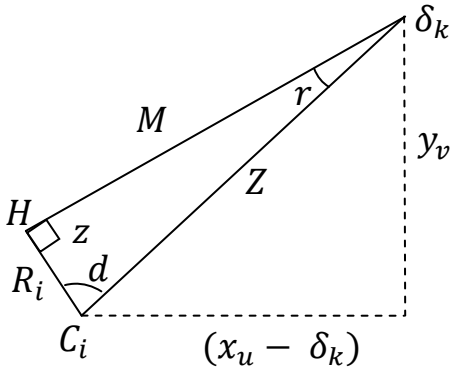

Figure A.6: Geometry for Proofs of Equations 4.24 and 4.25

Let $\Delta\delta_k C_i H$ be a right triangle. The, by the Pythagorean theorem, the length of the line segment $\overline{\delta_k H}$, denoted as M is $\sqrt{Z^2 - R_i^2}$

A.10: Proof of Equation (4.25)

Let $\Delta\delta_k C_i H$ be a right triangle. By the definition of sine,

$$\sin(r) = \frac{R_i}{Z} \tag{23}$$

Solving for r,

$$r = \sin^{-1}\frac{R_i}{Z} \tag{24}$$

Appendix B: Results from Experiments

B.1 Results from Single Sensor Experiments

B.1.1 DOE Design

Table B.1: Experimental Design for a $4^2 2^{5-4}$, or otherwise known as a 2^{9-4}, Design

RunOrder	StdOrder	A	B	C	D	E	F	G	H	J
11	1	-	-	-	-	-	+	+	+	+
5	2	+	-	-	-	-	+	-	-	-
14	3	-	+	-	-	-	-	+	-	-
13	4	+	+	-	-	-	-	-	+	+
20	5	-	-	+	-	-	-	-	+	-
2	6	+	-	+	-	-	-	+	-	+
8	7	-	+	+	-	-	+	-	-	+
26	8	+	+	+	-	-	+	+	+	-
16	9	-	-	-	+	-	-	-	-	+
4	10	+	-	-	+	-	-	+	+	-
12	11	-	+	-	+	-	+	-	+	-
28	12	+	+	-	+	-	+	+	-	+
17	13	-	-	+	+	-	+	+	-	-
10	14	+	-	+	+	-	+	-	+	+
15	15	-	+	+	+	-	-	+	+	+
23	16	+	+	+	+	-	-	-	-	-
27	17	-	-	-	-	+	-	-	-	-
3	18	+	-	-	-	+	-	+	+	+
1	19	-	+	-	-	+	+	-	+	+
30	20	+	+	-	-	+	+	+	-	-
18	21	-	-	+	-	+	+	+	-	+
32	22	+	-	+	-	+	+	-	+	-
6	23	-	+	+	-	+	-	+	+	-
24	24	+	+	+	-	+	-	-	-	+
22	25	-	-	-	+	+	+	+	+	-
19	26	+	-	-	+	+	+	-	-	+
9	27	-	+	-	+	+	-	+	-	+
31	28	+	+	-	+	+	-	-	+	-
7	29	-	-	+	+	+	-	-	+	+
21	30	+	-	+	+	+	-	+	-	-
29	31	-	+	+	+	+	+	-	-	-
25	32	+	+	+	+	+	+	+	+	+

B.1.2 Raw Data Results

Table B.2: Results from the $4^2 2^{5-4}$ Experiment

StdOrder	Q^*	x^*	y^*	Processing Time (seconds)
1	0.34781	48	7	61.4533
2	0.26172	29.5-70.5	5	94.0392
3	0.24762	29-71	4	193.42
4	0.28631	37.5	7	2262.026
5	0.48684	31.5	3	1414.194
6	0.44157	33	4	264.0281
7	0.4252	31	5	126.0804
8	0.45235	29	4	40.7619
9	0.49184	49.5	4	1060.2149
10	0.45477	51	4	352.6645
11	0.43279	49.5	5	164.0512
12	0.45056	47	4	32.0185
13	0.47182	67	4	24.3025
14	0.45795	70.5	5	244.4389
15	0.4312	68	4	571.5725
16	0.43556	66.5	3	756.6166
17	0.39633	34.5-65.5	5	838.0124
18	0.40808	64	10	600.2492
19	0.38231	51	10	256.9355
20	0.36868	34-66	5	26.231
21	0.62414	34	5	34.3591
22	0.59105	36.5	5	174.3793
23	0.5815	33	5	387.0698
24	0.60072	32	5	1168.4944
25	0.63959	49	5	43.8466
26	0.60319	52	5	137.0071
27	0.58549	49	5	290.5038
28	0.60434	47	5	1548.9443
29	0.63036	65.5	5	2389.0686
30	0.5931	67	5	208.3632
31	0.57082	66	5	103.3004
32	0.59384	63	5	64.758

B.1.3 Minitab Results for Q^*

Results after non-significant terms were removed:

Factorial Fit: Qstar versus A, B, C, D, E, H

```
Estimated Effects and Coefficients for Qstar (coded units)

Term        Effect      Coef   SE Coef       T      P
Constant             0.47967  0.002504  191.58  0.000
A         -0.00887  -0.00443  0.002504   -1.77  0.090
B         -0.02818  -0.01409  0.002504   -5.63  0.000
C          0.08916   0.04458  0.002504   17.81  0.000
D          0.09656   0.04828  0.002504   19.28  0.000
E          0.13735   0.06868  0.002504   27.43  0.000
H          0.01330   0.00665  0.002504    2.66  0.014
A*B        0.02580   0.01290  0.002504    5.15  0.000
C*D       -0.09890  -0.04945  0.002504  -19.75  0.000
D*E        0.01193   0.00596  0.002504    2.38  0.026


S = 0.0141631   PRESS = 0.00933666
R-Sq = 98.86%   R-Sq(pred) = 97.59%   R-Sq(adj) = 98.39%


Analysis of Variance for Qstar (coded units)

Source              DF    Seq SS    Adj SS    Adj MS       F      P
Main Effects         6  0.297512  0.297512  0.049585  247.19  0.000
  A                  1  0.000629  0.000629  0.000629    3.14  0.090
  B                  1  0.006353  0.006353  0.006353   31.67  0.000
  C                  1  0.063599  0.063599  0.063599  317.05  0.000
  D                  1  0.074594  0.074594  0.074594  371.87  0.000
  E                  1  0.150924  0.150924  0.150924  752.39  0.000
  H                  1  0.001414  0.001414  0.001414    7.05  0.014
2-Way Interactions   3  0.084714  0.084714  0.028238  140.77  0.000
  A*B                1  0.005323  0.005323  0.005323   26.54  0.000
  C*D                1  0.078253  0.078253  0.078253  390.11  0.000
  D*E                1  0.001138  0.001138  0.001138    5.67  0.026
Residual Error      22  0.004413  0.004413  0.000201
Total               31  0.386640
```

