OPTIMIZATION OF DISTRIBUTED NETTED SENSOR FIELDS WITH APPLICATION TO UNDERSEA SURVEILLANCE

A Thesis in
Industrial Engineering
by
Megan Christine Katic

© 2011 Megan Christine Katic

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2011
The thesis of Megan Christine Katic was 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

Christopher C. Byrne  
Assistant Professor of Mathematics and Research Associate

Paul Griffin  
Peter and Angela Dal Pezzo Department Head Chair  
Head of the Department of Industrial and Manufacturing Engineering

*Signatures are on file at the Graduate School.
ABSTRACT

Distributed-Netted Systems (DNS) can be used in many different environments. In particular, several Undersea Distributed-Netted Sensors (UDNS) have been explored in software simulation and hardware prototypes. While specific approaches have been studied in great detail, the literature lacks a broad view that explores which sensing modalities and sensor field compositions yield the best performance. The literature fails to answer questions such as is mobility cost effective, how fast is fast enough, are fewer high performance sensors preferred over many less expensive lower performance sensors, etc. We introduce a novel geometric method that allows rapid analysis of competing approaches including heterogeneous mixtures of several different sensor types. Our approach will allow planners to select Pareto efficient solutions for undersea surveillance applications. Within our approach, the sensors are modeled as disks which represent their acquisition footprint. The sensor will be able to detect, classify, localize, and track (DCLT) any submerged contact that enters this region; we refer to the sensor’s ability to DCLT as ‘acquiring’ the contact. 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 one side of the region and the angle they create with the border (also referred to as the heading). Probability density functions are used to describe these offsets and headings, and their combination results in a probability mass function (PMF). A coverage function is developed to determine which of these offsets and headings are ‘covered’ by a sensor located at a particular point in the area. The goal of the approach is to maximize the probability of acquisition, which is calculated as the Frobenius inner product of the Hadamard product of the PMF and the coverage function. The probability of acquisition of a DNS is plotted against the cost of the DNS and a Pareto analysis is performed to determine which DNS should be implemented.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>II</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>VI</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>VII</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>X</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>XIII</td>
</tr>
<tr>
<td><strong>CHAPTER 1: INTRODUCTION</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Undersea Surveillance</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Motivation for a Abstract representation View</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Research Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Organization of Thesis</td>
<td>3</td>
</tr>
<tr>
<td><strong>CHAPTER 2: BACKGROUND LITERATURE SURVEY</strong></td>
<td>5</td>
</tr>
<tr>
<td>2.1 Distributed-Netted Sensors Overview</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Types of Sensors</td>
<td>6</td>
</tr>
<tr>
<td>2.3.1 Passive Sonar</td>
<td>8</td>
</tr>
<tr>
<td>2.3.2 Active Sonar</td>
<td>9</td>
</tr>
<tr>
<td>2.3.3 Electromagnetic Detection Systems</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Comparison of Stationary and Mobile Platforms</td>
<td>12</td>
</tr>
<tr>
<td>2.5 Technological Problems Associated with DNS</td>
<td>13</td>
</tr>
<tr>
<td>2.6 Approaches for Solving the Sensor Positioning Problem</td>
<td>15</td>
</tr>
<tr>
<td>2.6.1 Track Coverage Problem</td>
<td>15</td>
</tr>
<tr>
<td>2.6.2 Set Covering Problem</td>
<td>16</td>
</tr>
<tr>
<td>2.6.3 Occupancy Grid Approach</td>
<td>17</td>
</tr>
<tr>
<td>2.6.4 Geometric Optimization Approach</td>
<td>19</td>
</tr>
<tr>
<td>2.6.5 Need for the New Geometric Approach</td>
<td>21</td>
</tr>
</tbody>
</table>
CHAPTER 3: DNS PERFORMANCE PROBLEM WITH PARETO ANALYSIS .......... 22

3.1 Problem Definition and Discussion .......................................................... 22

3.2 Assumptions .............................................................................................. 25

  3.2.1 Assumptions on the Sensors ................................................................. 25
  3.2.2 Assumptions on the Area of Interest .................................................... 25
  3.2.3 Assumptions on the Submerged Contacts ........................................... 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 Environment 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 Sensor Acquisition Radius ........................................ 38

4.6 Sensor’s Coverage Function (Discrete Case) ......................................... 39

  4.6.1 Range of $\delta_k$ Values that Can Result in a Coverage Function Value of One ........ 39
  4.6.2 Calculations of the Ranges of $\theta_j$ When $Z \geq D$ ............................ 40
  4.6.3 Calculations of the Ranges of $\theta_j$ When $Z < D$ ............................ 45
  4.6.4 Coverage Function Calculations ....................................................... 47

4.7 The Geometric Approach (Discrete Case) ............................................. 49

4.8 The Geometric Approach (Continuous Case) ....................................... 52
CHAPTER 5: METHODOLOGY FOR MULTIPLE SENSORS ........................................ 54
5.1 Context and Assumptions ............................................................................... 54
5.2 The Geometric Approach for Two Sensors .................................................. 54
5.2 Geometric Approach for Simultaneous Placement of N sensors ................. 58
5.3 Explanation of Pareto Analysis ..................................................................... 60

CHAPTER 6: EXPERIMENTATION .................................................................... 62
6.1 MATLAB Program ....................................................................................... 62
6.2 Results from Experiments of a Single Sensor ............................................. 62
   6.2.1 Complete Results of Run One with Plots ............................................. 63
   6.2.2 Summary of Results ........................................................................... 66
6.3 Results from Experiments of Multiple Sensors .......................................... 71
   6.3.1 Complete Results of DNS #1 with Plots ............................................ 72
   6.3.2 Summary of Results ........................................................................... 75
6.4 Discussion on Lifecycle Cost ...................................................................... 78
6.5 A Pareto Analysis of Two Sensor DNS ..................................................... 79
6.6 Analysis of the Number of Sensors vs. Increase in Q* .............................. 80

CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH ................................. 82
7.1 Contributions .............................................................................................. 82
7.2 Future Research ......................................................................................... 82

REFERENCES ................................................................................................. 84

APPENDIX A: PROOFS FOR EQUATIONS .................................................... 90
  A.1: Proof of Equation (4.14) .......................................................................... 90
  A.2: Proof of Statement 1 (Congruent Triangles) .......................................... 91
  A.3: Proof of Equation (4.16) .......................................................................... 92
  A.4: Proof of Equation (4.17) .......................................................................... 93
  A.5: Proof of Equation (4.18) .......................................................................... 93
  A.6: Proof of Equation (4.19) .......................................................................... 94
  A.7: Proof of Equation (4.20) .......................................................................... 95
A.8: Proof of Equation (4.21) ................................................................. 96
A.9: Proof of Equation (4.24) ................................................................. 96
A.10: Proof of Equation (4.25) .............................................................. 96

APPENDIX B: RESULTS FROM EXPERIMENTS ........................................ 97
B.1 Results from Single Sensor Experiments ............................................ 97
B.1.1 DOE Design ................................................................................. 97
B.1.2 Raw Data Results ......................................................................... 98
B.1.3 Minitab Results for $Q^*$ .............................................................. 99
B.1.4 Minitab Results for $x^*$ .............................................................. 101
B.1.5 Minitab Results for $y^*$ .............................................................. 103
B.1.6 Minitab Results for Computer Processing Time ............................. 105

APPENDIX C: MATLAB CODE ................................................................ 107
LIST OF TABLES

Table 2.1: Summary of the Influences of the Approaches on the Geometric Approach............20

Table 5.1: Sequential vs. Simultaneous Placement of Sensors.........................................60

Table 6.1: Summary of the Factor Levels ...........................................................................63
Table 6.2: Coding Schemes for Qualitative Four-level Factors........................................63
Table 6.3: Two-sensor DNS Under Consideration for Example Problem.........................71
Table 6.4: Results for all DNS from the MATLAB Program.............................................76
Table 6.5: Lifecycle Costs for Sensors..............................................................................79

Table B.1: Experimental Design for a $4^22^54$, or otherwise known as a $2^{9-4}$, Design........97
Table B.2: Results from the $4^22^54$ Experiment.............................................................98
LIST OF FIGURES

Figure 1.1: Thesis Organization...........................................................................4

Figure 2.1: SNR at two threshold settings T1 and T2..............................................7
Figure 2.2: Passive Sonar..................................................................................8
Figure 2.3: Active Sonar....................................................................................9
Figure 2.4: Coverage cones for track coverage..................................................15
Figure 2.5: Arc of S_i’s Perimeter Covered by S_j.............................................17

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 $\delta_k$ 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 $\delta_k$................................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 $\Delta \delta_k C_i H$ 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: Example of a Coverage Function................................................49
Figure 4.15: Computational Framework..........................................................50

Figure 5.1: 3D Matrices for N sensors..............................................................58
Figure 5.2: Metric space for DNS Analysis......................................................61
Figure 6.1: MATLAB 3D and Pseudocolor Plots for the Probability of Occurrence Matrix….64
Figure 6.2: MATLAB 3D and pseudocolor plots for the CF of a sensor whose $R_i = 15\text{nm}$ and position is (51, 10)…………………………………………………………………………………………………65
Figure 6.3: MATLAB 3D and Pseudocolor Plots for the $Q_i$ Matrix…………………..65
Figure 6.4: $Q_i$ Values For every $(x_u, y_v)$ Combination………………………………66
Figure 6.5: Normal Probability Plot for $Q_1^*$………………………………………………67
Figure 6.6: Estimated Effects and Coefficients for $Q_1^*$………………………………67
Figure 6.7: Normal Probability Plot for $x_1^*$………………………………………………68
Figure 6.8: Estimated Effects and Coefficients for $x_1^*$………………………………68
Figure 6.9: Normal Probability Plot for $y_1^*$………………………………………………69
Figure 6.10: Estimated Effects and Coefficients for $y_1^*$………………………………69
Figure 6.11: Normal Probability Plot for Processing Time…………………………….70
Figure 6.12: Estimated Effects and Coefficients for Processing Time……………………70
Figure 6.13: MATLAB 3D and Pseudocolor Plots for the Probability of Occurrence Matrix for DNS #1………………………………………………………………………………72
Figure 6.14: MATLAB 3D and pseudocolor plots for the CF of a sensor 1 positioned at (49, 5)…………………………………………………………………………………………73
Figure 6.15: MATLAB 3D and pseudocolor plots for the CF of a sensor 2 positioned at (79, 5)…………………………………………………………………………………………73
Figure 6.16: MATLAB 3D and pseudocolor plots for the TCF for DNS #1………………74
Figure 6.17: MATLAB 3D and Pseudocolor Plots for the $Q$ Matrix……………………74
Figure 6.18: Spatial Coverage Map for DNS #1………………………………………………75
Figure 6.19: MATLAB 3D and pseudocolor plots for the CF of a sensor 2 positioned at (77, 4)…………………………………………………………………………………………77
Figure 6.20: MATLAB 3D and pseudocolor plots for the TCF for DNS #2b…………….77
Figure 6.21: $Q_m^* \text{ vs. } LC$ Graph for Pareto Analysis……………………………80
Figure 6.22: $Q_m^* \text{ vs. Number of Sensors}………………………………………81

Figure A.1: Geometry for Proof of Equation 4.14 .................................................................90
Figure A.2: Geometry for Proof of Statement 1………………………………………………91
Figure A.3: Geometry for Proofs of Equations 4.16 and 4.17…………………………………92
Figure A.4: Geometry for Proofs of Equations 4.18 and 4.19.................................................93
Figure A.5: Geometry for Proof of Equation 4.20.................................................................95
Figure A.6: Geometry for Proofs of Equations 4.24 and 4.25..............................................96

