CONTROL AND OPTIMIZATION OF TRACK COVERAGE IN UNDERWATER SENSOR NETWORKS

by

Kelli A. Crews Baumgartner

Department of Mechanical Engineering and Materials Science
Duke University

Date: ___________________________

Approved:

____
Silvia Ferrari, Ph.D., Supervisor

____
Devendra Garg, Ph.D.

____
Brian Mann, Ph.D.

____
Jeffrey Scruggs, Ph.D.

____
Xiaobai Sun, Ph.D.

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Mechanical Engineering and Materials Science
in the Graduate School of
Duke University

2007

ABSTRACT

(Engineering—Mechanical)

CONTROL AND OPTIMIZATION OF TRACK COVERAGE IN UNDERWATER SENSOR NETWORKS

by

Kelli A. Crews Baumgartner

Department of Mechanical Engineering and Materials Science
Duke University

Date: ______________________________

Approved:

____
Silvia Ferrari, Ph.D., Supervisor

____
Devendra Garg, Ph.D.

____
Brian Mann, Ph.D.

____
Jeffrey Scruggs, Ph.D.

____
Xiaobai Sun, Ph.D.

An abstract of a dissertation submitted in partial
fulfillment of the requirements for the degree
of Doctor of Philosophy in the Department of
Mechanical Engineering and Materials Science in the Graduate School of
Duke University

2007

Abstract

Sensor network coverage refers to the quality of service provided by a sensor network surveilling a region of interest. So far, coverage problems have been formulated to address area coverage or to maintain line-of-sight visibility in the presence of obstacles (i.e., art-gallery problems). Although very useful in many sensor applications, none of the existing formulations address coverage as it pertains to target tracking by means of multiple sensors, nor do they provide a closed-form function that can be applied to the problem of allocating sensors for the surveilling objective of maximizing target detection while minimizing false alarms. This dissertation presents a new coverage formulation addressing the quality of service of sensor networks that cooperatively detect targets traversing a region of interest, and is readily applicable to the current sensor network coverage formulations. The problem of track coverage consists of finding the positions of n sensors such that the amount of tracks detected by at least k sensors is optimized. This dissertation studies the geometric properties of the network, addressing a deterministic track-coverage formulation and binary sensor models. It is shown that the tracks detected by a network of heterogeneous omnidirectional sensors are the geometric transversals of non-translates families of disks. A novel methodology based on cones and convex analysis is presented for representing and measuring sets of transversals as closed-form functions of the sensors positions and ranges.

As a result, the problem of optimally deploying a sensor network with the aforementioned objectives can be formulated as an optimization problem subject to mission dynamics and constraints. The sensor placement problem, in which the sensors are placed such that track coverage is maximized for a fixed sensor network, is formulated as a nonlinear program and solved using sequential quadratic programming. The sensor deployment, involving a dynamic sensor network installed on non-maneuverable sonobuoys deployed in the ocean, is formulated as an optimization problem subject to inverse dynamics. Both a finite measure of the cumulative coverage provided by a sensor network over a fixed period of time and the oceanic-induced current velocity field are accounted for in order to optimize the dynamic sensor network configuration. It is shown that a state-space representation of the motions of the individual sensors subject to the current vector field can be derived from sonobuoys oceanic drift models and obtained from CODAR measurements. Also considered in the sensor model are the position-dependent acoustic ranges of the sensors due to the effects from heterogenous environmental conditions, such as ocean bathymetry, surface temporal variability, and bottom properties. A solution is presented for the initial deployment scheme of the non-maneuverable sonobuoys subject to the ocean's current, such that sufficient track coverage is maintained over the entire mission. As sensor networks are subject to random disturbances due to unforseen heterogenous environmental conditions propagated throughout the sensors trajectories, the optimal initial positions solution is evaluated for robustness through Monte Carlo simulations. Finally, the problem of controlling a network of maneuverable underwater vehicles, each equipped with an onboard acoustic sensor is formulated using optimal control theory. Consequently, a new optimal control problem is presented that integrates sensor objectives, such as track coverage, with cooperative path planning of a mobile sensor network subject to time-varying environmental dynamics.

Contents

List of Tables

List of Figures

Nomenclature

Symbols

A	:	Object's cross-sectional area
A_C	:	Measure of distinct area covered by n sensors
$A_C^{\max}$	:	Total area coverage
A_i	:	Area covered by sensor i, represented as a disk
A_p	:	Total segment area of the disk outside of $\mathcal{A}$
A_s	:	Total segment area of the overlapping portion of two disks
A_0	:	Union of n sensor areas
$\mathcal{A}$	:	Area of interest
a_y	:	Slope of $\mathcal{R}_\alpha(b_y)$
α	:	Angle formed by $\mathcal{R}$ with the x-axis
b_y	:	Intercept of $\mathcal{R}_\alpha(b_y)$
B_i	:	Logical array of positive track detection by sensor C_i
$\text{cone}(X)$	:	Cone generated by X
c_i	:	Chord length of circular segment i
C_d	:	Coefficient of drag on an object
$C(\mathbf{x}_i, r_i)$	:	Disk with radius r_i centered at $\mathbf{x}_i$
$\{C_1, ..., C_n\}$	:	Set of n disjoint non-translates disks in $\mathbb{R}^2$
$\det(\,\cdot\,)$	:	Matrix determinant
d_i	:	CPA range from i^{th} sensor-to-target

δb	:	Size of discretization increments
$\Delta \mathbf{v}_i$	:	Component of the fluid relative velocity vector past the object for $i = u, \ell$
f_d	:	Total drag on an object
F	:	Target source level
γ_i	:	Angle formed by $\hat{\mathbf{l}}_i$ with the x-axis
h_i	:	Perpendicular distance between the origin of sensor i and c_i
h_{ij}	:	Distance between the origins of sensors i and j
$\hat{h}_i$	:	Upper unit vector generating $K(C_i, y_0)$
$\hat{h}^*$	:	Upper unit vector generating $K_k(S_k, y_0)$
I_R	:	Index set of the reference axes-of-interest
I_S	:	Index set of S
$\mathbf{I}$	:	Identity matrix
$\mathcal{I}_y$	:	Interval representing the side of $\mathcal{A}$ along the y-axis
k	:	Number of detections required per track
$K(C_i, \mathbf{y}_0)$	:	Cone generated by C_i with origin y_0
$K_k(S_k, \mathbf{y}_0)$	:	k-Coverage cone of S_k with origin y_0
$\mathcal{K}_k(S, \mathcal{A})$	: :	Set of tracks traversing $\mathcal{A}$ and intersecting at least k sensors in S
$\mathcal{K}_k(S, \mathbf{y}_0)$	:	Set of tracks through y_0 and intersecting at least k sensors in S
$\hat{l}_i$	:	Lower unit vector generating $K(C_i, y_0)$
$\hat{l}^*$	:	Lower unit vector generating $K_k(S_k, y_0)$
L_1	:	Width of $\mathcal{A}$
L_2	:	Height of $\mathcal{A}$
λ_i	:	Angle formed by $\hat{\mathbf{h}}_i$ with the x-axis
$\Lambda(S, \mathbf{y}_0)$	:	Set of unit vectors $\hat{\mathbf{l}}_i$ generating the coverage cones of all disks

in S with origin $\mathbf{y}_0$

m	:	Number of possible k-subsets in S
$M_{\imath\jmath}$	:	Cross product matrix of $\hat{\mathbf{l}}^*$ and $\hat{\mathbf{h}}^*$
μ	:	Lebesgue measure on a set
n	:	Number of sensors in S
N_1	:	Number of increments in $\mathcal{I}_y$ and $\mathcal{I}_{y'}$
N_2	:	Number of increments in $\mathcal{I}_x$ and $\mathcal{I}_{x'}$
$\Omega(S, \mathbf{y}_0)$	:	Set of unit vectors $\hat{\mathbf{h}}_i$ generating the coverage cones of all disks in S with origin y_0
$Pr^k_{\mathcal{A}}(X_S)$	:	Probability function of track detection
$\psi(S_k, \mathbf{y}_0)$	:	Opening angle of $K_k(S_k, \mathbf{y}_0)$
Q_i^-	:	Clockwise rotation matrix by an angle θ_i
Q_i^+	:	Counterclockwise rotation matrix by an angle θ_i
r_i	:	Range of sensor i
R_S	:	Set of sensor ranges for S
$\mathbf{R}$	:	Real-symmetric positive definite control weighting matrix
$\mathcal{R}_\alpha(b_y)$	:	Ray with y-intercept b_y and slope $\tan\alpha$
ρ	:	Fluid density
$\rho(S_k, \mathbf{y}'_0)$	:	Opening angle of $K_k(S_k, \mathbf{y}'_0)$
s_i	:	Arc of the circular segment i
S	:	Set of n sensors
S_k	:	Set of k sensors
S_k^j	:	j^{th} k-subset of S
$t_{i,CPA}$	:	Time period of the CPA for sensor i
T	:	(1) Set of single positive detection measurements (2) Set of CODAR training data

T_k	:	Logical array of k-track coverage by network S
$\mathcal{T}^k_{\mathbf{y}_0}(X_S)$	:	Measure of the set of tracks in $\mathcal{K}_k(S, y_0)$
$\mathcal{T}^k_{\mathcal{A}}(X_S)$	:	Measure of the set of tracks in $\mathcal{K}_k(S, \mathcal{A})$
$\mathcal{T}^{max}_{\mathcal{A}}$	:	Total track-coverage
θ_i	:	Half the opening angle of $K(C_i, y_0)$
θ_{ij}	: :	Central angle of the segment area i due to the overlap of disks i and j
ϑ	:	Detection threshold
$\mathbf{u}_i$	:	(1) Water velocity components for $i = u, \ell$ (2) Control components for sensor i
V	:	Magnitude of the fluid relative velocity vector past the object
$\boldsymbol{\upsilon}$	:	Sonobuoy velocity vector
$\mathbf{v}_i$	:	Relative position vector of sensor i
$\hat{\mathbf{v}}_i$	:	Unit relative position vector
$\mathbf{x}_i$	:	Position of sensor i
X_S	:	Set of sensor position vectors for S
$\xi(S_k, \mathbf{x}'_0)$	:	Opening angle of $K_k(S_k, \mathbf{x}'_0)$
$\mathbf{y}_0$	:	Position vector of the y-intercept b_y
$\mathbf{y}^a$	:	Input sample a of T
$\mathbf{z}^a$	:	Output sample a of T
$\zeta(S_k, \mathbf{x}_0)$	:	Opening angle of $K_k(S_k, \mathbf{x}_0)$
$\lvert \cdot \rvert$	:	Absolute value
$\lVert \cdot \rVert$	:	Matrix 2-norm

Acronyms

BN	:	Bayesian network

CODAR	:	Coastal Ocean Dynamics Applications Radar
CPA	:	Closest-point-of-approach
DSM	:	Direct shooting method
GPM	:	Gauss pseudospectral method
GPOCS	:	Gauss pseudospectral optimal control software
LTI	:	Linear, time-invariant
NLTV	:	Nonlinear, time-varying
RAM	:	Range-dependent acoustic model
ROI	:	Region-of-interest
SQP	:	Sequential quadratic programming
SVM	:	Single-vehicle minimum energy solution

Acknowledgements

I would first like to thank my advisor, Dr. Silvia Ferrari. Your help, financial support, and dedication to me over the course of my graduate studies were invaluable. Your consistent motivation kept me focused every day and on-task. I believe my experiences at Duke have well-prepared me for the "real-world".

I would also like to give special mention those faculty who have served on my Ph.D. qualifier, preliminary exam, and final defense committees for their time and effort: Dr. Michael Lavine, Dr. Dan Cole, Dr. Ron Parr, Dr. Jeff Krolik, Dr. Xiaobai Sun, Dr. Brian Mann, Dr. Devendra Garg, and Dr. Jeff Scruggs. In addition, I would like to thank those who have contributed to my better understanding of this research: Dr. Tom Wettergren, Dr. Josh Kohut, and Dr. Anil Rao. A special thank you to my labmates and fellow graduate students, who were always available and willing to help along the way: Mark Jensenius, Chenghui Cai, Gianluca Di Muro, Guoxian Zhang, and Kurt Wulff. A special thank you to Meredith Spiker, as I really enjoyed our runs, lunches, and smoothie breaks!

I wish to dedicate this dissertation to those who have shown me that I can do anything, my family. To my parents, Marty and Linda. I give you the credit for my work ethic, promoting my imagination as a kid, but most of all, for giving me the freedom and encouragement to follow my ambitions. To my better half, Rob. Your support, understanding, and love made this happen. Even though it required (too) many dinners of cereal, you kept me both motivated and sane, and I thank you for not letting me ever give up on my dreams. To my little buddy Drew. You amaze me and I look forward to helping you discover the world, as both daddy and I hold your little hands as we go along. And last but not least, to Aspen- let's go for a run!

Chapter 1

Introduction

"*If you cause your ship to stop, and place the head of a long tube in the water and place the outer extremity to your ear, you will hear ships at a great distance from you.*" -Leonardo da Vinci, 1490 A.D. [7]

Simple, low power sensors distributed throughout an environment can provide situational awareness at moderate cost. This technology lends itself to surveillance and monitoring tasks [8], such as environmental (e.g., monitoring ocean temperatures and tracking animal species) and military (e.g., surveillance and reconnaissance), which require coverage of large two-dimensional regions of interest with little or no prior knowledge of the target tracks. To ensure the distributed system is both practical and affordable, passive proximity sensors with individual detection capabilities are often employed to obtain limited measurements from each target, possibly at different moments in time, for the purpose of detection coverage. These sensors only report a simple acoustic energy observation, from which a relative distance measurement from the sensor to the target, referred to as the sensor-to-target closest-point-of-approach (CPA), may be inferred. Multiple sensor detections are used to form an hypothetical target track by fusing together the detection events from several sensors in what is referred to as a *track-before-detect* approach to moving target detection and tracking [3].

Although the problem of target tracking by means of multiple sensors arises in many applications, coverage formulations introduced so far in the literature address line-of-sight area coverage in the presence of obstacles (i.e., art-gallery problem) or point or area coverage, to ensure that every point in a two-dimensional region is within the range of at least one sensor in the network [8]. Although they are very useful in other sensor applications, none of the existing coverage formulations address the problem of coverage as it pertains to target-formation by means of multiple sensors, or *track coverage*. The problem of placing sensors in order to optimize track-formation capabilities over a region of interest so far has been addressed by placing sensors via randomized algorithms, grid configurations, or Poisson probability distributions [9–12]. As a result, the many sensor applications that include surveillance over a region for purposes of detecting a moving target are effectively limited by the current coverage formulations.

A novel coverage problem, referred to as track coverage, is presented in this dissertation. This problem is relevant when multiple sensors are used to track a moving target through non-directional measurements, such as CPA detections, according to the track-before-detect approach or are subject to frequent false alarms. The track coverage function is formulated as a problem in computational geometry using line transversals such that the quality of service of a given network configuration with respect to a pre-defined area of interest can be rapidly assessed. The overall objective of this research is to develop an optimal deployment strategy for a sensor network that is both practical, affordable, and realistic for the purposes of target tracking and detection. The simple proximity sensors to be deployed must optimize the overall track coverage over a specified region of interest and a fixed period of time. To achieve this, the following problems are to be addressed in this dissertation:

1. Place sensors in order to achieve maximum track-formation capabilities over a homogeneous region of interest.

2. Distribute autonomous sensors within a heterogenous, dynamic environment

(e.g., ocean environment, oceanic current) such that the overall track coverage over a fixed period of time is maximized.

3. Model the environmental effects that affect the acoustic sensor detection range (i.e., range of the sensor's omnidirectional field-of-view), such that the strategy accounts for sensor ranges as a function of their position within the area of interest.

4. Evaluate the robustness of the optimal deployment solution by evaluating the optimality of solutions after incorporating various sources of uncertainty through Monte Carlo simulations.

5. Optimize the trajectories of a group of underwater vehicles for cooperative coverage in the presence of ocean dynamics using optimal control theory.

The track coverage function is formulated with respect to the sensors respective locations and ranges. Since many distributed sensors are subject to environmental forcing (e.g., sonobuoys distributed in the ocean and drifting according to the current), track coverage is also optimized subject to non-maneuverable sonobuoy dynamics, as well as a mobile, controllable underwater gliders. In most applications, the sensor network is subject to random disturbances, such as unforseen and uncontrollable variations in the initial sensor location, in the ocean current, or in the maximum sensor range due to heterogenous environmental conditions. In this case, the robustness of the solution obtained from the track coverage optimization is investigated by incorporating uncertainty that is propagated throughout time. Also, the ocean bathymetry, surface temporal variability, and bottom properties influence the maximum range of an acoustic sensor and its location over time. Thus, the optimization of the track coverage function is extended to include a moving sensor field, position-dependent sensor ranges as due to heterogenous environmental conditions, and finally random disturbances and partially-unknown state due to changes in the ocean current and range model.

1.1 Problem Formulation and Assumptions

The track coverage problem consists of placing a set of omnidirectional sensors in a region of interest (ROI) for detecting moving targets, such that the amount of tracks that are cooperatively detected is maximized. The present formulation is in two-dimensional Euclidian space and relies on the following assumptions: (*i*) target maintains constant heading, speed, and amplitude; (*ii*) the region of interest $\mathcal{A} \subset \mathbb{R}^2$ is a rectangle with dimensions $L_1 \times L_2$ Km^2; (*iii*) the omnidirectional sensor field-of-view can be approximated by a disk centered at the sensor location, where the radius of the disk (range) is known, as in [3,11,13,14]; and (*iv*) a sensor may detect a target only if the target track intersects its field-of-view. These assumptions apply to the remaining chapters, unless indicated otherwise.

1.2 Research and Dissertation Outline

The main body of the dissertation is organized into five chapters. Chapter 2 provides the background for the main concepts used throughout the problem formulation, such as various coverage formulations, geometric transversals, and target tracking by multiple detections. These provide the foundation and motivation for the quality of service measurements derived in Chapter 3. In Section 3.1, the track coverage optimization problem is formulated and the track coverage function is derived. This closed form function can rapidly measure the coverage of any sensor network within a region of interest based only upon the sensor locations within the region of interest and the sensors fields-of-view. Then, the other quality-of-service metrics of a sensor network that are employed throughout this dissertation, that is area coverage and energy consumption, are derived in Sections 3.2-3.3. The track coverage optimization problem formulated in Section 3.1 is solved for various networks in Chapter 4 and compared to several other popular deployment algorithms used frequently in the literature.

Then, in Chapters 5 and 6, the static track coverage formulation is extended to include a non-fixed environment, which leads to a new problem in dynamic computational geometry pertaining to the geometric transversals of a moving family of objects. For example, when sensors are deployed in the ocean, they move according to the ocean current-induced velocity. Chapter 5 formulates the optimal deployment problem of a sensor network comprised of sonobuoys. As the current-induced drift of the sonobuoys is known to have a detrimental impact on the performance of the sensor network, the problem is formulated to include the state-space buoy equations of motion due to the ocean-induced drift velocities. These current velocities are obtained by real measurements, and approximated by a NN (Section 5.1.1). Also, environmental conditions, such as, bathymetry, surface temporal variability, and bottom properties, are known to influence the maximum range of an acoustic sensor (Section 5.1.2). Thus, the optimization of the track coverage function of a non-maneuverable, moving sensor network, is solved with respect to the initial positions that maximize coverage over a period of time, thereby minimizing the negative effects of the ocean environment. In most applications, especially sensors deployed in the ocean, the sensor network is subject to random disturbances, such as unforseen and uncontrollable variations in both the sensor location and maximum range due to heterogenous environmental conditions. Therefore, the robustness of the nominal solution of the optimal deployment problem due to parameter uncertainty is evaluated through Monte Carlo simulations (Section 5.3.2). Then, Chapter 6 addresses computing optimal trajectories for controllable underwater gliders that are deployed to detect moving targets in an oceanic region of interest by means of onboard omnidirectional acoustic sensors through optimal control theory. Consequently, a new optimal control problem is presented that integrates sensor objectives such as track coverage with cooperative path planning of a mobile sensor network subject to time-varying environmental dynamics.

Chapter 2

Background

2.1 Sensor Coverage and Search Problems

Coverage can be considered a measure of the quality of service of a sensor network deployed to perform a specified mission over time. The most popular definition of coverage, point coverage, refers to the two-dimensional space and ensures that every point in the specified space is within range of at least one sensor in the network [8]. The sensor is assumed to cover an area given by a disk with radius equal to the sensor range, and its center placed at the sensor position. Then, the network coverage considers the union of all areas covered by the sensors, referred to as *area coverage* [14, 15], where all points in the 2-D space are within the radius of at least one sensor. This formulation can be used to deploy sensors by solving a class of problems commonly referred to as *packing problems*. The circle-packing optimization problem is posed as: given a set of unequal disks and a rectangular area, find the sensor locations so that all disks can be packed into the container without overlapping. Proposed solutions to this problem include genetic algorithms [16] and a greedy heuristic algorithm that iteratively places disks according to the maximum-hole degree rule [17]. The latter algorithm is particularly interesting as it addresses the value of *corner placement*, where corner positions possess the highest value, followed by the side placements, with the middle positions being the least valuable. By following this placement strategy, the hole degree, or uncovered area that occurs once an object has been placed, is kept to a minimum. Even though packing problems address only one aspect of the multi-faceted problem of sensor network tracking posed here, the observation in [17] pertaining to corner placement is readily apparent in the initial results of this study, discussed in Section 4.1. Another common formulation of coverage is described by the *art-gallery problem*, characterized as line-of-sight visibility, where a sensor sees the target if the line segment between the two does not intersect any obstacle [18–20]. This formulation is concerned with placing the sensors such that the targets in a given area of interest, including the obstacles, are in the line of sight of at least one of the sensors. Although these formulations can be useful for various sensor applications, they are not entirely applicable to target tracking by means of a distributed and collaborative sensor network, a problem in itself with limited study.

This research focuses on target surveillance applications for target detection, a class of problems known as "search" problems. Objectives of search strategies include maximizing the total probability of successful searches [21, 22] or minimizing the expected cost (usually referring to time) until the object is found [21,23]. One search strategy, referred to as"alert-confirm" [24] entails scanning a portion of an overall area. If the sensor detects a possible target (an alert) in a broad scan, the sensor then focuses on that point to confirm the target location. Early work in the 1940's focused on deployment strategies based on search platforms, e.g., radar and sonar on a boat or plane, to best find enemy targets [25, 26]. This early research naturally extends to the optimal search tactics developed for distributed sensor networks [22], searches that develop sequentially over time [27], and developing a network of fixed sensors for a collaborative search [28]. The most limiting drawback in early classical sensor applications is independent and noncollaborative sensors in a network (or searchers) seeking targets. Many applications in tracking and surveillance benefit when increased coverage is provided by a distributed set of sensors over coverage provided by a single sensor platform. In [29], an incremental greedy search strategy was implemented for detecting a single static target located in one cell comprising a particular area. The results show that an optimal search strategy can be obtained by incrementally placing sensors such that each sensor maximizes the immediate gain in either the most-likely or second-most-likely place to contain an object. However, it does not provide a significant algorithm for achieving an optimal sensor configuration.

Another popular search technique for optimal sensor placement is genetic algorithms, which has commonly been applied to the field of structures. One example is modal identification on a large space structure [30]. Another example involves fault detection involves using neural networks to locate and classify faults while a genetic algorithm (GA) determines an optimal (or near optimal) sensor distribution [31]. Both applications display positive results in their sensor configurations. Although the application to structures is very different from the application for target surveillance, genetic algorithms can be compared to the methods posed in this research when applied to the optimal sensor placement problem.

The problem of target tracking by a sensor network arises in many applications, including surveillance systems, monitoring endangered species, and manufacturing, and as a result has received considerable attention. Tracking refers to the estimation of the state (e.g., position, velocity, acceleration) of a moving object by way of a sensor or sensors positioned on a stationary or moving platform. The track of a target once detected by the sensor(s) in search mode is formed by its state trajectory being estimated from the set of measurements acquired from each sensor detection. The measurements of multiple sensors are combined to estimate a target's state and maintain the track as precisely as possible, using data fusion. Popular algorithms for data fusion include the Nearest Neighbor algorithm, Probabilistic Data Association, and Multiple Hypothesis Tracking [32–37]. Another popular fusion algorithm for combining information from multiple sensor nodes for optimal detection decisions is the Neyman-Pearson test with the likelihood ratio tests [38–43]. This method, which seeks a decision rule to have maximum probability of detection while not allowing the probability of false alarm to exceed a certain value, gives excellent theoretical results, but is not generally applicable when no *a-priori* knowledge of the target track is known, as is the case for proximity sensors or a variable environment. Efforts have been made to develop a training approach that learns the typically unknown statistical performance probabilities of detections and false alarms [44–46]. However, these approaches have been effectively hindered by a large region of interest for which to observe the same target over the same time interval by many sensors, and are also limited by the communications bandwidth available between the sensors in the network. These efforts are not applicable to the proximity sensors because once the data is associated with one target and fused by an algorithm described above, it is used together with past observations to estimate the target tack by means of well-known Kalman-filter equations [47] as done in [33]. They rely on frequent and accurate measurements from sensors, typically obtained over the same time interval, such as, in the case of air-traffic control radars.

To make distributed sensor networks practical and affordable, each sensor must be relatively simple. Proximity sensors are typically deployed when they may be lost over time, there is no *a-priori* knowledge of the target track, and the measurements are limited and collected at different times while the target moves across a large region of interest. A central fusion center that fuses only the limited energy information attained from the proximity sensors is incorporated such that the sensor network has very low communication bandwidth and is easily applied to a variety of target types with little modification. A central fusion system collects only peak energy information from each individual proximity sensor, and assumes that the relationship between the energy recorded and the distance between the target and the sensor follows a known relationship and is easily acquired numerically, as done in [3]. This limited-information event-based approach for forming an estimate of a target track in a distributed sensor network showed that when considering a single target moving at a constant heading, speed, and source amplitude through the sensor field, it is possible to derive a reduced set of potential target path tracks from only the proximity information from multiple sensors and the closest-point-of-approach (CPA) [3]. An additional benefit of this approach when combined with track path clustering is its robustness to false detections or false alarms [3]. Thus, for every potential target, multiple detections must be obtained from sensors distributed throughout the region of interest [3, 13, 48].

2.2 Geometric Transversals

This dissertation focuses on the geometric properties of the network, addressing a deterministic track-coverage formulation and binary sensor models (as in [9, 14, 49]). In order to maximize the coverage provided in $\mathcal{A}$ by n sensors, the space of all tracks is derived through geometric transversal theory, see [2]. A *line transversal*, also referred to as a stabber, is a straight line that intersects every member in a family of objects, while a *common transversal* for a family of sets is a line which intersects every set in a family. When a set of geometric objects in $\mathbb{R}^d$ have a k-*transversal*, the objects are said to be simultaneously intersected by a k-dimensional flat (or translate of a linear subspace). When $k = 0$, the objects are referred to as point transversals. The field of geometric transversals has originated with Helly's Theorem.

Helly's Theorem [50] *$\mathcal{S}$ is a family of n convex sets in $\mathbb{R}^d$ that has a common intersection point if and only if every $d+1$ convex sets have a common intersection point*

Since the introduction of Helly's Theorem in 1923, much study has been given to establishing the necessary and sufficient conditions for the existence of transversals, as well as developing algorithms for determining common transversals. However, algorithms for determining common transversals are known in special cases only. Finding point transversals for families of half-spaces is the focus of linear programming [51]. When $k = 1$, i.e., stabbing lines, the problem is formulated similarly to $k = 0$, and $\mathcal{S}$ has a line transversal if there exists a line that intersects every member of $\mathcal{S}$. However, this computation becomes increasingly more difficult compared to $k = 0$ [2]. Although the general problem for finding k-transversal to a family of n polytopes in $\mathbb{R}^d$ can be formulated in terms of a system of algebraic inequalities, or LMIs, the current methods for solving such systems, such as computer algebra, are inefficient due to their generality as well as ignoring the underlying geometric nature of the problem.

A brief overview is given of the more popular algorithms for constructing the space of transversals for a family S of n simple objects (polygons with a constant number of edges, e.g., disks) in $\mathbb{R}^2$. Motivated by the practical visibility problem in the plane, one of the first proposed line transversal algorithms constructs the stabbing region for n line segments in $\mathbb{R}^2$ in $\mathcal{O}(n \log n)$ by a general "divide and conquer" technique [52]. The n line segments are partitioned into two groups of $n/2$ segments to calculate the space of transversal for each group, with the two merging at the end to form the entire space of transversals for the original n line segments. This early algorithm was then extended to a family of simple convex sets whose boundaries intersect pairwise at most s times [53], finding line transversals for homothets (involving both scaling and translating) of simple planar objects [54], and finding line transversals for disks of equal radius [46], all of which run in $\mathcal{O}(n \log n)$. The algorithm in [52] was also successfully extended to plane transversals of convex polytopes in $\mathbb{R}^3$ but was unable to improve upon the computational complexity in higher dimensions [55].

One of the most relevant results to the track coverage problem is an algebraic decision tree methodology that finds a single line transversal for a translates family of n line segments in $\mathbb{R}^2$, or n equal disks in $\mathbb{R}^2$ [56]. Although existing algorithms (e.g., [56, 57]) cannot construct closed-form representations of line transversals, they could be applied to determine the tracks intercepted by a family of omnidirectional sensors, provided *all* sensors have the same range and their positions are known. It was pointed out in [2] that geometric transversals algorithms could be greatly improved by considering the underlying geometric nature of the problem. Therefore, the track coverage function is formulated in Section 3.1 as a measure of the geometric transversals (potential target tracks) that intersect the non-translate families of disks (omnidirectional sensors fields-of-view).

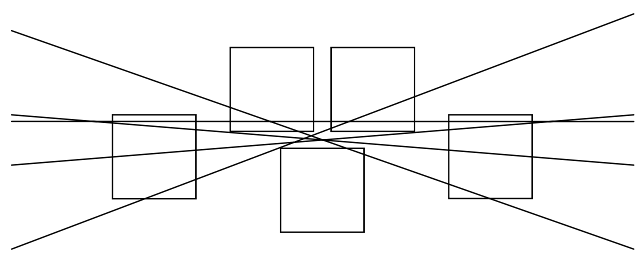


Figure 2.1: Examples of line transversals for a family of five square polygons, with $k = 3$ (taken from [2]).

2.3 Track-Before-Detect Approach in Surveillance Systems

The problem of target tracking by a sensor network arises in many applications, including surveillance systems, monitoring of endangered species, and manufacturing. As a result, it is receiving considerable attention. Tracking refers to the estimation of the state (e.g., position, velocity, acceleration) of a moving object by means of multiple sensor measurements. Once a detection is declared by sensors in search mode, a target track is formed by estimating its state from the set of measurements acquired over time, through Kalman filtering. In this dissertation, it is assumed that a central fusion center collects only peak energy information from a set of proximity sensors and then computes all sensor-target distances as shown in [3]. Using only limited sensed information, such as closest-point-of-approach (CPA) detections, it is possible to hypothesize a set of target paths for one or more targets moving at a constant heading and non-zero speed through the sensor field [3].

Passive (or listening) sensor systems detect the acoustic energy emitted by the target. In this context, we are concerned with proximity sensors, which report a simple acoustic energy observation, from which a relative omnidirectional distance measurement to the target may be inferred. For example, when a target moves at a constant speed and heading through $\mathcal{A}$, each sensor reports to a central fusion processor its respective location in two-dimensional space and a single value for the received signal level at the sensor-to-target CPA. The event-based tracking algorithm in [3] assumes each sensor from the same target receives an isotropic energy attenuated by the environment according to the power law from the target,

$$e_i(t) = cF(d_i(t))^{-\alpha} \tag{2.1}$$

$$e_i(t) \mid_{t_{i,CPA}} = e_i^{CPA} = cF(d_i)^{-\alpha} \tag{2.2}$$

where $t_{i,CPA}$ is the time period of the CPA for the i^{th} sensor; d_i is the CPA range (distance) from the i^{th} sensor-to-target and is approximated by a disk centered at the sensor location $\mathbf{x}_i$; F, which in the sonar literature is also frequently denoted as TS, represents a target source level independent of both time and location; c is the target independent scaling constant that is based on the physics of the problem; and the nondimensional exponential attenuation coefficient α depends on the particular physical mechanism of the energy that is being received and the environment. As time is not reported, it is assumed that multiple detections from a single target occur within a specified time period and the target does not maneuver in that time. Introducing the constant scaling factor $B = (cF)^{-1/\alpha}$ and a minimum detection threshold ϑ, it was shown in [3] that given a set of τ_i error-free measurements

$$\tau_i = (e_i^{CPA})^{-1/\alpha} = B \cdot d_i > 0 \tag{2.3}$$

for each of the sensors located at $\mathbf{x} = [\mathbf{x}_1^T \ ... \ \mathbf{x}_n^T]^T$, then for the set of single positive detections

$$T = \{\tau_i | \tau_i = B \cdot d_i \geq \vartheta\} \tag{2.4}$$

the target path is a line that is jointly tangent to all disks $C_i(S) \equiv \{\chi : \|\chi - \mathbf{x}_i\| \leq d_j\}$, where $\chi \in \mathbb{R}^{2\times 1}$ and $\|\cdot\|$ is the Euclidean norm. Fig. 2.2 illustrates the two possible tracks formed from two sensor measurements, τ_1 and $\tau_2 \in T$, where $d_i = \tau_i/B$. Since B is constant for both sensors, it may remain unknown without affecting d_i [3]. Consequently, reliable target detection typically requires $k \geq 2$ sensor detections that are used in a track-before-detect approach.

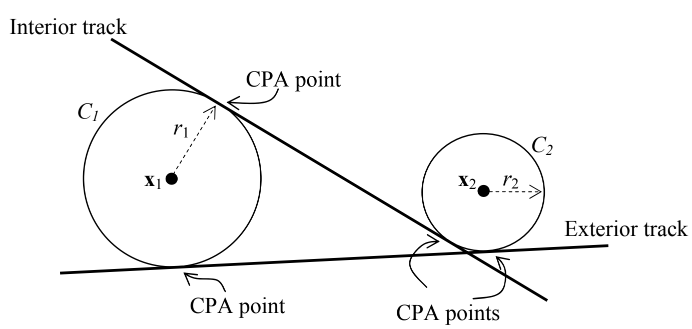


Figure 2.2: Geometry of interior and exterior tracks formed from two CPA detections obtained by two omnidirectional sensors, placed at $\mathbf{x}_1$ and $\mathbf{x}_2$ (adapted from [3], reflections are omitted for simplicity).