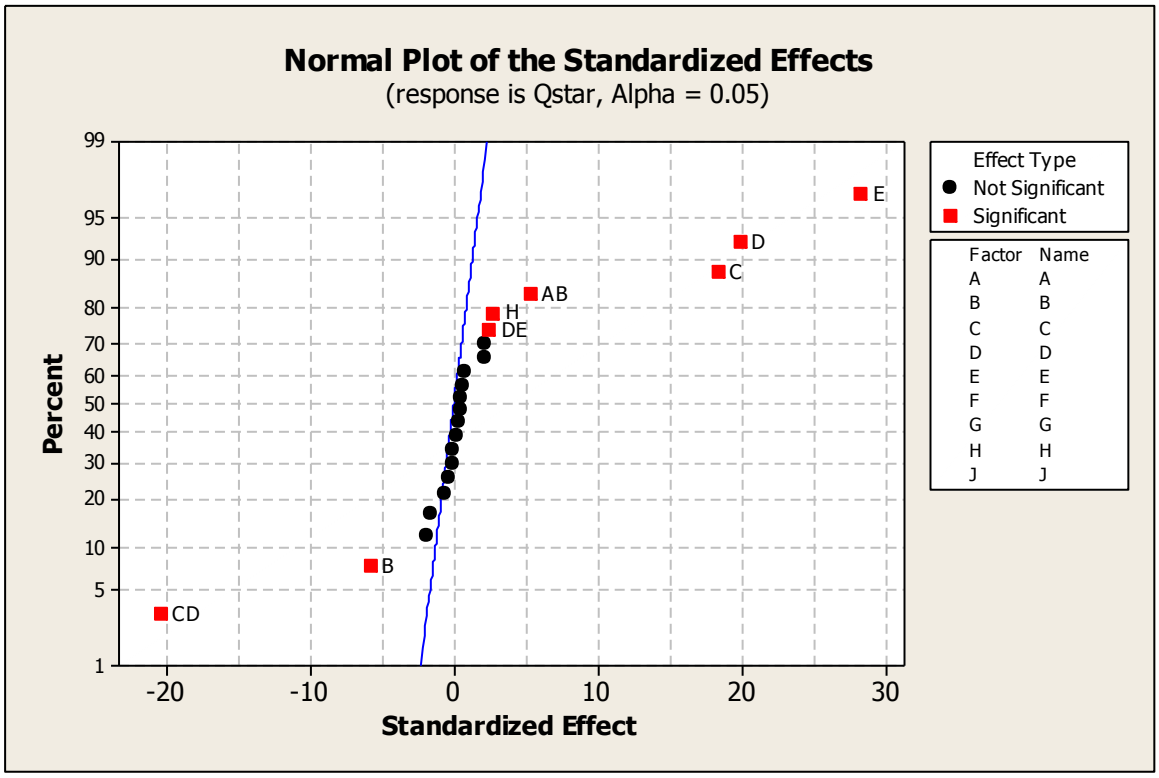

Figure B.1: Normal Probability Plot for all factors for Q_1^* (with no replicates)

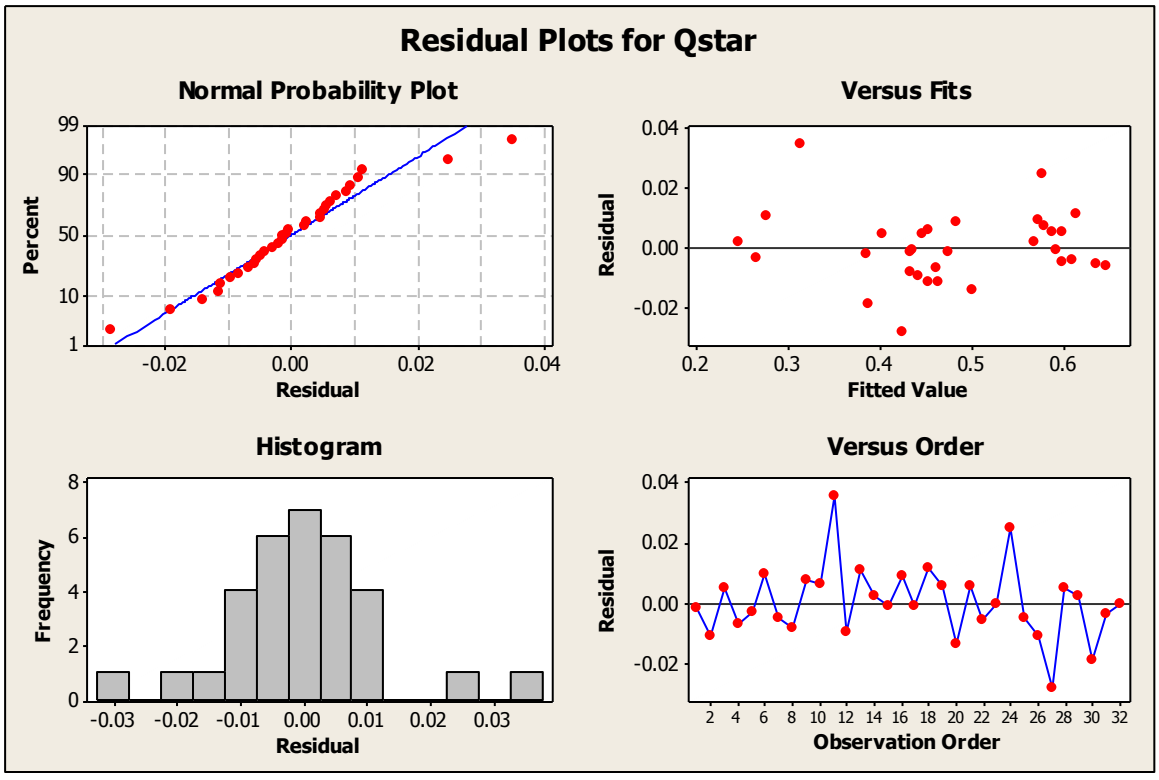

Figure B.2: Residual Plots for Q_1^* (once non-significant factors are removed)

B.1.4 Minitab Results for x^*

Results after non-significant terms were removed:

Factorial Fit: xstar versus C, D, H, J

```
Estimated Effects and Coefficients for xstar (coded units)

Term      Effect    Coef  SE Coef      T      P
Constant          47.344   0.8886  53.28  0.000
C          4.500   2.250   0.8886   2.53  0.018
D         21.250  10.625   0.8886  11.96  0.000
H          4.562   2.281   0.8886   2.57  0.017
J          4.688   2.344   0.8886   2.64  0.014
C*D       12.937   6.469   0.8886   7.28  0.000
C*J       -4.625  -2.312   0.8886  -2.60  0.015


S = 5.02693     PRESS = 1035.06
R-Sq = 89.91%   R-Sq(pred) = 83.46%   R-Sq(adj) = 87.48%


Analysis of Variance for xstar (coded units)

Source              DF  Seq SS  Adj SS   Adj MS       F      P
Main Effects         4  4116.8  4116.8  1029.20   40.73  0.000
  C                  1   162.0   162.0   162.00    6.41  0.018
  D                  1  3612.5  3612.5  3612.50  142.96  0.000
  H                  1   166.5   166.5   166.53    6.59  0.017
  J                  1   175.8   175.8   175.78    6.96  0.014
2-Way Interactions   2  1510.2  1510.2   755.08   29.88  0.000
  C*D                1  1339.0  1339.0  1339.03   52.99  0.000
  C*J                1   171.1   171.1   171.12    6.77  0.015
Residual Error      25   631.7   631.7    25.27
  Lack of Fit        1   162.0   162.0   162.00    8.28  0.008
  Pure Error        24   469.8   469.8    19.57
Total               31  6258.7
```