Figure B.1: Normal Probability Plot for all factors for $Q_1^*$ (with no replicates).................100
Figure B.2: Residual Plots for $Q_1^*$ (once non-significant factors are removed)......................100
Figure B.3: Normal Probability Plot for all factors for $x_1^*$ (with no replicates)....................102
Figure B.4: Residual Plots for $x_1^*$ (once non-significant factors are removed)....................102
Figure B.5: Normal Probability Plot for all factors for $y_1^*$ (with no replicates)...................104
Figure B.6: Residual Plots for $y_1^*$ (once non-significant factors are removed)...................104
Figure B.7: Normal Probability Plot for all factors for processing time (with no replicates)....106
Figure B.8: Residual Plots for processing time (once non-significant factors are removed).....106
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of unique sensors under consideration, ( i = 1,2,\ldots,N )</td>
<td>23</td>
</tr>
<tr>
<td>M</td>
<td>Number of DNS under consideration</td>
<td>24</td>
</tr>
<tr>
<td>( S )</td>
<td>Area of interest</td>
<td>29</td>
</tr>
<tr>
<td>( W )</td>
<td>Width of area of interest</td>
<td>29</td>
</tr>
<tr>
<td>( L )</td>
<td>Length of area of interest</td>
<td>29</td>
</tr>
<tr>
<td>( A )</td>
<td>Left boundary of width located at coordinates ((-1/2W,0))</td>
<td>29</td>
</tr>
<tr>
<td>( B )</td>
<td>Right boundary of width located at coordinates ((1/2W,0))</td>
<td>29</td>
</tr>
<tr>
<td>( \Delta x )</td>
<td>Increment (in nm) of the change in ( x ) values</td>
<td>29</td>
</tr>
<tr>
<td>( \Delta \delta )</td>
<td>Increment (in nm) of the change in the offset values</td>
<td>29</td>
</tr>
<tr>
<td>( \Delta y )</td>
<td>Increment (in nm) of the change in ( y ) values</td>
<td>29</td>
</tr>
<tr>
<td>( U )</td>
<td>Set of ( x ) values; Number of possible ( x_u ) values, ( U=W/\Delta x )</td>
<td>29</td>
</tr>
<tr>
<td>( V )</td>
<td>Set of ( y ) values; Number of possible ( y_v ) values, ( V=L/\Delta y )</td>
<td>29</td>
</tr>
<tr>
<td>( x_u )</td>
<td>( x ) coordinate of the center of Sensor, where ( u = 1,\ldots,U )</td>
<td>29</td>
</tr>
<tr>
<td>( y_v )</td>
<td>( y ) coordinate of the center of Sensor, where ( v = 1,\ldots,V )</td>
<td>29</td>
</tr>
<tr>
<td>TL</td>
<td>Transmission Loss</td>
<td>30</td>
</tr>
<tr>
<td>ANL</td>
<td>Ambient Noise Level</td>
<td>30</td>
</tr>
<tr>
<td>RNL</td>
<td>Reverberation Noise Level</td>
<td>30</td>
</tr>
<tr>
<td>MD</td>
<td>Depth of environment, ( D \in {0,1} )</td>
<td>30</td>
</tr>
<tr>
<td>( v )</td>
<td>Velocity of the submerged contact</td>
<td>31</td>
</tr>
<tr>
<td>( \delta_k )</td>
<td>Element of the offset matrix ( \delta ); Potential offset value, ( k = 1,\ldots,K )</td>
<td>31</td>
</tr>
<tr>
<td>( \theta_j )</td>
<td>Element of the heading matrix ( \theta ); Potential heading value, ( j = 1,\ldots,J )</td>
<td>31</td>
</tr>
<tr>
<td>RN</td>
<td>Radiated Noise by the submerged contact for passive sonar</td>
<td>31</td>
</tr>
<tr>
<td>TS</td>
<td>Target Strength of the submerged contact for active sonar</td>
<td>31</td>
</tr>
<tr>
<td>DT_i</td>
<td>Detection Threshold of Sensor ( i )</td>
<td>32</td>
</tr>
<tr>
<td>AG_i</td>
<td>Array Gain of Sensor ( i )</td>
<td>32</td>
</tr>
<tr>
<td>SL_i</td>
<td>Source Level of Sensor ( i )</td>
<td>32</td>
</tr>
<tr>
<td>FN_i</td>
<td>Flow Noise of Sensor ( i )</td>
<td>32</td>
</tr>
<tr>
<td>SN_i</td>
<td>Self Noise of Sensor ( i )</td>
<td>32</td>
</tr>
</tbody>
</table>
\( \nu_{\text{maxi}} \) Maximum velocity of Sensor \( i \), p. 32

\( \text{BC}_i \) Base Cost of Sensor \( i \), p. 32

\( \text{SC}_i \) Sensing Cost of Sensor \( i \), p. 32

\( Q_i \) Probability of Acquisition for a sensor, p. 33

\( T \) Patrol Time for the DNS, p. 33

\( \text{LC} \) Lifecycle Cost Metric, p. 34

\( O \) Probability matrix containing the probability of occurrence, p. 34

\( o_{jk} \) Probability of occurrence for \( \theta_j \) and \( \delta_k \), \( o_{jk} \in O \), \( o_{jk} = P(\theta_j \cap \delta_k) \), p. 34

\( \text{CF}_i(x_u, y_v) \) Coverage Function of a sensor located at \( (x_u, y_v) \), p. 34

\( c_{fi_{jk}}(x_u, y_v) \) Coverage Function value for \( \theta_j \) and \( \delta_k \), \( c_{fi_{jk}} \in \text{CF}_i(x_u, y_v) \), p. 34

\( Q_i(x_u, y_v) \) Probability of acquisition matrix for a sensor at \( (x_u, y_v) \), p. 34

\( \delta \) Set of the Submerged contact’s offset from the Threat axis,

\[ A \leq \delta_k \leq B, \delta = [\delta_k], \text{ p. 35} \]

\( \theta \) Set of the Submerged contact’s heading with respect to the border \( \overline{AB} \),

\[ 0 \leq \theta_j \leq \pi, \theta = [\theta_j], \text{ p. 35} \]

\( \Delta \theta \) Heading Value Increment (in radians), p. 35

\( \text{K} \) Set of offset values; Number of offset values, \( K=W/\Delta \), p. 35

\( \text{J} \) Set of heading values; Number of heading values, \( J=\pi/\Delta \), 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

\( \gamma \) Acquisition coefficient, p. 38

\( \alpha_{i\theta v}(\delta_k) \) Lower bound for range of headings of a sensor for offset \( \delta_k \), p. 40

\( \beta_{i\theta v}(\delta_k) \) Upper bound for range of headings of a sensor for offset \( \delta_k \), p. 40

\( \overline{\delta_k C_i} \) Line segment connecting sensor’s center to offset \( \delta_k \), 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 \( T \),

\[ D = \nu * T, \text{ p. 40} \]
G  Point on the circumference of $F_i$ that intersects the line of length $D$ that creates the angle $\alpha_{uv}$ 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

$\overline{\delta_kG}$  Line segment of length $D$ that creates the angle $\alpha_{uv}$ with the border $\overline{AB}$, p. 41

$\overline{\delta_kH}$  Line segment of length $D$ that creates the angle $\beta_{uv}$ with the border $\overline{AB}$, p. 41

$\Delta \overline{\delta_kC_iH}$  Triangle created by line the segments $\overline{\delta_kC_i}$, $\overline{\delta_kH}$, and $R_i$, p. 41

$\Delta \overline{\delta_kC_iG}$  Triangle created by line the segments $\overline{\delta_kC_i}$, $\overline{\delta_kG}$, and $R_i$, p. 41

$r$  Angle created by the line segments $\overline{\delta_kC_i}$ and $\overline{\delta_kH}$ or $\overline{\delta_kC_i}$ and $\overline{\delta_kG}$, p. 42

$z$  Angle created by the line segments $R_i$ and $\overline{\delta_kH}$ or $R_i$ and $\overline{\delta_kG}$, p. 42

$d$  Angle created by the line segments $R_i$ and $\overline{\delta_kC_i}$, p. 42

E  Length of $\overline{\delta_kG}$ and $\overline{\delta_kH}$ 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

$\textbf{CF}_i(x_i^*, y_i^*)$  Coverage function associated with $(x_i^*, y_i^*)$, p. 50

$\textbf{TCF}$  Total coverage function for a DNS, p. 55

$tcf_{ijk}$  Total coverage function value, $cf_{ijk} \in \textbf{CF}_i(x_u, y_u)$, p. 34

$Q_m^*$  Optimal probability of acquisition for DNS $m$, 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. The views expressed in this work are not necessarily the views of the ONR.

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-Netted Sensors (DNS) in undersea surveillance and the need for an abstract representation 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 interests and support its allies [1]. Sensor networks for ASW began with the introduction of the DNS 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]. With the United States interest in harbor defense, homeland security, and the protection of critical undersea infrastructure, there has been interest in researching the science of Undersea Distributed-Netted Sensors (UDNS) to provide low cost, unmanned approaches to undersea surveillance.

1.2 Motivation for a Abstract representation View

Most of the research concerning UDNS has consisted of simulation programs, prototype testing, and demonstrations 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 enable optimization across alternative sensor modalities and varying sensor field compositions. These programs also depend on the order and location at which the simulated submerged contacts enter the area of interest.
We propose an abstract representation of undersea surveillance which provides several benefits over the traditional Monte Carlo and sensor placement approaches. First, the new approach allows users to investigate many DNS in a relatively short amount of time. Secondly, the abstract sensor model provides a parametric representation of the sensors, which captures many different types of sensor systems. Data from three parameter vectors that describe the aquatic environment, the sensors, and the submerged contacts is inputted into the model to determine the radius of the disk that represents the sensor. All three of these vectors contain variables that affect the sensors abilities to acquire the contacts, whereas ‘acquire’ refers to the sensor’s collective abilities to detect, classify, localize, and tract (DCLT) the contacts.

Within our approach, 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 (PDFs) 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 configurations 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.

Our approach allows planners to analyze the DNS for a defined patrol time, which is the time during which sensors are monitoring the area. Also, it is assumed that submerged contacts travel at a constant course and speed. The patrol time and velocity of the submerged contacts are used to calculate the maximum distance that the contact may travel within the area of interest during the patrol time. The sensors are placed in the area of interest in response to the contacts’ speed and the PMF so that the sensors can acquire the contacts.

1.3 Research Objectives

The main objectives of this research are: (1) Develop a abstract representation approach in both a discrete and continuous case to determine the optimal DNS to DCLT submerged contacts in a defined two-dimensional area of either shallow or deep sea water; and (2) Build a computer program to test DNS 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 undersea surveillance, various types of sensors, technological challenges of DNS, 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.
Introduction
- Undersea Surveillance
- Motivation for Abstract representation View

Problem Definition
- Formal Definition of the Problem
- List of Assumptions

Methodology for Multiple Sensors
- Context and Assumptions
- Geometric Approach for Two Sensors
- Geometric Approach for N Sensors
- Explanation of Pareto Analysis

Literature Survey
- DNS & Sonar Systems
- Track Coverage Problem
- Set Covering Problem
- Occupancy Grid Approach
- Geometric Optimization Approach

Methodology for a Single Sensor
- Parameter Vectors
- Evaluation Metrics
- Acquisition Radii
- Coverage Function
- Geometric Approach for both Discrete & Continuous Cases

Experimentation
- MATLAB Program
- Results for single and multiple sensors
- Discussion of Lifecycle Costs
- Pareto Analysis

Conclusions
- Contributions
- Future Research

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 DNS 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 DNS 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 Distributed-Netted Sensors Overview

DNS can be utilized in many different environments, and the approach that is presented in this thesis is applicable to any of these environments. However, we focus our attention on undersea surveillance and sensing via UDNS. UDNS use gliders, sonobuoys, and unmanned underwater vehicles (UUVs) to provide unmanned options to monitor large areas of the ocean for short and long periods of time. The Persistent Littoral Undersea Surveillance Network (PLUSNet) is an autonomous UDNS 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 sensors depending on the mission of the platform. Gateway buoys are moored and are used to relay information between the submerged sensors and a control station located on shore. Seagliders also relay information within the system, but have diving capabilities which contribute to stealth in the form of intermittent surface expressions and environmental sensing. The main purpose of the fixed bottom sensors and other mobile UUVs are to detect and track the submarines [4, 5, 8]. The PLUSNet system will be revisited in §2.5 in regards to its technological challenges.