Chapter 3

Objective Functions for Assessing the Quality-of-Service of Mobile Sensor Networks in Surveilling Systems

3.1 A Geometric Transversal Approach to Analyzing Track Coverage in Sensor Networks

Suppose a network of n proximity sensors with different ranges $r_1, \ldots, r_n$ must be deployed in a region of interest $\mathcal{A}$ for the purpose of detecting moving targets. Then, the number of detections required per track is a constant parameter k, such that $1 \leq k \leq n$, and its value is decided based on the level of confidence required by the sensor system. For example, in applications with infrequent false alarms $k = 3$ is considered to be the minimum number of detections required for forming a reliable track and declare a detection. Figure 2.2, taken from [3], illustrates that two CPA detections obtained by proximity sensors may be caused by four possible tracks (the two in the figure, and their reflections). In systems where the sensor detections provide additional information about the target (such as, the position), multiple detections may still be required to track a moving target due to the presence of measurement errors and false alarms.

Therefore in this section we derive a track coverage function in order to address sensor deployment as an optimization problem (Chapters 4-6). In order to optimize the sensors placement, the amount of tracks they intercept is expressed as a function of the sensors coordinates in the plane $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and respective ranges $r_1, \ldots, r_n$. Under the given assumptions from Chapter 1.1, track coverage can be viewed as a new geometric transversal problem. Thus, in this dissertation, a novel approach is presented for representing geometric transversals by means of cones.

3.1.1 Cone Representation of Track Coverage

Based on the track-coverage problem formulation and assumptions (i)-(iv) in Section 1.1, the tracks detected by k sensors in an omnidirectional network S of size n can be viewed as the line transversals of a family of disks with different radii. In this section, we show that a set of line transversals can be represented by means of a coverage cone, which contains line transversals (or tracks) characterized by the same intercept. The coverage cone of a single sensor is defined in the following section, where we obtain the coverage cone of multiple sensors. Then, the track coverage over a rectangular area is represented by the union of coverage cones with intercepts along the perimeter. Finally, the coverage cone representation of line transversals is used in Section 3.1.2 to obtain a track-coverage function that quantifies the ability of a sensor network to perform cooperative detections.

Coverage Cone

Consider a sensor in the network S that is indexed by i and is located at $\mathbf{x}_i = [x_i \quad y_i]^T \in \mathbb{R}^2$ in the xy-plane. Let $C(\mathbf{x}_i, r_i) = C_i$ denote a disk with radius r_i centered at $\mathbf{x}_i$ that represents the field-of-view of this sensor. Assume that any target track can be described by a straight line, $y = a_y x + b_y$, with slope a_y and y-intercept b_y. As shown in [3], a CPA detection event takes place when the target path is tangential to a disk of radius $d_i \leq r_i$, centered at $\mathbf{x}_i$. Without loss of generality, we can assume that all disks and CPA detections are in the positive orthant $\mathbb{R}^2_+$. Then, we can represent tracks by rays or half-lines denoted by $\mathcal{R}_\alpha(b_y)$. Each ray originates at an intercept $y = b_y$ and forms an angle $\alpha = \tan^{-1}(a_y)$ with the x-axis. Let the vector $\mathbf{y}_0 \equiv [0 \quad b_y]^T$ denote the position of the y-intercept. Then, the position of the i^{th} sensor can be expressed by a relative position vector that is convenient for generating the sensor coverage cone, namely:

$$\mathbf{v}_i \equiv (\mathbf{x}_i - \mathbf{y}_0) = \begin{bmatrix} x_i \\ (y_i - b_y) \end{bmatrix} \tag{3.1}$$

Borrowing two basic definitions from convex analysis [58], a set K is said to be a *cone* if for all $x \in K$, where $x \in \mathbb{R}^2$, and $c > 0$, we have $cx \in K$. Also, given a nonempty subset X of $\mathbb{R}^n$, the *cone generated by* X is the set of all nonnegative combinations of the elements of X, denoted by cone(X). We define the *coverage cone* of the i^{th} sensor with respect to the intercept b_y to be the cone generated by C_i with origin $\mathbf{y}_0$, and we denote it by $K(C_i, \mathbf{y}_0)$. The coverage cone is a basic construct for the coverage function because it represents the set of tracks that can be detected by the i^{th} sensor.

Remark 3.1.1 *The coverage cone $K(C_i, \mathbf{y}_0)$ contains the set of all tracks $\mathcal{R}_\alpha(b_y)$ that intersect the sensor field-of-view $C_i(\mathbf{x}_i, r_i)$ in $\mathbb{R}^2_+$:*

The proof is provided in Appendix A, and an example of coverage cone is illustrated in Fig. 3.1.

Let θ_i denote half the opening angle of the coverage cone (Fig. 3.1). Since the extremals of K are tangential to C_i, the trigonometric relationships,

$$\sin\theta_i = \frac{r_i}{\|\mathbf{v}_i\|} = \frac{r_i}{\sqrt{x_i^2 + (y_i - b_y)^2}} \tag{3.2}$$

and,

$$\cos\theta_i = \frac{\sqrt{\|\mathbf{v}_i\|^2 - (r_i)^2}}{\|\mathbf{v}_i\|} \tag{3.3}$$

relate the opening angle to the sensor location $\mathbf{x}_i$ through $\mathbf{v}_i$ in (3.1). Then, the coverage cone $K \subset \mathbb{R}^2$ is finitely generated by two unit vectors $\hat{\mathbf{l}}_i$ and $\hat{\mathbf{h}}_i$, that is,

$$K(C_i, \mathbf{y}_0) = \text{cone}(\hat{\mathbf{l}}_i, \hat{\mathbf{h}}_i) = \{x \mid x = c_1\hat{\mathbf{l}}_i + c_2\hat{\mathbf{h}}_i,\ c_1, c_2 \geq 0\} \tag{3.4}$$

provided the unit vectors are obtained from $\mathbf{v}_i$ through rotation matrices,

$$\hat{\mathbf{h}}_i = \begin{bmatrix} \cos\lambda_i \\ \sin\lambda_i \end{bmatrix} = Q_i^+ \hat{\mathbf{v}}_i \equiv \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \tag{3.5}$$

and,

$$\hat{\mathbf{l}}_i = \begin{bmatrix} \cos\gamma_i \\ \sin\gamma_i \end{bmatrix} = Q_i^- \hat{\mathbf{v}}_i \equiv \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \tag{3.6}$$

where, $Q_i^- = (Q_i^+)^T$. Thus, the coverage cone $K(C_i, \mathbf{y}_0)$ is completely specified by the unit vectors $\hat{\mathbf{l}}_i$ and $\hat{\mathbf{h}}_i$, which are known functions of $\mathbf{x}_i$ and r_i.

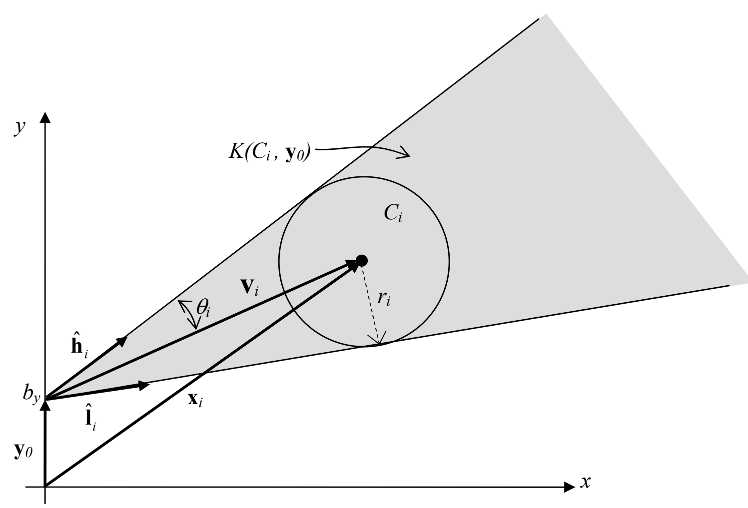


Figure 3.1: Coverage cone $K(C_i, \mathbf{y}_0)$ of a sensor located at $\mathbf{x}_i$, generated by the unit vectors $\hat{\mathbf{l}}_i$ and $\hat{\mathbf{h}}_i$.

k-Coverage Cone for Multiple Sensors

Multiple sensor detections typically are necessary to determine target tracks by means of proximity sensors, or in the presence of measurement errors and false alarms, as shown in [3]. Let k denote the minimum number of distinct sensor detections that are required by the system to reliably form a track. Two detections are said to be distinct when they are obtained by two different sensors. Thus, k detections are obtained when k sensors in the network $S = \{C_1, C_2, \ldots, C_n\}$ intersect the same track.

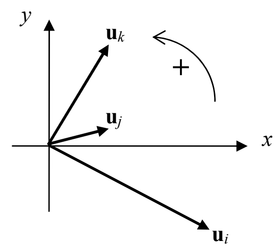


Figure 3.2: Example of three vectors ordered according to the xy-frame, where $\mathbf{u}_i \prec \mathbf{u}_j \prec \mathbf{u}_k$.

In this section, we show that the set of tracks that intersect at least k sensors in S, with y-intercept b_y, is contained by a so-called k-coverage cone. Vectors in $\mathbb{R}^2$ are ordered according to the orientation of the reference frame. Two vectors $\mathbf{u}_i$ and $\mathbf{u}_j$ are said to be *ordered* according to the xy-frame such that $\mathbf{u}_i \prec \mathbf{u}_j$ if when these vectors are translated to make their origins coincide, and $\mathbf{u}_i$ is rotated through the smallest angle possible to meet $\mathbf{u}_j$, this rotation is in the same direction as the orientation of the xy-frame (as illustrated in Fig. 3.2 and in [59]). Let $\Omega(S, \mathbf{y}_0)$ and $\Lambda(S, \mathbf{y}_0)$ denote the sets of unit vectors generating the coverage cones of all sensors in S with origin $\mathbf{y}_0$. That is, from (3.5)-(3.6), $\Omega(S, \mathbf{y}_0) = \{\hat{\mathbf{h}}_i \mid Q_i^{-}\hat{\mathbf{h}}_i = \hat{\mathbf{v}}_i, \forall i \in I_S\}$ and $\Lambda(S, \mathbf{y}_0) = \{\hat{\mathbf{l}}_i \mid Q_i^{+}\hat{\mathbf{l}}_i = \hat{\mathbf{v}}_i, \forall i \in I_S\}$, where I_S denotes the index set of S. Then, these two sets can be used to determine the k-coverage cone of S, as shown by the following result:

Proposition 3.1.2 *The set of all tracks $\mathcal{R}_\alpha(b_y)$ that are line transversals to a family of k non-translates disks $S = \{C_1, C_2, \ldots, C_k\} \equiv S_k$ with index set I_{S_k}, is contained* *by the finitely generated cone*

$$K_k(S_k, \mathbf{y}_0) = cone(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*) \tag{3.7}$$

Where, $\mathbf{y}_0 = [0 \quad b_y]^T$, $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath$ *and* $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\imath$ *with* $\jmath, \imath \in I_{S_k}$, *such that* $\hat{\mathbf{h}}_\jmath \preceq \hat{\mathbf{h}}_i \in \Omega(S_k, \mathbf{y}_0)$ *and* $\hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{l}}_i \in \Lambda(S_k, \mathbf{y}_0)$ *for* $\forall i \in I_{S_k}$, *and provided* $\hat{\mathbf{l}}_\imath \prec \hat{\mathbf{h}}_\jmath$. *If* $\hat{\mathbf{l}}_\imath \succ \hat{\mathbf{h}}_\jmath$, *then* $K_k(S_k, \mathbf{y}_0) = \emptyset$.

A proof is provided in Appendix B.

A simple example of k-coverage cone is illustrated in Fig. 3.3, where $k = 2$, and S_2 contains two sensors located at $\mathbf{x}_1$ and $\mathbf{x}_2$. In this example, the 2-coverage cone $K_2(S_2, \mathbf{y}_0)$ is generated by the unit vectors $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_2$ and $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_1$, since $\Omega(S_2, \mathbf{y}_0) = \{\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2\}$ and $\Lambda(S_2, \mathbf{y}_0) = \{\hat{\mathbf{l}}_1, \hat{\mathbf{l}}_2\}$, where $\hat{\mathbf{l}}_2 \succ \hat{\mathbf{l}}_1$ and $\hat{\mathbf{h}}_1 \prec \hat{\mathbf{h}}_2$.

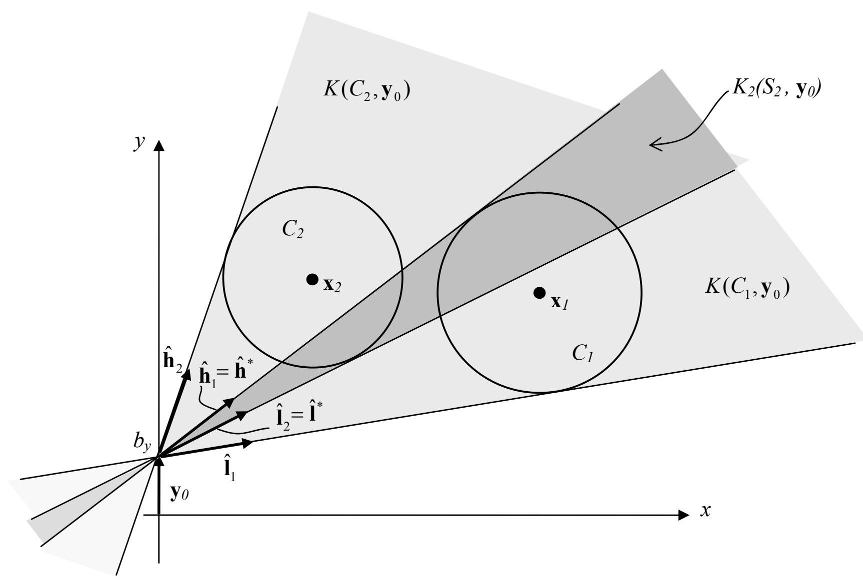


Figure 3.3: The k-coverage cone $K_2(S_2, \mathbf{y}_0)$ of the family $S_2 = \{C_1, C_2\}$ is shown in dark grey and is generated by the unit vectors $\hat{\mathbf{l}}^*$ and $\hat{\mathbf{h}}^*$ obtained from the sets of unit vectors generating $K(C_1, \mathbf{y}_0)$ and $K(C_2, \mathbf{y}_0)$ (shown in light grey).

The cone $K_k(S_k, \mathbf{y}_0)$ is referred to as the k-coverage cone of S_k with origin $\mathbf{y}_0$. An important feature of this approach is that the k-coverage cone is easily obtained from the sets of unit vectors Ω and Λ. Provided two unit vectors are in the first or fourth quadrant of a reference frame, they can be ordered by their direction sines (as shown in Appendix C). Therefore, if we let

$$\begin{aligned} \sin\lambda^* &= \inf\{\sin\lambda_i \mid \hat{\mathbf{h}}_i = [\cos\lambda_i \quad \sin\lambda_i]^T \in \Omega(S_k, \mathbf{y}_0),\ \forall i \in I_{S_k}\} \\ \sin\gamma^* &= \sup\{\sin\gamma_i \mid \hat{\mathbf{l}}_i = [\cos\gamma_i \quad \sin\gamma_i]^T \in \Lambda(S_k, \mathbf{y}_0),\ \forall i \in I_{S_k}\} \end{aligned} \tag{3.8}$$

then, $\hat{\mathbf{l}}^* = [\cos\gamma^* \quad \sin\gamma^*]^T$ and $\hat{\mathbf{h}}^* = [\cos\lambda^* \quad \sin\lambda^*]^T$. When the unit vectors are in the second or third quadrant, they can still be ordered by their direction sines by introducing a constant rotation (Appendix C). Therefore, the infimum and supremum in (3.8) can be determined by linear operations on the elements of Ω and Λ, respectively.

Consider now the tracks detected by at least k sensors in $S = \{C_1, \ldots, C_n\}$, with $1 \leq k < n$. These tracks are the line transversals of any k-subset of S. A *k-subset* is defined as a subset containing any k elements of a set with n elements [60]. By Proposition 3.1.2, all tracks $\mathcal{R}_\alpha(b_y)$ detected by a set of k sensors S_k are contained by the k-coverage cone of S_k. It follows that the set of all tracks $\mathcal{R}_\alpha(b_y)$ detected by at least k sensors in S is the union of the k-coverage cones of all k-subsets of S:

$$\mathcal{K}_k(S, \mathbf{y}_0) = \bigcup_{j=1}^{m} K_k(S_k^j, \mathbf{y}_0), \quad m = \left(\begin{array}{c} n \\ k \end{array} \right) \tag{3.9}$$

S_k^j denotes the j^{th} k-subset of S, and the number m of possible k-subsets is given by the binomial coefficient n *choose* k, as shown in (3.9). Since $\mathcal{K}_k$ is a union of possibly disjoint cones, it may not be a cone [58]. Nevertheless, it is just as useful because the same measure defined for a cone can be applied to it using the principle of inclusion-exclusion, as shown in Section 3.1.2. In the next section, we utilize the k-coverage cone to construct an approximate representation of the set of tracks that traverse the region-of-interest $\mathcal{A}$, and are detected by at least k sensors in S.

Track-Coverage over a Rectangular Area or Region-of-Interest (ROI)

The track coverage problem consists of placing a set of sensors for the purpose of detecting moving targets cooperatively in a region-of-interest (ROI). In this dissertation, the ROI is assumed to be a rectangle $\mathcal{A}$ of known dimensions $L_1 \times L_2$. Place the xy-frame of reference along two sides of $\mathcal{A}$, such that its origin $(0,0)_{xy}$ coincides with one vertex, and one side of $\mathcal{A}$ can be denoted by the interval $\mathcal{I}_y \equiv \{y \mid y \in [0 \quad L_2]\}$ (as shown in Fig. 3.4). Based on the previous section, the set of all tracks $\mathcal{R}_\alpha(b_y)$ that intersect this side of $\mathcal{A}$ at $b_y \in \mathcal{I}_y$ and are detected by at least k sensors in S is $\mathcal{K}_k(S, \mathbf{y}_0)$, and is given by (3.9). In order to obtain representations that are computationally tractable, we discretize the interval $\mathcal{I}_y$ in N_2 increments of size $\delta b = L_2/N_2$, and define $b_y^\ell \equiv \ell \cdot \delta b$, and $\mathbf{y}_0^\ell \equiv [0 \quad b_y^\ell]^T$. Then, the set of tracks that intersect the y-axis over the interval $\mathcal{I}_y$ and are detected by at least k sensors in S can be approximated by:

$$\mathcal{K}_k(S, \mathcal{I}_y) \approx \bigcup_{\ell=0,\ldots,N_2} \mathcal{K}_k(S, \mathbf{y}_0^\ell), \quad N_2 = L_2/\delta b \tag{3.10}$$

Where, each set $\mathcal{K}_k(S, \mathbf{y}_0^\ell)$ is given by (3.9). Clearly, by letting $\delta b \to 0$ the above approximation approaches the entire set of tracks intersecting $\mathcal{I}_y$.

The methodology is extended to all sides of $\mathcal{A}$ by placing a second frame of reference, $x'y'$, along the remaining sides of $\mathcal{A}$, such that its origin $(0,0)_{x'y'}$ is the vertex opposite to $(0,0)_{xy}$, as shown in Fig. 3.4. Then, each side of $\mathcal{A}$ is denoted by one of the following intervals: $\mathcal{I}_y$, $\mathcal{I}_x \equiv \{x \mid x \in [0 \quad L_1]\}$, $\mathcal{I}_{x'} \equiv \{x' \mid x' \in [0 \quad L_1]\}$, or $\mathcal{I}_{y'} \equiv \{y' \mid y' \in [0 \quad L_2]\}$. With this choice of reference frames an efficient representation of the target-tracks traversing $\mathcal{A}$ can be obtained by defining coverage cones with origins on each of the four axes, namely, $\mathbf{x}_0 = [b_x \quad 0]^T$, $\mathbf{y}'_0 = [0 \quad b_{y'}]^T$, and $\mathbf{x}'_0 = [b_{x'} \quad 0]^T$, where $b_x \in \mathcal{I}_x$, $b_{x'} \in \mathcal{I}_{x'}$, and $b_{y'} \in \mathcal{I}_{y'}$ (Fig. 3.4). The coverage cones of the i^{th} sensor with respect to each axis are denoted by $K(C_i, \mathbf{y}_0)$, $K(C_i, \mathbf{x}_0)$,

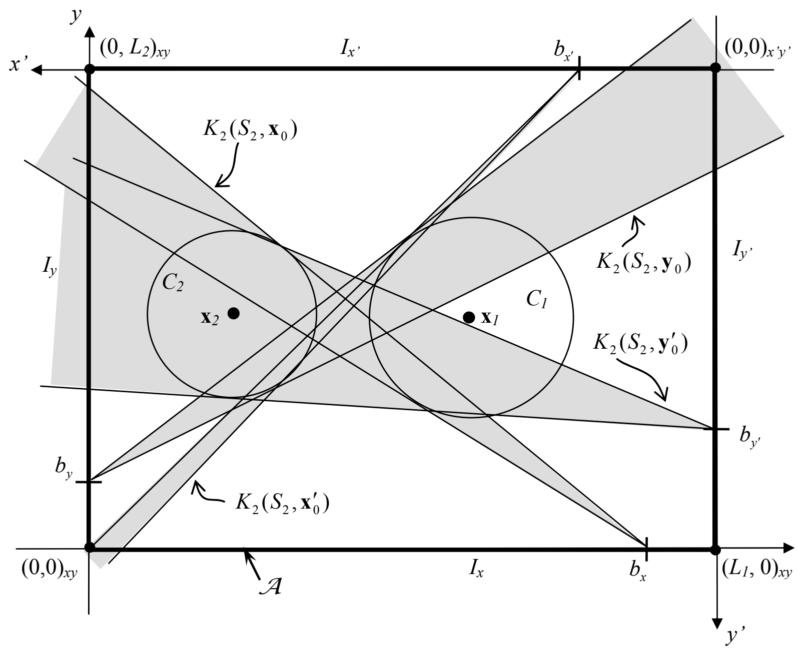


Figure 3.4: Reference frames used to define k-coverage cones with respect to each axis, as illustrated in the figure for $k = 2$ and $S_2 = \{C_1, C_2\}$.

$K(C_i, \mathbf{y}'_0)$, and $K(C_i, \mathbf{x}'_0)$, and are obtained by defining a relative-position vector for each axis. From hereon, denote the vector in (3.1) by $\mathbf{v}_i(\mathbf{y}_0)$, and let $\mathbf{v}_i(\mathbf{x}_0) = (\mathbf{x}_i - \mathbf{x}_0)$ denote the relative-position vector for x. The relative-position vectors for the x' and y' axes are defined as,

$$\mathbf{v}_i(\mathbf{x}'_0) = \mathbf{L} - \mathbf{x}_i - \mathbf{x}'_0, \quad \text{and} \quad \mathbf{v}_i(\mathbf{y}'_0) = \mathbf{L} - \mathbf{x}_i - \mathbf{y}'_0, \quad \text{for} \quad \forall i \in I_S \tag{3.11}$$

where, $\mathbf{L} \equiv [L_1 \quad L_2]^T$. The coordinate transformation $\mathbf{x}_i|_{x'y'} = (\mathbf{L} - \mathbf{x}_i)$ is used to express all sensor positions with respect to the same coordinate frame xy. Then, the k-coverage cone for multiple sensors methodology can be extended to all axes.

For simplicity, all intervals $\mathcal{I}_y$, $\mathcal{I}_x$, $\mathcal{I}_{y'}$, and $\mathcal{I}_{x'}$ are discretized by increments of the same size δb. Hence, from (3.10), the set of tracks traversing $\mathcal{A}$ and intersecting at least k sensors in S is given by,

$$\begin{aligned} \mathcal{K}_k(S,\mathcal{A}) &= \mathcal{K}_k(S,\mathcal{I}_y) \cup \mathcal{K}_k(S,\mathcal{I}_x) \cup \mathcal{K}_k(S,\mathcal{I}_{x'}) \cup \mathcal{K}_k(S,\mathcal{I}_{y'}) && (3.12) \\ &\approx \left(\cup_{\ell=0}^{N_2} \cup_{j=1}^{m} K_k(S_k^j, \mathbf{y}_0^\ell) \cup K_k(S_k^j, \mathbf{y}_0'^\ell)\right) && (3.13) \\ &\cup \left(\cup_{\ell=0}^{N_1} \cup_{j=1}^{m} K_k(S_k^j, \mathbf{x}_0^\ell) \cup K_k(S_k^j, \mathbf{x}_0'^\ell)\right), \end{aligned}$$

where m is equal to the binomial coefficient k *choose* n (as in (3.9)), $N_2 = L_2/\delta b$, and $N_1 = L_1/\delta b$.

Example: Assessing the track-coverage of a known sensor network configuration with $n = 20$ and $k = 3$

The cone representation of track coverage is demonstrated by considering a known sensor network configuration. When the sensors positions are known, the tracks detected can be verified by testing a designated sample [61]. In this example, the sensor network S is characterized by $n = 20$, $k = 3$, and ranges and positions shown in Fig. 3.5(a). The union of k-coverage cones $\mathcal{K}_k(S, \mathbf{y}_0^\ell)$, with $\mathbf{y}_0^\ell = [0\ 15]^T$, is illustrated in Fig. 3.5(a). The cone representation of area track-coverage, $\mathcal{K}_k(S, \mathcal{A})$, is computed using the methodology in Section 3.1.1 and plotted in parameter space in Fig. 3.5(b), where grey represents sets of tracks that are detected by at least k sensors.

When the sensors positions are known, a sample of tracks detected by S can be determined numerically by testing their intersections with S. Although this approach does not provide a closed-form representation of track-coverage, it is useful for validating its cone representation $\mathcal{K}_k(S, \mathcal{A})$. As shown in [3], a CPA detection takes place when the track $\mathcal{R}_\alpha(b_y)$ is tangential to a disk with a radius less or equal to a sensor range r_i. Then, the tracks intercepted by the i^{th} sensor, positioned at $\mathbf{x}_i$, are those whose parameters satisfy the inequality,

$$d_i = \left| \frac{(b_y + a_y x_i - y_i)}{\sqrt{a_y^2 + 1}} \right| \leq r_i \tag{3.14}$$

where, $a_y = \tan(\alpha)$. A brief proof is provided in Appendix D. This inequality can be used to develop a simple detection test for a designated set of tracks, denoted by $T_{\mathcal{R}}$. Since the sensors ranges and positions are all known, the inequality (3.14) can be evaluated for every track in $T_{\mathcal{R}}$ and for every sensor in S. Let B_i denote a logical array or truth table in which every element corresponds to one track in $T_{\mathcal{R}}$, and is either equal to 1 or 0, depending on whether the track has been detected (1) or missed (0) by the i^{th} sensor. Every element of B_i can be evaluated using (3.14), and an array B_i can be obtained for every sensor in S. Then, the logical array,

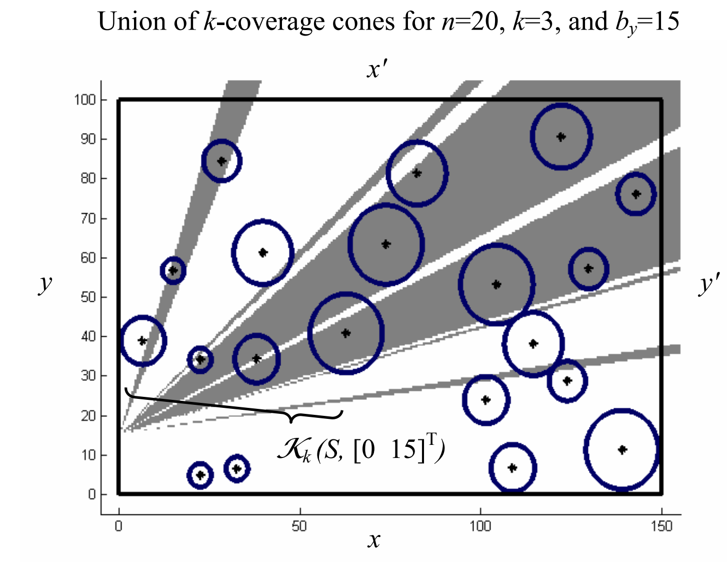
$$T_k = \left\{ \sum_{i \in I_S} B_i \geq k \right\} \tag{3.15}$$

indicates whether each track in $T_{\mathcal{R}}$ has been detected by at least k sensors in S.

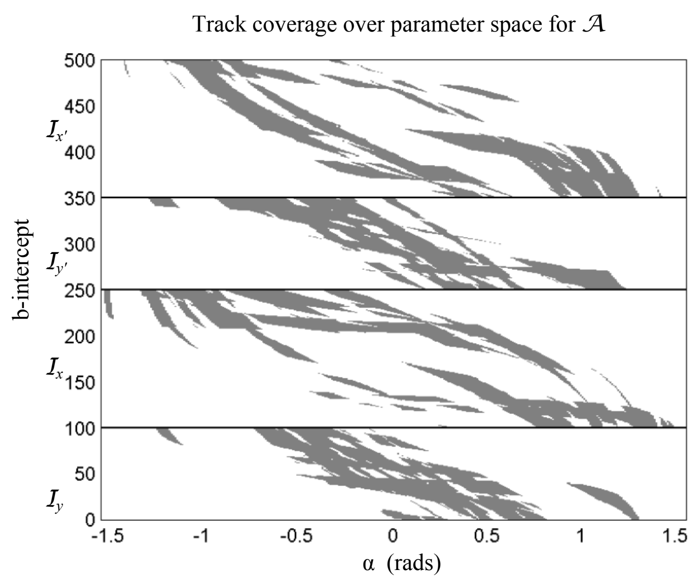
The array T_k obtained for the sensor network in Fig. 3.5(a) is plotted in parameter space in Fig. 3.5(b). The total number of detections per track is also plotted in Fig. 3.6(a) to verify the track coverage results in Fig. 3.6(b). It can be seen by comparing Fig. 3.6(b) to Fig. 3.5(b) that $\mathcal{K}_k(S, \mathcal{A})$ provides a faithful representation of the tracks that are cooperatively detected by S.

3.1.2 Track-Coverage Function

The cone representation of track coverage allows to generate the space of tracks that are cooperatively detected by a sensor network using sets of unit vectors. Another important use of coverage cones is the functional representation of the quality of service of the network. In this section, we derive a so-called track-coverage function

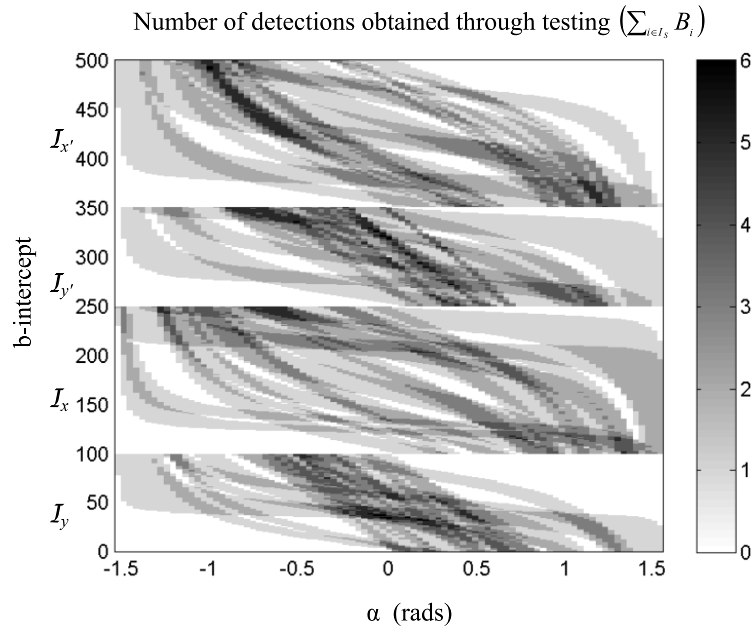


(a)

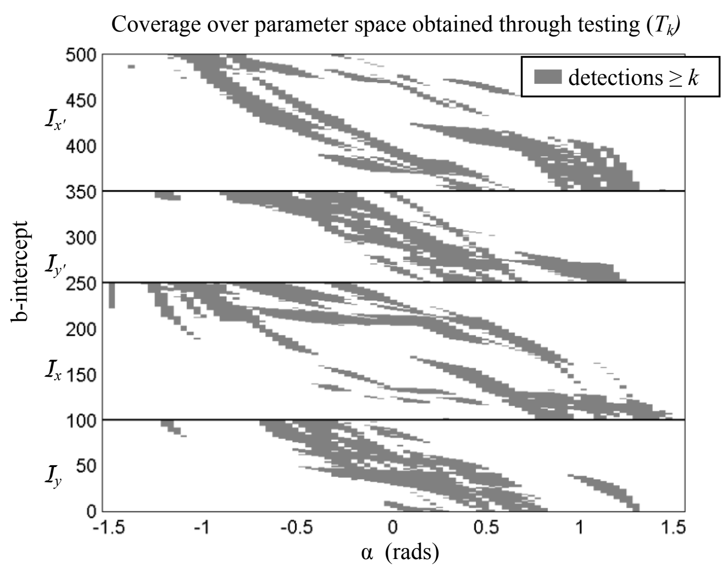


(b)

Figure 3.5: Track coverage $\mathcal{K}_k(S, \mathcal{A})$ (b) of a known sensor network configuration (a) with $n = 20$, $k = 3$. The union $\mathcal{K}_k(S, \mathbf{y}_0^\ell)$ is illustrated by the grey cones in (a) for $\mathbf{y}_0^\ell = [0\ 15]^T$.



(a)



(b)

Figure 3.6: Number of detections obtained through testing (a) and resulting track coverage (b) for the sensor network in Fig. 3.5(a).

that quantifies the ability of an omnidirectional sensor network S to detect straight tracks traversing $\mathcal{A}$, as a function of the sensors ranges $R_S = \{r_1, \ldots, r_n\}$, and positions $X_S = \{s_1, \ldots, s_n\}$.