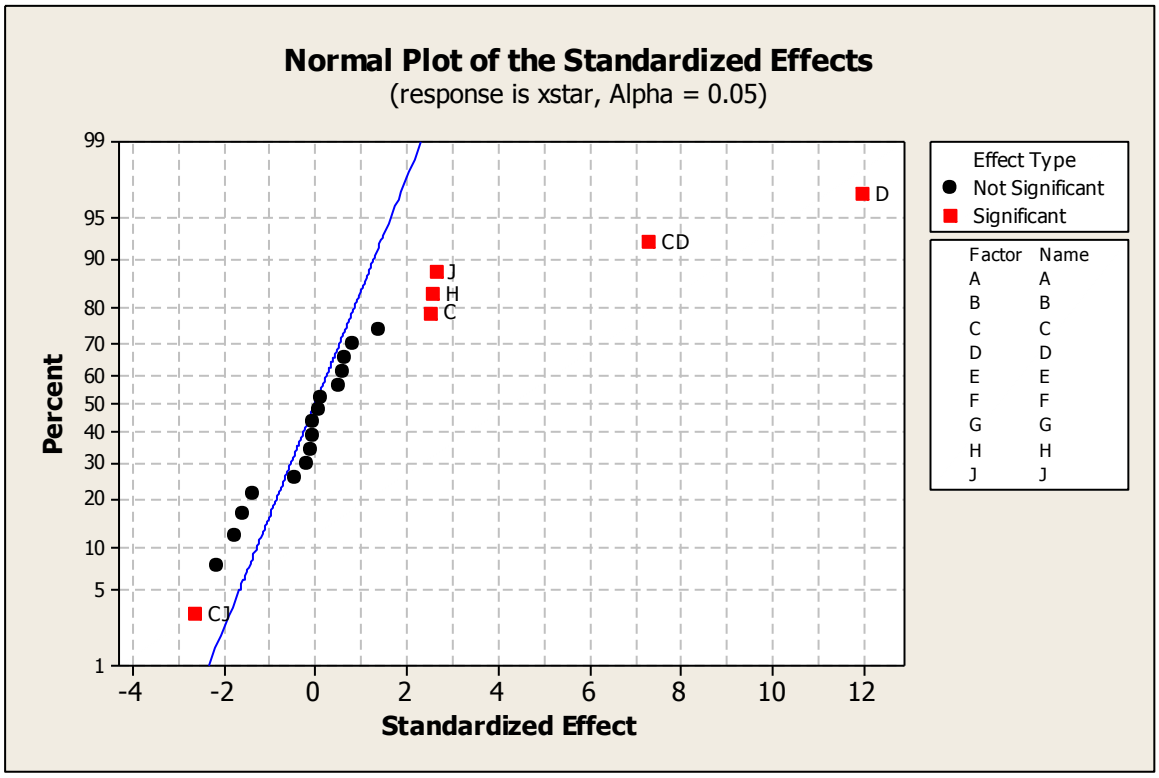

Figure B.3: Normal Probability Plot for all factors for x_1^* (with no replicates)

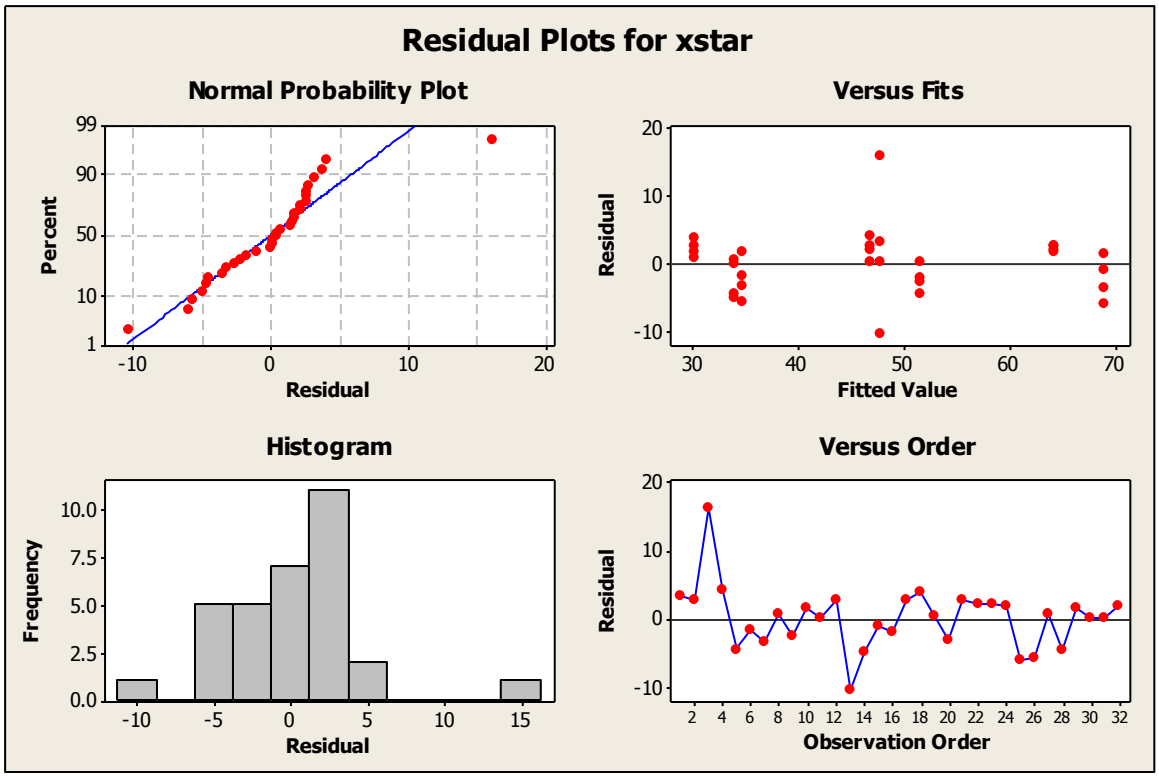

Figure B.4: Residual Plots for x_1^* (once non-significant factors are removed)

B.1.5 Minitab Results for y^*

Results after non-significant terms were removed:

Factorial Fit: ystar versus C, D, E, H, J

```
Estimated Effects and Coefficients for ystar (coded units)

Term       Effect     Coef  SE Coef      T      P
Constant            5.0625   0.1561  32.43  0.000
C         -1.1250  -0.5625   0.1561  -3.60  0.001
D         -1.0000  -0.5000   0.1561  -3.20  0.004
E          1.1250   0.5625   0.1561   3.60  0.001
H          1.0000   0.5000   0.1561   3.20  0.004
J          1.1250   0.5625   0.1561   3.60  0.001
C*D        1.0000   0.5000   0.1561   3.20  0.004


S = 0.883176    PRESS = 31.9488
R-Sq = 73.60%   R-Sq(pred) = 56.75%   R-Sq(adj) = 67.27%


Analysis of Variance for ystar (coded units)

Source              DF  Seq SS  Adj SS   Adj MS      F      P
Main Effects         5  46.375  46.375   9.2750  11.89  0.000
  C                  1  10.125  10.125  10.1250  12.98  0.001
  D                  1   8.000   8.000   8.0000  10.26  0.004
  E                  1  10.125  10.125  10.1250  12.98  0.001
  H                  1   8.000   8.000   8.0000  10.26  0.004
  J                  1  10.125  10.125  10.1250  12.98  0.001
2-Way Interactions   1   8.000   8.000   8.0000  10.26  0.004
  C*D                1   8.000   8.000   8.0000  10.26  0.004
Residual Error      25  19.500  19.500   0.7800
  Lack of Fit        9  16.500  16.500   1.8333   9.78  0.000
  Pure Error        16   3.000   3.000   0.1875
Total               31  73.875
```