DNS for undersea surveillance are primarily used for the DCLT of multiple targets as they move within a specified area of interest [9]. The PLUSNet is an excellent example of a UDNS that is rapidly deployable and is not reliant upon a fixed infrastructure, 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 DNS program at DARPA has researched the technological components and challenges of DNS. A DNS 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 DNS 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 DNS can be a homogenous or heterogeneous collection of a large number of low performing sensors or consist of only a few high performing 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 for undersea surveillance: active and passive sonar, and electromagnetic. Other types, such as seismic, radar, infrared, electro-optical, and synthetic aperture radar are also used in DNS, but are more common with environmental monitoring.

### 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 as well as the required equipment to sense objects. The active sonar requires a transducer that generates the transmitted 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 the signal-to-noise ratio (SNR) sound level at which signals are detected. 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 indicated 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. Receiver Operating Characteristics Curves (ROC) are frequently used to set a desired DT. Planners select the probability of detection and probability of false alarms that are acceptable for a sensor and then use the ROC curves to determine the required DT that would result in these probabilities [1].

![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 [10]](image-url)
We focus on acoustic sonar in this thesis, but there are many other types of sensors that can by modeled by the acquisition footprint. Electromagnetic (EM) sensors are also used for the direct detection of contacts. An example of an electromagnetic system is the Magnetic Anomaly Detection (MAD) system found on many aircrafts. MAD searches a given area for any anomaly in the Earth’s magnetic field that may be produced by a target [11]. EM sensors are discussed briefly in this chapter to provide a more comprehensive discussion on sensory types. Sensors can also be seismic, infrared, hydrodynamic, and radar, just to name a few.

### 2.3.1 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 or vector sensors 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. 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 [12]. Figure 2.3 shows how passive sonar works.
The advantages of passive sonar include: (1) it cannot be 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 [15]; and (3) there are no reverberation sounds that could cover the signals from the targets, however ambient noise exists from weather, marine life, and shipping. The main disadvantage of passive sonar is that the performance of the sonar depends on the level of RN from the target.

2.3.2 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. For a monostatic active sonar, 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. The array of transducers for active sonar can be maneuvered so that the sensor has directivity via a technique known as beamforming. Beamforming is the positioning of the transducers so that the sonar listens in a particular direction, which helps the sensor to listen more closely for a signal while ignoring sounds that may be coming from other directions [16]. Figure 2.2 shows how active sonar works.

![Active Sonar Diagram](image)

**Figure 2.3: Active Sonar [17]**

The DT must be set to disregard the source level (SL) of the radiated sounds from the projector and the reverberation noise level (RNL), which is the amount of 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 reflect 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, 18]. Also, the
device can be either low frequency or high frequency, depending on the needs of the human
observer. High frequency can give better spatial resolutions but it is attenuated more strongly by
the environment, which results in a short detection range. Low frequency can propagate a long
way in deep water.

Also, the signals generated are generally either continuous wave (CW) or frequency modulated
(FM) pulses. CW pulses provide good Doppler frequency resolution, which is the ability to
detect changes in the sound wave frequencies as an object moves past the sensor and determine
the direction and speed of the object. However CW pulses have 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 [18].

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, however this is highly dependent
upon the environmental conditions and the nature of the contact [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.

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 [19].
2.3.3 Electromagnetic Detection Systems

Electromagnetic (EM) sensors work in a similar fashion to acoustic sonar. In active form, 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 vibration 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 [20]. 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 environment [21].

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 sensor 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 [11]. 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 [22].
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. Drifting sensors, such as sonobuoys and ONR’s Deep Water Active Distributed System (DWADS), and bottom sensors, such as the Reliable Acoustic Path Vertical Line Array (RAP VLA) and SOSUS, are considered stationary sensors since typically they are not equipped with motion control devices [23]. However, the drifting sensors are subject to ocean currents and can move due to these 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 localize, classify, and track the submerged contact. Many DNS 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 [24].

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 [25] 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 [25]. The authors in [26] 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 often less expensive and have lower energy consumption. Whether mobile or stationary sensors are better for undersea surveillance is still a topic of interest in the field of undersea surveillance 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 DNS

Technological problems that are associated with DNS 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 DNS 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 DNS 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 [11] 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 [19]. 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 [9]. The authors in [27] 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 DNS. 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 [21], 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 [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 [21]. Whichever communication environment is employed, it must be able to endure the demands of time, bandwidth, and energy created by the DNS in dynamic environments [2].

The overall management of the DNS is a wide topic in the research community [28]. 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 environment access control (MAC) protocols are used to coordinate the communication across sensors. In [29], 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 DNS. 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 [30] 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 2.4. A $k$-coverage cone is formed by the intersection of $k$ separate coverage cones that share the same vertex.

![Coverage cones for track coverage](image)

**Figure 2.4: Coverage cones for track coverage [30]**
The work [24] 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 DNS.

Some of the concepts that are presented in [30] are related 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 [30]. 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 [31] 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. 2.5).
Voronoi diagrams are also used frequently in algorithms for the coverage problem. The work [32] 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 [33] implement the Voronoi diagrams in Lloyd algorithms to coordinate mobile sensors.

Some of the techniques presented in [31] 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. However, rather than determining if the area is k-covered, as in [30], 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 [34]. These maps are integrated for all the sensors in a network to develop the complete map of an area. The author in [34] suggests that the occupancy grid approach can be used to derive deterministic higher-level geometric representations of the DNS 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 [35], 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 [36] and [37] also use occupancy grid maps to plan the paths of mobile robots.

Although we do not sub-divide the area of interest into cells, we do apply the concept of ‘occupied’ or ‘covered’ cells for the Coverage Function (CF) of a sensor. The CF is a 2D matrix whose elements correspond to the combinations of headings and offset that can occur in the area of interest. Each element of the CF matrix can be thought of as a ‘cell’ of the matrix. The cell is ‘covered’ 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 [38] 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 $C_{\text{free}}$ and is decomposed into two types of cells. “A cell is defined as a closed and bounded subset of $C_{\text{free}}$ within which a robotic sensor path can be easily generated” [38].

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 $k$ has the property that for every configuration $q_i$ that is included in that cell $k$ 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 [36], 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

<table>
<thead>
<tr>
<th>Approach</th>
<th>Part of the Geometric Approach that was Influenced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track Coverage Problem</td>
<td>Coverage Cones and Definitions of Submerged Contact Tracks</td>
</tr>
<tr>
<td>Set Covering Problem</td>
<td>Coverage Function and Boundaries for Headings</td>
</tr>
<tr>
<td>Occupancy Grid Approach</td>
<td>Structure of the Area of Interest and Coverage Function</td>
</tr>
<tr>
<td>Geometric Optimization Approach</td>
<td>Acquisition Footprint, PMF for Headings and Offsets</td>
</tr>
</tbody>
</table>
2.6.5 Need for the New Geometric Approach

Provided a DNS, the approaches discussed above can optimally position the sensors, but they fail to address the problem of what DNS to choose in the first place. These approaches do not explore the different sensing modalities and sensor field compositions that yield the best performance. They also fail to address several key performance issues concerning the science of DNS, such as the cost effectiveness of mobility, the optimal speed of a sensor, and whether a DNS composed of a few high-performance sensors is better than a DNS consisting of many less expensive, low-performance sensors. Our novel geometric approach uses some of the concepts that are present in the approaches discussed above to place the sensors, but then extends beyond the analysis of a single DNS.

Our approach allows for a rapid analysis of many DNS, which can consist of heterogeneous mixtures of sensors. For each DNS, we place the sensors to maximize the acquisition of the submerged contacts. This ‘acquisition’ refers to a sensor’s ability to detect, classify, localize, and track (DCLT) the contacts. Literature has mainly focused on a subset of these abilities, especially detection and tracking as is evident in the various works presented in Chapter 2. Our approach takes a broader view by collectively referring to these abilities as a single ability to acquire the contacts via an abstract model of the sensors. Many different sensors can be quickly analyzed within a DNS simply by changing parameters in our model. The DNSs probabilities of acquisition are then plotted against their costs, and a Pareto analysis is then performed to find the optimal DNS for a given environment and type of submerged contacts. Our approach allows planners to compare many different DNS quickly to determine the optimal one.
Chapter 3: DNS 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 DNS is comprised of a variety of sensors that may be stationary or mobile and can be equipped with passive or active sonar or electromagnetic 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. Although sensor arrays have directivity, we choose an abstract model to represent the acquisition footprints of the sensors as disks since a sensor with directivity can sweep its beams through the space periodically.

The following three research problems are addressed in this thesis:

1. Develop an abstract DNS model that allows for rapid exploration of many different types of sensors by adjusting parameter values (Abstract Model).
2. Develop an optimization problem to determine the sensor locations within a DNS in order to achieve the best performance that is possible for the given environment (DNS Performance Problem).
3. Perform a Pareto analysis to determine the optimal DNS for the given environment (Pareto Analysis).

Problem Definition 3.1 (Abstract Model). Consider a set of parameters that describe the sensors, environment, and submerged contacts. These parameters can be adjusted to represent a variety of sensors. Each sensor is modeled as a disk, whose radius is calculated from the parameters.

Three parameter vectors are defined to identify the sensors, environment, and submerged contacts. The sensors are modeled as disks that represent the range at which the submerged contacts can be acquired. These disks are referred to as ‘acquisition footprints’ and their radii
reflect the ability of the sensors to acquire the contacts. A larger radius means that the sensor can acquire more contacts than a sensor with a smaller radius. A sensor’s detection threshold, array gain, source level, flow noise, and self noise can affect the radius. The environment parameters indicate whether the area of interest is in deep or shallow waters and defines the various noise levels in the water that would affect the acoustic sensors. The submerged contact parameters define the velocity, radiated noise, and target strength of the most difficult contact to acquire, which can also affect the radius of the acquisition footprint. These abstract models of the sensors are then used in the DNS performance problem.

**Problem Definition 3.2** (DNS Performance Problem). Consider a two-dimensional regular area of interest of either shallow water or deep sea. For this area, define a threat-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 $N$ 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. It does not matter the number of contacts that are expected since the PMF is used to describe the contacts.

We use a geometric approach implemented in a program developed in MATLAB to perform the track coverage optimization in the DNS Performance Problem. The length and width of the area are in terms of nautical miles (nm) and are defined by the user. It is assumed that submerged contacts can only enter the area through the border and travel along a threat axis. 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.
Given the area of interest and the PMF describing the submerged contacts’ tracks, the MATLAB program determines the placement of sensors for each DNS. We use the results of the program to perform a Pareto analysis to determine the optimal DNS for the given area of interest and the submerged contacts’ PMF.

**Problem Definition 3.3 (Pareto analysis)** Suppose that for each DNS tested its probability of acquisition and the lifecycle cost are recorded. Perform a Pareto analysis using these two metrics to determine the optimal DNS 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 that can be either heterogeneous or homogeneous, and a Pareto analysis concerning two sensor DNS is performed as an example. The approach can be applied to
N sensors, and the MATLAB program that is developed can handle N heterogeneous sensors. The concept of placing N sensors sequentially is explained in detail in this thesis. However, the sensors in the model cannot be bistatic, multistatic, of track residual signatures, or mobile. The abstract models for these sensors will be part of future research.

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 Sensors

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

1. The sensors are 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. A sensor is able to DCLT a submerged contact if that contact is within the acquisition footprint.

3.2.2 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 environment parameter vector and its effects on sensor footprints are included in the calculations of the radii of the footprints.
2. The observation space is 2D.
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. It is too time consuming for the submerged contact to travel around the border to enter at the sides.

4. Sensors are stationary due to one of the following reasons.
   a. They are anchored sensors.
   b. They are drifting sensors that are in an area with no current.
   c. They are station-keeping sensors in a constant current field, and the energy cost to keep the sensor stationary is included in the life cycle cost of the sensors.

### 3.2.3 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 we are concerned with acquiring the most challenging contact, and therefore only input the parameters associated with the most difficult contact.