We assign a Lebesgue measure μ on $[0,\ \pi]$ to any set of rays $K \subset \mathbb{R}^2$, such that $\mu\{\alpha : \mathcal{R}_\alpha \in K\}$. Then, the opening angle of the cone K is a measure on the set of rays contained by K. It follows from Remark 3.1.1 that the opening angle of the coverage cone $K(C_i, \mathbf{y}_0)$ is a measure on the set of tracks through $\mathbf{y}_0$ that are detected by the sensor C_i. Similarly, it follows from Proposition 3.1.2 that the opening angle of the k-coverage cone $K_k(S_k, \mathbf{y}_0)$ is a measure on the set of tracks through $\mathbf{y}_0$ that are detected by all sensors in S_k.

Based on Section 3.1.1, it is always possible to generate the k-coverage cone of a set S_k by means of two unit vectors $\hat{\mathbf{l}}^*$ and $\hat{\mathbf{h}}^*$. This convenient unit vector representation also allows to compute the opening angle of *any* coverage cone by means of the cross product. Let $\psi = \psi(S_k, \mathbf{y}_0)$ denote the opening angle of the k-coverage cone $K_k(S_k, \mathbf{y}_0)$ in (3.7), with origin $\mathbf{y}_0 = [0 \quad b_y]^T$, and $b_y \in \mathcal{I}_y$. This cone is finitely generated by two unit vectors $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\imath$ and $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath$ that are defined in terms of the relative-position vector (3.1). From the properties of the cross product

$$\sin\psi = \|\hat{\mathbf{l}}^* \times \hat{\mathbf{h}}^*\| \tag{3.16}$$

Thus, using (3.5)-(3.6), the opening angle can be written with respect to the sensors positions,

$$\psi = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \tag{3.17}$$

where

$$M_{\imath\jmath} \equiv \begin{bmatrix} \hat{\mathbf{l}}^{*T} \\ \hat{\mathbf{h}}^{*T} \end{bmatrix} = \begin{bmatrix} (\hat{\mathbf{v}}_\imath)^T Q_\imath^+ \\ (\hat{\mathbf{v}}_\jmath)^T Q_\jmath^- \end{bmatrix}, \quad \hat{\mathbf{v}}_i \equiv \frac{(\mathbf{x}_i - \mathbf{y}_0)}{||(\mathbf{x}_i - \mathbf{y}_0)||} \quad \text{for} \quad i = \imath, \jmath. \tag{3.18}$$

$H[\,\cdot\,]$ denotes the Heaviside function, and $\det(\,\cdot\,)$ denotes the matrix determinant. From Proposition 3.1.2, $\imath$ and $\jmath$ are the indices of the unit vectors $\hat{\mathbf{l}}_\imath$ and $\hat{\mathbf{h}}_\jmath$ with $\imath, \jmath \in I_{S_k}$, such that $\hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{l}}_i \in \Lambda(S_k, \mathbf{y}_0)$ and $\hat{\mathbf{h}}_\jmath \preceq \hat{\mathbf{h}}_i \in \Omega(S_k, \mathbf{y}_0)$ for $\forall i \in I_{S_k}$ (and obtained as shown in Appendix C). The Heaviside function in (3.17) ensures that if $\hat{\mathbf{l}}_\imath \succ \hat{\mathbf{h}}_\jmath$, then $\psi = 0$.

Consider now the case in which $1 \leq k \leq n$. We still wish to obtain a measure of the set of tracks through $\mathbf{y}_0$ that are detected by at least k sensors, namely $\mathcal{K}_k(S, \mathbf{y}_0)$. But, as shown in (3.9), this set is not always a cone. Thus, the Lebesgue measure μ on $\mathcal{K}_k(S, \mathbf{y}_0)$ is obtained through the principle of inclusion-exclusion [21, 62], as shown by the following result:

Theorem 3.1.3 *A measure on the set $\mathcal{K}_k(S, \mathbf{y}_0)$ for a family of non-translates disks $S = \{C_1, \ldots, C_n\} \subset \mathbb{R}^2_+$ is given by,*

$$\mathcal{T}^k_{\mathbf{y}_0}(X_S, R_S) = \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} \psi(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, \mathbf{y}_0), \quad m = \frac{n!}{(n-k)!k!} \tag{3.19}$$

where, the summation $\sum_{1 \leq i_1 < \ldots < i_j \leq m}$ is a sum over all the $[m!/(m-j)!\,j!]$ distinct integer j-tuples $(i_1, \ldots, i_j)$ satisfying $1 \leq i_1 < \ldots < i_j \leq m$. $S_k^{i_l}$ denotes the i_l^{th} k-subset of S, and the union $\{S_k^{i_1} \cup \ldots \cup S_k^{i_j}\}$ is a p-subset of S, with $k \leq p \leq n$.

A proof is provided in Appendix E. In the remainder of the dissertation, the union $\{S_k^{i_1} \cup \ldots \cup S_k^{i_j}\}$ is abbreviated as ${S_p}^{i_{1,j}}$.

The function $\mathcal{T}^k_{\mathbf{y}_0}$ provides a measure of the amount of tracks through y_0 that are detected by at least k sensors in a network S, as a function of the sensor positions, X_S, and ranges R_S. It can be seen from (3.19) that $\mathcal{T}^k_{\mathbf{y}_0}$ can be evaluated by summing the opening angles of the coverage cones of all p-subsets of S, with $k \leq p \leq n$. For a p-subset S_p, the coverage cone $K_p(S_p, \mathbf{y}_0)$ is generated by two unit vectors according to Proposition 3.1.2, and its opening angle $\psi(S_p, \mathbf{y}_0)$ is given by the cross product in (3.17). The above result is illustrated through a simple example in the next section, and will be used to derive a coverage function for $\mathcal{A}$ in Section 3.1.2.

Example: Track-coverage function for a single intercept, $\mathbf{y}_0$, and a network with $n = 3$ and $k = 2$

To illustrate the result in Theorem 3.1.3, we use a simple example involving a network $S = \{C_1, C_2, C_3\}$ positioned at $X_S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and ranges $R_S = \{r_1, r_2, r_3\}$, as shown in Fig. 3.7. Let $k = 2$ be the number of required detections for forming a track, and consider the tracks through $\mathbf{y}_0$. Then, from (3.19), the amount of tracks detected by at least two sensors in S is given by,

$$\begin{aligned} \mathcal{T}^2_{\mathbf{y}_0} &= \psi(S^1_2, \mathbf{y}_0) + \psi(S^2_2, \mathbf{y}_0) + \psi(S^3_2, \mathbf{y}_0) - \big[\psi(S^1_2 \cup S^2_2, \mathbf{y}_0) \\ &+ \psi(S^1_2 \cup S^3_2, \mathbf{y}_0) + \psi(S^2_2 \cup S^3_2), \mathbf{y}_0\big] + \psi(S^1_2 \cup S^2_2 \cup S^3_2, \mathbf{y}_0) \end{aligned} \tag{3.20}$$

where, from the definition of k-subset: $S^1_2 = \{C_1, C_2\}$, $S^2_2 = \{C_1, C_3\}$, and $S^3_2 = \{C_2, C_3\}$. But, the union of two or more k-subsets of S always produces a p-subset of S, with $k < p \leq n$. For instance, in this case $S^1_2 \cup S^2_2 = \{C_1, C_2, C_3\} = S$, and $S^1_2 \cup S^3_2 = S^2_2 \cup S^3_2 = S^1_2 \cup S^2_2 \cup S^3_2 = \{C_1, C_2, C_3\} = S$. Therefore, the above equation simplifies to:

$$\mathcal{T}^2_{\mathbf{y}_0} = \psi(S^1_2, \mathbf{y}_0) + \psi(S^2_2, \mathbf{y}_0) + \psi(S^3_2, \mathbf{y}_0) - 2\psi(S, \mathbf{y}_0) \tag{3.21}$$

This result is illustrated in Fig. 3.7.

Using the cross product, the opening angles in (3.21) can be expressed as explicit functions of the sensors positions $X_S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, as illustrated in Appendix F. This property of the coverage function allows to formulate the placement of the sensors as an optimization problem, in which X_S is to be determined (as shown in Section 4).

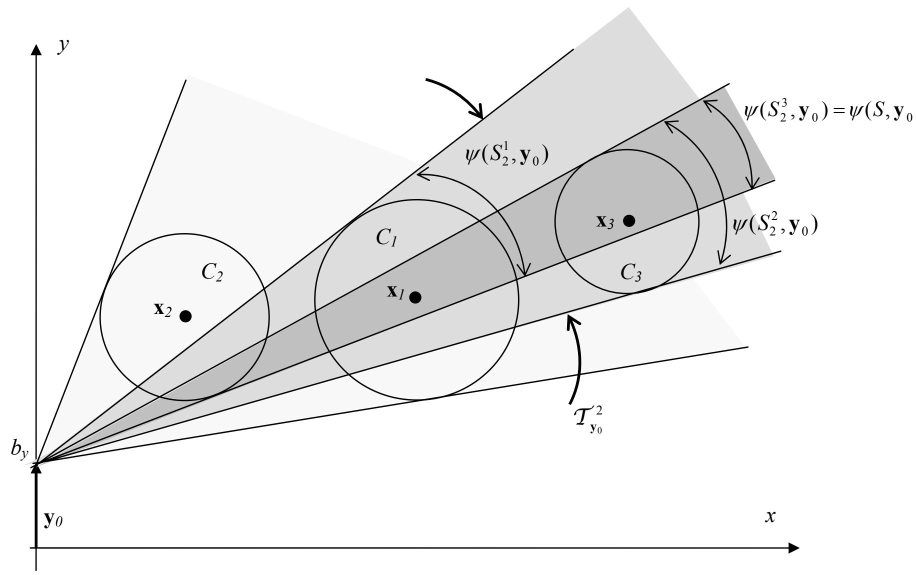


Figure 3.7: An example of coverage function, $\mathcal{T}^2_{\mathbf{y}_0}$, for three sensors $S = \{C_1, C_2, C_3\}$ located at $X_S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and $k = 2$.

Area Track-Coverage Function

The track-coverage function for a rectangular ROI $\mathcal{A}$ is obtained by considering the sets of tracks intersecting its four sides, $\mathcal{I}_y$, $\mathcal{I}_x$, $\mathcal{I}_{y'}$, and $\mathcal{I}_{x'}$, and leading to at least k detections by S. These sets can be represented by coverage cones, as illustrated in Section 3.1.1. Consider the set of tracks that intersect $\mathcal{I}_y$ and are detected by at least k sensors, $\mathcal{K}_k(S, \mathcal{I}_y)$, in (3.10). The sets in (3.10) are all disjoint because they contain rays with different intercepts, thus $\mathcal{K}_k(S, \mathbf{y}_0^{\ell_i}) \cap \mathcal{K}_k(S, \mathbf{y}_0^{\ell_j}) = \emptyset$ when $\ell_i \neq \ell_j$. Using the definition of Lebesgue measure for disjoint sets [63], it follows that a measure of the set $\mathcal{K}_k(S, \mathcal{I}_y)$, obtained from (3.10) and (3.19), is,

$$\mathcal{T}^k_{\mathcal{I}_y}(X_S, R_S) = \sum_{\ell=0}^{N_2} \mathcal{T}^k_{\mathbf{y}_0^\ell}(X_S, R_S) = \sum_{\ell=0}^{N_2} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} \psi({S_p}^{i_{1,j}}, \mathbf{y}_0^\ell) \quad (3.22)$$

where m, ${S_p}^{i_{1,j}}$, and the j-tuples $(i_1, \ldots, i_j)$ are all defined as in Theorem 3.1.3, and $N_2 = L_2/\delta b$. Thus, equation (3.22) is a measure approximating the amount of tracks that are detected by at least k sensors in S and intersect $\mathcal{I}_y$.

Then, obtain a measure on the set of tracks that traverse $\mathcal{A}$ and are detected by at least k sensors in S, denoted by $\mathcal{K}_k(S, \mathcal{A})$ and given by (3.12). The sets $\mathcal{K}_k(S, \mathcal{I}_y)$, $\mathcal{K}_k(S, \mathcal{I}_x)$, $\mathcal{K}_k(S, \mathcal{I}_{x'})$, and $\mathcal{K}_k(S, \mathcal{I}_{y'})$ in (3.12) are not disjoint, and a track intersecting one side of $\mathcal{A}$ always intersects one other side of $\mathcal{A}$. It follows that if we sum the measure μ on these four sets, every element in $\mathcal{K}_k(S, \mathcal{A})$ is counted twice. Thus, the true measure μ on $\mathcal{K}_k(S, \mathcal{A})$ can be obtained by dividing this sum by two. Let the opening angles of the k-coverage cones $K_k(S_k, \mathbf{x}_0)$, $K_k(S_k, \mathbf{y}'_0)$, and $K_k(S_k, \mathbf{x}'_0)$ be denoted by $\zeta(S_k, \mathbf{x}_0)$, $\xi(S_k, \mathbf{y}'_0)$, and $\rho(S_k, \mathbf{x}'_0)$, respectively. Then, a measure μ on $\mathcal{K}_k(S, \mathcal{A})$ is given by the following sum,

$$\begin{aligned} \mathcal{T}^k_{\mathcal{A}}(X_S, R_S) &= \frac{1}{2}\left[\mathcal{T}^k_{\mathcal{I}_y}(X_S, R_S) + \mathcal{T}^k_{\mathcal{I}_x}(X_S, R_S) + \mathcal{T}^k_{\mathcal{I}_{x'}}(X_S, R_S) + \mathcal{T}^k_{\mathcal{I}_{y'}}(X_S, R_S)\right] \\ &= \frac{1}{2}\sum_{\ell=0}^{N_2}\sum_{j=1}^{m}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le m}\left[\psi({S_p}^{i_{1,j}}, \mathbf{y}^{\ell}_0) + \xi({S_p}^{i_{1,j}}, \mathbf{y}'^{\ell}_0)\right] \qquad (3.23) \\ &+ \frac{1}{2}\sum_{\ell=0}^{N_1}\sum_{j=1}^{m}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le m}\left[\zeta({S_p}^{i_{1,j}}, \mathbf{x}^{\ell}_0) + \rho({S_p}^{i_{1,j}}, \mathbf{x}'^{\ell}_0)\right] \end{aligned}$$

where each term $\cup_{j=1}^{m} K_k(S_k, \cdot)$ in (3.12) has been written in terms of opening angles using Theorem 3.1.3. All opening angles are obtained from the cross product of two unit vectors (3.16), and can be written as explicit functions of the sensors positions and ranges, as shown in Appendix F.

In Chapters 4-6, the track-coverage function is used to optimize the deployment of sensor networks performing cooperative target detection. Also, in Chapter 4 the coverage cones and their opening angles are used to derive an upper bound for the track-coverage function, and the probability of cooperative target detection of the network.

3.2 Area Coverage in Omnidirectional Sensor Networks

A trivial increase to track coverage occurs when multiple sensors fields-of-views overlap, e.g., Fig. 3.8(a). However, multiple detections in the same location of the ROI may be multiple false alarms caused by the same false target or anomalous environmental conditions. Therefore, in order to prevent overlapping fields-of-view, two approaches are explored in this dissertation. The first approach imposes nonlinear state-constraints on X_S by prohibiting disks that represent sensors fields-of-view from overlapping one another. For example, the nonlinear constraint for two sensors $\mathbf{x}_i$ and $\mathbf{x}_j \in X_S$ with position vectors $\mathbf{x}_i = [x_i \quad y_i]^T$ and $\mathbf{x}_j = [x_j \quad y_j]^T$ and detection radii r_i and $r_j \in R_S$, is given by

$$\mathbf{c}(\mathbf{x}_i(t), \mathbf{x}_j(t)) \equiv -(x_i - x_j)^2 - (y_i - y_j)^2 + (r_i + r_j)^2 \leq 0, \quad i, j = 1, ..., n, \quad \forall i, j, \quad i \neq j \tag{3.24}$$

Then, for n sensors, the number of state constraints is $c_1 = \frac{n!}{2(n-2)!}$. Including this nonlinear overlapping constraint when optimizing the track-coverage function in affect limits the feasible track coverage solutions to the space of maximum area coverage.

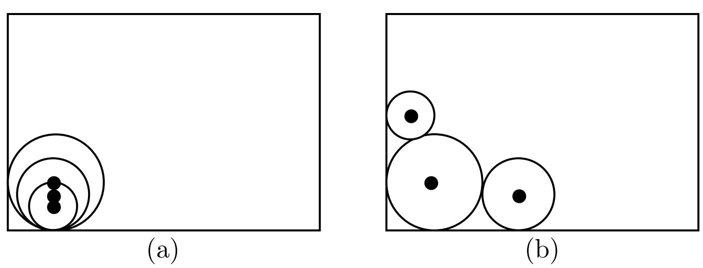


Figure 3.8: Simple example of $n = 3$ sensors in $\mathcal{A}$ that provide (a)maximum k-coverage but minimum area coverage, and (b) maximum k-coverage for the maximum area coverage solution.

The second approach consists of implementing an additional quality of service performance function, referred to as *area coverage* [14, 15]. By including the area coverage term in an objective function that is to be maximized in affect penalizes sensors whose fields-of-view intersect, thereby offsetting the trivial increase to the k-coverage. The measure of area coverage is defined by [14] as the union of area (units of length^2) of the area representing the field-of-view of each sensor,

$$A_C = \bigcup_{i=1,\ldots,n} A_i \tag{3.25}$$

where A_i is the area covered by the i^{th} sensor. Although the author mentions that sensors partially outside the ROI and those that overlap each other result in a decrease in total area coverage, these specific cases are not addressed mathematically. Therefore, an area coverage function that measures the distinct area in $\mathcal{A}$ covered by a set of n sensor is derived here.

Let $C_i(\mathbf{x}_i, r_i)$ continue to denote a disk with radius r_i centered at $\mathbf{x}_i$ that represents the field-of-view of this sensor. The area coverage for sensors constrained entirely within $\mathcal{A}$ and whose detection ranges do not overlap is simply the summation of the area of each disk,

$$A_0 = \sum_{i=1}^{n} H(A_{0,i}) \cdot \pi r_i^2 \tag{3.26}$$

where the heaviside function $H(\,\cdot\,)$ ensure that sensors entirely outside of $\mathcal{A}$ provides zero area coverage,

$$H(A_{0,i}) = \begin{cases} 0 & \text{if } x_i + r_i < 0 \mid y_i + r_i < 0 \mid x_i - r_i > L_1 \mid y_i - r_i > L_2 \\ 1 & \text{otherwise} \end{cases} \tag{3.27}$$

The three special cases that reduce (3.26) occur when the sensors fields-of-view (i) overlap, which results in redundant area coverage, (ii) are partially inside of $\mathcal{A}$, and (iii) are entirely outside of $\mathcal{A}$. Then, the total area coverage provided by a set of n sensors is formally expressed as,

$$A_C = A_0 - A_s - A_p \tag{3.28}$$

where A_s and A_p denote the total segment areas of the disks that are overlapping or partially outside of $\mathcal{A}$, respectively. A sensor whose field-of-view is entirely outside of $\mathcal{A}$ is considered by (3.27). Examples of each of these cases are depicted in Fig 3.9.

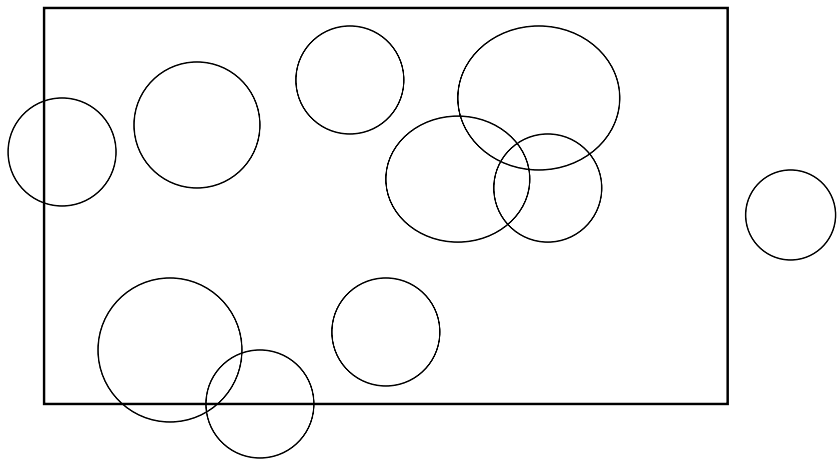


Figure 3.9: An example of a set of n=9 sensors deployed in $\mathcal{A}$, where the sensors overlap each other and are only partially in $\mathcal{A}$.

Each component of A_C in (3.28) is derived here in terms of the known variables $\mathbf{x}_i$ and r_i. The distance between the origins of the i^{th} and j^{th} sensors is given by the distance formula as,

$$h_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3.29}$$

When $h_{ij} < (r_i + r_j)$, the fields-of-view of these two sensors are overlapping, and the intersection area, a circular segment, must be subtracted from the total area of each sensor. A circular segment is the portion of a disk whose upper boundary is a (circular) arc s_i and whose lower boundary is a chord c_i making a central angle $\theta_i < \pi$ radians (180°). This is illustrated in Fig. 3.11(a) as the shaded region. A segment of a disk $A_{s,i}$ is calculated from a property in geometry [64] as follows,

$$A_{s,ij} = \frac{1}{2} r_i^2 \left(\theta_{ij} - \sin\theta_{ij}\right) \tag{3.30}$$

where the subscript ij denotes the i^{th} segment area due to the i^{th} and j^{th} sensor overlap. An example of this is illustrated in Fig. 3.10 by the darker gray shaded area. The reverse subscripts θ_{ji} refers to the j^{th} segment area due to this overlap, and is also illustrated in Fig. 3.10 by the lighter gray area. Let r_i be the radius of the disk and h_i the height of the triangle with sides $(r_i, h_i, \frac{1}{2}c_i)$. From Fig. 3.11, the central angle θ_{ij} is given by,

$$\theta_{ij} = 2\cos^{-1}\left(\frac{h_i}{r_i}\right) \tag{3.31}$$

where h_i is,

$$h_i = r_i \sin\left(\frac{\theta'_{ij}}{2}\right) \tag{3.32}$$

and θ'_{ij} is the opening angle between r_i and r_j for the triangle with sides (r_i, r_j, h_{ij}). The opening angle θ'_{ij} is given by the *law of cosines* as

$$\cos\theta'_{ij} = \frac{r_i^2 + r_j^2 - h_{ij}^2}{2r_i r_j} \Leftrightarrow \theta'_{ij} = \cos^{-1}\left(\frac{r_i^2 + r_j^2 - h_{ij}^2}{2r_i r_j}\right) \tag{3.33}$$

Substituting (3.31)-(3.33) into (3.30), the area for the segment of the i^{th} sensor overlapping the j^{th} sensor is

$$\begin{aligned} A_{s,ij} &= H(A_{s,ij}) \cdot \frac{r_i^2}{2}\left[\theta_{ij} - \sin\theta ij\right] \\ &= H(A_{s,ij}) \cdot \frac{r_i^2}{2}\left[2\cos^{-1}\left(\sin\left(\frac{\theta'_{ij}}{2}\right)\right) - \sin\left(2\cos^{-1}\left(\sin\left(\frac{\theta'_{ij}}{2}\right)\right)\right)\right] \qquad (3.34) \\ &= H(A_{s,ij}) \cdot \frac{r_i^2}{2}\left[2\cos^{-1}\left(\sin\left(\frac{1}{2}\cos^{-1}\left(\frac{r_i^2 + r_j^2 + (x_i - x_j)^2 + (y_i - y_j)^2}{2r_i r_j}\right)\right)\right)\right. \\ &- \left.\sin\left(2\cos^{-1}\left(\sin\left(\frac{1}{2}\cos^{-1}\left(\frac{r_i^2 + r_j^2 + (x_i - x_j)^2 + (y_i - y_j)^2}{2r_i r_j}\right)\right)\right)\right)\right] \end{aligned}$$

where θ_{ij} is a function of the sensor positions $\mathbf{x}_i$ and $\mathbf{x}_j$ and ranges r_i and r_j, i.e., $\theta_{ij} = \theta_{ij}(\mathbf{x}_i, r_i, \mathbf{x}_j, r_j)$. It follows that $A_{s,ij} = 0$ when two sensors do not overlap due to the heaviside function $H(A_{s,ij})$,

$$H(A_{s,ij}) = \begin{cases} 0 & \text{if } (r_i + r_j)^2 < h_{ij}^2 \\ 1 & \text{otherwise} \end{cases} \tag{3.35}$$

Then (3.34) is calculated for all the combination of n sensors taken two at a time, where order *does* matter, which leads to the total segment area, i.e., redundant area coverage,

$$A_s = \sum_{i=1}^{n} \sum_{j=1;i\neq j}^{n} H(A_{s,ij}) \cdot \frac{r_i^2}{2} \left(\theta_{ij} - \sin\theta_{ij}\right) \tag{3.36}$$

Due to symmetry of the i^{th} and j^{th} disks, (3.34) for $A_{s,ij}$ and $A_{s,ji}$ reduces to,

$$A_{s,ji} = \frac{r_j^2}{r_i^2} A_{s,ij} \tag{3.37}$$

Then, (3.36) is simplified to

$$A_s = \cdot \sum_{i=1}^{n-1} \sum_{j=2}^{n} H(A_{s,ij}) \cdot \frac{r_i^2}{2} \left(\theta_{ij} - \sin\theta_{ij}\right) \left(1 + \frac{r_i^2}{r_j^2}\right) \tag{3.38}$$

The reduction of area coverage for a sensor partially outside of $\mathcal{A}$ is derived similarly to A_s. The main difference is the value of h_i, which is now the perpendicular distance between the sensor origin and the reference axis-of-interest,

$$h_i = \|\mathbf{x}_i - \mathbf{x}_r\| = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}, \quad \forall\, r \in I_R \tag{3.39}$$

where $I_R = \{y, x, y', x'\}$ denotes the index set of the reference axes-of-interest, with specific vector values of $\mathbf{x}_r$ of the four axes, namely, $\mathbf{x}_y = [x_i,\ 0]^T$, $\mathbf{x}_x = [0,\ y_i]^T$, $\mathbf{x}_{y'} = [L1,\ 0]^T$, and $\mathbf{x}_{x'} = [0,\ L2]^T$. Then, the segment area of the disk outside of $\mathcal{A}$ is calculated by substituting (3.31), (3.32), and (3.39) into (3.30) for $j = r$, which is given for the i^{th} sensor and r-axis by,

$$
\begin{aligned}
A_{p,ir} &= H(A_{p,ir}) \cdot \frac{r_i^2}{2} \left[\theta_{ir} - \sin\theta_{ir}\right] \\
&= H(A_{p,ir}) \cdot \frac{r_i^2}{2} \left[2\cos^{-1}\left(\frac{(x_i - x_r)^2 + (y_i - y_r)^2}{r_i}\right)\right. \\
&- \left.\sin\left(2\cos^{-1}\left(\frac{(x_i - x_r)^2 + (y_i - y_r)^2}{r_i}\right)\right)\right]
\end{aligned}
\tag{3.40}
$$

It is important to note that when a sensor does not overlap a primary axis, $A_{p,ir} = 0$ due to the heaviside function $H(A_{p,ir})$,

$$
H(A_{p,i}) = \begin{cases} 0 & \text{if } -r_i < x_i < r_i \mid -r_i < y_i < r_i \mid L_1 - r_i < x_i < L_1 + r_i \\ & \quad \mid L_2 - r_i < x_i < L_2 + r_i \\ 1 & \text{otherwise} \end{cases}
\tag{3.41}
$$

Then, for all n sensors and four axes, the segmented are outside of $\mathcal{A}$ is given by,

$$
A_p = \sum_{i=1}^{n} \sum_{\forall r \in I_R} H(A_{p,ir}) \cdot \frac{r_i^2}{2} \left(\theta_{ir} - \sin\theta_{ir}\right)
\tag{3.42}
$$

where $\theta_{ir} = \theta_{ir}(\mathbf{x}_i, r_i, \mathbf{x}_r)$.

The total area coverage for a network of n sensors area coverage is calculated by substituting (3.26), (3.38), and (3.42) into (3.28). It has been shown here that A_C is a function of the sensors positions X_S and ranges R_S for a specific ROI. However, the maximum area coverage, where every point in $\mathcal{A}$ is encompassed within at least one sensor field-of-view, has an upper bound,

$$
A_C^{\max} = L_1 \cdot L_2
\tag{3.43}
$$

that is independent of k and n. In Chapter 6, the track-coverage function is used to optimize the deployment of sensor networks performing cooperative target detection subject to the strict constraint of maximum area coverage (3.44). Then, in Chapter 5, the objective function is a weighted sum of the track coverage (3.23) and the area coverage (3.28) functions.

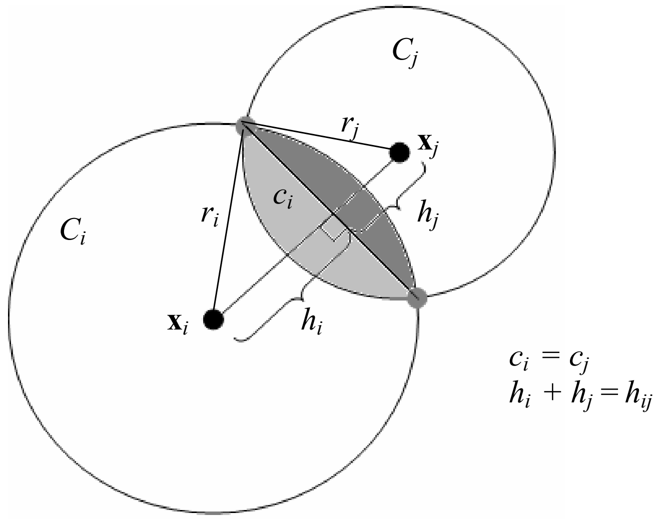


Figure 3.10: Geometry and notation of two overlapping sensors. The darker area represents $A_{s,i}$ while the lighter area represents $A_{s,j}$.

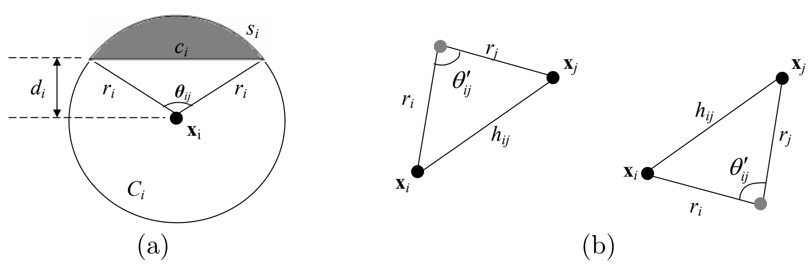


Figure 3.11: The geometry of (a) the segment of the i^{th} sensor, and (b) the triangle with sides (r_i, r_j, h_{ij}).

3.3 Vehicle Energy Consumption

In some applications, when the sensors are each attached to a controllable platform, such as an underwater glider, control is applied in order to maneuver the group of gliders along desired trajectories. The use of energy-optimal trajectories can extend the period of operation for autonomous underwater vehicles with limited energy resources by reducing the energy consumption. By including an energy term into the objective function, the system more effectively exploits the natural dynamics for vehicle transport, which has been used in a number of research areas, such as space mission designs for space mission design and low energy orbits [65].

Consider the energy source driving a network of underwater gliders. Let $\mathbf{u}(t) \in \mathbb{R}^q$ be the velocity of the n underwater gliders due to the control. As the velocity of the gliders are directly proportional to the energy, minimizing energy consumed by the group of gliders over a period of time is viewed as minimum-control-effort problem [66]. Then, the minimum expenditure of control effort is sought to transfer a system from an arbitrary initial state $\mathbf{x}(t_0) = [\mathbf{x}_1^T(t_0), ...\mathbf{x}_n^T(t_0)]^T \in \mathbb{R}^{2n}$ to final state $\mathbf{x}(t_f) = [\mathbf{x}_1^T(t_f), ...\mathbf{x}_n^T(t_f)]^T \in \mathbb{R}^{2n}$ along a trajectory (or path) $\mathbf{x}(t)$. Then, the general form of the performance measure for energy expenditure along an entire trajectory $\mathbf{x}(t)$ is,

$$J_E = \int_{t_0}^{t_f} \left[\mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right] dt \tag{3.44}$$

$$= \int_{t_0}^{t_f} \|\mathbf{u}(t)\|_R^2 dt \tag{3.45}$$

where $\mathbf{R} \in \mathbb{R}^{q \times q}$ is a real symmetric positive definite (i.e., $\mathbf{z}^T\mathbf{R}\mathbf{z} > 0 \forall \mathbf{z} \neq \mathbf{0}$ [66]) weighting matrix representing the relative importance of the energy of different sensors. The elements of $\mathbf{R}$ may be functions of time if it is desired to vary the weights on control-effort expenditure during the time-interval $[t_0, t_f]$. Typically, $\mathbf{R}$ is equal to the identity matrix $\mathbf{I}$ as this provides equal weight to all sensors over the entire trajectory.

3.4 Chapter Summary

This chapter formulates the quality of service measures to be used throughout this dissertation. Section 3.1 presents a novel track coverage formulation addressing the quality of service of sensor networks performing cooperative target detection. In many surveillance applications, simple (e.g., proximity) sensor networks are employed to detect passive unauthorized targets, such as aircraft and submarines, that may traverse a region of interest along a straight path. When sensor measurements are limited and subject to false alarms a track-before-detect approach is employed to form a feasible track before the target is positively detected. Thus, multiple and distinct sensor detections must be obtained from each target in what is referred to as cooperative detection. This section focuses on the geometric properties of these networks and formulates the problem of cooperative sensors detection of moving targets through the theory of geometric transversals. The network coverage is approached using a novel methodology based on the theory of cones. This methodology allows to represent sets of geometric transversals in closed-form, and to assign a Lebesgue measure that is a function of the sensors positions. Then, in Section 3.2, an area coverage function, which is commonly implemented in the coverage literature, is formally formulated to measure the amount of distinct area covered by the sensors field-of-view. This measure is necessary when sensors are employed in a moving environment, such as the ocean, in order to compensate the trivial solution to the track-coverage function (Chapter 5). The final performance function formulated in Section 3.3 measures the total energy consumption of a sensor network due to onboard control. When sensors are deployed on a controllable platform, such as an underwater glider (Chapter 6), limited energy resources and the affects of energy on the performance of each sensor provide the motivation for including the energy consumption term to be minimized.