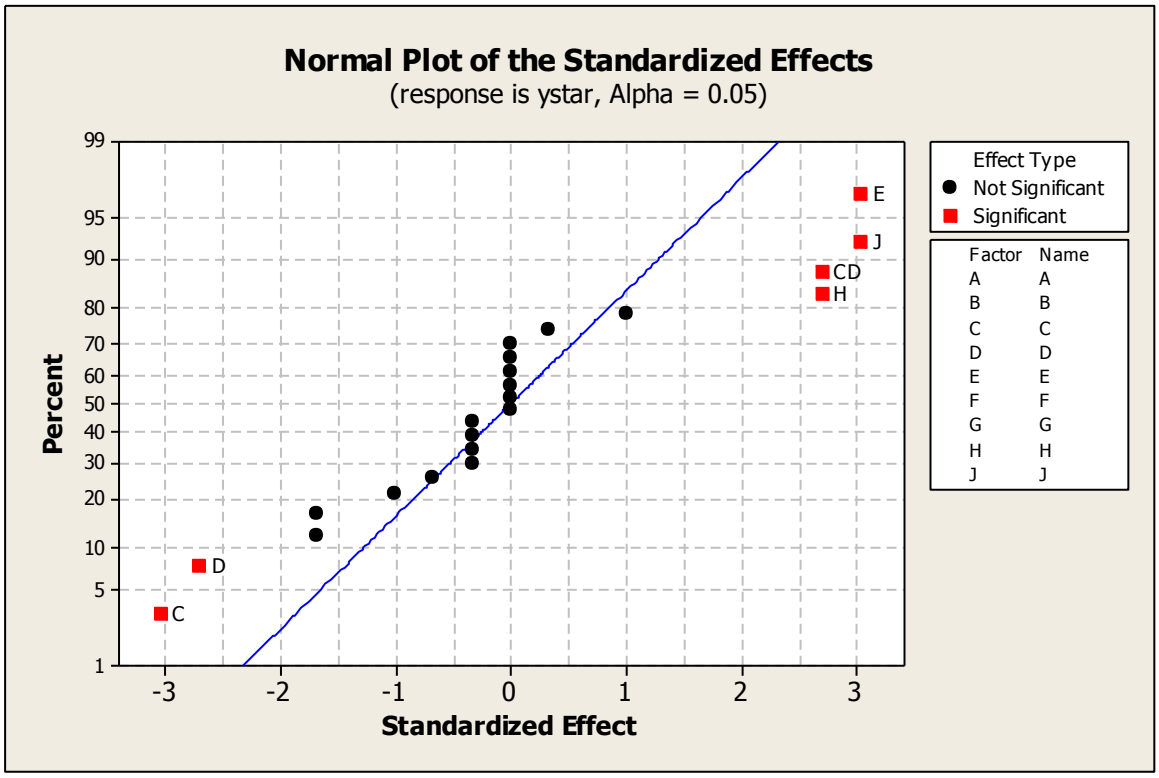

Figure B.5: Normal Probability Plot for all factors for y_1^* (with no replicates)

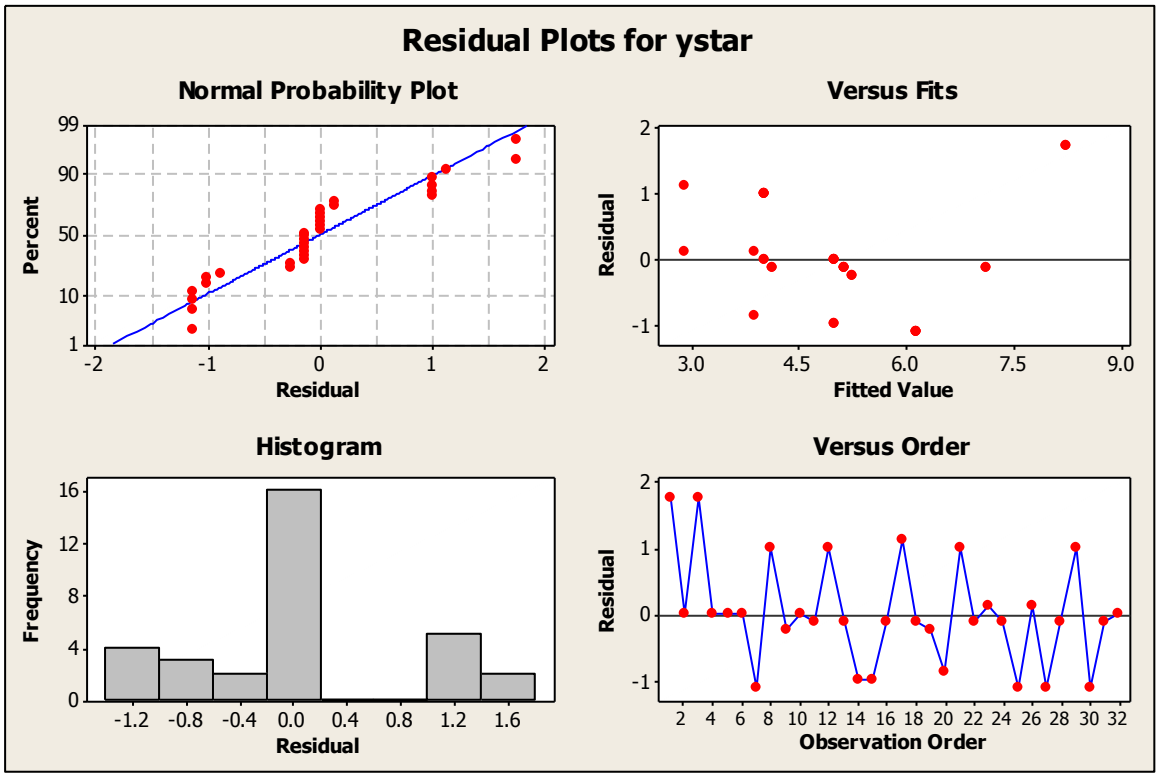

Figure B.6: Residual Plots for y_1^* (once non-significant factors are removed)

B.1.6 Minitab Results for Computer Processing Time

Results after non-significant terms were removed:

Factorial Fit: ProcessingTime versus A, B, C, D, E, F, G, H, J

Estimated Effects and Coefficients for ProcessingTime (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		497.9	9.085	54.80	0.000
A	1.0	0.5	9.085	0.06	0.955
B	3.3	1.6	9.085	0.18	0.861
C	0.6	0.3	9.085	0.03	0.973
D	3.1	1.6	9.085	0.17	0.866
E	38.1	19.1	9.085	2.10	0.058
F	-792.3	-396.2	9.085	-43.61	0.000
G	-596.4	-298.2	9.085	-32.82	0.000
H	326.2	163.1	9.085	17.95	0.000
J	199.6	99.8	9.085	10.98	0.000
A*B	474.8	237.4	9.085	26.13	0.000
A*C	-267.1	-133.5	9.085	-14.70	0.000
A*D	-163.8	-81.9	9.085	-9.01	0.000
A*E	-52.9	-26.4	9.085	-2.91	0.013
B*C	-195.1	-97.5	9.085	-10.74	0.000
B*D	-119.3	-59.6	9.085	-6.56	0.000
B*E	-75.6	-37.8	9.085	-4.16	0.001
C*D	91.0	45.5	9.085	5.01	0.000
C*E	97.9	48.9	9.085	5.39	0.000
D*E	159.4	79.7	9.085	8.77	0.000

S = 51.3945 PRESS = 225399
R-Sq = 99.75% R-Sq(pred) = 98.20% R-Sq(adj) = 99.35%

Analysis of Variance for ProcessingTime (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	9	9049595	9049595	1005511	380.67	0.000
A	1	9	9	9	0.00	0.955
B	1	85	85	85	0.03	0.861
C	1	3	3	3	0.00	0.973
D	1	78	78	78	0.03	0.866
E	1	11614	11614	11614	4.40	0.058
F	1	5022453	5022453	5022453	1901.44	0.000
G	1	2845425	2845425	2845425	1077.24	0.000
H	1	851324	851324	851324	322.30	0.000
J	1	318604	318604	318604	120.62	0.000
2-Way Interactions	10	3421343	3421343	342134	129.53	0.000
A*B	1	1803670	1803670	1803670	682.85	0.000
A*C	1	570539	570539	570539	216.00	0.000
A*D	1	214633	214633	214633	81.26	0.000
A*E	1	22365	22365	22365	8.47	0.013
B*C	1	304419	304419	304419	115.25	0.000
B*D	1	113818	113818	113818	43.09	0.000
B*E	1	45773	45773	45773	17.33	0.001
C*D	1	66263	66263	66263	25.09	0.000
C*E	1	76632	76632	76632	29.01	0.000
D*E	1	203233	203233	203233	76.94	0.000
Residual Error	12	31697	31697	2641		
Total	31	12502635				