2. A submerged contact’s heading distribution is independent of its offset.

3. Submerged contacts travel at a constant course and speed.
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 environment, 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 DNS 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 Threat 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 $\Delta x$ and $\Delta y$ respectively. There are a total of $U$ possible values of x, where $U=W/\Delta x$ and $\Delta x$ is the increment value of the x values. There are also a total of $V$ possible values of y, where $V=L/\Delta y$ and $\Delta y$ 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 $(x_u,y_v)$. 
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 Environment Parameter Vectors

The environment 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 environment has on sound propagation and sonar detection. There is one environment parameter vector per experiment.

\[
\begin{bmatrix}
\text{Transmission Loss} \\
\text{Ambient Noise Level} \\
\text{Reverberation Noise Level} \\
\text{Deep vs Shallow}
\end{bmatrix}
= 
\begin{bmatrix}
TL \\
ANL \\
RNL \\
MD
\end{bmatrix}
\]
The TL is the weakening of sound between the source and some point at a distance in the sea. The ANL of the environment 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 environment 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 environment 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 ($\delta_k$), heading ($\theta_j$), 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.

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

Throughout the program, the velocity of a submerged contact will remain constant. A submerged contact will enter the area of interest at a particular $\delta_k$ and $\theta_j$. The offset refers to the distance
between the Cartesian Frame’s origin 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 ($v_{\text{maxi}}$), 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 sensor.

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

The DT$_i$ is determined for each sensor based on the inputs for the environment 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<sub>i</sub>. FN<sub>i</sub> is created by the hydrodynamic flow around the sensor, while the SN<sub>i</sub> is created by the sensor itself.

The ν<sub>maxi</sub> refers to the maximum velocity that the sensor can reach and shows the mobility capability of the sensor; a stationary sensor will have a ν<sub>maxi</sub> = 0. In order to ensure that the optimal solutions are tractable, cost must be included in the model. The BC<sub>i</sub> is according to the type of sensor (stationary vs. mobile), the sensing capabilities (e.g. passive sonar vs. electromagnetic), and the life of the sensor. The SC<sub>i</sub> refers to the cost of deployment, energy consumption, maintenance, and retrieval.

4.3 Evaluation Metrics

There are two metrics for which each DNS is evaluated. The probability of acquisition metric is maximized in the optimization model. The lifecycle cost is recorded for each DNS 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 DNS.

4.3.1 Probability of Acquisition

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

Event A: A submerged contact will enter the area S at δ<sub>k</sub>, θ<sub>j</sub> 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<sub>j</sub> and offset δ<sub>k</sub> is denoted as o<sub>jk</sub>, and occurrence matrix O is the collection of these combinations, that is O = [o<sub>jk</sub>]. The development of 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,

\[
\mathbf{O} = \begin{bmatrix}
\theta_1 & 0.25 \\
\theta_2 & 0.25
\end{bmatrix}
\]

(4.1)

The coverage function (CF) for a sensor located at position \((x_u, y_v)\) is denoted as \(\mathbf{CF}_i(x_u, y_v)\) and has elements \(c_{f_{ijk}}(x_u, y_v)\), that is \(\mathbf{CF}_i(x_u, y_v) = [c_{f_{ijk}}(x_u, y_v)]\). 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 \(\theta_j\) and \(\delta_k\) 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:

\[
\mathbf{CF}_i(x_u, y_v) = \begin{bmatrix}
\theta_1 & 1 \\
\theta_2 & 0
\end{bmatrix}
\]

(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 \(\mathbf{Q}_i(x_u, y_v) = [q_i(x_u, y_v)]\). The matrix \(\mathbf{Q}_i(x_u, y_v)\) is the Hadamard product of \(\mathbf{O}\) and \(\mathbf{CF}_i(x_u, y_v)\).

\[
q_i(x_u, y_v) = P(A \cap B) = P(A) * P(B) = o_{jk} * c_{f_{ijk}}(x_u, y_v) \quad \forall u \in U, \forall v \in V
\]

(4.3)

From the example, matrix \(\mathbf{Q}_i(x_u, y_v)\) is calculated as follows:

\[
\mathbf{Q}_i(x_u, y_v) = \mathbf{O} \circ \mathbf{CF}_i(x_u, y_v) = \begin{bmatrix}
0.25 & 0.25 \\
0.25 & 0.25
\end{bmatrix} \begin{bmatrix}
1 \\
0
\end{bmatrix} = \begin{bmatrix}
0.25 & 0.25 \\
0 & 0
\end{bmatrix}
\]

(4.4)

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

\[
Q_i(x_u, y_v) = \sum_j \sum_k (q_i(x_u, y_v)) = \sum_j \sum_k (o_{jk} * c_{f_{ijk}}(x_u, y_v))
\]

(4.5)
Therefore, the $Q_i(x_u, y_v)$ for our example is 0.75. For an entire DNS, 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 DNS 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 DNS:

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

### 4.4 Submerged Contact Tracks

The tracks for submerged contacts may be considered straight, which entails constant course and speed. A submerged contact’s $\nu$ 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 $\pi$, 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 $\pi/100$ 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 $\theta = [\theta_j]$ is the set of all the possible values of the heading, while
the vector $\delta = [\delta_k]$ is the set of all the possible values of the offset. The PDF values are calculated at each possible heading value in $\theta$ and $\delta$.

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](image)

$$
OE_k + UE_k = \sum_{k=1}^{C} \frac{1}{2} (P_k - P_{k-1}) \Delta \delta + \sum_{k=C+1}^{K} \frac{1}{2} (P_k - P_{k-1}) \Delta \delta
$$

$$
= \frac{1}{2} (P_K - P_1) \Delta \delta = 0
$$

(4.7)

The PMF of all the heading and offset combinations is defined by the occurrence matrix $O$.

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

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

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 $(x_i, y_i)$ with radius $R_i$. This radius is determined via the sonar equation and several of the parameters from the environment, 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 DNS in littoral environments. Equation 4.10 is used when $MD = 1$ (indicator for deep sea), and equation 4.11 is used when $MD = 0$ (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).
Once the detection range $r$ is determined, an acquisition coefficient 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, $SL = 230 \, dB$, $TS = 25 \, dB$, $NL = 70 \, dB$, $AG = 30 \, dB$, 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 $R_1 = 0.80*15nm = 12nm$. 

Figure 4.3: Nomogram for Computing Transmission Loss [1]
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:

\[ RN - TL = NL - AG + DT \]  

(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 Sensor Acquisition Radius

There is a limited amount of literature on how to calculate the range of detection for electromagnetic 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 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 \( C_i(x_{ul}, y_{ul}) \) 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 based 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 \( O \) matrix.

4.6.1 Range of \( \delta_k \) Values that Can Result in a Coverage Function Value of One

Any submerged contacts whose offset, \( \delta_k \), 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.

\[
D = v * T
\]  

(4.13)

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

\[
Z = \sqrt{y_{ul}^2 + (x_{ul} - \delta_k)^2}
\]  

(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

\[ Z = R_i + D \]  \hspace{1cm} (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.

![Diagram showing the relationship between sensor position, offset, and coverage](image)

Figure 4.4: Maximum \( \delta_k \) value at which a sensor at \( C_i \) can have a CF value of one

### 4.6.2 Calculations of the Ranges of \( \theta_j \) When \( Z \geq D \)

For values of \( \delta_k \) that result in \( Z \leq R_i + D \), the lower and upper bounds, which are denoted as \( \alpha_{tuv} \) and \( \beta_{tuv} \) 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 \( \delta_k \). 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 \( \delta_k G \) creates the angle \( \alpha_{tuv} \) with the border, and the line segment \( \delta_k H \) creates the angle \( \beta_{tuv} \) with the border. The coverage cone created by the sensor’s acquisition radius and the offset \( \delta_k \) is bisected by the line segment \( \delta_k C_i \). 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 $\overrightarrow{\delta_k G}$ and $\overrightarrow{\delta_k H}$ 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 $\delta_k$](image)

Suppose that $Z \geq D$, then the values of $\alpha_{uvw}$ and $\beta_{uvw}$ depend on the distance $D$. In figure 4.6, the distance $D$ acts as the radius of a semi-circle originating at $\delta_k$. 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 $\overrightarrow{\delta_k G}$ and $\overrightarrow{\delta_k H}$ 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 $Z \geq D$](image)
The angle \( r \) will be used in the calculations of the angles \( \alpha_{iuv} \) and \( \beta_{iuv} \). Angle \( z \) is determined first using equation 4.16 and then used in equation 4.17 to calculate the angle \( r \).

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

\[
r = \sin^{-1} \left[ \frac{R_i \sin (z)}{Z} \right] \tag{4.17}
\]
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 $\alpha_{iuv}$ and $\beta_{iuv}$, respectively. Equation 4.19 will always hold true regardless of the relationship between $\delta_k$ and $x_u$.

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

$$\beta_{iuv} = \alpha_{iuv} + 2r$$  \hspace{1cm} (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 $\alpha_{iuv}$ change as the relationship between $\delta_k$ and $x_u$ changes.

$$\alpha_{iuv} = \pi - \tan^{-1}\left(\frac{y_i}{\delta_k - x_u}\right) - r$$  \hspace{1cm} (4.20)
Equation 4.20 reduces to equation 4.21 when $\delta_k = x_u$, which results from taking the arctangent of an undefined value.

$$\alpha_{uv} = \frac{\pi}{2} - r$$ (4.21)

In summary, the functions of $\alpha_{iv}$ and $\beta_{iv}$ are as follows for when $Z \geq D$:

$$\alpha_{iv} (\delta_k) = \begin{cases} \tan^{-1} \left( \frac{y_v}{x_u - \delta_k} \right) - \sin^{-1} \left[ R_i \sin \left( \cos^{-1} \left( \frac{R_i^2 + D^2 - Z^2}{2R_i D} \right) \right) \right] & \text{for } A \leq \delta_k < x_u \\ \frac{\pi}{2} - \sin^{-1} \left[ R_i \sin \left( \cos^{-1} \left( \frac{R_i^2 + D^2 - Z^2}{2R_i D} \right) \right) \right] & \text{for } \delta_k = x_u \\ \pi - \tan^{-1} \left( \frac{y_v}{\delta_k - x_u} \right) - \sin^{-1} \left[ R_i \sin \left( \cos^{-1} \left( \frac{R_i^2 + D^2 - Z^2}{2R_i D} \right) \right) \right] & \text{for } x_u < \delta_k \leq B \end{cases}$$ (4.22)
4.6.3 Calculations of the Ranges of $\theta_j$ When $Z < D$

Suppose that $Z < D$, then the values of $\alpha_{iuv}$ and $\beta_{iuv}$ 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 $\alpha_{iuv}$ and $\beta_{iuv}$ are no longer located on the circumference of the semi-circle with radius $D$ and centered at $\delta_k$. 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 $\Delta\delta_kC_iH$ and $\Delta\delta_kC_iG$ become right triangles. Figure 4.12 shows the coverage cone created when $Z < D$.

![Diagram](image-url)
The length of $\delta_k G$ and $\delta_k H$ is calculated by equation 4.24.

$$E = \sqrt{Z^2 - R_i^2}$$  \hspace{1cm} (4.24) 

Since the angle $\angle$ is a right angle, equation 4.17 simplifies to

$$r = \sin^{-1} \left[ \frac{R_i}{Z} \right]$$  \hspace{1cm} (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.

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

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

### 4.6.4 Coverage Function Calculations

The coverage function, \( \text{CF}_i(x_u, y_v) \) of a stationary sensor located at the coordinate \((x_u, y_v)\) is a \( J \times K \) matrix that contains Boolean values for a combination of \( \theta_j \) and \( \delta_k \) values. That is \( \text{CF}_i(x_u, y_v) = [cf_{ijk}(x_u, y_v)] \). An element of \( \text{CF}_i(x_u, y_v) \) has a value of 1 for a particular \( \delta_k \) and \( \theta_j \) if that \( \theta_j \) is between the \( \alpha_{iuv} \) and \( \beta_{iuv} \) associated with that \( \delta_k \):

\[
cf_{jk}(x_u, y_v) = \begin{cases} 
1 & \text{if } \alpha_{iuv} \leq \theta_j \leq \beta_{iuv} \text{ for } \delta_k \\
0 & \text{otherwise}
\end{cases}
\tag{4.28}
\]

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. An example of a coverage function for a sensor whose \( R_i = 15\)nm and position is \((51, 10)\) in a 100 nm\(^2\) area is shown in figure 4.14. The red indicates which offset and heading values are covered by the sensor located at \((51, 10)\), while the blue indicates the combinations that are not covered.