Chapter 4

Track Coverage Optimization and Probability of Detection

The quality of service objective functions derived in Chapter 3.1 are implemented accordingly to address the following problem:

Problem 4.0.1 (Track Coverage Optimization) *Given a parameter* $1 \leq k \leq n$ *and a network* S *of* n *omnidirectional sensors with ranges* $R_S = \{r_1, \ldots, r_n\}$*, find the sensor positions* $X_S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ *inside a region-of-interest* $\mathcal{A}$ *such that the amount of tracks detected by at least* k *sensors in* S *is maximized.*

Using the track-coverage function obtained in Section 3.1.2, Problem 4.0.1 can be formulated as a nonlinear program (NLP). In order to obtain distinct sensor detections in the ROI, the sensors positions are constrained to lie in $\mathcal{A}$ and to prevent overlapping. Then, the set of optimal sensor positions X_S^* is given by the solution $\{\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*\}$ of the following NLP:

$$\text{maximize} \quad \mathcal{T}_{\mathcal{A}}^k(X_S, R_S), \tag{4.1}$$

$$\text{subject to} \quad (x_i - x_j)^2 + (y_i - y_j)^2 > (r_i + r_j)^2, \ \forall i, j \in I_S \tag{4.2}$$

$$0 < x_i < L_1, \ \forall i \in I_S \tag{4.3}$$

$$0 < y_i < L_2, \ \forall i \in I_S \tag{4.4}$$

where $\mathbf{x}_i^* = [x_i^* \quad y_i^*]^T$, and the objective function $\mathcal{T}_{\mathcal{A}}^k(X_S, R_S)$ is given by (3.23). Also, the NLP (4.1)-(4.4) can be easily modified to add sensors optimally to an existing network. In fact, suppose f sensors already exist in $\mathcal{A}$, and there is an opportunity for replenishing the network with q additional sensors. The NLP (4.1)-(4.4) can be written for a network $S = \{C_1, \ldots, C_f, C_{f+1}, \ldots, C_n\}$ with $n = q + f$ sensors, where now $\{s_1, \ldots, s_f\}$ are known constants, and $\{\mathbf{x}_{f+1}, \ldots, \mathbf{x}_n\}$ are the variables. Then, its solution $\{\mathbf{x}^*_{f+1}, \ldots, \mathbf{x}^*_n\}$ represents the set of sensor positions for optimally replenishing the network.

It is shown in Appendix G that the track-coverage function (3.23) has an upper bound,

$$\mathcal{T}^{\max}_{\mathcal{A}} = \left(\frac{L_1 + L_2}{\delta b} + 2\right)\pi \geq \mathcal{T}^k_{\mathcal{A}}(X_S, R_S), \quad \text{for} \quad \forall\ X_S, k, n \tag{4.5}$$

that is independent of k and n. This upper bound represents the track coverage provided by a sensor network that detects all tracks through $\mathcal{A}$ at least k times, where $\mathcal{A}$ is $L_1 \times L_2$. Therefore, it is referred to as *total track coverage*. In large sensor networks total track coverage may be achieved by concentric configurations placed around the perimeter of $\mathcal{A}$. However, in many applications the available sensors are not sufficient to provide total track coverage and $\mathcal{T}^k_{\mathcal{A}}$ can be maximized by determining the optimal placement X^*_S with known ranges R_S from (4.1)-(4.4).

The coverage cone representation of track coverage is also used to derive the probability of detection of targets in $\mathcal{A}$ as a function of X_S and R_S. In applications where there is no prior knowledge of target tracks any ray $\mathcal{R}_\alpha(b_y)$ has the same probability of representing an actual target track. Then, the probability that a target traversing $\mathcal{A}$ along a straight path is detected by at least k sensors in S is,

$$\begin{aligned} Pr^k_{\mathcal{A}}(X_S, R_S) \;&=\; \frac{\delta b}{2\pi(L_2 + \delta b)} \sum_{\ell=0}^{N_2} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} [\psi({S_p}^{i_{1,j}}, y^\ell_0) + \xi({S_p}^{i_{1,j}}, y'^\ell_0)] + \\ &+\; \frac{\delta b}{2\pi(L_1 + \delta b)} \sum_{\ell=0}^{N_1} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} [\zeta({S_p}^{i_{1,j}}, x^\ell_0) + \rho({S_p}^{i_{1,j}}, x'^\ell_0)] \end{aligned} \tag{4.6}$$

where m, ${S_p}^{i_{1,j}}$, and the j-tuples $(i_1, \ldots, i_j)$ are defined as in Theorem 3.1.3. A proof is provided in Appendix H. As in the previous sections, the opening angles ψ, ξ, ζ, and ρ are given by the functions in Appendix F, and are computed for every coverage cone of the p-subsets in (4.6), with $k \leq p \leq n$. The derivation in Appendix H can be modified to account for non-uniform probabilities of the tracks' heading and intercept. Due to space limitations, this topic will be the subject of a separate dissertation.

4.1 Static Optimization of the Track Coverage Function

The methodology developed in the previous sections is implemented to optimize track coverage with respect to the sensors positions. We show that a number of sensor deployment problems can be formulated as an NLP optimizing the track coverage function (3.23). In every case, the NLP solution, X_S^*, is determined by the sequential quadratic programming (SQP) algorithm [67, 68]. Multiple random initializations are utilized to avoid local maxima. In Section 4.1.1, the track coverage provided by networks deployed using the SQP algorithm is compared to that obtained by random and grid deployment strategies, which have been proposed by several authors, including [9] and [69]. The SQP algorithm is shown to improve track coverage by up to two orders of magnitude, compared to the former techniques. Also, a fast and efficient greedy algorithm implementing the coverage function (3.23) is shown to produce deployment strategies that are considerably more effective than random or grid deployment. In Section 4.1.2, the NLP solution is used to deploy sensors until a desired detection performance is achieved. The results show that this approach employs significantly smaller networks (e.g., with 50% fewer sensors) than the path-exposure deployment strategy proposed in [70]. In Section 4.1.3, the NLP is modified as explained in Section 4 in order to optimally replenish an existing sensor network. These results show that when a network is replenished by SQP the resulting track coverage is almost doubled compared to random or grid strategies.

Using on-board thrustors and GPS systems, even simple disposable sensors can be accurately positioned and replenished to maintain satisfactory surveillance. In some cases, these sensors can even reposition themselves once or twice during their lifetime and, consequently, the network performance can be significantly improved by using the methodology developed in this dissertation. In Section 4.1.4, an NLP is used to optimally reposition sensor networks with maneuvering capabilities. By repositioning each sensor within a region dictated by its power and energy limitations, it is possible to improve the track coverage of a sensor network by up to 69.4%.

Table 4.1: Sensor networks size and range

$\boldsymbol{n}$	$\boldsymbol{R_S}$ (Km)
10	{ 3, 3, 5, 5, 6, 6, 8, 8, 10, 10 }
15	{ 3, 3, 3, 5, 5, 5, 6, 6, 6, 8, 8, 8, 10, 10, 10 }
20	{ 3, 3, 3, 3, 5, 5, 5, 5, 6, 6, 6, 6, 8, 8, 8, 8, 10, 10, 10, 10}
40	{ 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5, 5, 5,... 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 8, 8, 8, 8, 8, 8, 8, 8}

4.1.1 Formulation of the Track Coverage Optimization Problem

The effectiveness of deploying sensors using the solution of the track coverage optimization problem (Problem 4.0.1) is demonstrated for the sensor networks in Table 6.1. The number of required detections, k, is made to vary between 2 and 4, and the dimensions of $\mathcal{A}$ are $L_1 \times L_2 = 150 \times 100$ (Km). In Table 4.2, the track coverage of sensors placed at the SQP solution X_S^* of the NLP (4.1)-(4.4) is compared to that obtained by random and grid deployments. In this table, the coverage function

$\mathcal{T}^k_{\mathcal{A}}(X_S)$ is normalized by $\mathcal{T}^{\max}_{\mathcal{A}}$, such that track coverage can be compared for different sensor networks and parameters . These results show that sensors placed at the SQP solution achieve significantly higher track coverage than the grid and random deployments, which have been previously proposed in the literature for placing sensors for cooperative target detection (e.g., see [9, 69]). In fact, the same network deployed via SQP can provide a track coverage up to 10 or 15 times higher than the coverage provided by random or grid deployments (Table 4.2, $n = 15$ and $k = 3$).

Table 4.2: Normalized track coverage as a function of network parameters and deployment strategy

Network Parameters	$\boldsymbol{\mathcal{T}^k_{\mathcal{A}}(X_S)/\mathcal{T}^{\max}_{\mathcal{A}}}$ **(SQP Improvement %)**			
$\boldsymbol{n, k}$	**SQP**	**Random**	**Grid**	**Greedy**
10,2	0.304	0.169 (79.9%)	0.158 (92.4%)	0.300 (1.33%)
10,3	0.158	0.033 (379%)	0.0392 (303%)	0.151 (4.64%)
10,4	0.0700	3.00×10^{-3} (2.23×10^3%)	4.90×10^{-4} (1.42×10^4%)	0.0680 (2.94%) 0.0680 (2.94%)
15,3	0.286	0.0764 (274%)	0.0912 (214%)	0.250 (14.4%)
15,4	0.172	0.0179 (861%)	0.0117 (1.37×10^3%)	0.149 (15.4%)
20,3	0.364	0.183 (98.9%)	0.169 (115%)	0.325 (12.0%)
40,3	0.578	0.440 (31.4%)	0.450 (28.4%)	0.471 (22.7%)
40,4	0.423	0.202 (109%)	0.226 (87.2%)	0.354 (19.5%)

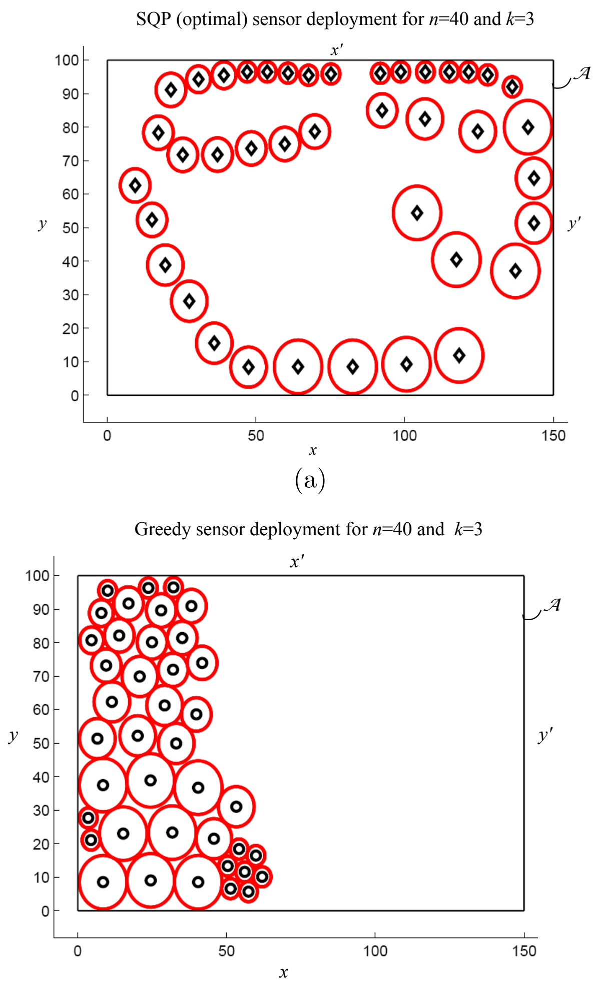
Also, a fast and effective greedy algorithm has been obtained by implementing the track coverage function in a packing algorithm proposed in [17], as shown in Algorithm 1. The original algorithm consists of packing unequal disks in a rectangle by placing them one at a time based on heuristic criteria and on their maximum hole degree performance [17]. The heuristic criteria are that the first disk is placed in the bottom-left corner of $\mathcal{A}$, and each subsequent disk must border two items (one side of $\mathcal{A}$, or another disk) and avoid overlapping. It is found that by implementing the coverage function (3.23) in lieu of the maximum hole degree performance function

[17], the resulting deployment is considerably more effective than the grid and random deployments. In fact, the track coverage of sensor networks deployed by the greedy algorithm lies within 1.3-22.7% of the optimal track coverage $\mathcal{T}_{\mathcal{A}}^k(X_S^*)$ (Table 4.2). Therefore, the greedy algorithm may be used in lieu of SQP when computation time is a concern.

Algorithm 1 Pseudocode of greedy track coverage algorithm.

```
order n sensors in S according to decreasing radii;
place first sensor in bottom-left corner of A;
for i = 2 to n do
   generate all eligible positions for sensor C_i;
   Require: C_i touches two items;
   for (every eligible placement of C_i) do
      calculate the coverage T_A^k({x_1, ..., x_{i-1}, x_i});
   end for
   select eligible placement x_i with maximum track-coverage;
end for
```

The SQP and greedy sensor deployments are plotted in Fig. 4.1 for $n = 40$ and $k = 3$, and can be compared to the grid and random deployments plotted in Fig. 4.2. The greedy algorithm tends to cluster sensors providing near-optimal track coverage when area coverage is low, but may cause track-coverage holes for small k, as shown in Fig. 4.1(b). This behavior could be prevented by optimizing a tradeoff between $\mathcal{T}_{\mathcal{A}}^k$ and ${\mathcal{T}_{\mathcal{A}}}^1$. The track coverage $\mathcal{K}_k(S, \mathcal{A})$ in (3.12) is plotted over parameter space in Fig. 4.3 for the SQP and grid deployments of a network with $n = 40$ and $k = 4$. Although the two sensor networks perform detections in similar regions of parameter space, the SQP deployment displays far less coverage holes leading to an 87.2% increase in track coverage (Table 4.2).



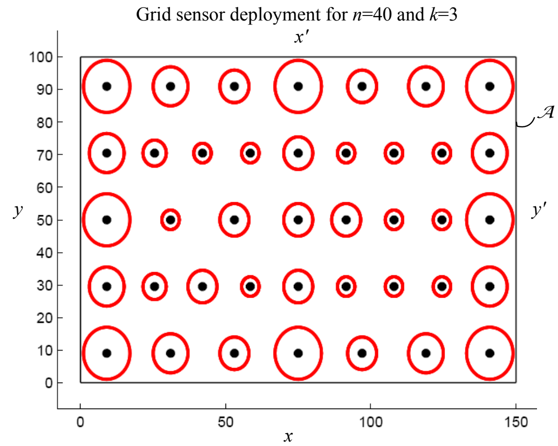
(a)

(b)

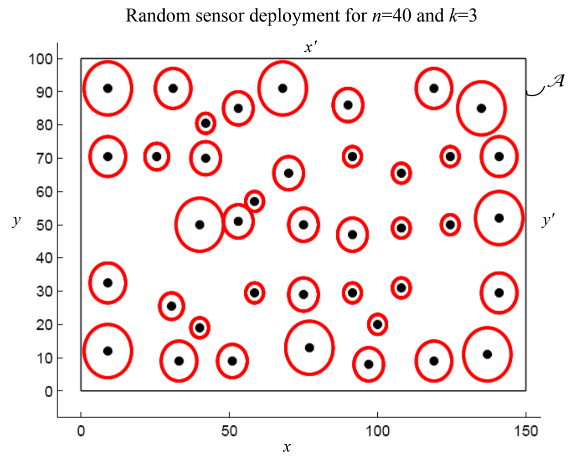
Figure 4.1: Deployment of a sensor network with $n = 40$ and $k = 3$ obtained by the SQP solution ($\diamond$) in (a), and by the greedy algorithm ($\circ$) in (b).

4.1.2 Application to Sensor Deployment for Achieving a Desired Detection Performance

The problem of deploying sensors for cooperatively detecting targets that traverse a region of interest has been previously considered in [70]. In this work, a sequential deployment algorithm was developed to achieve a desired detection performance by using a minimal number of sensors. In this section, we implement the same deployment strategy as [70] and show that, by deploying the sensors using the SQP



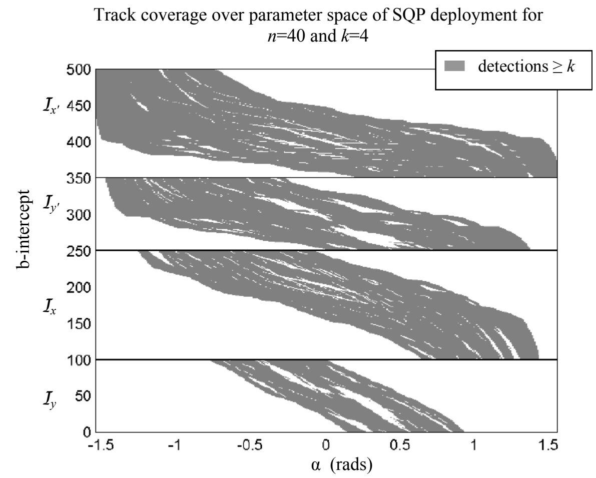
(a)



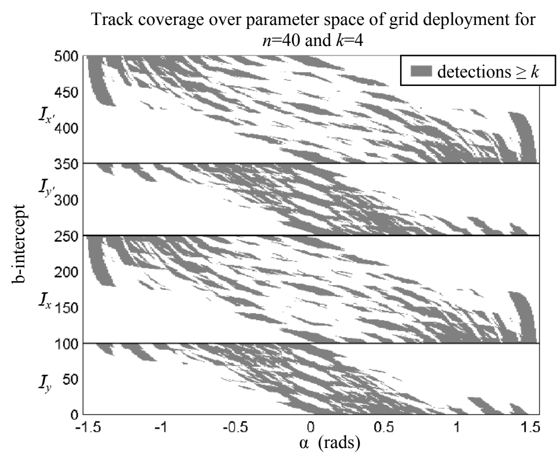
(b)

Figure 4.2: (a) Grid and (b) random deployments for a sensor network with $n = 40$ and $k = 3$.

solution of the NLP (4.1)-(4.4), the number of sensors $\hat{n}$ that is required to achieve a desired probability of detection $\hat{P}_{\mathcal{A}}^{k}$ can be decreased by up to 50%. In the first example, all sensors have the same range $r_i = 5$ Km, and the desired probability of detection is $\hat{P}_{\mathcal{A}}^{3} = 0.41$. When the sensor network is deployed using the sequential algorithm from [70], as shown in Fig. 4.4(a), the minimal number of sensors required is $\hat{n}_{\text{SEQ}} = 40$. Instead, when the sensor network is deployed using the SQP solution, the number of sensors required is only $\hat{n}_{\text{SQP}} = 30$ (Fig. 4.4(b)). In the second example, the desired probability of detection is $\hat{P}_{\mathcal{A}}^{3} = 0.18$, and the size of the network



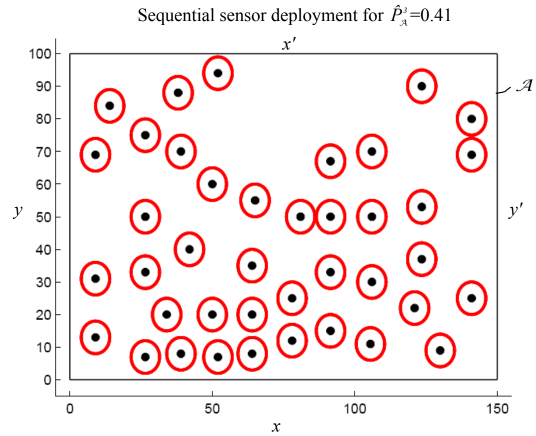
(a)



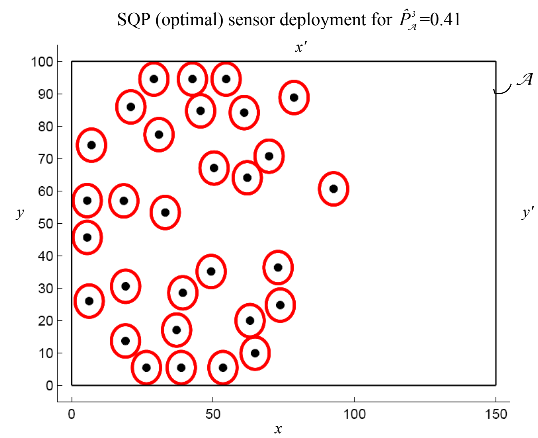
(b)

Figure 4.3: Track coverage $\mathcal{K}_k(S, \mathcal{A})$ of a sensor network with $n = 40$ and $k = 4$, deployed by (a) SQP and (b) grid strategies.

is increased according to the sensor networks (ranges) in Table 6.1. In this case, the sequential deployment algorithm from [70] requires a minimum of $\hat{n}_{\text{SEQ}} = 20$ to achieve the desired probability of detection. Whereas, the SQP deployment achieves the desired detection performance with only $\hat{n}_{\text{SQP}} = 10$ sensors (i.e., 50% less than the sequential algorithm).



(a)



(b)

Figure 4.4: Sequential deployment of $n = 40$ sensors in (a) and optimal deployment of $n = 30$ sensors in (b), all with range $r_i = 5$ Km.

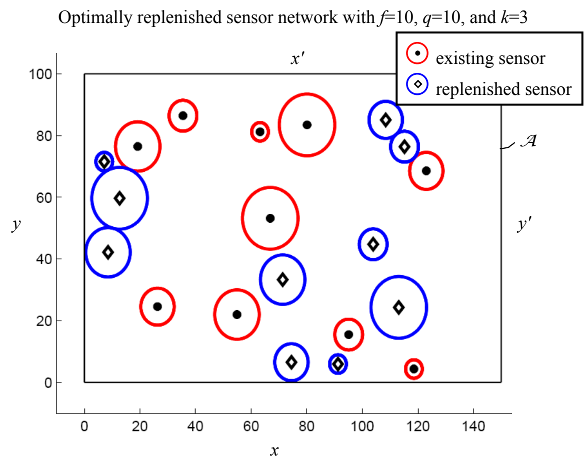
4.1.3 Application to Optimal Replenishment of Sensor Networks

In some applications, sensors cannot all be placed at desired locations or may be displaced over time. In this section, the NLP (4.1)-(4.4) is used to deploy a set of sensors for the purpose of replenishing an existing network that has suboptimal track coverage performance. It is assumed that the positions of f existing sensors in $\mathcal{A}$ are known, but the sensors have no repositioning capabilities. Therefore, the track coverage of the network is to be improved by adding an additional set of q sensors.

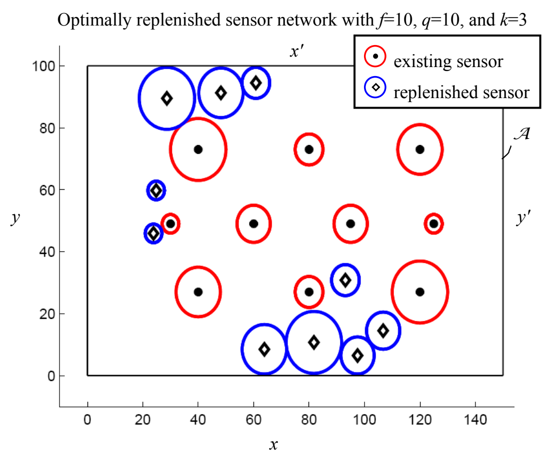
Consider an example with parameters $f = 10$, $q = 10$, and $k = 3$. The existing sensor network is shown by dots in Fig. 4.5(a), and provides a normalized track coverage $\mathcal{T}_{\mathcal{A}}^k/\mathcal{T}_{\mathcal{A}}^{\max} = 0.033$. The new set of $q = 10$ sensors is added at the optimal solution X_S^* of the NLP (4.1)-(4.4), as shown by the diamonds in Fig. 4.5(a). By replenishing the sensor network using this optimal deployment strategy, the track coverage of the entire network (with $n = q + f$ sensors) is improved by 715.2%. Whereas, when the q sensors are added using a random sequential strategy (adapted from [70]), the track coverage is improved by only 35.9%. In another example, the existing sensor network is in a grid configuration, as shown by dots in Fig. 4.5(b), and provides $\mathcal{T}_{\mathcal{A}}^k/\mathcal{T}_{\mathcal{A}}^{\max} = 0.039$. When an additional $q = 10$ sensors are deployed by the SQP algorithm (as shown by diamonds in Fig. 4.5(b)), the track coverage is improved by 635.9%. Whereas, the same set of sensors deployed by a random sequential strategy improves the track coverage by only 47.2%. Thus, by replenishing a sensor network with the methodology presented in this dissertation, its track coverage is improved significantly compared to existing deployment schemes.

4.1.4 Application to Optimal Repositioning of Sensor Networks

In applications where sensors are maneuverable (e.g., sensors are equipped with thrusters) an optimal deployment strategy can be obtained by including the allowed repositioning region in the NLP constraints. Without loss of generality, assume that all sensors have the same repositioning capabilities, and let w denote half the width of a square region within which each sensor can maneuver with the thrusters and power available, as shown in Fig. 4.6(a). Then, the NLP (4.1)-(4.4) is modified by replacing the constraints (4.3)-(4.4) with the following equations



(a)



(b)

Figure 4.5: Optimal replenishment of an existing sensor network with $f = 10$ sensors ($\bullet$) in a (a) random or (b) grid configuration, with $q = 10$ replenished sensors symbolized by diamonds ($\diamond$).

$$x_i - w < x_i < x_i + w, \ \forall i \in I_S \tag{4.7}$$

$$x_i - w > 0, \ \forall i \in I_S \tag{4.8}$$

$$x_i + w < L_1, \ \forall i \in I_S \tag{4.9}$$

$$y_i - w < y_i < y_i + w, \ \forall i \in I_S \tag{4.10}$$

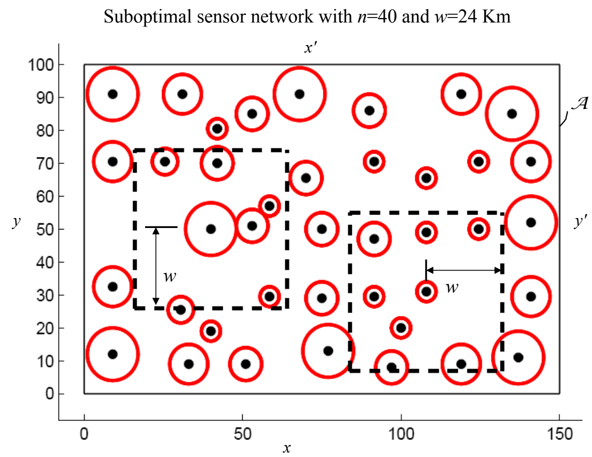
$$y_i - w > 0, \ \forall i \in I_S \tag{4.11}$$

$$y_i + w < L_2, \ \forall i \in I_S \tag{4.12}$$

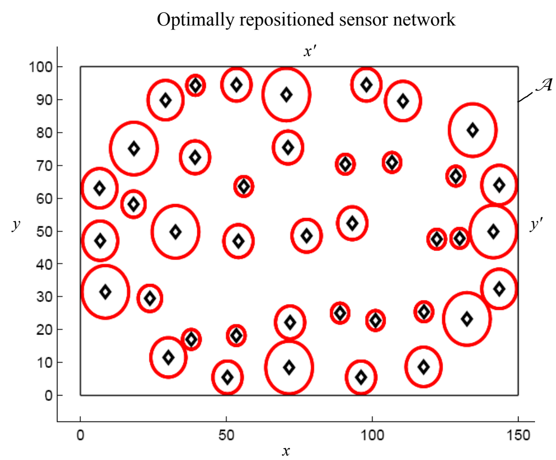
The SQP solution X_S^* of the resulting NLP constitutes the new positions to be assumed by the maneuvering sensors in order to improve the overall track coverage of the sensor network. Consider the sensor network in Fig. 4.6(a), with suboptimal track coverage $\mathcal{T}_{\mathcal{A}}^k/\mathcal{T}_{\mathcal{A}}^{\max} = 0.44$, $n = 40$, $k = 3$, and $w = 24$ Km. When these sensors are repositioned using the SQP solution, as shown by the diamonds in Fig. 4.6(b), the track coverage of the network is improved by 27.7%. As another example, consider the sensor network illustrated in Section 3.1.1, Fig. 3.5(a), with suboptimal track coverage $\mathcal{T}_{\mathcal{A}}^k/\mathcal{T}_{\mathcal{A}}^{\max} = 0.183$, $n = 20$, $k = 3$, and $w = 24$ Km. When these sensors are repositioned using the SQP solution, the track coverage is improved by 69.4%. Thus, the methodology presented in this dissertation can be used to improve the track coverage of a sensor network by allowing existing sensors in $\mathcal{A}$ to maneuver subject to power and energy constraints.

4.2 Chapter Summary

The novel track coverage formulation in Section 3.1 is optimized using a nonlinear program (NLP). The numerical results show that optimal deployment can increase track coverage by up to two orders of magnitude compared to existing grid and random deployment schemes. This methodology can decrease the number of sensors required to provide a desired probability of detection by up to 50% compared to existing path-exposure techniques. Also, it can significantly improve track coverage by replenishing or repositioning an existing sensor network that displays suboptimal performance due to errors in its initial placement or to sensors being displaced over time by winds or oceanic currents.



(a)



(b)

Figure 4.6: A suboptimal sensor network (a) in which every sensor has the capability of maneuvering within a region of width $2w$ (dashed line) is optimally repositioned using SQP in (b).

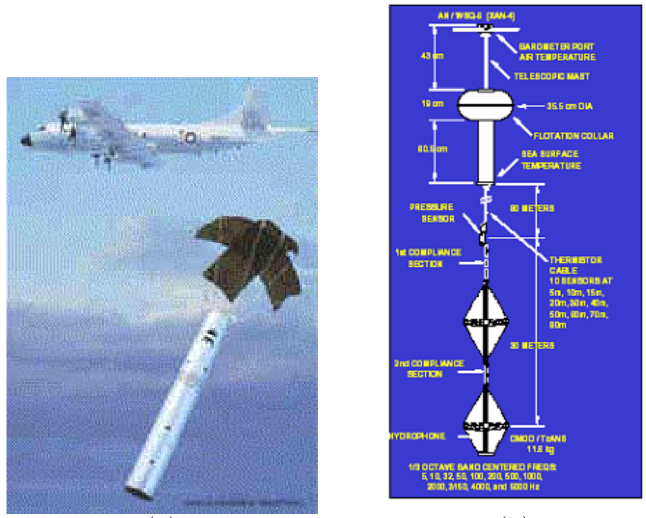
Chapter 5

Optimal Deployment of Acoustic Sensor Networks in an Oceanic Environment

In Chapter 3 (Section 3.1), the track coverage function is formulated with respect to the ranges R_S and fixed locations X_S of the sensors within $\mathcal{A}$. Since many distributed sensors are not naturally stationary within their environment (e.g., sensors distributed in the ocean), track coverage must be formulated to address moving sensor networks. It has long been recognized in practice that a moving sensor network, such as one comprised of sonar buoys drifting due to the oceanic current, can have a detrimental impact on the effectiveness of a distributed sensor network for maintaining surveilling coverage of an ROI. A drifting sensor network typically develops significant track-coverage holes over time. A coverage hole is defined as a region in parameter space where tracks are not detected by at least k sensors. Another undesirable outcome is the increased redundant coverage, which takes place when more than k sensors detect the same set of tracks. If the sensors have no control inputs, e.g., they are non-maneuverable free-floating sensors, the trajectories that maximize the overall sensor network performance over a period of time are in terms of the initial conditions that represent the initial location of the sensors. The drift dynamics induced by an oceanic environment are accounted for by utilizing oceanographic models and measurements of the ocean current, which produce a known forcing vector field in the buoy equations of motion [4]. Then, both a finite measure of the cumulative coverage provided by a sensor network and the drift dynamics of the environment must be accounted for in order to optimize the dynamic sensor network configuration. The track coverage function optimization problem stated in Problem 4.0.1 is reformulated in this chapter to include drift dynamics and measure the track coverage of a sensor network over a fixed period of time.

A typical free-floating sensor network is comprised of sonobuoys that are deployed from an aircraft in canisters (Fig. 5.1(a)). Then, upon impact of the water an inflatable float with a radio transmitter remains on the surface for communication, while the passive acoustic sensor and stabilizing equipment descends to a (preset) depth. The schematic of a sonobuoy example AN/WSQ-6 is shown in Fig. 5.1(b). To minimize the impact that the currents have on the performance of the sonobuoy system, the current vector field is modeled and accounted for by the track-coverage function optimization. One popular approach to measuring oceanic currents, referred to as the Lagrangian approach, employs a buoy known as a drifter that rides at the ocean surface. Tracking this drifter (by satellite, radar, radio, sound [71–76]) then provides a description of the ocean current. Other methods for obtaining current measurements include radar-based measurements, such as Coastal Ocean Dynamics Applications Radar (CODAR) [77], and satellites [5]. In view of these recent technological developments, a methodology is developed here for optimally placing a set of proximity sensors whose dynamics are formulated in terms of the surface current-velocities specified by a known vector field.

In this chapter, a novel sensor deployment problem is presented with the objective of providing maximum track coverage of a rectangular region of interest over time by means of moving sensors. The approach developed in this chapter leads to a new problem in dynamic computational geometry pertaining the geometric transversals of moving families of objects. It is shown that a state-space representation of the motions of the individual sensors subject to the current vector field can be derived from sonobuoys oceanic drift models. Also, the heterogenous environmental conditions, such as bathymetry, surface temporal variability, and bottom properties, are



(a) (b)

Figure 5.1: (a) Aircraft deployment of a sonobuoy, and (b) the schematic of the AN/WSQ-6 taken from [4].

known in practice to influence the field-of-view of an acoustic sensor with respect to its position within $\mathcal{A}$ (Section 5.1.2). In addition to the environmental effects on the sensor range, a sensor network is subject to random disturbances, such as unforseen and uncontrollable variations in both the sensor location and field-of-view. The uncertainties surrounding sensor movement and positioning in the ocean include actual currents, severe weather, and accuracy of initial deployment, due to heterogenous environmental conditions. Both the uncertainty and position-dependent sensor range affect the overall track coverage provided by the sensor network. Therefore, the nominal (i.e., ideal) solution to the optimal initial positions for maximum track coverage over a fixed period of time, which accounts for both the oceanic current-velocities and the position-dependent sensor ranges, is tested for robustness by incorporating uncertainty into the system.