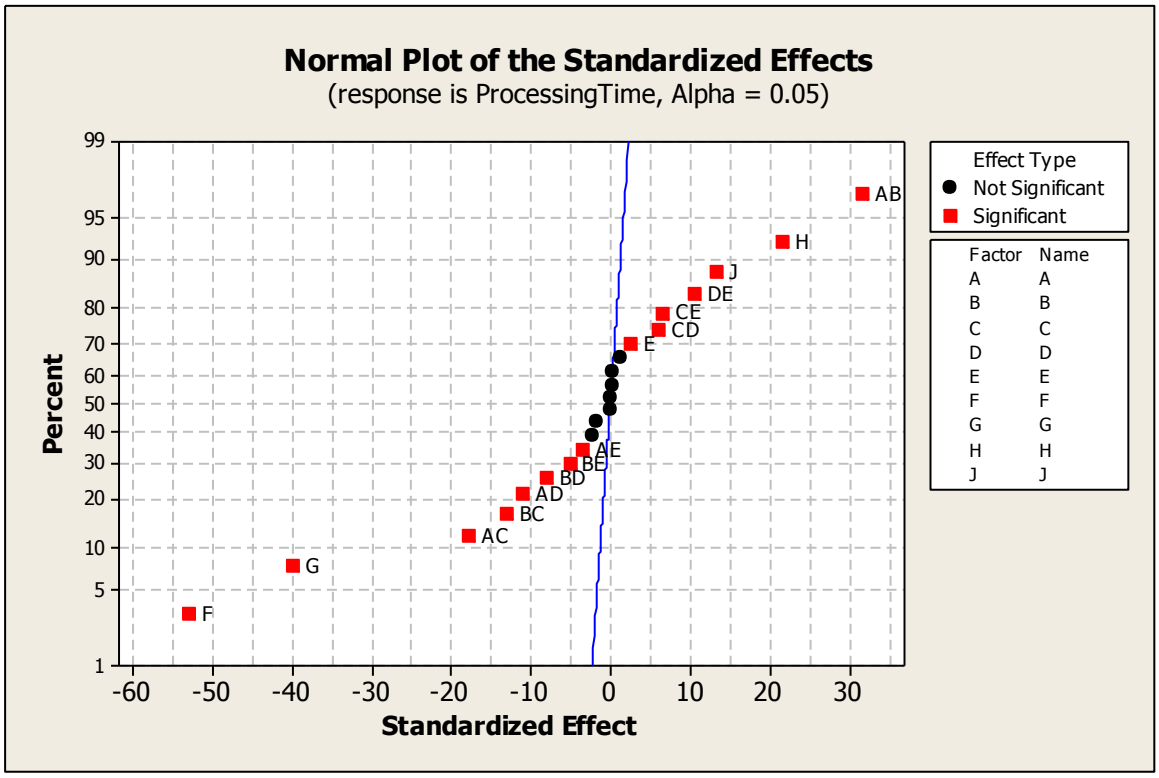

Figure B.7: Normal Probability Plot for all factors for processing time (with no replicates)

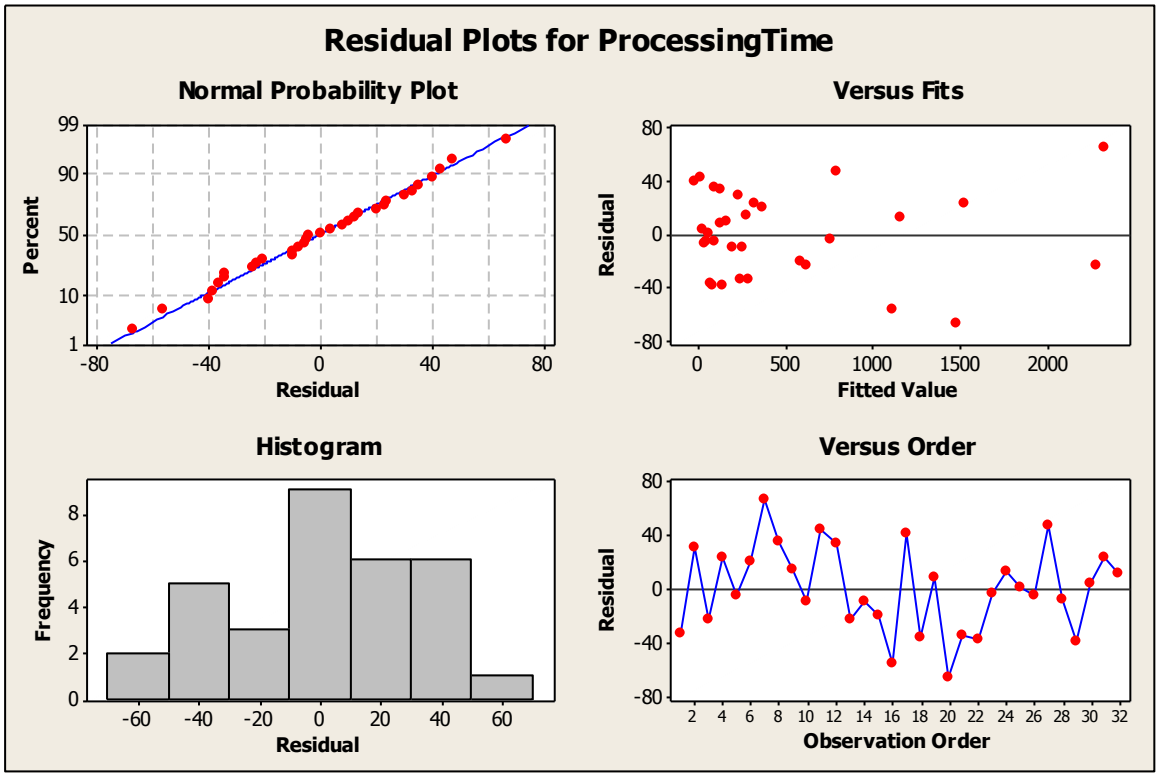

Figure B.8: Residual Plots for processing time (once non-significant factors are removed)