Notice that the coverage function covers the most number of headings for the offset values that are close to the sensor’s x coordinate. Logically, the number of heading values for a specific offset value the can be covered by the sensor decreases as the offset values move further away
from the sensor’s x coordinate. The coverage function that covers the highest number of offset
and heading combinations that have a high probability of occurrence is the best coverage
function to choose for the sensor, and the sensor is then placed at the x, y coordinates associated
with that coverage function. The probability of acquisition will be greater with a coverage
function that covers many combinations of offsets and headings that have a high probability of
occurrence.

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

1: For a sensor with acquisition radius \(R_i\)
2: Set \(\delta_k = A\) and \(\theta_j = 0\)
3: While \(\delta_k < B\)
4: Calculate \(Z\)
5: Calculate \(\alpha_{iuv}(\delta_k)\) and \(\beta_{iuv}(\delta_{ik})\)
6: While \(\theta_j < \pi\)
7: If \(Z > R_i + D\) Then
8: \(c_{ijk}(x_u, y_v) = 0\) for \(\theta_j, \delta_k\)
9: Else If \(Z \leq R_i\) Then
10: \(c_{ijk}(x_u, y_v) = 1\) for \(\theta_j, \delta_k\)
11: Else If \(R_i < Z \leq R_i + D\) Then
12: If \(\alpha_{iuv} \leq \theta_j \leq \beta_{iuv}\) Then
13: \(c_{ijk}(x_u, y_v) = 1\) for \(\theta_j, \delta_k\)
14: Else
15: \(c_{ijk}(x_u, y_v) = 0\) for \(\theta_j, \delta_k\)
16: End If
17: End If
18: Increase \(\theta_j\) by \(\Delta\theta\)
19: End While
20: Increase \(\delta_k\) by \(\Delta\delta\)
21: End While
22: 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 4.15. The double lined boxes represent the inputs to the model, while the single lined boxes care calculations within the geometric approach.
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 \(Q_i(x_u, y_v)\). This matrix \(Q_i(x_u, y_v)\) for sensor \(i\) is formed by taking the Hadamard product of the occurrence matrix \(O\) and \(CF_i(x_u, y_v)\) matrices and has the a size of \(J \times K\).

\[
Q_i(x_u, y_v) = O \circ CF_i(x_u, y_v)\\
= \begin{bmatrix}
o_{11} & \cdots & o_{1K} \\
\vdots & \ddots & \vdots \\
o_{j1} & \cdots & o_{jK}
\end{bmatrix} \begin{bmatrix}
cf_{11} & \cdots & cf_{1K} \\
\vdots & \ddots & \vdots \\
cf_{j1} & \cdots & cf_{jK}
\end{bmatrix} = \begin{bmatrix}
o_{11}cf_{11} & \cdots & o_{1K}cf_{1K} \\
\vdots & \ddots & \vdots \\
o_{j1}cf_{j1} & \cdots & o_{jK}cf_{jK}
\end{bmatrix}
\]  
(4.29)
The probability of acquisition for sensor $i$ located at $(x_w, y_v)$ is the sum of the elements of $Q_i(x_w, y_v)$ (also known as the Frobenius inner product):

$$Q_i(x_w, y_v) = \sum_j \sum_k (a_{jk} \ast c_{ijk}(x_w, y_v))$$ (4.30)

To determine the optimal position of the sensor, an exhaustive search of $Q_i(x_w, y_v)$ 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 $Q_i(x_w, y_v)$ is calculated, it is compared to $Q^*_i(x^*_i, y^*_i)$, and if $Q_i(x_w, y_v) > Q^*_i(x^*_i, y^*_i)$, then $Q_i(x_w, y_v)$ becomes the new $Q^*_i(x^*_i, y^*_i)$. Once all $(x_w, y_v)$ combinations have been investigated, $Q^*_i(x^*_i, y^*_i)$ is reported. The $CF^*_i(x^*_i, y^*_i)$ 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 environment and submerged contact are defined the same for all the DNS under consideration. Therefore, the matrix $O$ is the same for all the DNS tested.

As an example, suppose that we are investigating the performance of a sensor with $R_i = 2\text{nm}$ in a 10x10nm area of water to track a submerged contact travel at 5 knots during a patrol time of 2 hours. Suppose that $\Delta \delta = \Delta x = \Delta y = 5\text{nm}$ 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 $O$ matrix:

$$O = \begin{bmatrix} 0 & 5 & 10 \\ \pi/2 & 1/9 & 1/9 & 1/9 \\ \pi & 1/9 & 1/9 & 1/9 \end{bmatrix}$$ (4.31)

Nine combinations of $(x_w, y_v)$ values are investigated to determine the optimal location of the sensor. A coverage function is calculated for each $(x_w, y_v)$ combination using algorithm 1. The probability of acquisition is then calculated for the $(x_w, y_v)$ 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

1: Define environment parameter vector
2: Define submerged contact parameter vector
3: Define distributions for the heading and offset of the submerged contact
4: Calculate the \( O \) matrix for the submerged contacts
5: Set \( Q_i^*(x_i^*, y_i^*) = 0 \)
6: For a sensor \( i \) with acquisition radius \( R_i \)
7: Set \( x_u = A \) and \( y_v = 0 \)
8: While \( x_u < B \)
9: While \( y_v < L \)
10: Calculate the \( CF_i(x_u, y_v) \) \{see Algorithm 1\}
11: Calculate \( Q_i \) for sensor positioned at \((x_u, y_v)\)
12: If \( Q_i(x_u, y_v) > Q_i^*(x_i^*, y_i^*) \) Then
13: \( Q_i^*(x_i^*, y_i^*) = Q_i(x_u, y_v) \)
14: End If
15: Increase \( y_v \) by \( \Delta y \)
16: Increase \( x_u \) by \( \Delta x \)
17: End While
18: End For
19: 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 \( \theta \) and \( \delta \). 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 \( \theta \) and \( \delta \) values. When the PDF for the heading is uniform, the CF is then only dependent on the \((x_u, y_v)\) coordinates and the \( \delta \) values.

\[
CF = f(x_u, y_v, \delta) = \frac{\beta - \alpha}{\pi} \tag{4.32}
\]
From equation 4.19,

$$\beta - \alpha = 2r = 2 \sin^{-1} \left[ \frac{R_l \sin(z)}{z} \right]$$  \hspace{1cm} (4.33)

Written in terms of $x_u, y_v,$ and $\delta,$ equation 4.33 becomes

$$\beta - \alpha = 2r = 2 \sin^{-1} \left\{ \frac{R_l \sin \left[ \cos^{-1} \left( \frac{R_l^2 + D^2 - y_v^2 + (x_u - \delta)^2}{2R_l D} \right) \right]}{\sqrt{y_v^2 + (x_u - \delta)^2}} \right\}$$  \hspace{1cm} (4.34)

And equation 4.32 then becomes

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

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

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

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

The global optimum $Q^*_l$ can then be found using $\nabla g$ 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 DNS. The problem is that of determining where to place these sensors so that they maximize their total probability of acquisition for the DNS. 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 DNS 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 DNS.

5.2 The Geometric Approach for Two Sensors

We first look at the simplest example of a DNS- 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 $(x_u, y_u)$ 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 $O$ matrix is as follows:

$$
O = \begin{bmatrix}
0 & 5 & 10 \\
\frac{\pi}{2} & \frac{1}{9} & \frac{1}{9} \\
\pi & \frac{1}{9} & \frac{1}{9} \\
\end{bmatrix}
$$  \hspace{1cm} (5.1)

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

$$
CF_1(0, 0) = \begin{bmatrix}
0 & 5 & 10 \\
\frac{\pi}{2} & 1 & 0 \\
\pi & 1 & 1 \\
\end{bmatrix}
$$  \hspace{1cm} (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 $O$ matrix to reflect the combinations that have been covered by the first sensor.

This new $O$ matrix is created by replacing the $o_{jk}$ of the $O$ 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 $O$ matrix (see algorithm 3). With this new $O$ matrix, the second sensor is then placed using algorithm 2. By transforming the $O$ 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 $O$ matrix becomes

$$
O = \begin{bmatrix}
0 & 5 & 10 \\
\frac{\pi}{2} & 0 & 0 \\
\pi & \frac{1}{9} & 1 \\
\end{bmatrix}
$$  \hspace{1cm} (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 DNS.

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 $O$ 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.

$$CF_1(5, 0) = \begin{bmatrix} 0 & 5 & 10 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.4)$$

$$CF_1(10, 0) = \begin{bmatrix} 0 & 5 & 10 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

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

$$TCF = CF_1(x^*_1, y^*_1) \cup CF_2(x^*_2, y^*_2) \quad (5.6)$$

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

$$Q = O \circ TCF \quad (5.7)$$

$$Q^*_m = \sum_j \sum_k (o_{jk} \ast tcf_{jk}) \quad (5.8)$$
Suppose that a third homogeneous sensor is available to be placed. The $O$ 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 $O$ matrix. For instance, let the second sensor from our example be located at $(10, 0)$. Then the $O$ matrix is updated to

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

(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 DNS, and the result is a lower bound for the total $Q_m^*$ of the DNS. 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 of placing the sensors sequentially.

<table>
<thead>
<tr>
<th>Algorithm 3. Pseudo code for the Geometric Approach for N Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Define environment parameter vector</td>
</tr>
<tr>
<td>2: Define submerged contact parameter vector</td>
</tr>
<tr>
<td>3: Define distributions for the heading and offset of the submerged contact</td>
</tr>
<tr>
<td>4: Calculate the original $O$ matrix for the submerged contacts</td>
</tr>
<tr>
<td>5: While $i \leq N$</td>
</tr>
<tr>
<td>6: Determine the optimal location $(x_{i\omega}, y_{i\omega})$ for sensor $i$</td>
</tr>
<tr>
<td>7: Transform the $O$ matrix by using the $CF_i(x_i^<em>, y_i^</em>)$</td>
</tr>
<tr>
<td>8: if $cf_{ijk}(x_i^<em>, y_i^</em>) = 1$, then $o_{jk} = 0$ for the new $O$ matrix</td>
</tr>
<tr>
<td>9: if $cf_{ijk}(x_i^<em>, y_i^</em>) = 0$, then $o_{jk}$ remains the same for the new $O$ matrix</td>
</tr>
<tr>
<td>10: End For</td>
</tr>
<tr>
<td>11: Calculate the $TCF$</td>
</tr>
<tr>
<td>12: Use the $TCF$ to calculate the total $Q_m^*$ for the DNS</td>
</tr>
<tr>
<td>13: Report $(x_i^<em>, y_i^</em>)$ for all the sensors, and $Q_m^*$ for the DNS</td>
</tr>
</tbody>
</table>
5.2 Geometric Approach for Simultaneous Placement of $N$ sensors

Instead of positioning the sensors sequentially, 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

Set of Coverage Functions for a Unique Sensor with $R_1$

Set of Coverage Functions for a Unique Sensor with $R_N$

Total CF = $CF_1(x_{u1}, y_{v1}) \cup CF_2(x_{u2}, y_{v2}) \ldots$

$CF_N(x_{u[N]}, y_{v[N]})$
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:

\[
\text{Max } \sum_j \sum_k (o_{jk} \cdot tcf_{jk})
\]

subject to

\[
tcf_{jk} = c_{f_{i,j,k}}(x_{u[i]}, y_{v[i]}) \lor c_{f_{i,j,k}}(x_{u[i+1]}, y_{v[i+1]}) \lor \ldots c_{f_{i,j,k}}(x_{u[N]}, y_{v[N]})
\]

\[
(x_{u[i]}, y_{v[i]}) \neq (x_{u[i+1]}, y_{v[i+1]}) \neq (x_{u[i+2]}, y_{v[i+2]}) \neq \ldots (x_{u[N]}, y_{v[N]})
\]

This optimization problem chooses the coverage functions whose union creates a TCF whose Hadamard product with the \(O\) 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. This approach is time consuming and computationally heavy.