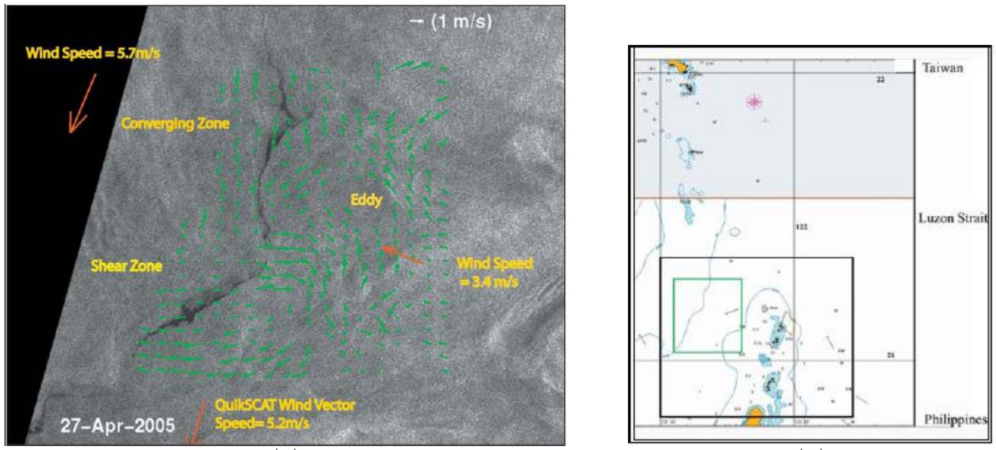
Thus the sensor network research for optimal track coverage investigated here seeks to extend the optimization of the track coverage function to the case of a moving sensor field, i.e., sensors moving according to oceanic drift, the case of position-dependent *sensor range as due to heterogenous environmental conditions, and finally to the case of random disturbances and partially-unknown state, as due to changes in the ocean current and range model.*

5.1 Methodology

5.1.1 Sonobuoy Equations of Motion

The ocean-current velocity profile induced by an oceanic environment is acquired by oceanographic models [4], satellite [5], or by Coastal Ocean Dynamics Applications Radar (CODAR) [77]. Surface currents can be measured through oceanographic models from *past* measurements acquired from previously deployed sonobuoys in the ocean, as explained in [4]. The measurement of surface currents by CODAR, a high frequency radar system, employs a transmitter that sends out radio waves that scatter off the ocean surface and then return to a receiver antenna. Using this information and the principles of the Doppler shift, CODAR is able to calculate the speed and direction of the surface current.

Another method for obtaining the ocean surface current vector components utilizes state-of-the-art satellite technology. Currently, the most efficient way of deriving the surface currents consists of performing *feature tracking*, which overlaps multiple synthetic aperture radar (SAR) images taken from different satellites over a short period of time [5]. SAR is a side-looking imaging radar that transmits a series of short, coherent pulses to the ground. Then, the high-resolution image is produced by detecting small Doppler shifts to the moving radar. The image-collecting sensors on each satellite have very different dynamic ranges of data, and filtered data with the same dynamic range are essential for feature tracking. The SAR data obtained from different satellites is matched by means of a 2-dimensional band-pass data filter that is localized in both frequency and time, and employs wavelet transforms [5]. For example, Figure 5.2(a) taken from [5], shows the ocean surface drift (green arrows) derived from the wavelet analysis of two satellites' SAR data over the Luzon Strait near the Philippines (Figure 5.2(b)).



(a) (b)

Figure 5.2: (a) Ocean surface drift (green arrows) derived from two satellites' SAR data over the Luzon Strait, and (b) the location map with the SAR image coverage area shown in the large box taken from [5].

Once a current vector field has been obtained by one of the above methods, it can be employed in buoy equations of motion that have been validated through experiments in the ocean, and are taken from [4]. The sonobuoy response to a 3-dimensional current profile is represented by a two orthogonal planar current profile characterized by the drag equation

$$f_d = \frac{1}{2}\rho C_d A V^2 \tag{5.1}$$

where ρ is the fluid density, C_d is the object's coefficient of drag, and A is the object's cross-sectional area, all of which are assumed to be constant. f_d is the total drag on a sphere obtained from the steady-state solution to Stokes' problem along the local current velocity vector. V, the magnitude of the fluid relative velocity vector past the object, which is a function of both position and time but is excluded in the notation for simplicity, has the following upper and lower sonobuoy components (denoted by subscripts u and ℓ, respectively and illustrated in Fig. 5.3(a)),

$$\Delta\mathbf{v}_u \equiv \mathbf{u}_u - \boldsymbol{v} \tag{5.2}$$

$$\Delta\mathbf{v}_\ell \equiv \boldsymbol{v} - \mathbf{u}_\ell, \tag{5.3}$$

assuming that the velocity profile in the vertical direction can be approximated as shown in Figure 5.3(b). Each velocity vector can be described in the plane as $\Delta\mathbf{v}_i = [\Delta v_{x_i} \quad \Delta v_{y_i}]^T$, the water velocity components $\mathbf{u}_i = [u_{x_i} \quad u_{y_i}]^T$, and the sonobuoy velocity components $\boldsymbol{v} = [v_x \quad v_y]^T$.

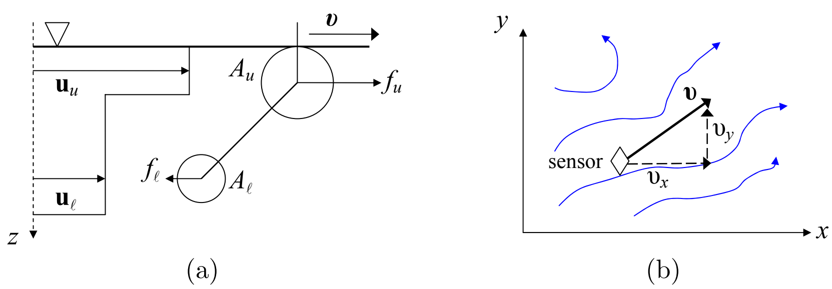


Figure 5.3: (a) The upper and lower components of a sonobuoy in which a force balance of $f_u = f_\ell$ is applied, (5.5)-(5.6), and (b) is the view from above.

In order to describe the sononbuoy velocity by a differential equation,

$$\dot{\mathbf{x}} = \boldsymbol{v}(x, y, t), \tag{5.4}$$

a force balance is applied to the upper and lower spheres that approximate the sonobuoy, as shown in Figure 5.3. It follows that the equations in the $x-$ and $y-$directions are:

$$C_{d_u} A_u (\Delta \mathrm{v}_{x_u})^2 = C_{d_\ell} A_\ell (\Delta \mathrm{v}_{x_\ell})^2 \tag{5.5}$$

$$C_{d_u} A_u (\Delta \mathrm{v}_{y_u})^2 = C_{d_\ell} A_\ell (\Delta \mathrm{v}_{y_\ell})^2. \tag{5.6}$$

Introducing the constant $\beta = \sqrt{(C_{d_\ell} A_\ell)/(C_{d_u} A_u)}$, the relative velocities can be written as $\Delta \mathrm{v}_{x_u} = \beta \Delta \mathrm{v}_{x_\ell}$ and $\Delta \mathrm{v}_{y_u} = \beta \Delta \mathrm{v}_{y_\ell}$. Then, the velocity of the buoy in (5.2)-(5.3) is,

$$\boldsymbol{\upsilon} = \begin{bmatrix} \frac{u_{x_u} + \beta u_{x_\ell}}{1+\beta} \\ \frac{u_{y_u} + \beta u_{y_\ell}}{1+\beta} \end{bmatrix} \tag{5.7}$$

Now, let $u_{x_\ell} = \alpha_x u_{x_u}$ and $u_{y_\ell} = \alpha_y u_{y_u}$, and for simplicity assume that $\alpha_x = \alpha_y = \alpha$, with $0 \leq \alpha \leq 1$. Then (5.7) can be written as,

$$\boldsymbol{\upsilon} = \begin{bmatrix} \left(\frac{1+\beta\alpha}{1+\beta}\right) u_{x_u} \\ \left(\frac{1+\beta\alpha}{1+\beta}\right) u_{y_u} \end{bmatrix} = \gamma \begin{bmatrix} u_{x_u} \\ u_{y_u} \end{bmatrix}, \tag{5.8}$$

with the scalar $\gamma \equiv (1+\beta\alpha)/(1+\beta) \leq 1$. Then, it follows that by assuming the buoys move with the surface current, i.e., $\Delta \mathbf{v}_i = 0$, the buoy equation of motion (5.4) in terms of the nonlinear, time-varying (NLTV) currents at the i^{th} sonobuoy location is written in terms of the i^{th} sensor location,

$$\dot{\mathbf{x}}(t) = \boldsymbol{\upsilon}(\mathbf{x}_i(t), t) = [\boldsymbol{\upsilon}_x(\mathbf{x}_i(t), t) \quad \boldsymbol{\upsilon}_y(\mathbf{x}_i(t), t)]^T \tag{5.9}$$

where $\mathbf{x}_i = [x_i \quad y_i]^T$, and the ocean current velocity vector for the entire sensor network is denoted as $\boldsymbol{\upsilon} = [\boldsymbol{\upsilon}_1^T \ ... \ \boldsymbol{\upsilon}_n^T]^T$. Then the unforced (i.e., no control) system dynamics (5.10) is written in terms of the entire sensor network as

$$\dot{\mathbf{x}}(t) = \boldsymbol{\upsilon}(\mathbf{x}(t), t), \quad \dot{\mathbf{x}} \in \mathbb{R}^{2n} \tag{5.10}$$

Modeling of the Nonlinear, Time-Varying Oceanic Currents

In practice, the ocean-induced current velocity represented by (5.10) is a nonlinear, time-varying function. The current velocity profile $\boldsymbol{\upsilon}(x, y, t)$ can be obtained by the methods discussed in Section 5.1.1, specifically Coastal Ocean Dynamics Applications Radar (CODAR) [77], which is the method used here. CODAR data was obtained from COOL [77] at Rutgers University in tabular form, which describes the current velocity profile off the coast of NJ (shown in Fig. 5.6(a)) for the coordinates over several days. Examples of the measured current-velocities are illustrated in Fig. 5.4, which clearly shows the currents in the $x-$ and $y-$direction are in fact nonlinear. Then, by comparing υ_x and υ_y in Figs. 5.4(a)-5.4(b), respectively, to the currents taken approximately two days later illustrated in Figs. 5.4(c)-5.4(d) over the same ROI, the time-varying behavior of the currents is also illustrated.

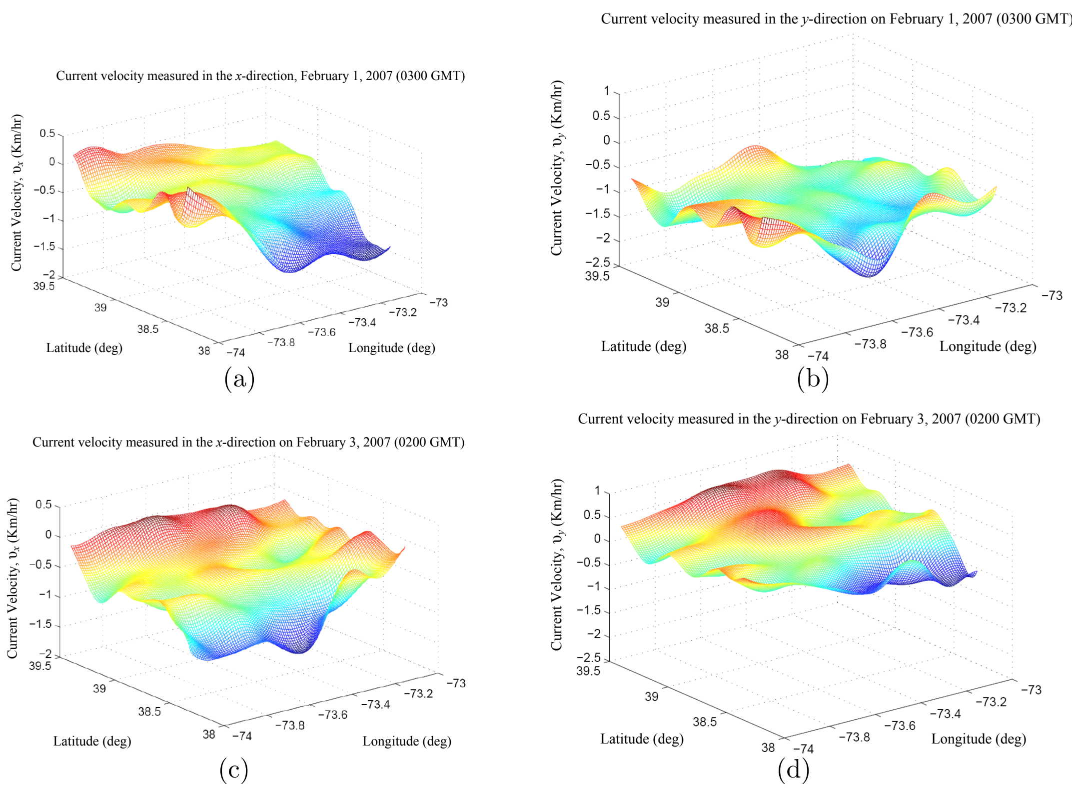


Figure 5.4: Current velocity measured by CODAR in the (a) x-direction on February 1, 2007 (0300 GMT), (b) y-direction on February 1, 2007 (0300 GMT), (c) x-direction on February 3, 2007 (0200 GMT), (d) y-direction on February 3, 2007 (0200 GMT).

The true underlying functional that describes the spatial and temporal characteristics of the current velocity field is unknown for a large area of the ocean. Without making prior assumptions on its functional form, the current velocity profile data is approximated by means of a nonlinear neural network. The CODAR training data consists of a set of input/output samples $T = \{\mathbf{y}_a, \mathbf{z}_a\}_{a=1,\dots,p}$ such that $\mathbf{z}^a = \boldsymbol{\upsilon}(\mathbf{y}^a)$ from (5.10), where $\mathbf{y}_a = [x(t),\ y(t),\ t]^T$ and $\mathbf{z}_a = [\upsilon_x,\ \upsilon_y]^T$. The ranges of T are listed in Table 6.1. A two-layer feedforward neural network is created and illustrated in Fig. 5.5 for a three-element input vector $\mathbf{y}(t)$, s hidden neurons from the hyperbolic tangent sigmoid transfer function,

$$\mathbf{h}(\mathbf{y}(t)) = \frac{1}{1 + e^{-(\mathbf{w}_1 \cdot (\mathbf{y})^T + \mathbf{b}_1)}} \tag{5.11}$$

and two linear output neurons,

$$\boldsymbol{\upsilon}(\mathbf{y}(t)) = \mathbf{w}_2 \cdot \mathbf{h}(\mathbf{y}(t)) + \mathbf{b}_2 \tag{5.12}$$

where (5.12) is the approximated NLTV ocean current in (5.10). The NN input and output weights $\mathbf{w}_1 \in \mathbb{R}^{s\times 3}$ and $\mathbf{w}_2 \in \mathbb{R}^{2\times s}$, and input and output biases $\mathbf{b}_1 \in \mathbb{R}^{s\times 1}$ and $\mathbf{b}_2 \in \mathbb{R}^{2\times 1}$, are obtained through supervised learning in batch mode from the CODAR data. The automated regularization training algorithm ('trainbr' [78]) updates the weight and bias values according to Levenberg-Marquardt optimization through Bayesian Regularization [79, 80]. This training algorithm is used here as it is known to generalize well for function approximation problems with possibly that may contain noisy data. The effectiveness of the NN approximator in (5.11)-(5.12) for $s = 100$ hidden nodes is verified through the following simulations. The trajectories of three gliders with zero on-board control are randomly placed in the region of interest in the ocean with measured CODAR information. As shown in Fig. 5.6, the gliders' trajectories simulated over a period of 5-days confirm that the approximator provides a satisfactory closed-form representation for the real CODAR measurements.

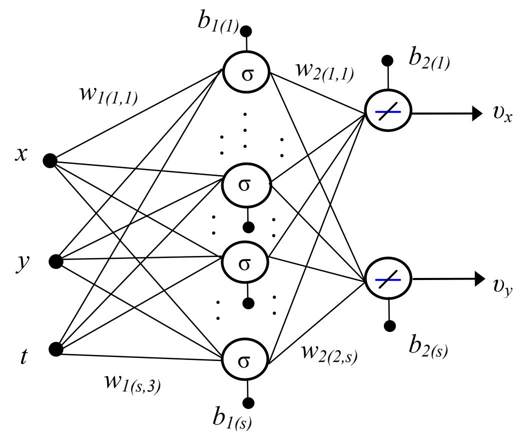


Figure 5.5: Neural network architecture with the elements of the input and output weighting matrices, $\mathbf{w}_1$ and $\mathbf{w}_2$, denoted by $w_i(j,\ell)$ for i=1,2 and j,ℓ are the matrix indices.

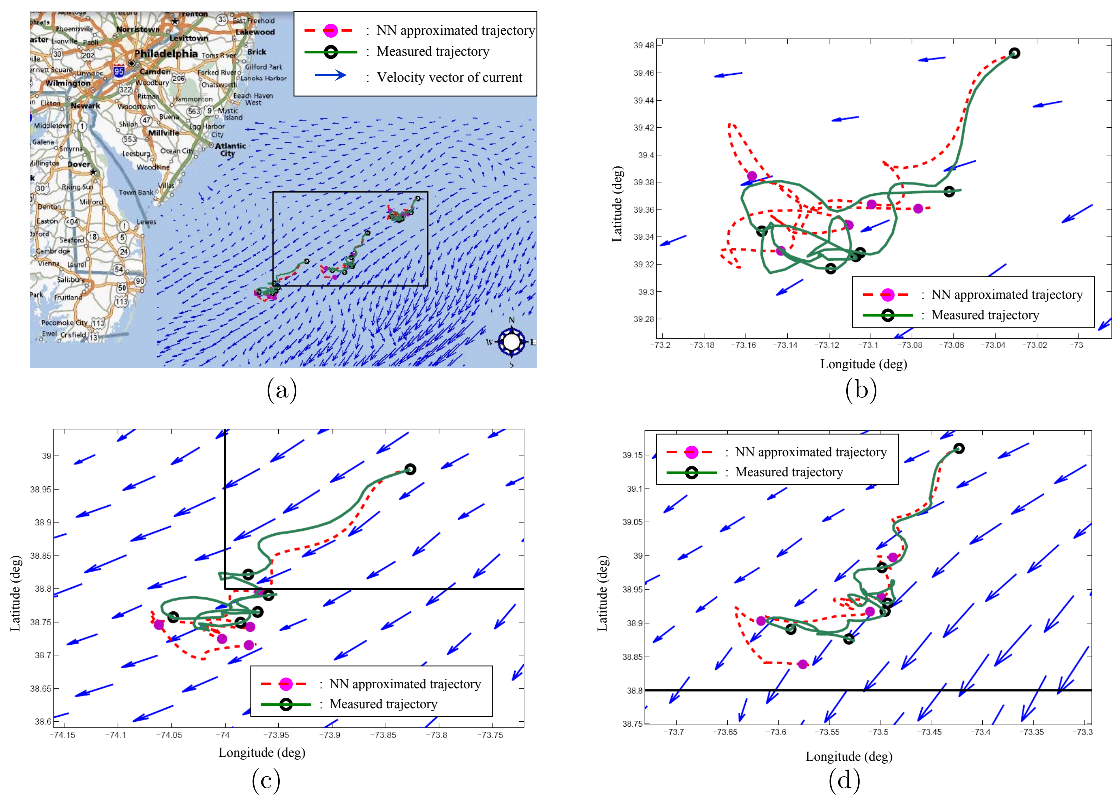


Figure 5.6: NLTV velocity field off the coast of NJ with coordinates $(-74.1°, -72.7°)$ longitude and $(38.6°, 39.5°)$ latitude is measured by CODAR and approximated by the NN. The approximation is validated through simulation for three sonar buoys deployed in $\mathcal{A}$ (a), where (b)-(d) depict the zoomed in trajectory comparisons.

Table 5.1: Range of the NN input/output samples obtained from CODAR measurements

Variable	Range
x	$-74.6° - \ -72.54°$ longitude
y	$38.6° \ - \ 39.5°$ latitude
v_x	-1.93 Km/hr $-$ 2.41 Km/hr
v_y	-2.37 Km/hr $-$ 1.81 Km/hr
t	0 $-$ 120 hours, Feb. 1, 2007 (0000 GMT) - Feb. 5, 2007 (2300 GMT)

5.1.2 Environmental Effects on the Acoustic Sensor Range

While an acoustic sensor field-of-view placed in the ocean can be approximated by a disk, the radius (range) is known to depend upon the local environmental conditions surrounding the sensors, because they affect the sound propagation process [81]. The acoustic effects from the sea surface act as a near-perfect reflector scattering much of the sound energy due to its temporal variability. The ocean bottom also has high variability acoustic properties that range from perfectly reflective to almost total attenuation, with sea floor roughness accounting for significant scattering. Thus, an improved strategy for sensor deployment would account for the sensors range dependency upon the environmental conditions, which may vary considerably within $\mathcal{A}$.

The sensor field-of-view is now approximated by a disk centered at the sensors origin whose range is no longer a constant but is a function of the local environmental conditions influencing the sensor measurements, i.e.:

$$r_i = r(\mathbf{x}_i(t)) \equiv r^e(\mathbf{x}_i(t)) \tag{5.13}$$

A BN approach has been developed in [1] to determine (5.13) within $\mathcal{A}$ for the variables listed in Table 5.2. BNs organize the body of knowledge for a given system by mapping deterministic relationships among all relevant variables. They can be used to estimate unknown variables (in this case, the range) and make predictions by combining probabilistic data with heuristic arguments. A BN model of acoustic-wave propagation is trained using the range-dependent acoustic model (RAM) [82,83], where the resulting BN structure learned from the RAM data is illustrated in Fig. 5.7(a).

The BN model can be combined with the sonar equation [81] to obtain the function describing maximum range of the sensor field-of-view, $r^e(\mathbf{x}_i)$, due to the environmental conditions with reasonable accuracy [1]. This function describes the maximum distance between the target and the sensor that may lead to a positive CPA detection by a sensor located at $\mathbf{x}_i$. The target strength, TS, and the detection threshold, denoted as ϑ or in sonar literature as DT, are known quantities given by the target of interest and sensor characteristics, and are assumed to be location invariant. The noise level, NL, over a ROI is assumed to have a Gaussian distribution with range interval [66, 78] dB, and is estimated using the ambient-noise spectra [81] for the ship-traffic in the ROI. Then, using the passive sonar equation,

$$SL + DI_s - PL - (NL - DI) = DT \tag{5.14}$$

the maximum value of PL leading to a detection that surpasses the threshold DT can be determined for a sensor with known directivity index, DI, and target-source directivity, DI_s. Subsequently, the maximum value of PL and any known environmental conditions near $\mathbf{x}_i(t)$ are provided as evidence, denoted by the set $e = \{Z, SF, BD, V, F, PL\}$, to the BN model in Fig. 5.7(a). The probability that the range assumes any one of its possible values is given by Bayes' Rule [84, 85],

$$Pr(R|e) = \frac{Pr(e|R)Pr(R)}{Pr(e)} \tag{5.15}$$

Then, the sensor range $r^e(\mathbf{x}_i(t))$ is estimated according to the highest probability of its probability distribution $Pr(R|e)$, such as, $r^e(\mathbf{x}_i(t)) = \text{argmax } Pr(R|e)$ [1]. The estimated range of the sensor field-of-view $r^e(\mathbf{x}_i(t))$ computed by this method for the same $\mathcal{A}$ used in Fig. 5.6(a) is plotted in Fig. 5.7(b), with respect to the latitude and longitude coordinates. It can be seen that the sensor performance varies significantly with respect to the sensor location as a consequence of the environmental conditions.

Table 5.2: List of variables of BN acoustic model from [1], where an instantiation refers to the value taken by the variable.

Variable Type	Variable (units)	Number of Instantiations: [Interval]
Target Position	Range, R (m)	depends on application, e.g., 40: [100:100:4000]
	Depth, Z (m)	depends on application, e.g., 10: [50:50:500]
Environment	Sea Floor, SF	3: flat, uphill, downhill, up and down
	Bottom Density, BD (g/cc)	10: [1.5:0.1:2.4]
	Bottom Sound Speed, V (m/s)	10: quadratic experiential function of BD
Source Parameters	Source Frequency, SF (Hz)	depends on source characteristics, e.g., 20: [10:10:200]
Output	Propagation Loss, PL (dB)	depends on discretization method, e.g., 10-20 instantiations (see [1])

5.1.3 Optimization of Cumulative Track Coverage Over a Fixed Period of Time

The optimization of the track coverage provided by a sensor network over a fixed period of time consists of optimizing the space of line transversals of a moving family of disks. As the sensors are non-maneuverable, the trajectories of the sensors depend only on their initial conditions, namely, their initial positions in $\mathcal{A}$. The coverage function (3.23) is used to obtain a measure of the cumulative coverage over time in terms of an integral objective function of the Lagrange type. This objective function is derived through a dynamic computational geometry approach that expresses a

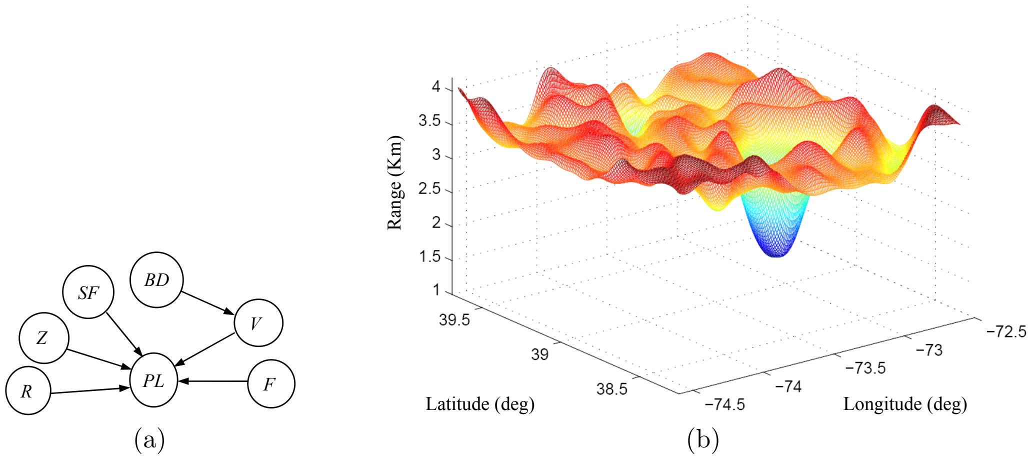


Figure 5.7: (a) BN model of acoustic wave propagation learned from RAM for sensor parameters and environmental variables defined in Table 5.2, and (b) sensor range over an oceanic ROI.

Lebesgue measure on the space of line transversals in closed form. Since the objective function is not quadratic and is composed of several terms, the solution of this optimal control problem becomes increasingly difficult as $\mathcal{A}$ becomes larger and the number of sensors increases.

In this chapter, the sensor network is assumed to be governed by (5.23), and the goal is to find the initial conditions for which the resulting trajectories provide maximum cumulative coverage. As discussed in Section 3.2, k-coverage increases as sensors fields-of-view overlap. When sensors are moving due to the ocean-induced current velocities but are non-maneuverable, it is feasible to impose the nonlinear constraint (3.24) only on the initial positions, which is considered the controllable aspect of the problem, and not the trajectories. In order to describe the coverage over a period-of-time with respect to the initial positions, the objective function must then include the area coverage measure (Section 3.2), as this penalizes sensors whose fields-of-view overlap, but does not impose a hard constraint that may reduce the feasible solution space unnecessarily. Therefore, the objective function is a weighted sum of the track and area coverages provided by n sensors,

$$\begin{aligned}\mathcal{J}(\mathbf{x}(t), \mathbf{r}(\mathbf{x}(t)), t) &= \int_{t_0}^{t_f} \left\{ W_{\mathcal{T}} \mathcal{T}_{\mathcal{A}}^k \left[X_S(t), R_S(X_S(t))\right] + W_C A_C \left[X_S(t), R_S(X_S(t))\right] \right\} dt \\ &\equiv \int_{t_0}^{t_f} \left\{ W_{\mathcal{T}} \mathcal{T}_{\mathcal{A}}^k \left[\mathbf{x}(t), \mathbf{r}(\mathbf{x}(t))\right] + W_C A_C \left[\mathbf{x}(t), \mathbf{r}(\mathbf{x}(t))\right] \right\} dt \end{aligned} \tag{5.16}$$

where the set $X_S(t)$ as a vector is denoted as $\mathbf{x}(t) = [\mathbf{x}_1^T(t), ..., \mathbf{x}_n^T]^T \in \mathbb{R}^{2n}$, and the set $R_S(\mathbf{x}(t))$ as a vector is denoted as $\mathbf{r}(\mathbf{x}(t)) = [r_1(\mathbf{x}_1(t)), ..., r_n(\mathbf{x}_n(t))]^T \in \mathbb{R}^n$, and $W_{\mathcal{T}}$ and W_C represent the scalar weights on the track and area coverage, respectively. Because the track coverage and area coverage functions, (3.23) and (3.28), respectively, are implicit functions of the sensor position and range within $\mathcal{A}$ (Chapter 3), then these functions are naturally time-varying when extended to the problem of measuring the track coverage over a fixed period of time for location-dependent sensor ranges, i.e., $\mathcal{T}_{\mathcal{A}}^k(X_S, R_S) = \mathcal{T}_{\mathcal{A}}^k(X_S(t), R_S(X_S(t))$ and $A_C(X_S, R_S) = (X_S(t), R_S(X_S(t))$. The sensor positions $\mathbf{x}(t)$ in the objective function (5.16) can be related to their initial positions by integrating the system dynamics (5.10),

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \boldsymbol{v}(\mathbf{x}(t), t) dt. \tag{5.17}$$

where $\boldsymbol{v}(\mathbf{x}(t), t)$ are obtained from the NN approximator discussed in Section 5.1.1.

The objective of the optimization problem of maximizing the cumulative coverage, $\mathcal{J}$ in (5.16), is to initially place n sensors in an ROI such that their ability to cooperatively detecting moving targets over a fixed period of time is optimized. Using the objective function (5.16), this problem can be reformulated as a NLP. In order to obtain distinct sensor detections in the ROI, the sensors initial positions are constrained to lie in $\mathcal{A}$ and to prevent overlapping. Then, the initial positions

$\mathbf{x}(t_0) = \mathbf{x}_0$ is given by the solution $\mathbf{x}_\mathbf{0}^*$ of the following NLP:

$$\max_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}(t), \mathbf{r}^e(\mathbf{x}(t))), \tag{5.18}$$

$$\text{subject to} \;\; \mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \boldsymbol{\upsilon}(\mathbf{x}(t), t)dt \tag{5.19}$$

$$0 > (r_{i,0}^e + r_{j,0}^e)^2 - (x_{i,0} - x_{j,0})^2 - (y_{i,0} - y_{j,0})^2, \quad \forall i, j \in I_S \tag{5.20}$$

$$0 < x_{i,0} < L_1 \quad \forall i \in I_S \tag{5.21}$$

$$0 < y_{i,0} < L_2 \quad \forall i \in I_S \tag{5.22}$$

where $\mathbf{x}_0^* = [\mathbf{x}_1^{*\,T}(t_0)...\mathbf{x}_n^{*\,T}(t_0)]^T$, notation is simplified to $r^e(\mathbf{x}_i(t_0)) = r_{i,0}^e$, $x_i(t_0) = x_{i,0}$, and $y_i(t_0) = y_{i,0}$, and the objective function is given by (5.16).

Optimal Deployment of a Moving Sensor Network Based on Linear, Time-Invariant Equations of Motion

In order to illustrate the problem of computing the initial positions that maximize track coverage of an entire trajectory over a fixed period of time, a preliminary simplifying assumption to the buoy equation of motion (5.10) is to reduce it to a linear, time-invariant (LTI) state space model,

$$\frac{d\mathbf{x}(t)}{dt} = \boldsymbol{\upsilon} = \mathbf{A}\mathbf{x}(t), \tag{5.23}$$

Then, the LTI system dynamics are incorporated into preliminary analysis of optimal deployment of moving sensors in Sections 5.1.3-5.1.3 in order to illustrate the optimal deployment problem posed in the previous section. These results also motivate the much more computationally expensive optimal deployment of sensors according to the nonlinear, time-varying system dynamics, and the inclusion of the area coverage term in the objective function. As this is a preliminary example, the ranges are assumed constant.

For this example, the objective function is formulated in terms of the cumulative track coverage, as follows,

$$\mathcal{J}(\mathbf{x}(t), t) = \int_{t_0}^{t_f} \mathcal{T}_{\mathcal{A}}^k[\mathbf{x}_1(t), ..., \mathbf{x}_n(t)]dt. \tag{5.24}$$

Since the governing equation of the system dynamics (5.23) is linear, the sensor positions $\mathbf{x}(t)$ can be related to their initial positions by the transition matrix, $\mathbf{\Phi}(t, t_0)$:

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0). \tag{5.25}$$

Where, $\mathbf{\Phi} = e^{\mathbf{A}(t-t_0)}$, and the elements of $\mathbf{A}$ are constant parameters obtained from the known current vector field. A general form for $\mathbf{\Phi}(t, t_0)$ is derived in Section 5.1.3. Then, (5.25) is substituted in (5.24), and the integral is maximized with respect to $\mathbf{x}(t_0) = \mathbf{x}_0$ in order to obtain the optimal initial position, $\mathbf{x}_0^*$. Then, the optimization problem in (5.18)-(5.22) is restated for this example as,

$$\max_{\mathbf{x}_0} \mathcal{J}, \tag{5.26}$$

$$\text{subject to} \;\; \mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0) \tag{5.27}$$

$$(x_{i,0} - x_{j,0})^2 + (y_{i,0} - y_{j,0}^2 > (r_{i,0} + r_{j,0})^2, \;\; \forall i, j \in I_S \tag{5.28}$$

$$0 < x_{i_0} < L_1, \;\; \forall i \in I_S \tag{5.29}$$

$$0 < y_{i,0} < L_2, \;\; \forall i \in I_S \tag{5.30}$$

In order to solve (5.26)-(5.30) for the initial sensor positions, the NLP solution, $\mathbf{x}_0^*$, is determined by the sequential quadratic programming (SQP) algorithm [67, 68].