Appendix C: MATLAB Code

```
%GeometricApproach.m
%(c) 2011 m.c.katic
%
%The purpose of this file is to calculate determine the optimal location
%for a sensor using the geometric approach.
%
%Area of Interest Variables
%   W = width of area of interest (in nm)
%   Length = length of area of interest (in nm)
%   Kincrement = increment of the offset value (also for x value)
%   Jincrement = increment of the heading value
%   Lincrement = increment of the y value
%   T = patrol time
%
%
%Submerged Contact Parameter Vector Variables
%   V = velocity (in nm/second)
%   delta = offset from the thread axis (in nm)
%   theta = heading/angle contact makes with the border,0<=theta<=pi
%
%Sensor Parameter Vector Variables
%   N = number of unique sensors under consideration
%   Vmax = maximum velocity (in nm/second)
%   BC = base cost (in dollars)
%   SC = sensing cost (in dollars)
%
%Other variables (calculated)
%   A = left boundary of width located at coordinates (-1/2W,0)=(-1/2)*W
%   B = right boundary of width located at coordinates (1/2W,0)=(1/2)*W
%   K = total number of offset values = W/Kincrement
%   J = total number of heading values = pi/Jincrement
%   L = total number of y values =
%
%Occurence Matrix Variables:
%   OffsetSet = vector of set of offset values
%   OffsetDist = vector for the pdf for offset values
%   HeadingSet = vector of set of heading values
%   HeadingDist = vector for the pdf for heading values
%   OMatrix = the probability of occurrence matrix
%
%Variables for the coverage function calculations
%   offset = offset value under investigation
%   heading = heading value under investigation
%   Z = distance between the offset value and the center of the sensor
%   CF(j,k) = coverage function matrix of size J x K
%   Radius, R = radius of the acquisition footprint (from
%       AcquisitionRadiusCalculation.m)
%   D = maximum distance that a submerged contact can travel in the patrol
%       time
%   alpha(k) = lower bound for the heading values associated with the
%       offset k
%   beta(k) = upper bound for the heading values associated with the
%       offset k
%   x = x value of the sensor's center location
```

```
%   y = y value of the sensor's center location
%
%
%_____________________________________________________________________________
%_____________________________________________________________________________
%----Start Inputs, Occurrence Matrix Formation, and Radius Calculations----
clear

%_____________________________________________________________________________
%-----------------------Start Inputs------------------------------------
%input area of interest variables
disp(' ')
disp('_____________________Start Program_____________________________________')
W = input('Enter the width of the area of interest (nm):');
if isempty(W)
    disp('You must enter the width')
end

Length = input('Enter the length of the area of interest (nm):');
if isempty(Length)
    disp('You must enter the length')
end

Kincrement = 1;

Lincrement = 1;

Jincrement = 0.031416; %separates pi into 100 equal segments

%calculate A, B, K, J, and L
A = 0;
B = W;
K = (W/Kincrement)+1;
U = (W/Kincrement)+1;
J = 101;%separate pi into 100 equal segments
L = (Length/Lincrement)+1;

disp(' ')
disp('-----------------------------------------------------------------')
str = ['There are ', num2str(K), ' discrete offset values (and x values)'];
disp(str)
str = ['There are ', num2str(J), ' discrete heading values'];
disp(str)
str = ['There are ', num2str(L), ' discrete y values'];
disp(str)

%Define the set of all possible offset values as well as x values
for k=1:K;
    OffsetSet(k) = A +((k-1)*Kincrement);
end
%display the OffsetSet
   % OffsetSet


%Define the set of all possible heading values
```

```
for j=1:J;
    HeadingSet(j) = 0 +((j-1)*Jincrement);
end
%display the HeadingSet
  % HeadingSet

%Define the set of all possible x values
for u=1:U;
 XSet(u) = A +((u-1)*Kincrement);
end

%Define the set of all possible y values
for l=1:L;
    YSet(l) = 0 +((l-1)*Lincrement);
end

%display the XSet
   % XSet
%display the YSet
   % YSet
disp('------------------------------------------------------------------')

%input the Sensor Parameter Vector Variables
disp(' ')
disp('Enter the following information for the sensor parameter vector(s).')
N = input('Enter the number of unique sensors under consideration:');
if isempty(N)
    disp('You must enter the number of unique sensors under consideration')
end

T = input('Enter the patrol time (hours):');
if isempty(T)
    disp('You must enter the patrol time')
end

disp(' ')
disp('Enter the following information for the submerged contact parameter
vector.')
%input the Submerged Contact Parameter Vector Variables
V = input('Enter the velocity of the submerged contact:');
if isempty(V)
    disp('You must enter the velocity of the submerged contact')
end

%calculate the maximum distance a submerged contact can travel
D = V*T;

%--------------------------------End Inputs--------------------------------
%__________________________________________________________________________

%__________________________________________________________________________
%-------------------Start Acqusition Range Calculations--------------------

%Ask user if they have info to calculate acquisition range
disp(' ')
```

```
for i= 1:N
    %basic information needed for every sensor no matter what type


    str = ['Enter the following information for sensor ', num2str(i), '.'];
    disp(str)
    %{
    Vmax(i) = input('Enter the maximum velocity for the sensor:');
    if isempty(Vmax(i))
         disp('You must enter the maximum velocity for the sensor');
    end
    %}

    BC(i) = input('Enter the base cost for the sensor:');
    if isempty(BC(i))
         disp('You must enter the base cost for the sensor');
    end

    SC(i) = input('Enter the sensing cost for the sensor:');
    if isempty(SC(i))
         disp('You must enter the sensing cost for the sensor');
    end

    TC(i) = BC(i) + SC(i);

    Radius(i) = input('Enter the acquisition radius for the sensor: ');
    if isempty(Radius(i))
        disp('You must enter the acqusition radius for the sensor.');
    end

end
TDSNCost = sum(TC);

%-------------------End Acqusition Range Calculations----------------------
%_____________________________________________________________________________


%_____________________________________________________________________________
%-----------------Start of Occurrence Matrix Formation---------------------

disp(' ')
disp('You can choose any of the following built-in pdfs/formations for the
offset')
disp('or heading values. You can also choose to manually enter the
probabilities.')
disp(' ')
disp('1 = uniform distribution')
disp('2 = triangular distribution (enter C value)')
disp('3 = double triangular distribution')
disp('4 = manually enter values')

OffsetDistNum =input('Enter the number of the formation for the offset
probability: ');
```

```
if OffsetDistNum == 1%user chooses uniform distribution
    for k = 1:K;
        OffsetDist(k)= 1/K;
    end
elseif OffsetDistNum == 2%user chooses triangular distribution
    c = input('Enter in the index of the offset value for the C value for the
triangular distribution: ');
    C = OffsetSet(c);
    %calculate the estimates of the probabilities
    for k = 1:K;
        if OffsetSet(k) <= C
            OffsetEstDist(k) = OffsetSet(k) * 2/((B-A)*C);
        elseif OffsetSet(k) > C
            OffsetEstDist(k) = OffsetSet(k) *(-2/((B-A)*(B-C))) + 2*B/((B-
A)*(B-C));
        end
        OffsetEstDist(k)= OffsetEstDist(k)*Kincrement;
    end
    %OffsetEstDist
    %OED = sum(OffsetEstDist)
    %calculate delta
    delta = (1 - sum(OffsetEstDist))/K;
    %calculate the true probabilities
    for k = 1:K;
        OffsetDist(k) = OffsetEstDist(k) + delta;
    end

elseif OffsetDistNum == 3%user chooses double triangular distribution
    disp('Please enter the offset index for the following points for the two
triangular distributions.')
    cone = input('C1: ');
    C1 = OffsetSet(cone);
    ctwo = input('C2: ');
    C2 = OffsetSet(ctwo);
    cthree = input('C3: ');
    C3 = OffsetSet(cthree);
    for k = 1:K;
        if OffsetSet(k) <= C1
            OffsetEstDist(k) = OffsetSet(k) /((C2-A)*C1);
        elseif (OffsetSet(k) > C1) && (OffsetSet(k) <= C2)
            OffsetEstDist(k) = OffsetSet(k)*(-1/((C2-A)*(C2-C1))) + C2/((C2-
A)*(C2-C1));
        elseif (OffsetSet(k) > C2) && (OffsetSet(k) <= C3)
            OffsetEstDist(k) = OffsetSet(k)/((B-C2)*(C3-C2))- C2/((B-C2)*(C3-
C2));
        elseif (OffsetSet(k) > C3)
            OffsetEstDist(k) = OffsetSet(k)*(-1/((B-C2)*(B-C3))) + B/((B-
C2)*(B-C3));
        end
        OffsetEstDist(k)= OffsetEstDist(k)*Kincrement;
    end
   %OffsetEstDist
   %OED = sum(OffsetEstDist)
    %calculate delta
    delta = (1 - sum(OffsetEstDist))/K;
```

```
    %calculate the true probabilities
    for k = 1:K;
        OffsetDist(k) = OffsetEstDist(k) + delta;
    end

elseif OffsetDistNum == 4 %if the user wants to manually enter probabilites
for offset values
%Prompt user to enter the probabilities of occurrence for offset values
disp(' ')
disp('Enter the probabilities of occurrence for every offset value.')

for k = 1:K;
    str = ['Enter the probability that the offset value of ',
num2str(OffsetSet(k)),' will occur: '];
    OffsetDist(k) = input(str);
end

disp('------------------------------------------------------------------')
end

%display the offset distribution
%disp('The following is the distribution for the offset values:')
%OffsetDist
%Off = sum(OffsetDist);
%Off
%disp('------------------------------------------------------------------')


HeadingDistNum = input('Enter the number of the formation for the heading
probability: ');
if HeadingDistNum == 1%user chooses uniform distribution
    for j = 1:J;
        HeadingDist(j)= 1/J;
    end
elseif HeadingDistNum == 2%user chooses normal distribution
    c = input('Enter in the heading index value for the C value for the
triangular distribution: ');
    C = HeadingSet(c);
    %calculate the estimates of the probabilities
    for j = 1:J;
        if HeadingSet(j) <= C
            HeadingEstDist(j) = HeadingSet(j)* 2/(3.14*C);
        elseif HeadingSet(j) > C
            HeadingEstDist(j) = HeadingSet(j)* (-2/(3.14*(3.14-C)))
+(2/(3.14-C));
        end
        HeadingEstDist(j)= HeadingEstDist(j)*Jincrement;
    end
    %HeadingEstDist
    %HED = sum(HeadingEstDist)
    %calculate delta
    delta = (1 - sum(HeadingEstDist))/J;
    %calculate the true probabilities
    for j = 1:J;
        HeadingDist(j) = HeadingEstDist(j) + delta;
    end
```

```
elseif HeadingDistNum == 3%user chooses double triangular distribution
    disp('Please enter the heading index for the following points for the two
triangular distributions.')
    cone = input('C1: ');
     C1 = HeadingSet(cone);
    ctwo = input('C2: ');
     C2 = HeadingSet(ctwo);
    cthree = input('C3: ');
     C3 = HeadingSet(cthree);
    for j = 1:J;
        if HeadingSet(j) <= C1
            HeadingEstDist(j) = HeadingSet(j) /(C2*C1);
        elseif (HeadingSet(j) > C1) && (HeadingSet(j) <= C2)
            HeadingEstDist(j) = HeadingSet(j)*(-1/(C2*(C2-C1))) + 1/(C2-C1);
        elseif (HeadingSet(j) > C2) && (HeadingSet(j) <= C3)
            HeadingEstDist(j) = HeadingSet(j)/((3.14-C2)*(C3-C2))- C2/((3.14-
C2)*(C3-C2));
        elseif (HeadingSet(j) > C3)
            HeadingEstDist(j) = HeadingSet(j)*(-1/((3.14-C2)*(3.14-C3))) +
3.14/((3.14-C2)*(3.14-C3));
        end
        HeadingEstDist(j)= HeadingEstDist(j)*Jincrement;
    end
    %HeadingEstDist
    %HED = sum(HeadingEstDist)
    %calculate delta
    delta = (1 - sum(HeadingEstDist))/J
    %calculate the true probabilities
    for j = 1:J;
        HeadingDist(j) = HeadingEstDist(j) + delta;
    end


elseif HeadingDistNum == 4 %if the user wants to manually enter probabilities
for heading values
%Prompt user to enter the probabilities of occurrence for heading values
disp(' ')