Several experiments were performed to determine the difference between results from the sequential and simultaneous placement of sensors and their associated probability of acquisition. For each experiment, two homogenous sensors of radius 2nm were placed in an area of interest, whose dimensions are changed for each experiment. The PDF for the offset followed a triangular distribution whose peak coincides with the middle of the width of the area, while the heading PDF was uniform. The offset increment was set at 1nm, and the heading increment was set at 0.031415. The sensors were placed sequentially following algorithm 3. An exhaustive search was then used for the simultaneous placement of the sensors. The probability of acquisition was calculated for every combination of placements for the two sensors and the locations of the sensor that resulted in the highest probability of acquisition war recorded. Table 5.1 summarizes the results from these experiments.
Table 5.1: Sequential vs. Simultaneous Placement of Sensors

<table>
<thead>
<tr>
<th>Experiment</th>
<th>W</th>
<th>L</th>
<th>Sensor 1 Location</th>
<th>Sensor 2 Location</th>
<th>$Q^*$</th>
<th>Sensor 1 Location</th>
<th>Sensor 2 Location</th>
<th>$Q^*$</th>
<th>Difference between $Q^*$ Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>(4,1)</td>
<td>(8,0)</td>
<td>0.9158</td>
<td>(3,0)</td>
<td>(7,0)</td>
<td>0.9583</td>
<td>4.25%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>10</td>
<td>(9,1)</td>
<td>(14,1)</td>
<td>0.6966</td>
<td>(7,1)</td>
<td>(12,1)</td>
<td>0.7261</td>
<td>2.95%</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>10</td>
<td>(14,1)</td>
<td>(20,1)</td>
<td>0.5598</td>
<td>(11,1)</td>
<td>(18,1)</td>
<td>0.5792</td>
<td>1.94%</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>10</td>
<td>(19,1)</td>
<td>(26,1)</td>
<td>0.4713</td>
<td>(16,1)</td>
<td>(24,2)</td>
<td>0.4872</td>
<td>1.59%</td>
</tr>
</tbody>
</table>

The results show that the difference between the sequential and simultaneous placement of sensors is small. The difference decreases when the area becomes larger since the offset increment remains the same. Both sensors are positioned approximately in the same locations regardless of the type of placement. These results show that although the sequential placement does provide suboptimal results, the results are close to optimal. However, when many DNS are under consideration, it will be beneficial to achieve the optimal results by simultaneously placing the sensors. Therefore, as part of the future work of this research, the simultaneous placement of sensors should be developed in more detail.

5.3 Explanation of Pareto Analysis

Once the total $Q^*_m$ of the DNS is calculated for all DNS under consideration, a Pareto analysis is performed using the two metrics previously discussed in §4.3. The analysis is used to decide which DNS 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 DNS 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 DNS, their performance and cost are plotted on the graph.

A DNS is a Pareto improvement over another DNS if it has the same or higher $Q^*_m$ without increasing the cost. For instance, DNS A is a Pareto improvement over DNS B in figure 5.2 since both DNSs 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 DNS on the Pareto frontier would be strictly dominate over A, B, and C.

Figure 5.2: Metric space for DNS Analysis

When analyzing the set of $M$ DNSs, the best DNS is closest to the Pareto Frontier. This frontier may be unknown, but the $M$ DNS can be compared to one another to determine the optimal system. An example of the Pareto analysis for a set of five two-sensor DNS 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 DNSs 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 \(Q_l\) is then calculated and compared to the current \(Q_l^*\). If the new value of \(Q_l > Q_l^*\), \(Q_l\) replaces \(Q_l^*\). The program performs an exhaustive search of the \((x_u, y_v)\) combinations and reports the coordinates associated with \(Q_l^*\). 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 \(4^{25-4}\) 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: \(Q_l^*, x_l^*, y_l^*, \) and the processing time. A coding scheme is used to convert the four-level factor into two two-level factors [39]. 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

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

Table 6.2: Coding Schemes for Qualitative Four-level Factors

<table>
<thead>
<tr>
<th>Two-Level Factors</th>
<th>Four-Level Factor</th>
<th>Two-Level Factors</th>
<th>Four-Level Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>Heading PDF Type</td>
<td>C</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>2</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>4</td>
<td>+</td>
</tr>
</tbody>
</table>

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 $R_t = 15\text{nm}$, $\Delta \delta = 0.5\text{nm}$, $\Delta \theta = 0.03\text{ radians}$, $T = 4\text{ hours}$, and $v = 15 \text{nm/h (knots)}$. 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 $Q^*_x = 0.38231$ when the sensor is located at position (51, 10). As expected, the sensor was placed close to the border near the Threat axis. The sensor was positioned near the Threat 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](image)

The program then determines the $CF_t$ 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 $R_l = 15$nm and position is (51, 10)

The $Q_l$ matrix that is used to determine $Q_l$ 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 $Q_l$ values calculated at each $(x_w, y_w)$ coordinate for the single sensor. The red in figure 6.4 corresponds to a higher $Q_l$ value. Notice that the $Q_l$ 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 $Q_l$ Matrix
6.2.2 Summary of Results

All results from the $4^22^{5-4}$ 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 $O$, which affects the $Q^*_1$. 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.
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.

![Normal Plot of the Standardized Effects](image)

**Figure 6.7:** Normal Probability Plot for $x_1^*$

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

**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, $R_l$ 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).
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 result 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.

![Normal Plot of the Standardized Effects](image)

**Figure 6.11: Normal Probability Plot for Processing Time**

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

**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 DNS that include a variety of sensor combinations. Table 6.3 summarizes the DNS under consideration in our example. Homogenous and heterogeneous DNS 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. 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 DNS. The $Q^*$ for the DNS are recorded in table 6.4 along with the $x_i^*$, and $y_i^*$ for each sensor.

For all DNS, the length and width of the area of interest are each 100nm, $\Delta \delta = 1\text{nm}$, $\Delta \theta = 0.03$ radians, $T = 4$ hours, and $v = 15\text{ nm/h}$ (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>
<thead>
<tr>
<th>DNS</th>
<th>Sensor 1 Type</th>
<th>Sensor 2 Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Passive ($R_l = 15$)</td>
<td>Passive ($R_l = 15$)</td>
</tr>
<tr>
<td>2a</td>
<td>Passive ($R_l = 10$)</td>
<td>Passive ($R_l = 15$)</td>
</tr>
<tr>
<td>2b</td>
<td>Passive ($R_l = 15$)</td>
<td>Passive ($R_l = 10$)</td>
</tr>
<tr>
<td>3</td>
<td>Active ($R_l = 10$)</td>
<td>Active ($R_l = 10$)</td>
</tr>
<tr>
<td>4a</td>
<td>Active ($R_l = 10$)</td>
<td>Active ($R_l = 12$)</td>
</tr>
<tr>
<td>4b</td>
<td>Active ($R_l = 12$)</td>
<td>Active ($R_l = 10$)</td>
</tr>
<tr>
<td>5a</td>
<td>Active ($R_l = 10$)</td>
<td>Passive ($R_l = 15$)</td>
</tr>
<tr>
<td>5b</td>
<td>Passive ($R_l = 15$)</td>
<td>Active ($R_l = 10$)</td>
</tr>
</tbody>
</table>
6.3.1 Complete Results of DNS #1 with Plots

The MATLAB plots for the DNS #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.

Using this original $O$ 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 $O$ matrix is transformed and the second sensor is placed at (79, 5), and has a $Q_2^* = 0.18269$. 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 1 positioned 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 \( Q \) matrix that is used to determine \( Q_l \) (see fig. 6.17). The Frobenius inner product of \( Q \) is \( Q_m^* = 0.8224 \). Notice that \( Q_m^* = Q_1^* + Q_2^* \) since \( Q_2^* \) is calculated with the transformed \( O \) 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 DNS #1

Figure 6.17: MATLAB 3D and Pseudocolor Plots for the $Q$ Matrix
Figure 6.18 shows a spatial coverage map with the placement of the two sensors for DNS #1. Notice that the sensors are place close to the border so that the sensors can capture more submerged contacts at more offset values in regardless of the heading values associated with that offset value. Since the offset PDF is triangular with $C = 50$, the first sensor is place so that it covers the offset values of higher probability of occurrence. The second sensor is placed to right side of the area of interest since the first sensor was placed slightly to the left of the center of the width of the area. This configuration performs better than a configuration that uniformly places the sensors within the area since more of the headings and offset combinations that have a higher probability of occurrence will not be covered.

![Spatial Coverage Map for DNS #1](image)

**Figure 6.18: Spatial Coverage Map for DNS #1**

### 6.3.2 Summary of Results

All five DNS were analyzed using the MATLAB program. Heterogeneous DNS were tested twice to determine if the order in which the sensors are placed had any effect on the performance of the DNS. 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 DNS took 65 seconds.
Table 6.4: Results for all DNS from the MATLAB Program

<table>
<thead>
<tr>
<th>DNS (m)</th>
<th>$Q_m$</th>
<th>($x_1^<em>, y_1^</em>$) for Sensor 1</th>
<th>($x_2^<em>, y_2^</em>$) for Sensor 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8224</td>
<td>(49, 5)</td>
<td>(79, 5)</td>
</tr>
<tr>
<td>2a</td>
<td>0.74428</td>
<td>(49, 4)</td>
<td>(75, 5)</td>
</tr>
<tr>
<td>2b</td>
<td>0.79638</td>
<td>(49, 5)</td>
<td>(77, 4)</td>
</tr>
<tr>
<td>3</td>
<td>0.70407</td>
<td>(49, 4)</td>
<td>(72, 4)</td>
</tr>
<tr>
<td>4a</td>
<td>0.712315</td>
<td>(49, 4)</td>
<td>(73, 4)</td>
</tr>
<tr>
<td>4b</td>
<td>0.74255</td>
<td>(49, 4)</td>
<td>(74, 4)</td>
</tr>
<tr>
<td>5a</td>
<td>0.74428</td>
<td>(49, 4)</td>
<td>(75, 5)</td>
</tr>
<tr>
<td>5b</td>
<td>0.79638</td>
<td>(49, 5)</td>
<td>(77, 4)</td>
</tr>
</tbody>
</table>

Some interesting results were found in the experiments. For DNS #2b, the union of the CF #1 (fig. 6.14) and CF #2 (fig. 6.19) results in a small gap of coverage for a few offset values (see fig. 6.20). 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 DNS #2b is the same as that for sensor 1 in DNS #1.

The order in which the sensors are placed does affect the performance of the DNS, 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.
Another interesting observation is that all the first sensors were placed in the same x coordinate, since it 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 DNS to determine the optimal DNS. For the heterogeneous DNS, the larger \( Q_m^{*} \) values from the two experiments for that DNS are used in the Pareto analysis.

6.4 Discussion on Lifecycle Cost

The lifecycle cost of a DNS 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 DNS 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) [31, 40].

The sensing cost for the sensors includes the cost of deployment, energy consumption, maintenance, and retrieval. The author in [41] 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 DNS for this thesis example are all sonobuoys, we assume that the deployment cost is the same for all DNS. 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 DNS 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 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 DNS have the same number of sensors, this cost is the same for all the DNS in our example.
The lifecycle costs for the five DNS in our example are summarized in table 6.5. The values from [40] 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 weight by a rate of 2lb/5nm. For some perspective, an EM sensor, such as the QMax EM3, can cost and weighs 500 lbs [42].