Example: Optimization of Dynamic Track Coverage for $n = k = 1$

A simple example with $n = 1$ sensor and $k = 1$ is presented in order to illustrate the solution approach outlined in the previous section. $\mathbf{\Phi}$ is derived for a general state-space matrix $\mathbf{A}$ representing the vector field and can easily be applied to $n > 1$ by increasing the dimensions appropriately. Assuming the buoys moves in a linear fashion with the surface current according to (5.23), $\mathbf{A}$ is a $2n \times 2n$ matrix that for one sensor can be defined as,

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \tag{5.31}$$

where the elements of $\mathbf{A}$ are obtained from the current vector field. The trajectory of one sensor can be described in terms of the initial sensor position using (5.25) as follows

$$\dot{\mathbf{x}} = \mathbf{A}\boldsymbol{\Phi}(t, t_0)\mathbf{x_0} \tag{5.32}$$

The eigenvalues, or roots, of the characteristic equation $\det(s\mathbf{I} - \mathbf{A})$ are found to be,

$$\lambda_1 = \frac{K_1 + \sqrt{K_1^2 - 4 \cdot K_2}}{2} \tag{5.33}$$

$$\lambda_2 = \frac{K_1 - \sqrt{K_1^2 - 4 \cdot K_2}}{2} \tag{5.34}$$

where $K_1 = d + a$ and $K_2 = ad - bc$. Therefore, the transition matrix becomes,

$$\boldsymbol{\Phi}(t, 0) = \begin{bmatrix} (c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}) & (c_3 e^{\lambda_1 t} + c_4 e^{\lambda_2 t}) \\ (c_5 e^{\lambda_1 t} + c_6 e^{\lambda_2 t}) & (c_7 e^{\lambda_1 t} + c_8 e^{\lambda_2 t}) \end{bmatrix}. \tag{5.35}$$

Because $\boldsymbol{\Phi}(0, 0) = \mathbf{I}$ and $\dot{\boldsymbol{\Phi}}(0, 0) = \mathbf{A}$, a system of eight simultaneous equations is used to obtain the eight unknowns in $\mathbf{c} = [c_1, ..., c_8]^T$ in terms of the constants a, b, c, d, λ_1, and λ_2, as shown in Table 5.3.

Substituting the values in Table 5.3 into (5.35), and substituting (5.35) into (5.32), the optimal initial conditions can be obtained by maximizing the resulting integral function (5.24). For example, for one sensor and $k = 1$, the integrand of the cost

Table 5.3: The constants of $\mathbf{\Phi}$

$c_1 = \frac{a-\lambda_2}{\lambda_1-\lambda_2}$	$c_3 = \frac{b}{\lambda_1-\lambda_2}$	$c_5 = \frac{c}{\lambda_1-\lambda_2}$	$c_7 = \frac{d-\lambda_2}{\lambda_1-\lambda_2}$
$c_2 = \frac{a-\lambda_1}{\lambda_2-\lambda_1}$	$c_4 = \frac{b}{\lambda_2-\lambda_1}$	$c_6 = \frac{c}{\lambda_2-\lambda_1}$	$c_8 = \frac{d-\lambda_1}{\lambda_2-\lambda_1}$

function (5.24) simplifies through the relationship $\sin\theta = r/\|\mathbf{v}\|$,

$$\begin{aligned}\mathcal{J} &= \frac{1}{2}\int_0^{t_f}\left[\sum_{b_y=0}^{L_2}\frac{r}{\|\mathbf{v}(t)\|_y} + \sum_{b_x=0}^{L_1}\frac{r}{\|\mathbf{v}(t)\|_x}\right.\\ &+ \left.\sum_{b_{y'}=0}^{L_2}\frac{r}{\|\mathbf{v}(t)\|_{y'}} + \sum_{b_{x'}=0}^{L_1}\frac{r}{\|\mathbf{v}(t)\|_{x'}}\right]dt, \qquad (5.36)\end{aligned}$$

where $\|\mathbf{v}\|$ is the position vector relative to the axes indicated by the subscript, for example, $\|\mathbf{v}(t)\|_x = \sqrt{(x-b_x)^2+y^2}$.

Track Coverage Optimization of the Linear, Time-Invariant System

The methodology developed in this chapter is used to optimize the track coverage of a moving sensor network with respect to an area of interest over a period of time. This problem is relevant to sensor networks floating and drifting in the ocean subject to the surface currents that are employed for detecting moving targets in a region of interest. A cumulative track coverage function is presented in Section 5.1.3 and is optimized subject to the LTI system dynamics. The $(k = 3)$ - track coverage of a network with $n = 10$ sensors, and ranges $R_S = \{3,\ 3,\ 5,\ 5,\ 6,\ 6,\ 8,\ 8,\ 10,\ 10\}$, is considered. The parameter k represents the number of CPA detections that are required for declaring a track detected. For comparison, the sensors are first placed according to the optimization of the static coverage function (3.23), without accounting for the buoys dynamics, as shown in blue in in Figure 5.8. When the cumulative track coverage function (5.24) is optimized subject to the drift dynamics (5.23), the sensor network is deployed at the positions shown in red in Figure 5.8. The resulting sensors trajectories, plotted in Figures 5.9(a) and 5.9(b), differ significantly due to the diversity in the oceanic currents that are experienced by the individual sensors. Consequently, the track coverage of the two sensor networks also differs significantly.

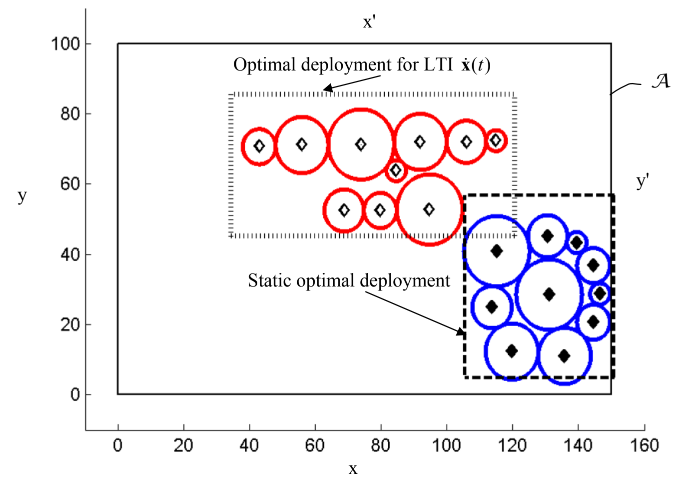
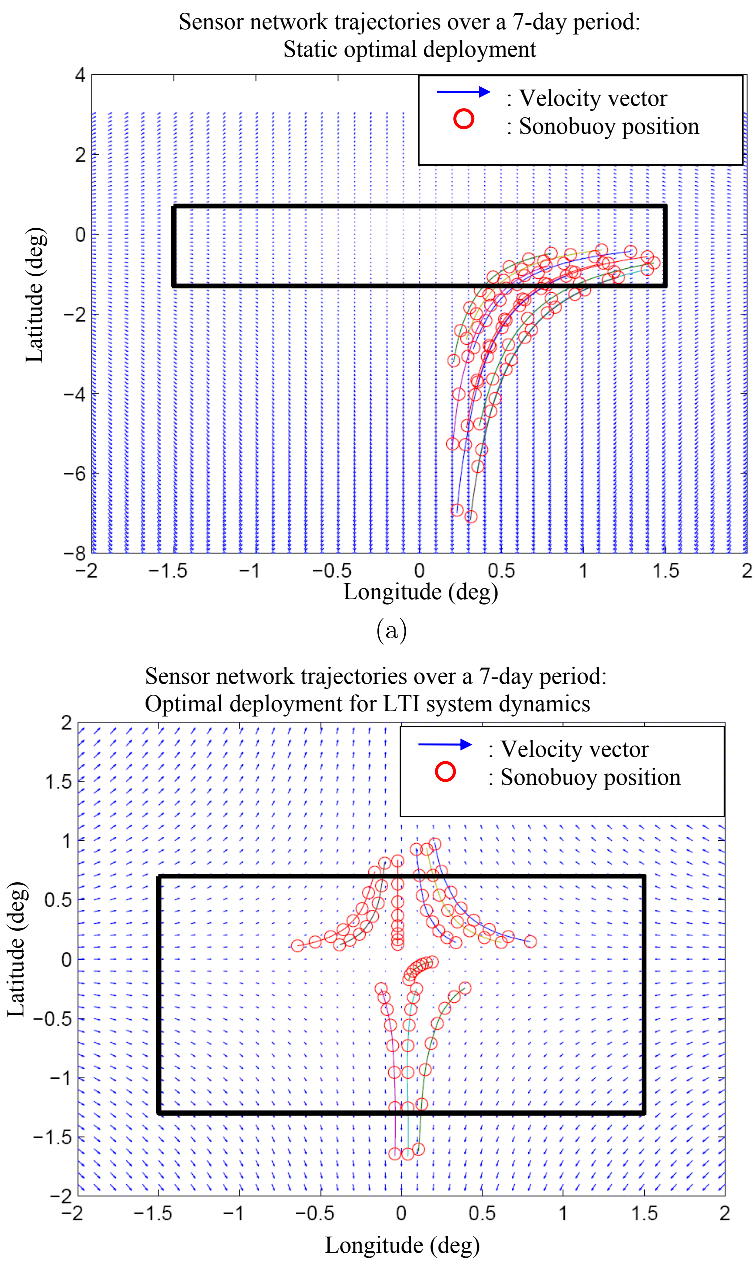


Figure 5.8: Comparing the initial sensor configurations by the static optimization (4.1)- (4.4) and the optimal deployment (5.26)-(5.30).

The time histories of the track coverage provided by the drifting sensor networks are plotted in Fig. 5.10. Although the two sensor networks are comprised of the same number of sensors and of the same individual performance (range), the different deployment results in significantly different drift patterns for the sensors over the 7-days mission (Figure 5.9). Consequently, it can be seen from Fig. 5.10 that the coverage provided by the sensors placed by optimizing the cumulative coverage function is much improved over time, despite the initial coverage being higher for the network placed by the static optimization. The maximum coverage provided by the sensor network placed by optimal control peaks at approximately 6 days, and displays a 43% decrease in coverage from initial deployment to the end of the mission. Whereas, the sensor network placed according to the static optimization peaks initially, but then decreases by 86% over the 7-days period. Finally, the cumulative coverage (Fig. 5.10)



(a)

(b)

Figure 5.9: The drift trajectories of $n = 10$ sensors for $k = 3$ detections placed according to the (a) the static optimal optimization and the (b) optimal deployment with respect to the LTI dynamics within an arbitrary reference frame.

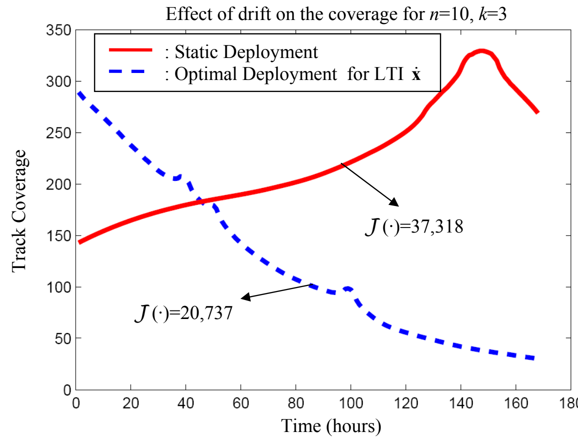


Figure 5.10: Coverage deterioration for sensors placed according to the optimal and static deployments.

reveals a 85% increase as a result of the initial deployment accounting for the drift dynamics.

5.2 Robustness Analysis of the Moving Sensor Network

A sensor network placed in an ocean environment is subject to various sources of uncertainty that cannot be accounted for *a priori*. The first source of uncertainty is due to the deployment of the sonobuoys, which is typically carried out by an aircraft. As a result, the actual deployment of the sonobuoys may include human-error and disturbances due to inclement weather conditions experienced by the sonobuoys as they travel in the air from the aircraft to the ocean surface. For example, strong wind can result in a systematic (bias) error in the initial deployment, such that all sensor locations deviate from the intended locations by the same error vector. Once the sensors are deployed, the GPS location device on each sonobuoy may relay a slightly different position of the sonobuoy than the actual position. Then, over the course of the mission, sonobuoys are likely to experience oceanic currents that differ from those estimated by the NN from CODAR measurements (Section 5.1.1), thereby affecting the entire trajectory. Also, the range of each sensor field-of-view is known to be subject to errors due to the classification estimate from the BN [1]. These sources of uncertainty are considered through Monte Carlo robustness analysis in order to estimate the expected performance reduction that the actual sensor system will experience when deployed by the methods developed in this section.

The optimal solution obtained for zero errors (ideal conditions) is referred to as the nominal trajectory. The above sources of uncertainty on the initial positions, currents, and sensor ranges are represented by the following random variables

$$\begin{aligned} \hat{\mathbf{x}}_0 &= \mathbf{x}_0 + \mathbf{d}_\epsilon + \mathbf{n}_\epsilon, \quad \hat{\mathbf{x}}_0 \in \mathbb{R}^{2n} && (5.37) \\ \hat{\boldsymbol{\upsilon}}(\hat{\mathbf{x}}(t), t) &= \boldsymbol{\upsilon}(\hat{\mathbf{x}}(t), t) + \boldsymbol{\upsilon}_\epsilon, \quad \hat{\boldsymbol{\upsilon}} \in \mathbb{R}^{2n} && (5.38) \\ \hat{\mathbf{r}}(\hat{\mathbf{x}}(t)) &= \mathbf{r}(\hat{\mathbf{x}}(t)) + \mathbf{r}_\epsilon, \quad \hat{\mathbf{r}} \in \mathbb{R}^{n} && (5.39) \end{aligned}$$

As the state $\mathbf{x}(t)$ cannot be propagated with certainty, it also is considered as a random variable. The errors expressed in (5.37)-(5.38) are propagated through time by integrating the dynamic equation (5.10) with respect to the random initial state variable $\hat{\mathbf{x}}_0$ and $\hat{\boldsymbol{\upsilon}}$ simultaneously to get the expected trajectory

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}_0 + \int_{t_0}^{t_f} \boldsymbol{\upsilon}(\hat{\mathbf{x}}(t), t) dt \qquad (5.40)$$

It is assumed here that each error is independently and identically distributed (i.i.d.) from a Gaussian (normal) distribution, and the $x-$ and $y-$directions of the sensor position and current are independent. Then, the uncertainty variables $\mathbf{d}_\epsilon$, $\mathbf{n}_\epsilon$, $\boldsymbol{\upsilon}_\epsilon$, and $\mathbf{r}_\epsilon$ each have an associated mean expected value and covariance matrix, given in the form *(mean, covariance)*,

$$(\mathbb{E}[\mathbf{d}_\epsilon] \;=\; m_d \cdot [1, ..., 1]^T = \mathbf{m}_d, \quad \mathbb{E}[\mathbf{d}_\epsilon \mathbf{d}_\epsilon^T] = p_d \mathbf{I}_{(2n\times 2n)} = \mathbf{P}_d) \tag{5.41}$$

$$(\mathbb{E}[\mathbf{n}_\epsilon] \;=\; m_n \cdot [1, ..., 1]^T = \mathbf{m}_n, \quad \mathbb{E}[\mathbf{n}_\epsilon \mathbf{n}_\epsilon^T] = p_n \mathbf{I}_{(2n\times 2n)} = \mathbf{P}_n) \tag{5.42}$$

$$(\mathbb{E}[\boldsymbol{\upsilon}_\epsilon] \;=\; m_\upsilon \cdot [1, ..., 1]^T = \mathbf{m}_\upsilon, \quad \mathbb{E}[\boldsymbol{\upsilon}_\epsilon \boldsymbol{\upsilon}_\epsilon^T] = p_\upsilon \mathbf{I}_{(2n\times 2n)} = \mathbf{P}_\upsilon) \tag{5.43}$$

$$(\mathbb{E}[\mathbf{r}_\epsilon] \;=\; m_r \cdot [1, ..., 1]^T = \mathbf{m}_r, \quad \mathbb{E}[\mathbf{r}_\epsilon \mathbf{r}_\epsilon^T] = p_r \mathbf{I}_{(n\times n)} = \mathbf{P}_r) \tag{5.44}$$

where the dimensions of the mean vectors and covariance matrices are given as $\mathbf{m}_d, \mathbf{m}_n, \mathbf{m}_\upsilon \in \mathbb{R}^{2n\times 1}$, $\mathbf{m}_r \in \mathbb{R}^{n\times 1}$ $\mathbf{P}_d, \mathbf{P}_n, \mathbf{P}_\upsilon \in \mathbb{R}^{2n\times 2n}$, $\mathbf{P}_r \in \mathbb{R}^{n\times n}$, and $\mathbf{I}$ is the identity matrix with noted dimensions.

The specific values of the scalar mean and covariance values in (5.41)-(5.44) are listed in Table 5.4. The measurement error on the initial positions, $\mathbf{n}_\epsilon$, is relatively small (on the order of a couple meters) thanks to the accuracy of GPS devices. On the other hand, the deployment error, $\mathbf{d}_\epsilon$, is considerably larger. Two different deployment models are investigated to accommodate for different scenarios that may occur in practice, such as calm weather with no wind versus strong winds. The first case includes $m_d = 0$ and a $3-\sigma$ initial position variation of 5 Km. The second case introduces a bias on the mean of $3-\sigma = 5$ Km, which translates a sensor network in $\mathcal{A}$ by the same distance. These cases are investigated separately in order to provide a clear picture of the sensor network performance sensitivity in the face of uncertainty. As for the expected uncertainty in the currents, it has been noted in [86] that the RMS differences between the CODAR and in situ instruments to be between 0.18 and 0.54 Km/hr, which is approximately a 10% difference. We implement a 15% $3-\sigma$ variation of the currents to account for both measurement error and unexpected severe weather. Finally, the $3-\sigma$ range error is 0.224 Km, as described in [1].

A sensor network that provides satisfactory coverage over time in the presence of system uncertainty is said to be robust. The probability of satisfactory $\mathcal{J}$ coverage

Table 5.4: List of the system Gaussian errors included in the Monte Carlo simulation

Variables	$\mathbf{3-\sigma}$ **Variation**	**Mean**
Initial positions:		
deployment error, $\mathbf{d}_\epsilon$ [Km]	± 5	$0 \ / \ 3-\sigma = \pm 5$
measurement error, $\mathbf{n}_\epsilon$ [Km]	± 0.005	0
Range, $\mathbf{r}_\epsilon$ [Km]	± 0.224	0
Ocean Current, $\boldsymbol{v}_\epsilon$ [Km/hr]	$\pm 15\%$	0

is defined as,

$$p_{\mathcal{J}} = Pr[\{\hat{\mathcal{J}}/\mathcal{J} \geq \varsigma\}], \quad 0 \leq \varsigma \leq 1 \tag{5.45}$$

where ς is the smallest allowable decrease in the objective function $\mathcal{J}$ (a deterministic quantity). The $p_{\mathcal{J}}$ should be close to 1, or some other chosen value within the the interval $[0, 1]$, as this represents a system that is (almost) entirely satisfactory. Otherwise, the system is not robust. The value $p_{\mathcal{J}}$ is unknown analytically, and may be estimated by Monte Carlo simulation. Let the expected value of the objective function in terms of (5.40) and (5.39),

$$\begin{aligned} \hat{\mathcal{J}} &= \mathcal{J}(\hat{\mathbf{x}}(t), \hat{\mathbf{r}}(\hat{\mathbf{x}}(t)), t) \qquad (5.46) \\ &= \int_{t_0}^{t_f} \left[W_{\mathcal{T}} \cdot \mathcal{T}_{\mathcal{A}}^k(\hat{\mathbf{x}}(t), \hat{\mathbf{r}}(\hat{\mathbf{x}}(t))) + W_A \cdot A_C(\hat{\mathbf{x}}(t), \hat{\mathbf{r}}(\hat{\mathbf{x}}(t))) \right] dt \end{aligned}$$

be evaluated N times. Then, the estimate of the probability of sufficient performance (5.45) becomes increasingly precise as N becomes large, i.e.,

$$Pr[\{\hat{\mathcal{J}}/\mathcal{J} \geq \varsigma\}] = \lim_{N \to \infty} \frac{M(\hat{\mathcal{J}}/\mathcal{J} \geq \varsigma)}{N} \tag{5.47}$$

where $M(\cdot)$ is the number of cases of sufficient coverage, that is, $\hat{\mathcal{J}}/\mathcal{J} \geq \varsigma$. For $N < \infty$, the probability of sufficient performance from Monte Carlo simulation is an estimate, denoted $\hat{p}_{\mathcal{J}}$. Then, the estimate $\hat{p}_{\mathcal{J}}$ is related to the true underlying probability of $p_{\mathcal{J}}$ through confidence intervals [87],

$$Pr[L < p_{\mathcal{J}} < U] = 1 - \alpha \tag{5.48}$$

where (L, U) is the interval estimate, $1 - \alpha$ is the confidence coefficient, and $\hat{p}_{\mathcal{J}}$ lies within (L, U) with $100 \cdot (1 - \alpha)\%$ confidence. For example, a 95% confidence interval implies that 95% of N samples lies within (L, U)

$$Pr[\hat{\mu}_{\mathcal{J}} - 1.96\frac{\hat{\sigma}_{\mathcal{J}}}{\sqrt{N}} < p_{\mathcal{J}} < \hat{\mu}_{\mathcal{J}} + 1.96\frac{\hat{\sigma}_{\mathcal{J}}}{\sqrt{N}}] = 0.95 \tag{5.49}$$

where $\hat{\mu}_{\mathcal{J}}$ and $\hat{\sigma}_{\mathcal{J}}$ are the sample mean and standard deviation of the objective function, respectively. It follows that the narrower is the interval, the more precise is the estimate $\hat{p}_{\mathcal{J}}$. Therefore, both $\hat{p}_{\mathcal{J}}$ and (L, U) can be used for assessing the robustness of the nominal system acquired through track coverage optimization of the moving sensor network.

5.3 Application to Optimal Deployment of a Sonobuoy Sensor Network

The methodology developed in this chapter is implemented such that the coverage objective function of a moving sensor network is maximized with respect to an area of interest over a period of time. This problem is relevant to a sensor network of sonobuoys that are floating and drifting in the ocean due to the surface current velocities that are employed for detecting moving targets in a region of interest. The system dynamics are approximated by a nonlinear, time-varying NN approximator function obtained from CODAR data (Section 5.1.1). Also, the sensor ranges are approximated by a nonlinear, position-dependent BN (Section 5.1.2). Then, the sensor deployment problem is formulated as an NLP (5.18)-(5.22), which is solved by the SQP algorithm [67, 68] using Runge-Kutte (4,5) numerical integration [88].

In Section 5.3.1, the track coverage provided by networks deployed using the SQP algorithm (and numerical integration) is compared to those obtained by the static solution from Chapter 4.1, i.e., solving the NLP (4.1)-(4.4) that we will refer to as the *static SQP deployment*, as well as the *random* and *grid deployment* strategies, all of which are discussed in Chapter 4. In Section 5.3.1, the nominal (i.e., ideal) solution of the NLP (5.18)-(5.22) from the SQP algorithm is shown to improve track coverage by up to two orders of magnitude, compared to the former techniques, for networks of various size and k required detections. Then, in Section 5.3.2, the nominal solutions obtained in Section 5.3.1 are tested for robustness, as discussed in Section 5.2, through Monte Carlo simulation. Uncertainty is introduced into the nominal initial positions, and the errors on the initial positions, current velocity, and sensor ranges are propagated through time to obtain the trajectories of each sensor subject to errors. The cases of no bias and bias uncertainty included in the initial positions are investigated separately to illustrate the sensitivity of the nominal initial position solution. The results show that the decrease in sensor network performance in the face of uncertainty is less than 6% for both cases. Although the bias error results in larger initial and final position envelopes, the resulting performance for the two models are very similar. Also in both cases performance is much improved over the static SQP, grid, and random deployments.

5.3.1 MultiObjective Optimization

The effectiveness of deploying sensors using the solution of the NLP (5.18)-(5.22) is demonstrated for four networks listed in Table 5.5. The dimensions of $\mathcal{A}$ are $L_1 \times L_2 = 90 \times 82.5$ Km^2, while the mission period-of-time is 4 days. The respective weights of the objective function W_C and $W_{\mathcal{T}}$ are given as

$$W_C = \frac{\alpha_C}{A_C^{\max}} = \frac{15}{7,426} = 0.0020 \tag{5.50}$$

$$W_{\mathcal{T}} = \frac{\alpha_{\mathcal{T}}}{\mathcal{T}_{\mathcal{A}}^{\max}} = \frac{2.5}{1,097} = 0.0023 \tag{5.51}$$

where track-coverage is given slightly more weight than area coverage. Then, in Table 5.5, the performance measures of track coverage, area coverage, and the objective function are calculated for the trajectories from the sensors placed according to the SQP solution $\mathbf{x}_0^*$ to the NLP and compared to the static SQP, random, and grid deployments. As a result of significantly different drift patterns experienced by the four sensor networks, sensors placed at the SQP solution achieve significantly higher performance than the other three deployment methods. In fact, the same network deployed via the SQP solution for $n = 15$ and $k = 3$ can provide significantly higher track coverage (+519%) and area coverage (+553%) over the static SQP deployment. A comparison of the coverage over time is illustrated in Fig. 5.11 for $n = 20$ and $k = 3$, whose deployment positions and trajectories are shown in Fig. 5.12. The coverage provided by the the SQP solution is consistent over time, as evident from Fig. 5.11, compared to the other deployment methods whose coverage decrease significantly as the mission progresses. Even though the initial coverage is higher for the static SQP deployment, the sensors quickly drift outside of $\mathcal{A}$ and significantly overlap (Fig. 5.12(b)), providing much less coverage.

5.3.2 Robustness Analysis via Monte Carlo Simulations

A Monte Carlo (MC) simulation is utilized to statistically assess the robustness of the nominal SQP solutions discussed in Section 5.3.1. The Gaussian errors on the initial positions, ocean-current velocities, and ranges, with means and variances listed in

Table 5.5: Comparison between the different deployment methods of sensor networks for coverage over a 4-day mission period of time, where ($\cdot$) represents the (SQP % Improvement) over each deployment method.

$\mathbf{n}, \mathbf{k}$:	**Performance Measures**	**SQP**	**Static**	**Grid**	**Random**
$10, 2$	Track Coverage	1.671×10^4	8,823	4,866	4,951
	Area Coverage	3.091×10^4	1.119×10^4	1.831×10^4	1.843×10^4
	Objective Function	**100.5**	**42.73** (135%)	**48.09** (109%)	**48.51** (107%)
$15, 3$	Track Coverage	1.630×10^4	3,145	1,763	2,057
	Area Coverage	5.368×10^4	8,223	2.830×10^4	3.094×10^4
	Objective Function	**145.6**	**23.78** (512%)	**61.17** (138%)	**67.19** (117%)
$20, 3$	Track Coverage	2.245×10^4	1.326×10^4	2,800	6,281
	Area Coverage	7.174×10^4	2.965×10^4	3.607×10^4	3.471×10^4
	Objective Function	**196.1**	**90.14** (118%)	**79.25** (147%)	**84.43** (132%)
$25, 4$	Track Coverage	1.860×10^4	1.358×10^4	932.3	2,648
	Area Coverage	8.892×10^4	3.211×10^4	4.163×10^4	4.494×10^4
	Objective Function	**222.0**	**95.97** (132%)	**86.21** (158%)	**96.82** (129%)

Table 5.4 are simulated for $M = 5,000$ MC trials for two different cases: when initial position uncertainty has no bias (i.e., $m_d = 0$) or includes bias (i.e., $m_d \sim \mathcal{N}(0, 2.78$ from $(3-\sigma) = 5$ in Table 5.4). Comparing these two different cases in affect provides a clear picture to the actual sensitivity of the performance to the uncertainty.

First, we will investigate the system robustness for the no bias case. The three nominal performance measures (track coverage, area coverage and objective function) are compared to the mean of the performance measures obtained from the MC simulation in Table 5.6. The largest decrease of the nominal performance occurs for $n = 15$ and $k = 3$, although this decrease is approximately 6%. Compared to the performance of the other deployment methods in Table 5.5, the decrease in performance is 84%, 58%, and 54% for the static SQP, grid, and random deployments, respectively.

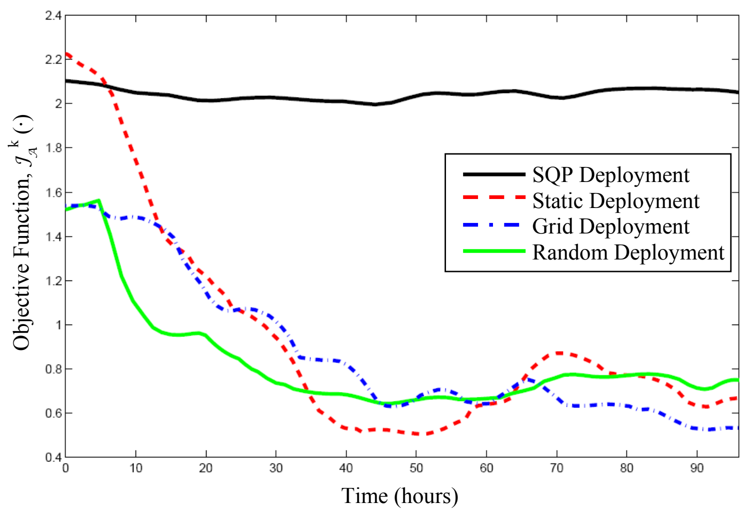


Figure 5.11: Performance of the four deployment methods over a period of 4-day mission for $n = 20$ and $k = 3$.

In addition, the relatively narrow 95% confidence level of the objective function for the three examples indicates that the mean cost has been rather precisely measured. Even though sensors do not follow the nominal paths (from the trajectory envelopes in Fig. 5.14(d) for 5000 MC trials of two sensors), sensors will provide coverage over other areas of $\mathcal{A}$, as evident by the light shading in the parameter space plots 5.14(a)-5.14(c). Therefore, performance of the sensor network over time is consistent (Fig. 5.13), although each trial may provide coverage over different areas of $\mathcal{A}$.

The MC robustness analysis is also performed for sensors deployed with a non-zero bias error as depicted in Fig. 5.15(c). The simulations including uncertainty (Figs. 5.15(b)-5.15(c)) lead to both overlapping fields-of-view and to be partially outside of $\mathcal{A}$. However, the performance for each of these systems does not decrease significantly over time and is very close to the nominal performance, as illustrated in Fig. 5.15(d). The results for the $M = 5000$ MC trials that include the bias error in the initial positions are listed in Table 5.7 for three different sensor networks. Of the three examples investigated, only one of the listed results actually provide less expected performance compared to the no bias example, i.e., $\mathcal{J}_{\mathcal{A}}^{k} = 186.2$ compared to $\mathcal{J}_{\mathcal{A}}^{k} =$

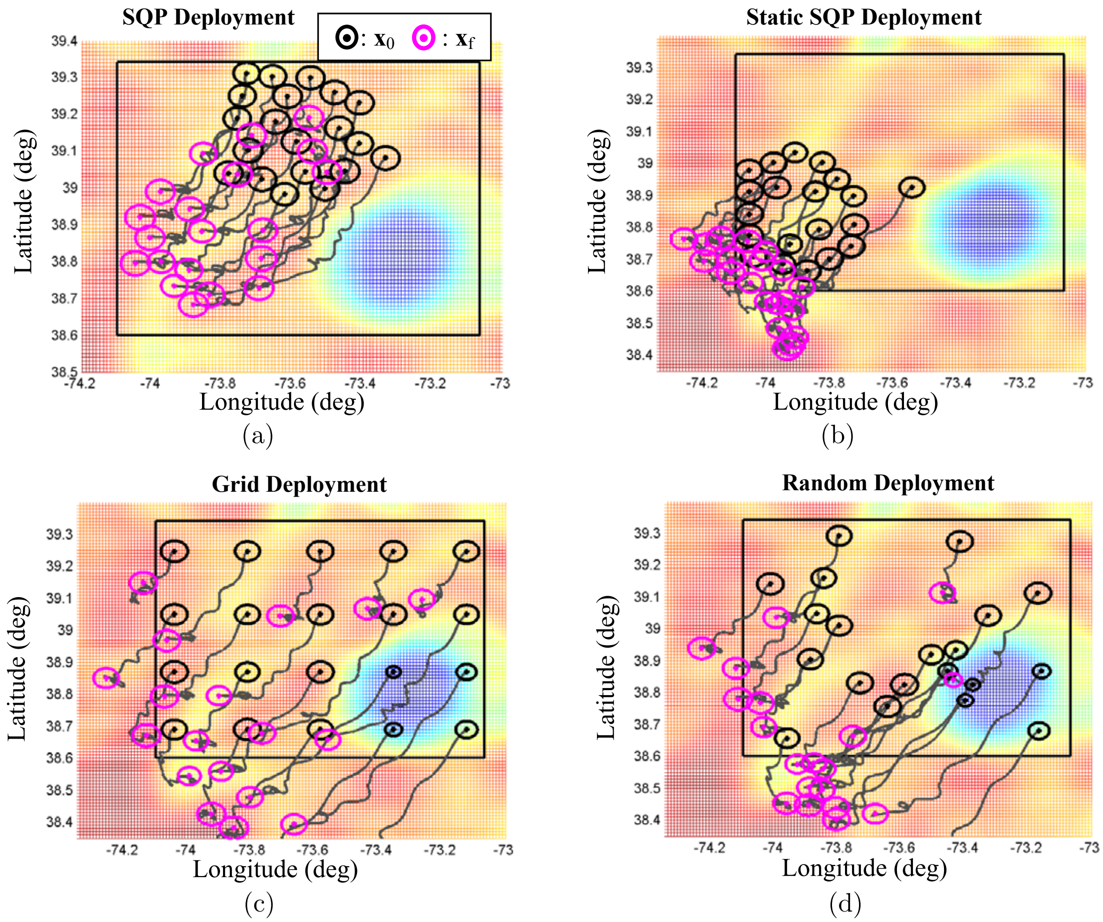


Figure 5.12: The trajectories of $n = 20$ sensors over the 4-day missions according to the (a) SQP, (b) static SQP, (c) grid, and (d) random deployments.