disp('Enter the probabilities of occurrence for every heading value.')

for j = 1:J;
    str = ['Enter the probability that the heading value of ',
num2str(HeadingSet(j)),' will occur: '];
    HeadingDist(j) = input(str);
end
disp('-----------------------------------------------------------------------')
end

%HeadingDist
%OffsetDist


%display the heading distribution
%disp('The following is the distribution for the heading values:')
%HeadingDist
%Head = sum(HeadingDist);
%Head
```

```
%disp('------------------------------------------------------------------------')

disp('Computing.....')
%disp('------------------------------------------------------------------------')
%disp('The following is the probability of occurrence matrix:')
%Calculate the outer product between the OffsetDist Vector and
%the HeadingDist Vector to obtain the OccurrenceMatrix
OMatrix = HeadingDist'*OffsetDist;
%OMatrix
%surf(OMatrix)
O = sum(sum(OMatrix));
%O
%disp('------------------------------------------------------------------------')

%----------------- End of  Occurrence Matrix Formation---------------------
%_____________________________________________________________________________

%_____________________________________________________________________________
%-----------------Start Geometric Approach-------------------------------

%The code in this section populates the matrices for the coverage function
%for a sensor at all possible x and y values.
%
%Variables
%   Qstar = optimal probability of acquisition
%   xstar = x coordinate associated with Qstar
%   ystar = y coordinate associated with Qstar
%   Q = probability of acquisition
%   QM = the matrix created by multiplying the Occurrence matrix by the CF
%   QSet = set of Q values
%   multi = counter for number of (x,y) optimal positions; used for
%   displaying results
%   xstarset = set of xstar values if more than one optimal position
%   ystarset = set of ystar values if more than one optimal position
%

Qstar = 0;%initialize the optimal probability of acquisition to 0
xstar = 0;%initialize the x coordinate associated with Qstar as 0
ystar = 0;%initialize the y coordinate associated with Qstart as 0

OMatrixNew = OMatrix;%Original OMatrix used to palce first sensor

for i=1:N;
    tic
    R = Radius(i);
    for u=1:U;
        x = XSet(u);%set x value
        for l=1:L;
            y = YSet(l);%set y value
            %Populate the coverage function matrix associated with the x,y
            for k=1:K;%evaluate coverage function at x,y
                %calculate the Z value associated with the offset where Z =
sqrt(y^2 +(x-offset)^2)
                Z = sqrt(y^2 + (x-OffsetSet(k))^2);
                %calculate the alpha and beta values at the offset value
```

```
                if Z>=D
                  if (OffsetSet(k) >=A) && (OffsetSet(k)<x)
                     alpha = atan(y/(x-OffsetSet(k)))-
asin((R*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z);
                  elseif OffsetSet(k) ==x
                     alpha = (pi/2) - asin((R*sin(acos((R^2+D^2-
Z^2)/(2*R*D))))/Z);
                  elseif (OffsetSet(k) >x) && (OffsetSet(k) <=B)
                     alpha = pi - atan(y/(OffsetSet(k)-x)) -
asin((R*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z);
                  end
                  beta = alpha + (2*asin((R*sin(acos((R^2+D^2-
Z^2)/(2*R*D))))/Z));
                elseif (Z>=R) && (Z<D)
                  if (OffsetSet(k) >=A) && (OffsetSet(k)<x)
                     alpha = atan(y/(x-OffsetSet(k))) - asin(R/Z);
                  elseif OffsetSet(k) ==x
                     alpha = (pi/2) - asin(R/Z);
                  elseif (OffsetSet(k) >x) && (OffsetSet(k) <=B)
                     alpha = pi - atan(y/(OffsetSet(k)-x)) - asin(R/Z);
                  end
                  beta = alpha + (2*asin(R/Z));
                end

                if Z>(R+D)%if the sensor is farther than R+D away from the
offset, it can't acquire anything from that offset value
                  for j=1:J;
                     CF(j,k) = 0;
                  end%end for loop for heading values
                end

                if (Z>=0) && (Z<R)% if the sensor is <R away from the offset,
all headings are acquired at that offset value
                  for j=1:J;
                     CF(j,k) = 1;
                  end%end for loop for heading values
                end

                if (Z>=R) && (Z<(R+D))
                  for j=1:J;
                     if (HeadingSet(j) >= alpha) && (HeadingSet(j) <= beta)
                         CF(j,k) = 1;
                     else
                         CF(j,k) = 0;
                     end
                  end%end the for loop for heading values
                end%end the for loop for when R < Z < (R+D)

             end%end the for loop as the offset values change

             %Calculate the probability of acquisition for the sensor
positioned at x,y
             QM = OMatrixNew.*CF;
             %QM %display QM matrix
             Q = sum(sum(QM));
             %Q %display Q value for the x,y position
```

```
            %sum(QM) produces a vector of the sum of the columns in QM
            %then the sum(sum(QM)) reports the sume of the elements of the
            %vector producted from sum(QM)

            QSet(l,u) = Q;
            %QSet %display QSet

            if Q > Qstar%keep track of the best Q and its coordinates
                Qstar = Q;
                xstar = x;
                ystar = y;
                QMstar = QM;
                if i ==1
                    CFstar1 = CF;
                elseif i ==2
                    CFstar2 = CF;
                end
            end

        end%end the for loop for the coverage function as y changes
    end%end the for loop for the coverage function as x changes


    %Record if multiple (x,y) values result in the optimal Q value
    multi = 0;
    for u=1:U;
        for l=1:L;
            if QSet(l,u) >= Qstar
                multi = multi +1;
                xstarset(multi) = XSet(u);
                ystarset(multi) = YSet(l);
            end
        end
    end

    %give results for the sensor i
    disp(' ')
    disp('------------------------------------------------------------------')
    %Report the best probabiltiy of acquisition
    disp('RESULTS:')
    str = ['The following information is for sensor ', num2str(i), '.'];
    disp(str)
    str = ['The best total probability of acquisition is ', num2str(Qstar),
'.'];
    disp(str)

    if multi ==1%only one optimal location
    str = ['This occurs at the coordinates (', num2str(xstar), ',',
num2str(ystar), ').'];
    disp(str)
    end

    if multi >1%more than one optimal location
        disp('There are multiple locations to place the sensor.')
        disp('The following are the coordinates for these locations:')
```

```
        for mul=1:multi;
        str = ['(',num2str(xstarset(mul)),',',num2str(ystarset(mul)),')'];
        disp(str)
        end
    end

    %record sensor i's optimal information for DSN summary
    %CF for the optimal position is already recorded in CF(:,:,i)
    if multi ==1
    Bestx(i) = xstar;
    Besty(i) = ystar;
    elseif multi >1%report first values from the list of multiple locations
    Bestx(i) = xstarset(1);
    Besty(i) = ystarset(1);
    end

    %Change the Original OMatrix to OMatrixNew by replacing the O(j,k)
    %values with zeros for the CF(j,k)=1
    if i ==1
    for k = 1:K;
       for j = 1:J;
          if CFstar1(j,k) == 1
              OMatrixNew(j,k) = 0;
          elseif CFstar1(j,k) == 0
              OMatrixNew(j,k) = OMatrix(j,k);
          end%end if statement
       end%end loop for j
    end%end loop for k
    end

    %OMatrixNew %display the new matrix
    Qstar = 0;%reset optimal probability of acquisition to 0
    xstar = 0;%reset the x coordinate associated with Qstar as 0
    ystar = 0;%reset the y coordinate associated with Qstart as 0
    clear QSet;


end

if N > 1
%Calculate the total probability of acquisition for the DSN
CFjoint = CFstar1|CFstar2;
TotalQM = OMatrix.*(double(CFjoint));
TotalQ = sum(sum(TotalQM));


%display results
disp('Given that:')
for i = 1:N;
str = ['Sensor ', num2str(i), ' is placed at ', '(', num2str(Bestx(i)), ',',
num2str(Besty(i)), ')'];
disp(str)
end
str =['The best probability of acquisition for the DSN is ', num2str(TotalQ),
'.'];
```

```
disp(str)
str = ['The total cost for this DSN is $', num2str(TDSNCost), '.'];
    disp(str)
    disp('-----------------------------------------------------------------')

end
    t(i) =toc;
    str = ['This run took ', num2str(t(i)), ' seconds'];
    disp(str)



   if N==1
   % shows results in four different graphs
   figure('Name','Graphs of Probability of Acquistion vs. x,y
Values','NumberTitle','off')
   subplot(2,1,1);surf(XSet,YSet,QSet,'EdgeColor','none')
   xlabel('x value');
   ylabel('y value');
   zlabel('Probability of Acquisition');
   box on;
   subplot(2,1,2);pcolor(XSet,YSet,QSet)
   xlabel('x value');
   ylabel('y value');

   figure('Name','Coverage Function of Optimal Sensor
Position','NumberTitle','off')
   subplot(2,1,1);surf(OffsetSet,HeadingSet,CFstar,'EdgeColor','none')
   ylabel('Heading');
   xlabel('Offset');
   zlabel('Coverage Function Value');
   zlim([0 1]);
   subplot(2,1,2);pcolor(OffsetSet,HeadingSet,CFstar)
   ylabel('Heading');
   xlabel('Offset');

   figure('Name','Probability of Acquisition Graphs','NumberTitle','off')
   subplot(2,1,1);surf(OffsetSet,HeadingSet,QMstar, 'EdgeColor','none')
   ylabel('Heading');
   xlabel('Offset');
   zlabel('Probability of Acquisition');
   subplot(2,1,2);pcolor(OffsetSet,HeadingSet,QMstar)
   ylabel('Heading');
   xlabel('Offset');

   figure('Name','Graphs of Probability of Occurrence','NumberTitle','off')
   subplot(2,1,1);surf(OffsetSet,HeadingSet,OMatrix, 'EdgeColor','none')
   ylabel('Heading');
   xlabel('Offset');
   zlabel('Probability of Occurrence');
   subplot(2,1,2);pcolor(OffsetSet,HeadingSet,OMatrix)
   ylabel('Heading');
   xlabel('Offset');

   end
```

```
    if N>1

    figure('Name','Graphs of Probability of Occurrence','NumberTitle','off')
    subplot(2,1,1);surf(OffsetSet,HeadingSet,OMatrix, 'EdgeColor','none')
    ylabel('Heading');
    xlabel('Offset');
    zlabel('Probability of Occurrence');
    subplot(2,1,2);pcolor(OffsetSet,HeadingSet,OMatrix)
    ylabel('Heading');
    xlabel('Offset');

    figure('Name','Coverage Function of Sensor 1','NumberTitle','off')
    subplot(2,1,1);surf(OffsetSet,HeadingSet,CFstar1,'EdgeColor','none')
    ylabel('Heading');
    xlabel('Offset');
    zlabel('Coverage Function Value');
    zlim([0 1]);
    subplot(2,1,2);pcolor(OffsetSet,HeadingSet,CFstar1)
    ylabel('Heading');
    xlabel('Offset');

    figure('Name','Coverage Function of Sensor 2','NumberTitle','off')
    subplot(2,1,1);surf(OffsetSet,HeadingSet,CFstar2,'EdgeColor','none')
    ylabel('Heading');
    xlabel('Offset');
    zlabel('Coverage Function Value');
    zlim([0 1]);
    subplot(2,1,2);pcolor(OffsetSet,HeadingSet,CFstar2)
    ylabel('Heading');
    xlabel('Offset');

    figure('Name','Coverage Function of DSN','NumberTitle','off')

subplot(2,1,1);surf(OffsetSet,HeadingSet,double(CFjoint),'EdgeColor','none')
    ylabel('Heading');
    xlabel('Offset');
    zlabel('Coverage Function Value');
    zlim([0 1]);
    subplot(2,1,2);pcolor(OffsetSet,HeadingSet,double(CFjoint))
    ylabel('Heading');
    xlabel('Offset');

    figure('Name','Probability of Acquisition Graphs','NumberTitle','off')
    subplot(2,1,1);surf(OffsetSet,HeadingSet,TotalQM, 'EdgeColor','none')
    ylabel('Heading');
    xlabel('Offset');
    zlabel('Probability of Acquisition');
    subplot(2,1,2);pcolor(OffsetSet,HeadingSet,TotalQM)
    ylabel('Heading');
    xlabel('Offset');
    end

%----------------------End Geometric Approach-------------------------------
%_______________________________________________________________________________
%_______________________________________________________________________________
%End program
```