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>
<thead>
<tr>
<th>DNS</th>
<th>Sensor 1 Base Cost ($)</th>
<th>Sensor 1 Sensing Cost ($)</th>
<th>Sensor 1 Total LC Cost ($)</th>
<th>Sensor 2 Base Cost ($)</th>
<th>Sensor 2 Sensing Cost ($)</th>
<th>Sensor 2 Total LC Cost ($)</th>
<th>Total LC for DNS ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>650</td>
<td>120</td>
<td>770</td>
<td>650</td>
<td>120</td>
<td>770</td>
<td>1,540</td>
</tr>
<tr>
<td>2</td>
<td>580</td>
<td>110</td>
<td>690</td>
<td>650</td>
<td>120</td>
<td>770</td>
<td>1,460</td>
</tr>
<tr>
<td>3</td>
<td>1,850</td>
<td>170</td>
<td>2,020</td>
<td>1,850</td>
<td>170</td>
<td>2,020</td>
<td>4,040</td>
</tr>
<tr>
<td>4</td>
<td>2,200</td>
<td>175</td>
<td>2,375</td>
<td>1,850</td>
<td>170</td>
<td>2,020</td>
<td>4,395</td>
</tr>
<tr>
<td>5</td>
<td>1,850</td>
<td>170</td>
<td>2,020</td>
<td>650</td>
<td>120</td>
<td>770</td>
<td>2,790</td>
</tr>
</tbody>
</table>

### 6.5 A Pareto Analysis of Two Sensor DNS

Data from tables 6.4 and 6.5 were plotted on the $Q_m^*$ vs. LC graph (see fig. 6.21) 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 DNS. From figure 6.21, DNS#3,4, and 5 cost more than the DNS #1 but do not provide more coverage than DNS#1 or 2. Therefore, all these DNS are strictly dominated by DNS #1 and #2. Between these two remaining possible DNS, 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 DNS #1 as the best system.
6.6 Analysis of the Number of Sensors vs. Increase in $Q^*$

There exist an optimal number of sensors for each DNS, where any additional sensors will increase the cost of the system, but will not increase the probability of acquisition. Figure 6.22 shows the results from an experiment, which consisted of adding homogenous sensors one at a time to a two-sensor DNS. The graph shows that the probability of acquisition increases in a non-linear fashion. The increase in the probability of acquisition is greater when only a few sensors are placed. However, the rate of the increase tends to decrease as more sensors are added. This means that adding one more sensor does not increase the probability of acquisition for the DNS as significantly once six sensors are part of the DNS.
Figure 6.22: $Q_m^*$ vs. Number of Sensors
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 abstract representation view of DNS 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 DNS were analyzed and compared to one another using a Pareto analysis in order to determine the optimal DNS for the given environment and submerged contacts.

7.2 Future Research

In the future, it will be beneficial to extend the abstract model developed in this thesis to include a wider variety of sensors, such as bistatic and multistatic active sonar, and tracking of residual signatures. This will allow for an even greater number of DNS that can be rapidly analyzed. The abstract model should be investigated more to determine how to replace the disk with a polygon. 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. Also, the continuous case can be developed more in depth for other distributions.

It will also be beneficial to investigate a closed-form function for the coverage function of sensors so that scaling rules can be developed. A closed-form coverage function will allow for a
more rapid approach to simultaneously placing N sensors without storing entire families of
coverage functions for every unique sensor.
References


Appendix A: Proofs for Equations

A.1: Proof of Equation (4.14)

Let point $\delta_k$ be located at coordinates $(\delta_k, 0)$ and point $C_i$ be located at coordinates $(x_u, y_i)$. By the Pythagorean Theorem, the length of the line segment $\delta_k C_i$, denoted as $Z$ is

$$\sqrt{y_i^2 + (x_u - \delta_k)^2}.$$
A.2: Proof of Statement 1 (Congruent Triangles)

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)

Let point $\delta_k$ be located at coordinates $(\delta_k, 0)$, point $C_i$ be located at coordinates $(x_i, y_i)$, 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 $\delta_k C_i$ have length $Z$, side $\delta_k H$ have length $D$, and side $HC_i$ have length $R_i$. Let $\angle z$ be opposite the side $\delta_k C_i$, $\angle d$ be opposite the side $\delta_k H$, and $\angle r$ be opposite the side $HC_i$. The angle $z$ is found by the cosine rule and algebra:

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

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

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

$$z = \cos^{-1} \left[ \frac{R_i^2 + D^2 - Z^2}{2R_iD} \right] \quad (4)$$
A.4: Proof of Equation (4.17)

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

\[
\frac{R_l}{\sin(r)} = \frac{Z}{\sin(z)} \tag{5}
\]

\[
\frac{\sin(r)}{R_l} = \frac{\sin(z)}{Z} \tag{6}
\]

\[
\sin(r) = \frac{R_l \sin(z)}{Z} \tag{7}
\]

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

A.5: Proof of Equation (4.18)

Let point \( \delta_k \) 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 \( \delta_k C_i \) have length \( Z \), side \( \delta_k x_u \)
have length $y_v$, and side $\overline{x_uC_i}$ have length $(x_u - \delta_k)$. The $\angle C_i \delta_k x_u$ contains the angles $\alpha_{iuv}$ 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

$$\tan(\angle C_i \delta_k x_u) = \frac{y_v}{(x_u - \delta_k)}$$  \hspace{1cm} (9)$$

By substitution,

$$\tan(\alpha_{iuv} + r) = \frac{y_v}{(x_u - \delta_k)}$$  \hspace{1cm} (10)$$

Then solving for $\alpha_{iuv}$,

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

$$\angle G \delta_k H = \angle G \delta_k C_i + \angle C_i \delta_k H$$  \hspace{1cm} (12)$$

By substitution,

$$\angle G \delta_k H = r + r = 2r$$  \hspace{1cm} (13)$$

Then, since

$$\angle x_u \delta_k H = \angle x_u \delta_k G + \angle G \delta_k H$$  \hspace{1cm} (14)$$

By substitution,

$$\beta_{iuv} = \alpha_{iuv} + 2r$$  \hspace{1cm} (15)$$
A.7: Proof of Equation (4.20)

Since $x_u B$ 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,

$$\tan(\angle x_u \delta_k C_i) = \frac{y_v}{\delta_k - x_u}$$  \hspace{1cm} (16)

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

By the addition of angles,

$$\angle x_u \delta_k B = \angle x_u \delta_k C_i + \angle C_i \delta_k G + \angle G \delta_k B$$  \hspace{1cm} (18)

By substitution,

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

Solving for $\alpha_{iuv}$,

$$\alpha_{iuv} = \pi - \tan^{-1} \left( \frac{y_v}{\delta_k - x_u} \right) - r$$  \hspace{1cm} (20)
A.8: Proof of Equation (4.21)

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

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

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

$$\alpha_{iuv} = \pi - \pi/2 - r = \pi/2 - r \quad (22)$$

A.9: Proof of Equation (4.24)

![Figure A.6: Geometry for Proofs of Equations 4.24 and 4.25](image)

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,

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

Solving for $r$,

$$r = \sin^{-1} \frac{R_i}{Z} \quad (24)$$
Appendix B: Results from Experiments

B.1 Results from Single Sensor Experiments

B.1.1 DOE Design

Table B.1: Experimental Design for a $4^22^{5-4}$, or otherwise known as a $2^{9-4}$, Design

<table>
<thead>
<tr>
<th>RunOrder</th>
<th>StdOrder</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>28</td>
<td>12</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>27</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>22</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>26</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>27</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>28</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>29</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>31</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>32</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>
### B.1.2 Raw Data Results

**Table B.2: Results from the $4^22^54$ Experiment**

<table>
<thead>
<tr>
<th>StdOrder</th>
<th>$Q^*$</th>
<th>$x^*$</th>
<th>$y$</th>
<th>Processing Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.34781</td>
<td>48</td>
<td>7</td>
<td>61.4533</td>
</tr>
<tr>
<td>2</td>
<td>0.26172</td>
<td>29.5-70.5</td>
<td>5</td>
<td>94.0392</td>
</tr>
<tr>
<td>3</td>
<td>0.24762</td>
<td>29-71</td>
<td>4</td>
<td>193.42</td>
</tr>
<tr>
<td>4</td>
<td>0.28631</td>
<td>37.5</td>
<td>7</td>
<td>2262.026</td>
</tr>
<tr>
<td>5</td>
<td>0.48684</td>
<td>31.5</td>
<td>3</td>
<td>1414.194</td>
</tr>
<tr>
<td>6</td>
<td>0.44157</td>
<td>33</td>
<td>4</td>
<td>264.0281</td>
</tr>
<tr>
<td>7</td>
<td>0.4252</td>
<td>31</td>
<td>5</td>
<td>126.0804</td>
</tr>
<tr>
<td>8</td>
<td>0.45235</td>
<td>29</td>
<td>4</td>
<td>40.7619</td>
</tr>
<tr>
<td>9</td>
<td>0.49184</td>
<td>49.5</td>
<td>4</td>
<td>1060.2149</td>
</tr>
<tr>
<td>10</td>
<td>0.45477</td>
<td>51</td>
<td>4</td>
<td>352.6645</td>
</tr>
<tr>
<td>11</td>
<td>0.43279</td>
<td>49.5</td>
<td>5</td>
<td>164.0512</td>
</tr>
<tr>
<td>12</td>
<td>0.45056</td>
<td>47</td>
<td>4</td>
<td>32.0185</td>
</tr>
<tr>
<td>13</td>
<td>0.47182</td>
<td>67</td>
<td>4</td>
<td>24.3025</td>
</tr>
<tr>
<td>14</td>
<td>0.45795</td>
<td>70.5</td>
<td>5</td>
<td>244.4389</td>
</tr>
<tr>
<td>15</td>
<td>0.4312</td>
<td>68</td>
<td>4</td>
<td>571.5725</td>
</tr>
<tr>
<td>16</td>
<td>0.43556</td>
<td>66.5</td>
<td>3</td>
<td>756.6166</td>
</tr>
<tr>
<td>17</td>
<td>0.39633</td>
<td>34.5-65.5</td>
<td>5</td>
<td>838.0124</td>
</tr>
<tr>
<td>18</td>
<td>0.40808</td>
<td>64</td>
<td>10</td>
<td>600.2492</td>
</tr>
<tr>
<td>19</td>
<td>0.38231</td>
<td>51</td>
<td>10</td>
<td>256.9355</td>
</tr>
<tr>
<td>20</td>
<td>0.36868</td>
<td>34-66</td>
<td>5</td>
<td>26.231</td>
</tr>
<tr>
<td>21</td>
<td>0.62414</td>
<td>34</td>
<td>5</td>
<td>34.3591</td>
</tr>
<tr>
<td>22</td>
<td>0.59105</td>
<td>36.5</td>
<td>5</td>
<td>174.3793</td>
</tr>
<tr>
<td>23</td>
<td>0.5815</td>
<td>33</td>
<td>5</td>
<td>387.0698</td>
</tr>
<tr>
<td>24</td>
<td>0.60072</td>
<td>32</td>
<td>5</td>
<td>1168.4944</td>
</tr>
<tr>
<td>25</td>
<td>0.63959</td>
<td>49</td>
<td>5</td>
<td>43.8466</td>
</tr>
<tr>
<td>26</td>
<td>0.60319</td>
<td>52</td>
<td>5</td>
<td>137.0071</td>
</tr>
<tr>
<td>27</td>
<td>0.58549</td>
<td>49</td>
<td>5</td>
<td>290.5038</td>
</tr>
<tr>
<td>28</td>
<td>0.60434</td>
<td>47</td>
<td>5</td>
<td>1548.9443</td>
</tr>
<tr>
<td>29</td>
<td>0.63036</td>
<td>65.5</td>
<td>5</td>
<td>2389.0686</td>
</tr>
<tr>
<td>30</td>
<td>0.5931</td>
<td>67</td>
<td>5</td>
<td>208.3632</td>
</tr>
<tr>
<td>31</td>
<td>0.57082</td>
<td>66</td>
<td>5</td>
<td>103.3004</td>
</tr>
<tr>
<td>32</td>
<td>0.59384</td>
<td>63</td>
<td>5</td>
<td>64.758</td>
</tr>
</tbody>
</table>
### B.1.3 Minitab Results for $Q^*$

Results after non-significant terms were removed:

**Factorial Fit: $Q^*$ versus A, B, C, D, E, H**

#### Estimated Effects and Coefficients for $Q^*$ (coded units)

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

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

#### Analysis of Variance for $Q^*$ (coded units)

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

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

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

Analysis of Variance for xstar (coded units)