Table 5.6: Performance results with *no bias* error, where (·) represents the SQP difference (%) with the performance mean

$\mathbf{n}, \mathbf{k}$	**Performance Measure**	**SQP**	**Performance mean**	**95% Confidence Level, no bias**
10,2	Track Coverage	1.671×10^4	1.508×10^4	$(1.507, 1.510)\times10^4$
	Area Coverage	3.091×10^4	3.063×10^4	$(3.061, 3.065)\times10^4$
	Objective Function	**100.5**	**96.27** (4.21%)	$(\mathbf{96.21}, \mathbf{96.33})$
15,3	Track Coverage	1.630×10^4	1.468×10^4	$(1.467, 1.470)\times10^4$
	Area Coverage	5.368×10^4	5.116×10^4	$(5.112, 5.119)\times10^4$
	Objective Function	**145.6**	**136.8** (6.04%)	$(\mathbf{136.7}, \mathbf{136.9})$
20,3	Track Coverage	2.245×10^4	2.046×10^4	$(2.044, 2.047)\times10^4$
	Area Coverage	7.174×10^4	6.907×10^4	$(6.903, 6.911)\times10^4$
	Objective Function	**196.1**	**186.2** (5.05%)	$(\mathbf{186.1}, \mathbf{186.3})$

185.9, respectively. However, this difference is less than 0.1% between the two cases, and less than a 5.2% decrease from the nominal performance. Another difference between these two cases can be seen in the slightly different confidence intervals. For the example with $n = 10$, the difference of the lower and upper confidence interval for the bias case is 0.05 units larger than the no bias case. However, this difference is practically negligible, considering the order of the objective function (10^2). In fact, the other two examples provide the same interval size between the bias and no bias cases. Although the performance of these two cases are nearly identical, it is obvious from Fig. 5.14(d) that the trajectory envelopes for the bias case are notably larger compared to the no bias case. The distribution of the initial and final position envelopes are also illustrated in Figs. 5.16-5.17 as contour plots. This confirms that the initial position envelope for the bias case provides a wider distribution than the no bias envelope, which are characterized by distinct peaks (much darker shading in the middle). Therefore, the nominal solution is shown to be both robust and insensitive to uncertainty.

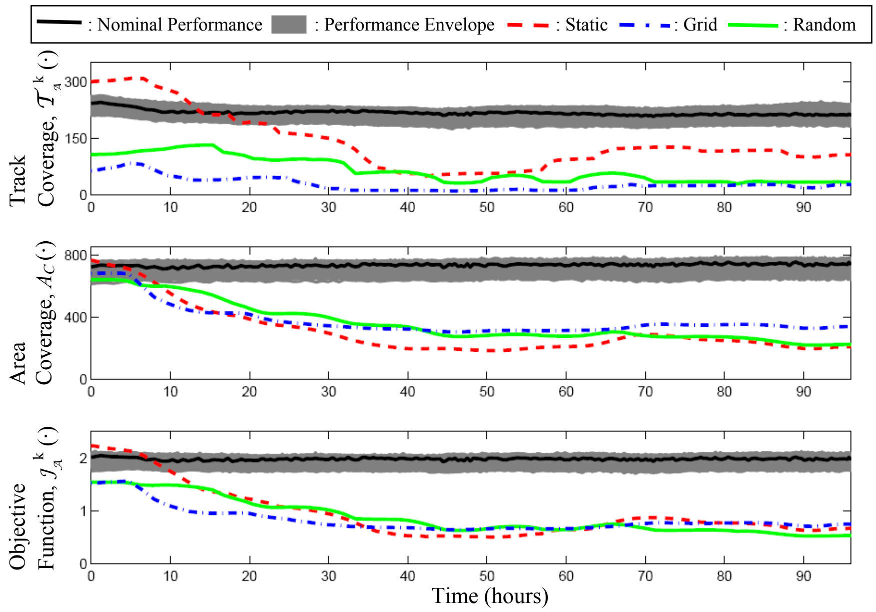


Figure 5.13: The performance envelope calculated from the actual trajectories of $n = 20$ sensors and $k = 3$ detections with propagated error and no bias included in the initial position uncertainty.

5.4 Chapter Summary

The methodology developed in this chapter is implemented to optimize track coverage of a moving sensor network with respect to an ROI over a fixed period of time. This problem is relevant to a moving sensor network of sonobuoys employed to detect moving targets in an ROI. In Section 5.1.1, the ocean current-velocities are approximated by an NLTV NN approximator obtained from measured CODAR data. Also, the expected performance, i.e., the range of the field-of-view, of each sensor is approximated by a BN that is a function of the sensor location in $\mathcal{A}$ (5.1.2). These closed-form models are incorporated implicitly into the weighted sum objective function of the cumulative track and area coverage (Section 5.1.3). This objective function is formulated as an NLP and optimized subject to the drift dynamics with respect to the initial positions of the sensors.

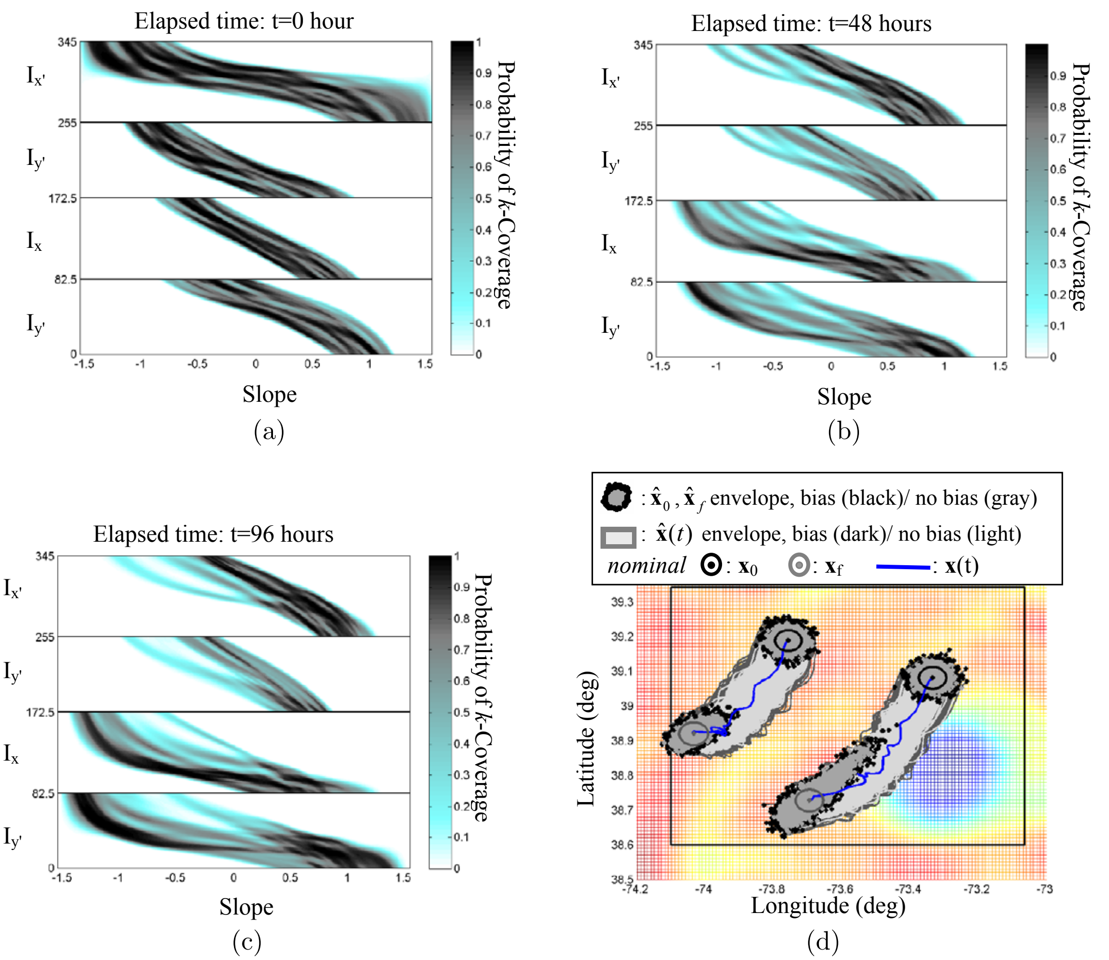


Figure 5.14: (a)-(c) the track-coverage in parameter space over time, (a) initial, (b) midpoint, and (c) final times in the mission for $n = 20$ and $k = 3$ (no bias). (d) Two examples (from $n = 20$) of the envelopes of the initial and final positions and trajectories for $M = 5000$ MC trials with the propagated error listed in Table 5.6.

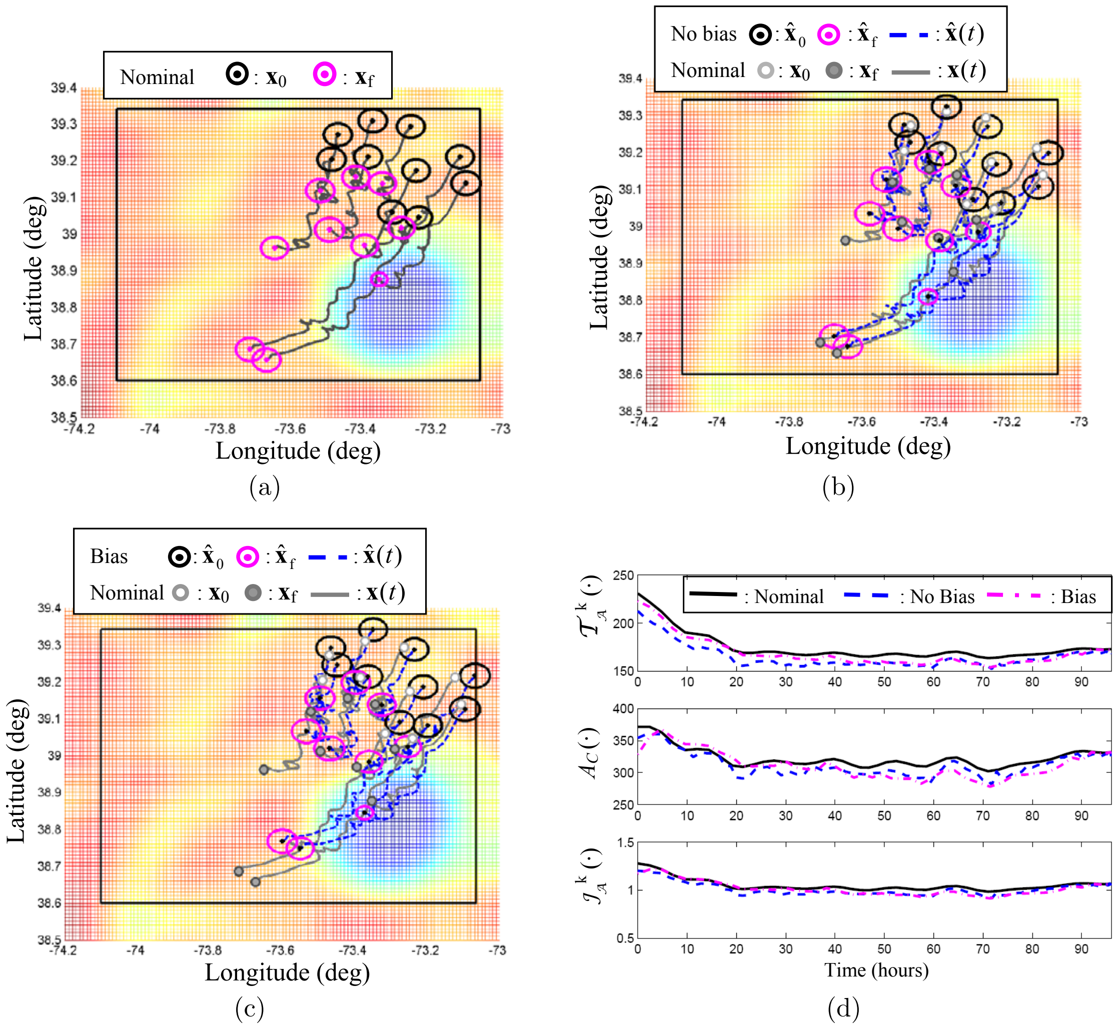
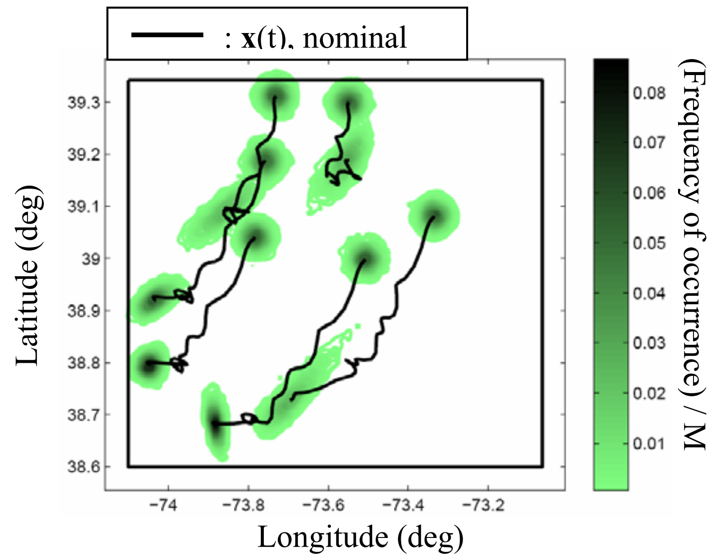
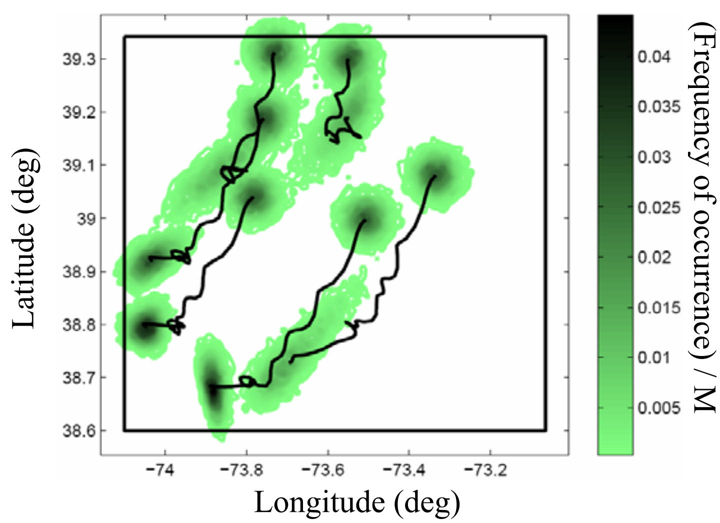


Figure 5.15: (a) The nominal trajectories for $n = 10$ and $k = 2$ are compared to an example of sensors placed with (b) no bias error and (c) bias error, where the performance measure for each sensor network is illustrated in (d).



(a)



(b)

Figure 5.16: The contour plots illustrates the distribution of the initial and final position envelopes for five sensors (two of which are shown in Fig. 5.14(d)) for n=20 k=3 example with (a) no bias, (b) bias error included in the initial positions.

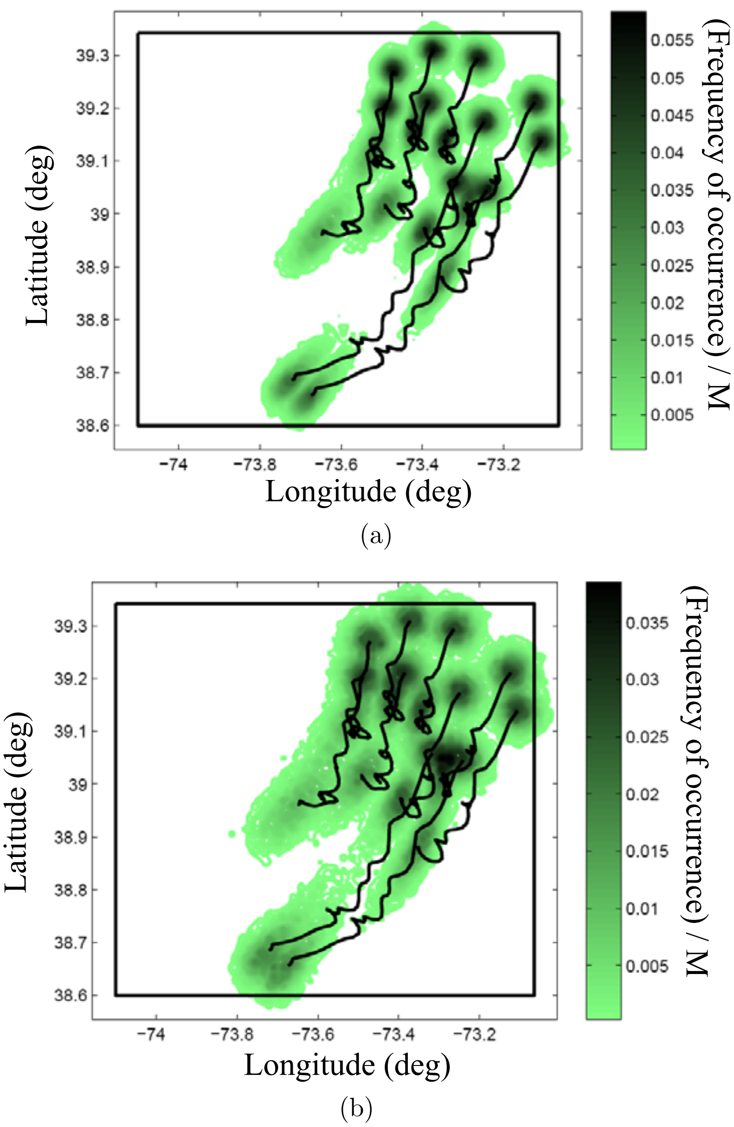


Figure 5.17: The contour plots illustrates the distribution of the initial and final position envelopes for $n = 10$ sensors (with nominal positions in Fig. 5.15(a)) for (a) no bias, (b) bias error included in the initial positions.

Table 5.7: Performance results with *bias* error, where ($\cdot$) represents the SQP difference (%) with the performance mean

n, k	**Performance Measure**	**SQP**	**Performance mean**	**95% Confidence Level, bias**
10,2	Track Coverage	1.671×10^4	1.509×10^4	$(1.507, 1.511) \times 10^4$
	Area Coverage	3.091×10^4	3.064×10^4	$(3.061, 3.066) \times 10^4$
	Objective Function	**100.5**	**96.29** (4.19%)	$(\mathbf{96.21}, \mathbf{96.38})$
15,3	Track Coverage	1.630×10^4	1.472×10^4	$(1.471, 1.474) \times 10^4$
	Area Coverage	5.368×10^4	5.110×10^4	$(5.106, 5.114) \times 10^4$
	Objective Function	**145.6**	**136.8** (6.04%)	$(\mathbf{136.7}, \mathbf{136.9})$
20,3	Track Coverage	2.245×10^4	2.048×10^4	$(2.046, 2.050) \times 10^4$
	Area Coverage	7.174×10^4	6.892×10^4	$(6.888, 6.896) \times 10^4$
	Objective Function	**196.1**	**185.9** (5.20%)	$(\mathbf{185.8}, \mathbf{186.0})$

An example of the optimal solution for LTI system dynamics is shown in Sections 5.1.3 for the $k = 3$ track coverage of a network with $n = 10$ sensors. These results show a considerable improvement to track-coverage over a period of time, although the overlapping sensors fields-of-view provide poor area coverage. Then, the final results, which incorporate the NLTV equations of motion, position-dependent range, and the weighted sum objective function results in significant improvement over various other deployment methods proposed proposed in the literature (i.e., grid and random) and the solution to the static track coverage problem that does not include the ocean current-velocities (from Chapter 4). As in most applications, especially when deployed in the ocean, a sensor network is subject to random disturbances, such as unforseen and uncontrollable variations in both the sensor location and field-of-view due to any number sources of error. The robustness of the nominal solution of the initial positions that maximize cumulative track coverage is verified through Monte Carlo Simulation for propagated uncertainty incorporated in the the initial deployment, trajectories, and range.

Chapter 6

Optimal Control of a Mobile Underwater Sensor Network

A group of mobile underwater vehicles, each with an onboard acoustic sensor, can reach a higher accuracy, and enable detection of low signature targets by combining measurements from multiple sensors [89]. However, cooperative detection of a moving target becomes increasingly difficult for a mobile sensor network as the path planning strategy must consider a group of independently moving vehicles for the unified purpose of coverage. In this chapter, a method is developed for computing the optimal trajectories of a group of underwater vehicles, such as gliders, in order to cooperatively provide optimal performance of a mission with respect to the quality of service sensor performance metrics introduced in Chapter 3. For this purpose, the interaction between the dynamical system comprised of the underwater gliders with onboard acoustic sensors that have omnidirectional fields-of-view is modeled in Section 6.2.2. The oceanic surface currents are modeled by ordinary differential equations that are approximated by a NN (Section 5.1.1). The influence of the environmental variables on the sensors fields-of-view can be modeled as a BN, as discussed in Section 5.1.2.

The approaches to cooperative path planning through control can be categorized into two classes: reactive and pregenerative. Much of the research in cooperative control for underwater vehicles is reactive and determines the control inputs in real-time to coordinate the autonomous sensors for objectives such as surveys, coverage and sampling [90–93]. The shortcoming of reactive control methods is that if the sensors move through a highly nonlinear, time-varying environment, the vehicles do not efficiently utilize the natural environmental dynamics as only the flow field velocity data at each specific sensor location is incorporated into the path planning. Potentially, a more favorable environment located elsewhere in the region is not recognized.

The pregenerative approach to cooperative vehicle control is carried out prior to the mission. Thus, in a oceanic environment, it must include a forecast model of the environmental dynamics in order to account for their influence on the trajectories of the gliders. An example of pregenerative control was presented in [94] to determine the trajectory of a single underwater vehicle traveling from specified initial and final positions, in the shortest time using minimal energy. By using a forecast model of the currents, the ideal path of the glider is one that provides the most velocity-assistance, thereby conserving energy and possibly reducing the travel-time. Although the glider may consume slightly more energy at times in order to reach the currents that will provide the most assistance, the total energy is minimized. The limited amount of pregenerative control methods that include a forecast environmental model for optimal control is limited to single vehicle dynamics and objectives that typically include energy and mission time, such as path planning for obstacle avoidance [94–96]. For example, the numerical solution obtained using Nonlinear Trajectory Generation seeks to find a tradeoff between the shortest path using minimal energy consumption for a single glider [94]. In [96], the energy-optimal trajectories for individual underwater gliders are computed using optimal control. While the author proposes the optimal control strategy for multiple vehicles, a non-cooperative cost function governs the problem, i.e., minimum time and energy trajectory that avoids obstacles. This concept of controlling a group of vehicles through individual solutions is the basis of an algorithm proposed in Chapter 6. Ideally, it is advantageous to have a path planning approach that is a mixture of pregenerative/reactive, where the problem is formulated such that the prediction of the temporal evolution of the oceanographic environment is acquired by a forecast model that is updated as the vehicle navigates the environment [95].

This chapter addresses computing optimal trajectories for controllable underwater gliders that are deployed to detect moving targets in an oceanic region of interest by means of onboard omnidirectional acoustic sensors. This constitutes a new optimal control problem that integrates geometric sensor objectives such as track coverage with cooperative path planning of a mobile sensor network subject to time-varying environmental dynamics. A parametric study revels the Pareto-front of the multi-objective cooperative control problem. The weighted sum cost function relates the competing objectives of minimizing energy while maximizing track coverage. The optimal control problem presented in this chapter can be viewed as a new problem in dynamic computational geometry pertaining to the geometric transversals of a moving family of geometric objects representing the sensors fields-of-view. In order to solve for the trajectory optimization problem through optimal control, it is important to have a closed form representation of the highly nonlinear oceanic currents. In Section 5.1.1 it was shown that the nonlinear, time-varying ocean current velocity are approximated by a neural network, which are included in into the glider systems dynamics in Section 6.2.1. Then, the multiobjective problem of maximizing track coverage while minimizing energy consumption is formulated as a multi-dimensional, constrained optimal control problem (Section 6.2.3). The quality-of-service measure of the mobile sensor network is formulated using the track coverage function represented in Section 3.1 and the energy consumption function in Section 3.3. Then, the continuous-time optimal control problem is approximated by a discrete parametric control problem in order to obtain a numerical solution that is locally and approximately optimal (Section 6.3). Finally, the parametric study in Section 6.4 show how the optimal trajectories differ based on different weights chosen for the competing objectives.

6.1 Background on Optimal Control

The deployment of a mobile underwater sensor network investigated in this research is formulated as follows. Determine the optimal state $\mathbf{x}^*(t) \in \mathbb{R}^n$ and control policy $\mathbf{u}^*(t) \in \mathbb{R}^m$ that maximizes the Bolza integral cost function,

$$J = \phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{6.1}$$

subject to a non-autonomous nonlinear system dynamic equation,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad t \in [t_0, t_f] \tag{6.2}$$

and a set of equality and inequality constraints on the state and control,

$$\mathbf{\Phi}(\mathbf{x}(t_0), t_0, t_f) \;=\; \mathbf{0}, \quad \mathbf{\Phi} \in \mathbb{R}^q \tag{6.3}$$

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \;\leq\; \mathbf{0}, \quad t \in [t_0, t_f], \quad \mathbf{C} \in \mathbb{R}^c \tag{6.4}$$

respectively.

The above optimal control problem (6.1)-(6.4) can be approached by the calculus of variations using Pontryagin's maximum principle [97], or by the dynamic programming approach, using the principle of optimality [98], applied to optimal control problems [99]. The necessary conditions for optimality are expressed as Euler-Lagrange equations, as derived from the calculus of variations [99]. However, as there typically is no closed form analytic solution to these equations when the system dynamics are nonlinear, it is necessary for computational tractability to solve the optimal control problem numerically. Numerical methods for solving optimal control problems can be categorized as either direct or indirect methods. In an indirect method, the first-order optimality conditions are derived using the minimum principle of Pontryagin [100]. These necessary conditions lead to a Hamiltonian boundary value problem (HBVP) which is then solved to determine candidate optimal trajectories, or extremal trajectories. On the other hand, a direct method discretizes the continuous-time problem about collocation points and then transcribes it into a finite-dimensional nonlinear program (NLP). The NLP is then solved using an appropriate optimization method, such as sequential quadratic programming (SQP) [67], [68]. Compared to indirect methods, direct methods are typically more popular as it is typically easier to find a solution to a NLP than a solution to the associated HBVP. Several direct methods of solution are discussed in Section 6.3.

6.2 Methodology

The cooperative trajectory optimization problem of a group of independent underwater gliders in a dynamic ocean environment consists of deploying and controlling a group of gliders, each with an onboard acoustical sensor, in order to maximize a cooperative measure of sensing performance. Underwater gliders are an example of vehicles for large-scale ocean operations, such as surveys and surveillance, in the ocean due to their low cost, simple and efficient design, and their capability to operate autonomously. They are winged, buoyancy-driven submersible vehicles that have high endurance and are strongly influenced by the currents. An example of an underwater glider is Eyak 02 from Alaska Native Technologies and is shown in Fig. 6.1 [6]. For these reasons, the glider model should take advantage of ocean current measurements and forecasts, such that the gliders can be steered efficiently and with minimal energy consumption. The drift dynamics induced by an oceanic environment are accounted for by utilizing oceanographic models and measurements from the ocean [4], which produce a known forcing vector field in the glider equations of motion [94]. As the gliders maneuver through the environment, the onboard acoustic sensors are continuously taking measurements within their respective fields-of-view. Representative models for both the gliders motion in the dynamic ocean environment and the expected field-of-view of each sensor are necessary for optimal path planning of a group of independent underwater gliders for the cooperative purpose of providing track coverage over a period of time. In this chapter, we show that an optimal control strategy can be used to optimize the sensor network performance, subject to a dynamic environment and mission constraints. The optimal control formulation relies on the assumptions from Chapter 1.1.

Figure 6.1: An acoustic underwater glider from Alaska Native Technologies (Eyak 02) [6].

6.2.1 Equations of Motion of the Sensor Network

In order to optimize the objective function (6.1) subject to system dynamics, a closed-form model of the dynamics (6.2) is derived from the kinematic models of n underwater gliders [94],

$$\dot{\mathbf{x}}_i(t) = \begin{bmatrix} v_x(x_i, y_i, t) \\ v_y(x_i, y_i, t) \end{bmatrix} + V_i(t) \begin{bmatrix} \cos\theta_i(t) \\ \sin\theta_i(t) \end{bmatrix}, \quad \forall\, i = 1, ..., n \tag{6.5}$$

where v_x and v_y are the time-dependent components of the ocean current velocity in the x- and y- direction, and V_i and θ_i are the control speed and direction for the i^{th}

sensor. Then, the control vector for the i^{th} sensor is defined as,

$$\mathbf{u}_i(t) = V_i(t) \cdot [\cos\theta_i(t) \quad \sin\theta_i(t)]^T \tag{6.6}$$

and the control vector for the entire sensor network is denoted as $\mathbf{u} = [\mathbf{u}_1 \ \ ... \ \ \mathbf{u}_n]^T$. The ocean current velocity in the system dynamic Eq. (6.5) is described in detail in Section 5.1.1, and is rewritten here for the i^{th} sensor location as,

$$\boldsymbol{v}_i(\mathbf{x}_i(t), t) = [v_x(\mathbf{x}_i, t) \ \ v_y(\mathbf{x}_i, t)]^T \tag{6.7}$$

where $\mathbf{x}_i = [x_i \quad y_i]^T$, and the ocean current velocity vector for the entire sensor network is denoted as $\boldsymbol{v} = [\boldsymbol{v}_1 \ \ ... \ \ \boldsymbol{v}_n]^T$. Then, the system dynamics (6.5) is given by (6.5) is written in terms of the sensor network as follows,

$$\dot{\mathbf{x}}(t) = \boldsymbol{v}(\mathbf{x}(t), t) + \mathbf{u}(t), \quad \dot{\mathbf{x}} \in \mathbb{R}^{2n} \tag{6.8}$$

6.2.2 Acoustic Sensor Detection Range

As shown in Sections 2.3 and 5.1.2, the maximum nominal value of the CPA distance at which the sensor can make a positive detection occurs when $d_i = \vartheta/B$. In addition to being affected by the environmental conditions (as discussed in Section 5.1.2), the range is also affected by the use of onboard control by the glider. Onboard control increases the local noise around the acoustic sensor, thereby decreasing the sensor range. The scalar detection range of a sensor can then be approximated by the sum of these two effects,

$$r_i = r(\mathbf{x}_i(t), \mathbf{u}_i(t)) = r^e(\mathbf{x}_i(t)) + r^u(\mathbf{u}_i(t)), \quad 0 < r_i \leq r^e, \quad r_u < 0 \tag{6.9}$$

where $r^u(\mathbf{u}_i(t))$ is the estimated *reduction* in the sensor range r^e brought about by the onboard control $\mathbf{u}_i(t)$. It is assumed here that the estimated reduction of range monotonically decreases as an exponential function of the glider control speed, $V_i = \|\mathbf{u}_i\|$,

$$r_u(\mathbf{u}_i(t)) = \frac{1}{ae^{-b\|\mathbf{u}_i\|}} + c, \quad a, b < 0 < c \tag{6.10}$$

where constants a, b, and c and are chosen, either heuristically by the user or by experimentation, based on sensor and glider design characteristics. For example, in order to bound (6.11) between $[-1, 0]$ for a maximum glider velocity $u_{\max} = 5.4$ Km/hr specified in [101], the constants are chosen heuristically to be $a = -13.88$, $b = -0.5$, and $c = 0.072$ and r^u is illustrated in Fig. 6.2. For simplicity, it is assumed here that the sensor range with respect to the environment is constant, i.e., $r^e(\mathbf{x_i}(\mathbf{t})) = r_{0,i}$ (assumption (ii)). Then, for the entire network $\mathbf{x}(t)$, the range vector is defined as,

$$\mathbf{r}(\mathbf{u}(t)) = \mathbf{r}_0 + \mathbf{r}^u(\mathbf{u}(t)), \quad \mathbf{r}_u \in \mathbb{R}^n \tag{6.11}$$

for $\mathbf{r}(\mathbf{u}(t)) \equiv [r_1(\mathbf{u}_1(t)) \ ... \ r_n(\mathbf{u}_n(t))]^T$.

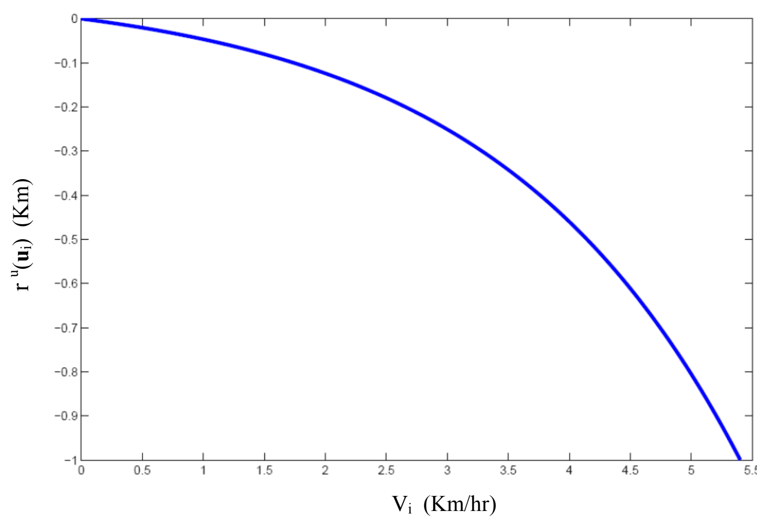


Figure 6.2: Reduction in range as a result of the magnitude of the applied control to the underwater glider.