Academic Vita of Megan Katic

Megan Katic
112 Circle Drive
Pittsburgh, PA 15237
katic.mc@gmail.com

Education:	The Pennsylvania State University M.S. Industrial Engineering and Operations Research (Spring 2011) B.S. Industrial Engineering (Spring 2011) Six Sigma Minor (Spring 2011) Schreyer Honors College
Thesis Title:	Sensor Placement for Acquiring Submerged Contacts
Thesis Advisors:	Dr. Soundar R.T. Kumara & Dr. Jeffrey J. Weinschenk
Honors Advisor:	Dr. Paul Griffin
Industry Experience:	Procter & Gamble- Alexandria Plant Outbound Logistics Summer Intern Alexandria, LA – Summer 2010
	Procter & Gamble- Supply Network Operations Summer Intern Cincinnati, OH – Summer 2009
Awards:	Wilbur L. & Judy L. Meier Award (2011) Joelle Award for Women in Engineering- Honorable Mention (2011) Achieving Women Award- Nominee (2011) College of Engineering Capstone Design –First Place (2010) Dean's List (2006-2011) Outstanding Individual Representing Service & Philanthropy (2008)
Leadership:	IIE Mentorship Program Co-Chair (2009-2011) IE Student Advisory Council Co-Chair (2009-2011) Pyramid THON Organization Family Relations Chair (2010-2011) SHC THON Secretary (2009-2010) Behrend's Reality Check Service Organization President (2007-2008) Behrend's Newman Club Treasurer (2007-2008) Member: Alpha Pi Mu, IIE, Phi Kappa Phi, SWE, National Society of Collegiate Shcholars, WEP