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Seq SS</th>
<th>Adj SS</th>
<th>Adj MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>4</td>
<td>4116.8</td>
<td>4116.8</td>
<td>1029.20</td>
<td>40.73</td>
<td>0.000</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>162.0</td>
<td>162.0</td>
<td>162.00</td>
<td>6.41</td>
<td>0.018</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>3612.5</td>
<td>3612.5</td>
<td>3612.50</td>
<td>142.96</td>
<td>0.000</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>166.5</td>
<td>166.5</td>
<td>166.53</td>
<td>6.59</td>
<td>0.017</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>175.8</td>
<td>175.8</td>
<td>175.78</td>
<td>6.96</td>
<td>0.014</td>
</tr>
<tr>
<td>2-Way Interactions</td>
<td>2</td>
<td>1510.2</td>
<td>1510.2</td>
<td>755.08</td>
<td>29.88</td>
<td>0.000</td>
</tr>
<tr>
<td>C*D</td>
<td>1</td>
<td>1339.0</td>
<td>1339.0</td>
<td>1339.03</td>
<td>52.99</td>
<td>0.000</td>
</tr>
<tr>
<td>C*J</td>
<td>1</td>
<td>171.1</td>
<td>171.1</td>
<td>171.12</td>
<td>6.77</td>
<td>0.015</td>
</tr>
<tr>
<td>Residual Error</td>
<td>25</td>
<td>631.7</td>
<td>631.7</td>
<td>25.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>1</td>
<td>162.0</td>
<td>162.0</td>
<td>162.00</td>
<td>8.28</td>
<td>0.008</td>
</tr>
<tr>
<td>Pure Error</td>
<td>24</td>
<td>469.8</td>
<td>469.8</td>
<td>19.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>31</td>
<td>6258.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Normal Plot of the Standardized Effects
(response is xstar, Alpha = 0.05)

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

Residual Plots for xstar

Normal Probability Plot

Versus Fits

Histogram

Versus Order

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)**

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coef</th>
<th>SE Coef</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td>5.0625</td>
<td>0.1561</td>
<td>32.43</td>
<td>0.000</td>
</tr>
<tr>
<td>C</td>
<td>-1.1250</td>
<td>-0.5625</td>
<td>0.1561</td>
<td>-3.60</td>
<td>0.001</td>
</tr>
<tr>
<td>D</td>
<td>-1.0000</td>
<td>-0.5000</td>
<td>0.1561</td>
<td>-3.20</td>
<td>0.004</td>
</tr>
<tr>
<td>E</td>
<td>1.1250</td>
<td>0.5625</td>
<td>0.1561</td>
<td>3.60</td>
<td>0.001</td>
</tr>
<tr>
<td>H</td>
<td>1.0000</td>
<td>0.5000</td>
<td>0.1561</td>
<td>3.20</td>
<td>0.004</td>
</tr>
<tr>
<td>J</td>
<td>1.1250</td>
<td>0.5625</td>
<td>0.1561</td>
<td>3.60</td>
<td>0.001</td>
</tr>
<tr>
<td>C*D</td>
<td>1.0000</td>
<td>0.5000</td>
<td>0.1561</td>
<td>3.20</td>
<td>0.004</td>
</tr>
</tbody>
</table>

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)**

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Seq SS</th>
<th>Adj SS</th>
<th>Adj MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>46.375</td>
<td>40.000</td>
<td>9.2750</td>
<td>11.89</td>
<td>0.000</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>10.125</td>
<td>10.125</td>
<td>10.125</td>
<td>12.98</td>
<td>0.001</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>10.000</td>
<td>10.000</td>
<td>10.000</td>
<td>10.26</td>
<td>0.004</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>10.125</td>
<td>10.125</td>
<td>10.125</td>
<td>12.98</td>
<td>0.001</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>8.000</td>
<td>8.000</td>
<td>8.0000</td>
<td>10.26</td>
<td>0.004</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>10.125</td>
<td>10.125</td>
<td>10.125</td>
<td>12.98</td>
<td>0.001</td>
</tr>
<tr>
<td>2-Way Interactions</td>
<td>1</td>
<td>8.000</td>
<td>8.000</td>
<td>8.0000</td>
<td>10.26</td>
<td>0.004</td>
</tr>
<tr>
<td>C*D</td>
<td>1</td>
<td>8.000</td>
<td>8.000</td>
<td>8.0000</td>
<td>10.26</td>
<td>0.004</td>
</tr>
<tr>
<td>Residual Error</td>
<td>25</td>
<td>19.500</td>
<td>19.500</td>
<td>0.7800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>9</td>
<td>16.500</td>
<td>16.500</td>
<td>1.8333</td>
<td>9.78</td>
<td>0.000</td>
</tr>
<tr>
<td>Pure Error</td>
<td>16</td>
<td>3.000</td>
<td>3.000</td>
<td>0.1875</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>31</td>
<td>73.875</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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)

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

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)

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Seq SS</th>
<th>Adj SS</th>
<th>Adj MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>9</td>
<td>9049595</td>
<td>9049595</td>
<td>1005511</td>
<td>380.67</td>
<td>0.000</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>0.00</td>
<td>0.955</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>85</td>
<td>85</td>
<td>85</td>
<td>0.03</td>
<td>0.861</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0.00</td>
<td>0.973</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>0.03</td>
<td>0.866</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>11614</td>
<td>11614</td>
<td>11614</td>
<td>4.40</td>
<td>0.058</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>5022453</td>
<td>5022453</td>
<td>5022453</td>
<td>1901.44</td>
<td>0.000</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>2845425</td>
<td>2845425</td>
<td>2845425</td>
<td>1077.24</td>
<td>0.000</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>851324</td>
<td>851324</td>
<td>851324</td>
<td>322.30</td>
<td>0.000</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>318604</td>
<td>318604</td>
<td>318604</td>
<td>120.62</td>
<td>0.000</td>
</tr>
<tr>
<td>2-Way Interactions</td>
<td>10</td>
<td>3421343</td>
<td>3421343</td>
<td>3421343</td>
<td>129.53</td>
<td>0.000</td>
</tr>
<tr>
<td>A*B</td>
<td>1</td>
<td>1803670</td>
<td>1803670</td>
<td>1803670</td>
<td>682.85</td>
<td>0.000</td>
</tr>
<tr>
<td>A*C</td>
<td>1</td>
<td>570539</td>
<td>570539</td>
<td>570539</td>
<td>216.00</td>
<td>0.000</td>
</tr>
<tr>
<td>A*D</td>
<td>1</td>
<td>214633</td>
<td>214633</td>
<td>214633</td>
<td>81.26</td>
<td>0.000</td>
</tr>
<tr>
<td>A*E</td>
<td>1</td>
<td>22365</td>
<td>22365</td>
<td>22365</td>
<td>8.47</td>
<td>0.013</td>
</tr>
<tr>
<td>B*C</td>
<td>1</td>
<td>304419</td>
<td>304419</td>
<td>304419</td>
<td>115.25</td>
<td>0.000</td>
</tr>
<tr>
<td>B*D</td>
<td>1</td>
<td>113818</td>
<td>113818</td>
<td>113818</td>
<td>43.09</td>
<td>0.000</td>
</tr>
<tr>
<td>B*E</td>
<td>1</td>
<td>45773</td>
<td>45773</td>
<td>45773</td>
<td>17.33</td>
<td>0.001</td>
</tr>
<tr>
<td>C*D</td>
<td>1</td>
<td>66263</td>
<td>66263</td>
<td>66263</td>
<td>25.09</td>
<td>0.000</td>
</tr>
<tr>
<td>C*E</td>
<td>1</td>
<td>76632</td>
<td>76632</td>
<td>76632</td>
<td>29.01</td>
<td>0.000</td>
</tr>
<tr>
<td>D*E</td>
<td>1</td>
<td>203233</td>
<td>203233</td>
<td>203233</td>
<td>76.94</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>12</td>
<td>31697</td>
<td>31697</td>
<td>2641</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>31</td>
<td>1250263</td>
<td>1250263</td>
<td>1250263</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>
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 locations
%for a set of sensors 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
% BC(i) = base cost of a sensor (in dollars)
% SC(i) = sensing cost of a sensor (in dollars)
% TC(i) = total cost of a sensor (in dollars)
% TDSNCost = total cost of the DSN (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 + 1
% J = total number of heading values = pi/Jincrement + 1
% L = total number of y values = Length/Lincrement + 1
%
%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

Variables for the geometric approach calculations
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

---Start Inputs, Occurrence Matrix Formation, and Radius Calculations---
clear

%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 = pi/100; %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;
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.')
%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
110

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

%---------------------------------------------------------------------End Inputs---------------------------------------------------------------------
%
%---------------------------------------------------------------------Start Acquisition Range Calculations---------------------------------------------------------------------
disp(' ')

for i= 1:N

    str = ['Enter the following information for sensor ', num2str(i), '.'];
    disp(str)

    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); %Calculate the total cost for the sensor

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

end
TDSNCost = sum(TC); %Calculate the total cost of the DSN

%---------------------------------------------------------------------End Acquisition 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
    for k = 1:K;
        OffsetDist(k) = 1/K;
    end
elseif OffsetDistNum == 2
    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
    % calculate delta
    delta = (1 - sum(OffsetEstDist))/K;
elseif OffsetDistNum == 3
    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
    delta = (1 - sum(OffsetEstDist))/K;
end
%calculate the true probabilities
for k = 1:K;
    OffsetDist(k) = OffsetEstDist(k) + delta;
end

elseif OffsetDistNum == 4 %if the user wants to manually enter probabilities
%for offset values
%Prompt user to enter the probabilities of occurrence for offset values
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); %Check that sum of the OffsetDist Matrix is 1
%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
    end
    HeadingEstDist(j) = HeadingEstDist(j) * Jincrement;
end
%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)
    elseif (HeadingSet(j) > C3)
    end
    HeadingEstDist(j) = HeadingEstDist(j)*Jincrement;
end

%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('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

%display the heading distribution
%disp('The following is the distribution for the heading values:')
%HeadingDist
%Head = sum(HeadingDist);%Check that sum of the HeadingDist Matrix is 1
%Head
%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)); %Check that sum of the OMatrix is 1
%O

%disp('-------------------------------------------------------------------
%disp('Computing.....
%-------------------------------------
%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.

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 Qstar as 0
OMatrixNew = zeros(J,K); %Initialize the OMatrixNew as a Zero matrix
OMatrixNew = OMatrix; %Original OMatrix used to place first sensor
CFjoint = zeros(J,K); %Initialize the CFjoint as a Zero matrix

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)));
                        beta = alpha + (2*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));
                        beta = alpha + (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)) -
                        (2*asin((R*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z));
                        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*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z));
                            beta = alpha + (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)) -
                            (2*asin((R*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z));
                            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*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z));
                                beta = alpha + (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)) -
                                (2*asin((R*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z));
                                beta = alpha + (2*asin((R*sin(acos((R^2+D^2-Z^2)/(2*R*D))))/Z));
                        end
                    end
                end
            end
        end
    end
end
end
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`

`elseif` (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`

`elseif` (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`%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 produced 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;
  CFstar = CF;
`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 probability 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)
endif 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

% Record the CFstar for sensor i display of results
CFstarset(:, :, i) = CFstar;

% Change the Original OMatrix to OMatrixNew by replacing the O(j,k)
%values with zeros for the CF(j,k)=1
OMatrixNew = OMatrixNew.*(1-CFstar);

%Calculate the new CFjoint matrix, which combines the CFstars for all %sensors
CFjoint = CFjoint|CFstar;

%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
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');
end

if N>1
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');
end

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

for i=1:N;
figure('Name','Coverage Function of Sensor', 'NumberTitle', 'off')
subplot(2,1,1);surf(OffsetSet,HeadingSet,CFstarset(:,:,i), 'EdgeColor', 'none')
ylabel('Heading');
xlabel('Offset');
zlabel('Coverage Function Value');
zlim([0 1]);
subplot(2,1,2);pcolor(OffsetSet,HeadingSet,CFstarset(:,:,i))
ylabel('Heading');
xlabel('Offset');
end
```matlab
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');
ylim([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