6.2.3 Objective Function of the Mobile Sensor Network

The sensor system employs a track-before-detect approach to form target tracks based on multiple CPA detections, in order to effectively minimize the probability of false alarms and, eventually, pursue enemy targets. Therefore, a minimum of k distinct detections must be performed per target, anywhere along its track within the area-of-interest $\mathcal{A}$, for a positive detection. Because the surveillance area is much larger than the area covered by the limited number of sensors, it is essential that the sensors move in a cooperative fashion to maximize the track coverage. Also, each glider has limited energy resources, which must also be taken into consideration. In this research, the glider trajectories in $\mathcal{A}$ must optimize a tradeoff of track coverage and vehicle energy consumption. The weighting coefficients manage these competing objectives, as some missions may require more energy consumption (as they may be longer in duration) as opposed to shorter missions that may want more track coverage as the limited amount of energy is of no concern.

The optimal control problem for an autonomous network of sensing underwater gliders is to find the control histories $\mathbf{u}(t)$ which optimize a weighted difference of track coverage and energy consumptions, over a fixed time interval, $[t_0,\ t_f]$, that is,

$$J(\mathbf{x}(t), \mathbf{u}(t), t) = \phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} \left\{ W_{\mathcal{T}} \mathcal{T}_{\mathcal{A}}^{k} \{\mathbf{x}(t), \mathbf{r}(\mathbf{u}(t))\} - W_u \left(\mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) \right) \right\} dt \quad (6.12)$$

where $W_{\mathcal{T}}$ and W_u represent the weights on the total mission track coverage and energy expenditure, respectively. $\mathbf{R}$ is the diagonal control weighting matrix representing the relative importance of the energy of different sensors. It follows from Section 5.1.3 that when the set R_S is given as an implicit function of the control, the range set becomes $R_s\{r_1(\mathbf{u}_1(t)), ..., r_n(\mathbf{u}_n(t))\}$, and in vector form $\mathbf{r}(\mathbf{u}(t)) = [r_1(\mathbf{u}_1(t)), ..., r_n(\mathbf{u}_n(t))]^T \in \mathbb{R}^n$. Because the track coverage function, (3.23) is an implicit function of the sensor position and range within $\mathcal{A}$ (Chapter 3), then these functions are naturally time-varying when extended to the problem of measuring the track coverage over a period of time for location-dependent sensor ranges, i.e., $\mathcal{T}_{\mathcal{A}}^{k}(X_S, R_S) = \mathcal{T}_{\mathcal{A}}^{k}(X_S(t), R_S(\mathbf{u}(t))$.

The final position of the sensors or other mission objectives at the final time t_f can be specified through the penalty function $\phi(\,\cdot\,)$, which could be defined as

$$\phi(\mathbf{x}(t_f)) = W_\phi[\mathbf{x}_f - \mathbf{x}(t_f)]^T[\mathbf{x}_f - \mathbf{x}(t_f)], \quad W_\phi, \phi \in \mathbb{R} \tag{6.13}$$

where $\mathbf{x}_f$ is the desired final position of the sensors, and $W_\phi < 0$ is the weight on the corresponding error. If final positions are not specified, then $\phi(\,\cdot\,) = 0$

6.2.4 Inequality Constraints on the State and Control

As discussed in Section 3.2, the trivial solution to the optimization of track coverage leads to sensors with overlapping fields-of-view, as this provides increased multiple detections. The first proposed method of preventing (or penalizing) the overlapping of sensors includes an additional term in the integrand of the cost function (with its own respective weight) that calculates the area coverage, or the area in $\mathcal{A}$ covered by at least one sensor. This method was investigated in Chapter 5, as the sensors were uncontrollable. The second method, which uses nonlinear state-constraints on the trajectories of n sensors to prevent overlapping disks representing sensors fields-of-view is applicable when the sensors are controllable and the solution is both the trajectories and control policy of the system. Restating (3.44) here, the nonlinear constraint for two sensors with position vectors $\mathbf{x}_i = [x_i \quad y_i]^T$ and $\mathbf{x}_j = [x_j \quad y_j]^T$, and characterized by detection radii r_i and r_j, is given by

$$\mathbf{c}(\mathbf{x}_i(t), \mathbf{x}_j(t)) \equiv -(x_i - x_j)^2 - (y_i - y_j)^2 + (r_i + r_j)^2 \leq 0, \quad i, j = 1, ..., n, \ \ \forall i, j, \ \ i \neq j \tag{6.14}$$

Then, for n sensors, the number of state constraints is $c_1 = \frac{n!}{2(n-2)!}$, where $c_1 \leq c$, and $\mathbf{c} \subset \mathbf{C}$ in Eq. (6.4). An additional path constraint is imposed to bound the gliders to remain in $\mathcal{A}$,

$$\mathbf{c}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{x}(t) - \mathbf{L} \\ -\mathbf{x}(t) \end{bmatrix} \leq \mathbf{0}, \quad \mathbf{c}(\mathbf{x}(t)) \subset \mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{6.15}$$

where $\mathbf{L} = [L_1,\ L_2, ..., L_1, L_2]^T \in \mathbb{R}^{2n\times 1}$. Finally, the constraint on the controls ensures that the speed of the glider is within the physical limitations,

$$\mathbf{c}(\mathbf{u}_i(t)) = \|\mathbf{u}_i\| - \mathbf{u}_{\max} \leq 0, \quad i = 1, ..., n, \quad \mathbf{c}(\mathbf{u}(t)) \subset \mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{6.16}$$

where $\mathbf{u}_{\max} = u_{\max} \cdot [1, ..., 1]^T \in \mathbb{R}^{n\times 1}$

6.3 Numerical Solutions of the Optimal Control Problem

6.3.1 Direct Shooting Approach

A direct shooting approach is implemented in this section to solve the optimal control problem (6.12), (6.3), (6.8), (6.14)-(6.16) numerically through parametric zero-order hold control of a uniformly discretized dynamic system. The control is approximated as piecewise-constant between equidistant collocation points, and the system dynamics are explicitly integrated by forward (Euler) integration of the nonlinear state-space equations. The infinite-dimensional optimal control problem is approximated by a finite-dimensional NLP

$$J(P) = \phi(\mathbf{x}_N) + \Delta t \sum_{T=0}^{N-1} \mathcal{L}[\mathbf{x}(T), \mathbf{r}(\mathbf{u}(T)), \mathbf{u}(T), T], \tag{6.17}$$

where $\mathcal{L}[\,\cdot\,]$ is the integrand of the cost function (6.12), with constraints

$$\mathbf{c}(P) \leq \mathbf{0} \tag{6.18}$$

$P \in \mathbb{R}^M$ is a finite set of parameters uniquely defining the controls $\mathbf{u}(t)$ and states $\mathbf{x}(t)$ of the system. For N equally-spaced switching times $(T_0, T_0 + \Delta t, ..., T_{N-1})$ the parameter set P for a sensor network is given as,

$$P = \{\mathbf{u}_{T_0}, ..., \mathbf{u}_{T_{N-1}}, \mathbf{x}_{T_0}, ..., \mathbf{x}_{T_N}\}, \quad M = 2n \times (2N + 1) \tag{6.19}$$

where the notation for $\mathbf{x}$ and $\mathbf{u}$ is simplified in discrete time as $\mathbf{x}_{T_i} = \mathbf{x}(T = i)$ and $\mathbf{u}_{T_i} = \mathbf{u}(T = i)$. The constraints (6.14)-(6.16) defined in function space are converted to finite-dimensional functions of the parameters in Eq. (6.18) of the NLP. The constraints must be satisfied at N discrete points in time, where the number of finite inequality constraints on the parameters is $c \cdot N$.

The continuous trajectory $\mathbf{x}(t)$ represented by integrating the ODE in Eq. (6.2),

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)dt, \quad \mathbf{x}(t) \in \mathbb{R}^{2n} \tag{6.20}$$

can be approximated by the following discretization

$$\begin{aligned} \mathbf{x}(T) &= \mathbf{x}(T-1) + \Delta t \cdot \mathbf{f}(\mathbf{x}(T-1), \mathbf{u}(T-1), T-1) & (6.21) \\ &= \mathbf{x}(0) + \Delta t \cdot \sum_{\tau=0}^{T-1} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau), \qquad \mathbf{x}(0) = \mathbf{x}_{T_0} & (6.22) \end{aligned}$$

where $\Delta t = t_f/N$. Substituting the nonlinear glider dynamic equation (6.8) into (6.22) over the time interval $[T, T+1)$ yields,

$$\mathbf{x}(T) = \mathbf{x}(0) + \Delta t \cdot \sum_{\tau=0}^{T-1} [\boldsymbol{\upsilon}(\mathbf{x}(\tau), \tau) + \mathbf{u}(\tau)] \tag{6.23}$$

The inherent continuous-time behavior of the dynamic constraint (6.8) is incorporated into the discretization of (6.23) by further constraining the control and currents to be constant along each time interval and change discontinuously only at the switching times. Then, the cumulative cost is obtained by Euler forward integration of the discretized performance index (6.17) and the state-space equations (6.23).

In Section 6.4, the NLP (6.17)-(6.19) is solved using SQP [67], [68], giving a suboptimal but feasible solution to the optimal control problem (6.12), (6.3), (6.8), (6.14)-(6.16). This solution can be arbitrarily close to the global solution by choosing the size of the parameter set P arbitrarily large (i.e., $M \to \infty$ and $\Delta t \to 0$) [102].

6.3.2 Gauss Pseudospectral Method of an NLP

A direct approach to solving the continuous time optimal control problem described in Eqs. (6.12), (6.3), (6.8), (6.14)-(6.16) is referred to as the Gauss Pseudospectral methom (GPM) [103–105]. GPM is an orthogonal collocation method where the collocation points are the Legendre-Gauss points of which the state and control is approximated. The following summarizes GPM for this one-phase optimal control problem, while the reader is referred to [103] for further details of the method.

Since the Bolza problem defined in Eqs. (6.12), (6.3), (6.8), (6.14)-(6.16) is defined over the time interval $t \in [t_0, t_f]$, and GPM points lie on the interval $\tau \in [-1, 1]$, the following transformation is used to express the problem in $t \in [t_0, t_N] = [-1, 1]$,

$$\tau = \frac{2t - (t_f + t_0)}{t_f - t_0} \tag{6.24}$$

It follows that by using Eq. (6.24), the scalar cost function (6.12) is given in terms of $\tau \in [-1, 1]$ as,

$$J = \phi(\mathbf{x}(-1), t_0, t_f) + \frac{t_f - t_0}{2} \int_{-1}^{1} \left(W_{\mathcal{T}} \cdot \mathcal{T}_{\mathcal{A}}^{k}(\mathbf{x}(\tau), \mathbf{r}(\mathbf{u}(\tau))) - W_u \cdot \mathbf{u}^T(\tau) \mathbf{R} \mathbf{u}(\tau) \right) d\tau \tag{6.25}$$

subject to the constraints,

$$\dot{\mathbf{x}}(\tau) = \frac{t_f - t_0}{2}\left(\boldsymbol{\upsilon}(\mathbf{x}(\tau),\tau) + \mathbf{u}(\tau)\right) \tag{6.26}$$

$$\boldsymbol{\Phi}(\mathbf{x}(-1),\tau_0,\tau_f) = \mathbf{0} \quad \in \mathbb{R}^q \tag{6.27}$$

$$\mathbf{C}(\mathbf{x}(\tau),\mathbf{u}(\tau),\tau) \leq \mathbf{0} \quad \in \mathbb{R}^c \tag{6.28}$$

The state, $\mathbf{x}(\tau)$, of the continuous problem is approximated using a basis of $N+1$ Lagrange interpolating polynomials, L_i, $(i = 0, ..., N)$, on the closed interval $\tau \in [-1, 1]$ as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^{N} \mathbf{X}(\tau_i) L_i(\tau), \tag{6.29}$$

where $L_i(\tau)$ for $i = 0, ..., N$ is defined as follows

$$L_i(\tau) = \prod_{j=0,\ j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \quad \Bigg| \quad L_i(\tau_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \quad i, j = 0, ..., N \tag{6.30}$$

The control is also approximated using a basis of N Lagrange interpolating polynomials L_i,$(i = 1, ..., N)$ as

$$\mathbf{u}(t) \approx \mathbf{U}(\tau) = \sum_{i=1}^{N} \mathbf{U}(\tau_i) L_i(\tau), \tag{6.31}$$

where L_i in Eq. (6.30) also satisfies the property in Eq. (6.30) except for the single difference of $i, j = 1, ..., N$.

Once J is transcribed to be $\tau \in [-1, 1]$, then it is approximated using a Gauss Quadrature [106], where the discretization occurs at the Gauss points. The approximation to the derivative of the state $\dot{\mathbf{x}}(\tau)$ in terms of $\mathbf{x}(\tau)$ at the collocation points

τ_i, we differentiate Eq. (6.29) to obtain a matrix multiplication of the following form:

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=0}^{N} \mathbf{X}(\tau_i)\dot{L}_i(\tau) = \sum_{i=0}^{N} D_{\ell i}(X)(\tau_\ell) = \left(\boldsymbol{\upsilon}(\mathbf{X}(\tau_\ell), \tau_\ell) + \mathbf{U}(\tau_\ell)\right), \ \ \ell = 1, ..., N, \tag{6.32}$$

where $D_{\ell i}$ are the elements in the $D \in \mathbb{R}^{N \times N+1}$ differential approximation matrix determined offline from the exact derivative of the Lagrange interpolating polynomials $L_i(\tau)$ as follows

$$D_{\ell i} = \dot{L}_i(\tau_k) = \sum_{\imath=0}^{N} \frac{\prod_{j=0, j\neq i,\imath}^{N}(\tau_\ell - \tau_j)}{\prod_{j=0, j\neq i}^{N}(\tau_i - \tau_j)} \tag{6.33}$$

The continuous time problem is transcribed into a NLP using the variables $\mathbf{X}_\ell = \mathbf{X}(\tau_\ell) \in \mathbb{R}^n$ for the state and $\mathbf{U}_\ell = \mathbf{U}(\tau_\ell) \in \mathbb{R}^m$ for the control and the ℓ^{th} Gauss point. When using GPM, the differential dynamic constraints are applied only at the N collocation points, whereas the state is approximated at $N+1$ interpolation points (including $\tau = -1$) [103]. The cost functional in Eq. (6.25) is approximated using the Gauss quadrature [106] as

$$J = \phi(\mathbf{X}_0, t_0, t_f) + \frac{t_f - t_0}{2} \sum_{\ell=1}^{N} w_\ell \left(W_{\mathcal{T}} \cdot \mathcal{T}_{\mathcal{A}}^{k}(\mathbf{X}_\ell, \mathbf{r}(\mathbf{U}_\ell)) - W_u \cdot \mathbf{U}_\ell^T \mathbf{R} \mathbf{U}_\ell\right) \tag{6.34}$$

where w_ℓ are the Gauss weights. The discretized differential dynamic constraint is transcribed into algebraic constraints via D as follows,

$$\sum_{i=0}^{N} D_{\ell i}\mathbf{X}_i - \frac{t_f - t_0}{2} \sum_{\ell=1}^{N} w_\ell \left(\boldsymbol{\upsilon}(\mathbf{X}_\ell, \tau_\ell) + \mathbf{U}_\ell\right) = \mathbf{0} \tag{6.35}$$

Finally, the boundary and path constraint, where the latter is evaluated at the LG points, are expressed as

$$\mathbf{\Phi}(\mathbf{X}_0, t_0, t_f) = \mathbf{0} \tag{6.36}$$

$$\mathbf{C}(\mathbf{X}_\ell, \mathbf{U}_\ell, \tau_\ell) \leq \mathbf{0}, \quad \ell = 1, ..., N \tag{6.37}$$

The cost function of Eq. (6.34) together with the algebraic constraints of Eqs. (6.34)-(6.36) comprise the NLP whose solution is an approximate solution to the continuous Bolza problem, and solved using the optimal control software GPOCS [105].

6.3.3 Single-Vehicle Minimal Energy Method

Much of the research in trajectory optimization of an underwater vehicle is to minimize energy for a single vehicle for fixed initial and final positions [94–96]. Using these methods, a fast and effective algorithm, referred to here as *single-vehicle minimal energy method* (SVM), is formulated here by solving only for the minimal energy trajectories of each glider individually. The cooperative measure of track coverage is incorporated into the individual optimal trajectories through the initial and final positions of the gliders. These positions are acquired by obtaining several local solutions to the static track coverage optimization problem state in Problem 4.0.1 (i.e., a solution for each random initialization) from Chapter 4. If the initial positions are fixed (e.g., a prior deployment which is not optimal), the final positions may be obtained by employing the optimal repositioning algorithm described in Section 4.1.4, for a value of w that constrains the sensors to remain in $\mathcal{A}$. If the initial sensor positions are not given (i.e., $\mathbf{x}_0$: free), then the initial and final positions are heuristically selected as a local optimum solution. For example, a heuristic strategy to selecting the initial and final positions utilizes the symmetry of the track coverage function. A static local optimal solution may be to place sensors centralized to one part (e.g., a corner) of $\mathcal{A}$. Then, the final position can be selected by translating these sensors to the opposite corner of $\mathcal{A}$. Then, the control strategy is for each sensor to travel from its initial position to the desired final position with minimal control-effort.

As to be expected, this algorithm provides less coverage than the solutions obtained by DSM and GPM. However, it provides very little computation and is shown to improve track coverage significantly over both the area coverage solution and the no control trajectories. Formally stated, the optimal control problem for the individual gliders is to find the control histories $\mathbf{u}(t)$ which minimizes the energy consumption, over a fixed time interval, $[t_0,\ t_f]$, that is,

$$J'(\mathbf{u}_i(t)) = \phi(\mathbf{x}_i(t_f)) - \int_{t_0}^{t_f} \left\{ W_u \cdot \left(\mathbf{u}_i^T(t)\mathbf{R}\mathbf{u}_i(t) \right) \right\} dt, \quad \mathbf{R} \in \mathbb{R}^{2\times 2} \tag{6.38}$$

$$\phi(\mathbf{x}_i(t_f)) = W_\phi \cdot [\mathbf{x}_{i,f} - \mathbf{x}_i(t_f)]^T [\mathbf{x}_{i,f} - \mathbf{x}_i(t_f)], \quad W_u, W_\phi, \phi \in \mathbb{R} \tag{6.39}$$

where $W_u > 0$ and $W_\phi > 0$ represent the weights on the total energy expenditure and the final position, respectively, and $\mathbf{x}_{i,f}$ and $\mathbf{x}_f(t_f)$ are the desired and actual final position of the i^{th} sensor. The control that minimizes J' is formulated as an NLP,

$$\min J'(\mathbf{u}_i(t)), \tag{6.40}$$

$$\text{subject to} \ \ \mathbf{x}_i(t) = \mathbf{x}_{i,0} + \int_{t_0}^{t} \boldsymbol{v}_i(\mathbf{x}_i(t), t) dt \tag{6.41}$$

$$0 < x_{i,0} < L_1 \tag{6.42}$$

$$0 < y_{i,0} < L_2 \tag{6.43}$$

where $\mathbf{x}_{i,0} = [x_i(t_0) \quad y_i(t_0)]^T$ and the objective function is given by (6.38). The NLP (6.40)-(6.43) is solved by the SQP algorithm [67, 68] using Runge-Kutte (4,5) numerical integration [88]. The Single-Vehicle Minimal Energy Method is presented as Algorithm 2.

6.4 Application to the Optimal Control of Underwater Gliders in Sensor Surveillance Systems

The methodology developed in the previous sections is implemented for trajectory optimization of a network of gliders, each equipped with an acoustic sensor network,

Algorithm 2 Pseudocode of the Single-Vehicle Minimal Energy Method

if $\mathbf{x}_0$ is fixed **then**
  Solve the NLP (4.1),(4.2),(4.7)-(4.12) for $\mathbf{x}_f^*$
else
  Solve the NLP (4.1)-(4.4) for multiple initializations
  Heuristically select $\mathbf{x}_0^*$ and $\mathbf{x}_f^*$ from the local static optimums
end if
for $i = 1$ to n **do**
  Solve for the control that minimizes J' by the NLP (6.40)-(6.43)
end for

for the purpose of the cooperative track-coverage measure over an area $\mathcal{A}$ with dimensions $L_1 \times L_2 = 90 \times 82.5$ (Km^2). The nominal ranges $\mathbf{r}_0$ that represent the unforced sensors fields-of-view are listed in Table 6.1 for the different sensor networks used in this study. Due to the highly nonlinear cost function J (6.12) and dynamics (6.8), one must strive to be convinced that the numerical solution obtained to the optimal control problem is a reasonable approximation to a (global) maximum. As there are no systematic methods for this [107], a heuristic method to the problem at hand consists of comparing the solutions of three independent approximation methods, DSM, GPM, and SVM, which were discussed in Section 6.3. While it is shown that the solutions obtained from these three methods for four different sensor networks are similar, the solution from DSM is slightly improved over GPM and even moreso over SVM. Then, the DSM algorithm is compared to the control policy obtained by the maximum area coverage algorithm, which has been proposed by several authors, including [14, 108, 109]. In Section 6.4.2 a Pareto-front approach to solving the weighted difference cost function (6.12) illustrates the effects of different weights on energy and track-coverage. Also investigated is the subproblem formulation of trajectory optimization for a mission that requires the sensor network to maintain a certain amount of track coverage. By including an additional constraint on the minimum track coverage and minimizing only energy consumption, the results in Section 6.4.3 show very different results for slight variations in the minimum allowable track coverage.

Table 6.1: Sensor networks size and nominal ranges

$\boldsymbol{n}$	Ranges, $\mathbf{r}_0$ (Km)
10	$[3,\ 3,\ 5,\ 5,\ 6,\ 6,\ 8,\ 8,\ 10,\ 10]^T$
15	$[4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4]^T$
20	$[4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4,\ 4]^T$

6.4.1 Optimal Glider Trajectories

A heuristic method for determining whether the numerical solution is in fact a reasonable approximation to the global maximum consists of comparing the solutions of three independent approximation methods presented in Section $\dot{\text{F}}$or example, the parametric-constant zero-order hold control solution from DSM in Section 6.3.1 is justified if the gliders trajectories are similar to those obtained by GPM, from Section 6.3.2. Then, the solutions obtained by these two methods are compared to the solution from SVM, where the control policy is solved for each glider individually. The results from these methods are summarized in Table 6.2. The performance measures of track-coverage, energy, and objective functional J for weights $(W_E,\ W_{\mathcal{T}}) = (1, 1)$. Only the results from two examples are listed for GPM as this was an extremely time-consuming routine due to the complexity of the cost function, and took considerably more time to run compared to DSM or SVM. As it is observed that GPM and DSM are in fact very similar, with DSM converging to a slightly better solution in both cases, no more simulations from GPM are necessary for comparison.

To illustrate similar solutions obtained by DSM and GPM, Fig. 6.3 shows the solution for $n = 10$ sensors and $k = 3$ required detections over a 3-day mission time. While the solutions to both GPM and DSM are to place the gliders initially in the upper-right-hand corner of $\mathcal{A}$ and similarly the gliders final positions are in the bottom-right-hand corner of $\mathcal{A}$, DSM converged to a better local solution (+2%). For the second example where $\mathbf{x}_0$ is fixed and the mission time is 5-days, the performance measures (energy, track coverage, and multiobjective cost) are illustrated in Figure 6.4. Although the control obtained by DSM is constrained to be a step function, the local solution is in fact better for DSM over GPM (+9%). Therefore, the simplifications to the numerical solution obtained by the numerical direct shooting method in Section 6.3 can be applied successfully to this highly nonlinear, large scale problem in order to achieve a suitable approximation to the (global) solution.

The third method, *single vehicle minimum energy solution* or SVM, consists of solving the minimal energy control solution of each glider according to Algorithm 2. The results of this simulation are listed in Table 6.2. As to be expected, SVM achieves less track coverage than GPM and SVM. By cooperatively controlling the gliders, improvements over SVM are as much as +71%. However, for $n = 10$, $k = 3$, the expected improvement of DSM is only 19%. Also, the the computation expense of SVM is considerably less than DSM and GPM, as the integrand of the cost function contains only the quadratic energy term, and the offline static optimization is completed once (twice for $\mathbf{x}_0$: free). Therefore, SVM provides a computationally less expensive alternative to DSM, although with less expected performance.

The effectiveness of the numerical solution in Section 6.3 from DSM is also compared to the *area coverage* formulation, which is a popular performance measure in the surveillance literature [14, 108, 109] and is derived in Section 3.2. Solving for the maximum area coverage problem can be solved by defining a new cost function that replaces the track coverage term in (6.12) with the area coverage function (3.28). An equivalent (and more efficient) method of solution consists of minimizing the NLP (6.17)-(6.19) for weights $W_u = 1$ and $W_\phi = W_\mathcal{T} = 0$, which includes the nonoverlapping sensors fields-of-view constraint. Then, the performance measures (track

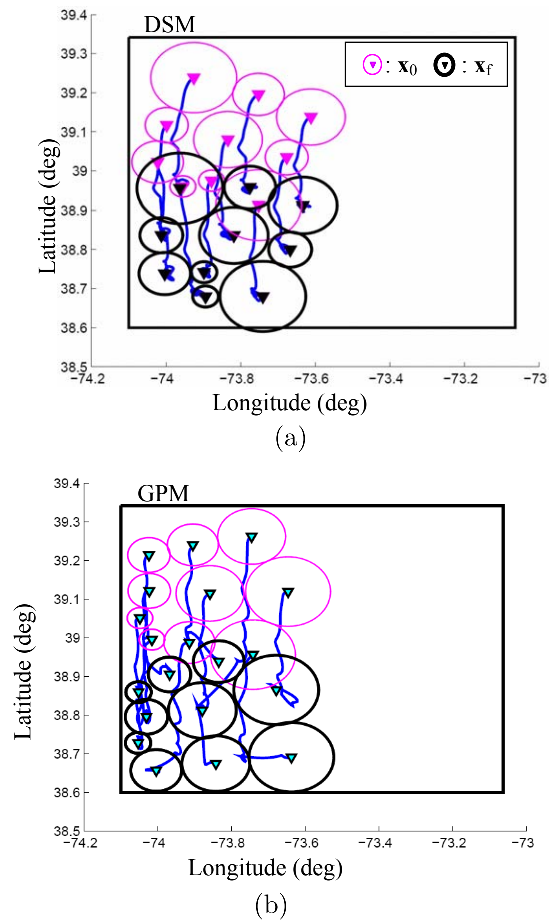


Figure 6.3: Comparison of the solutions to the optimal initial positions and trajectories for the (a) direct shooting method solved by DSM and (b) GPM.

Table 6.2: Performance measures of network parameters and numerical solution, where $(\cdot)$ refers to (DSM Improvement, %).

$(\mathbf{n}, \mathbf{k})$: $\mathbf{x}_0$, t_f	**Performance Measures**	**DSM**	**GPM**	**SVM**
(10,3):	Track Coverage	2.929×10^4	2.802×10^4	2.360×10^3
$\mathbf{x}_0$ fixed,	Energy	924.2	1,406	22.03
5-days	**Cost**, J	2.836×10^4	2.661×10^4 (+9%)	2.358×10^4 (+21%)
(10,3):	Track Coverage	1.821×10^4	1.809×10^4	1.527×10^4
$\mathbf{x}_0$ free,	Energy	164.9	336.6	34.90
3-days	**Cost**, J	1.805×10^4	1.776×10^4 (+2%)	1.523×10^4 (+19%)
(15,3):	Track Coverage	1.742×10^4	–	1.040×10^4
$\mathbf{x}_0$ fixed,	Energy	1,307	–	212.8
3-days	**Cost**, J	1.742×10^4	–	1.019×10^4 (+71%)
(20,4):	Track Coverage	1.818×10^4	–	1.449×10^4
$\mathbf{x}_0$ free,	Energy	268.9	–	162.3
3-days	**Cost**, J	1.791×10^4		1.433×10^4 (+25%)

coverage, energy, and multiobjective cost) are listed in Table 6.3 for comparison. As expected, the track-coverage DSM trajectory optimization significantly improves the track coverage of $\mathcal{A}$ as opposed to the area coverage solution. It is interesting to note that the total amount of area coverage, i.e., the time integral of (3.28), calculated for the two solutions is approximately the same, with only a slight decrease achieved by DSM due to the increase in energy consumption; however, the area coverage solution provides significantly less track coverage (less than half on average). For example, the largest difference between the total area coverage for the maximum area coverage solution and DSM occurs for the sensor network ($n = 15,\ k = 3$), where only a 3.7% increase in area coverage is achieved by the maximum area coverage solution. This is compared to the increase of 121% in track coverage that is achieved by the track coverage solution DSM over the maximum area coverage solution. Therefore, the for-

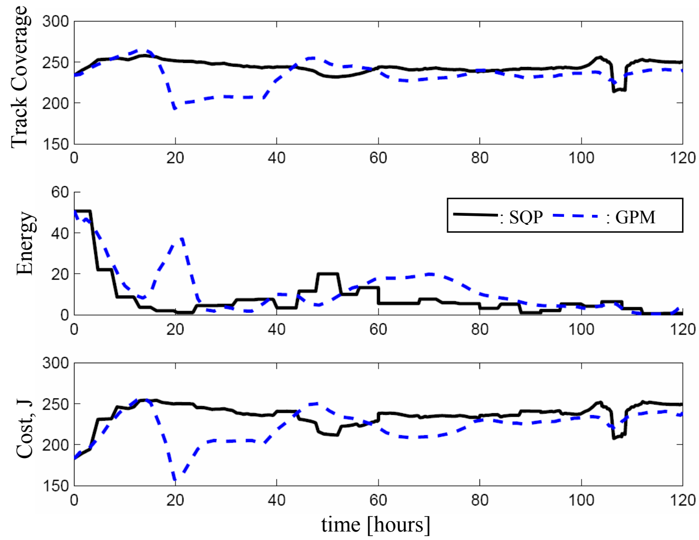


Figure 6.4: A comparison of the three performance measures of the solutions to the trajectory optimization obtained by DSM and the the direct shooting method solved by DSM for the example $n = 10$, $k = 3$, $\mathbf{x}_0$ fixed and mission time is 5-days.

mulation and solution to the track-coverage optimization can be directly applied to area coverage problems, whereas it is not true vice versa. Comparing the trajectories of the n sensors for the two solutions (Figs. 6.5(a) and 6.5(b)), it is apparent why the track coverage is so poor for the area coverage solution. Track coverage is best achieved when the limited number of sensors are in a clustered configuration which maximizes the k detections required for a positive detection. The final comparison of DSM is to the zero-control scenario, where sensors are simply left to move with the currents for a given sensor configuration (i.e., $\mathbf{x}_0$ fixed). Without control, the sensors fields-of-view overlap and the gliders drift outside of $\mathcal{A}$ (Fig. 6.5(c)). This results in significant coverage holes, and improvement of the DSM trajectory optimization is nearly 9 times that of zero-control. It should also be noted that the solutions obtained by SVM provide significant improvement compared to both the area coverage solution and the zero-control trajectories.

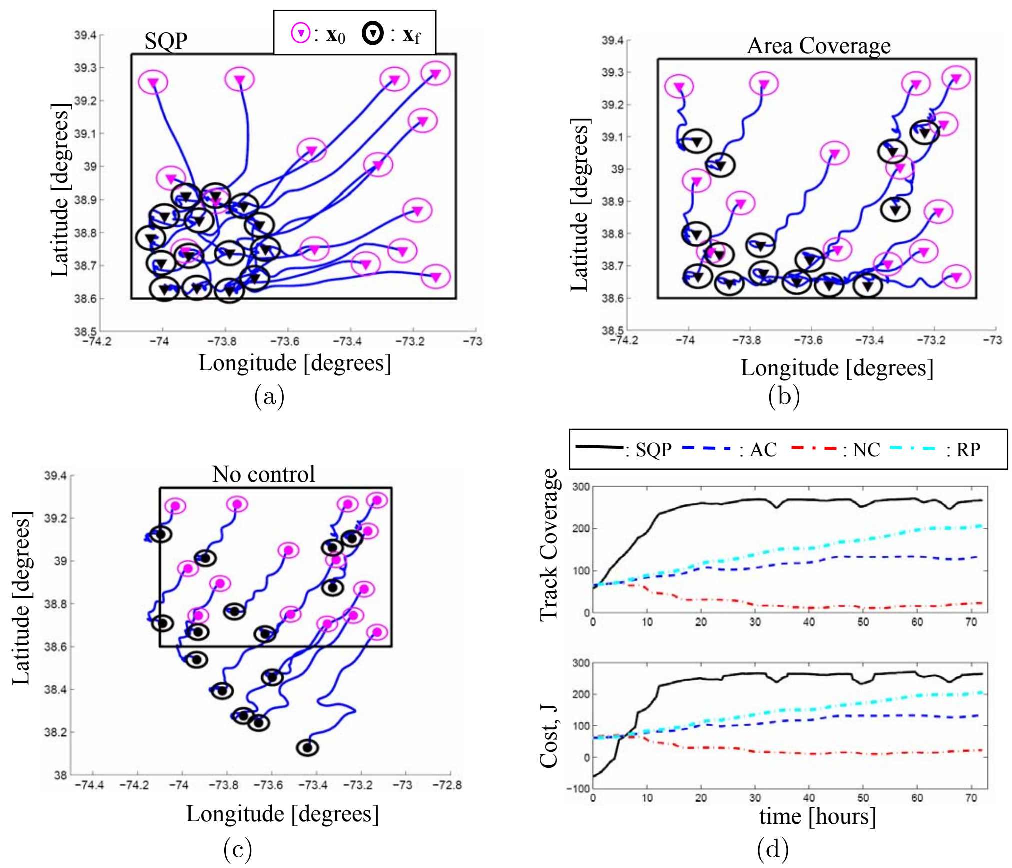


Figure 6.5: Comparison of the trajectories from (a) trajectory optimization via DSM, (b) area coverage, and (c) zero-control, while (d) shows that the DSM solution achieves significantly higher track-coverage over the other three methods.

Table 6.3: Performance measures for the network parameters (n, k) and control policy over a mission period-of-time, where $(\cdot)$ refers to (DSM Improvement, %).

$(\mathbf{n}, \mathbf{k})$: $\mathbf{x}_0$, t_f	**Performance Measures**	**DSM**	**Maximum Area Coverage**	**Zero-Control**
(10,3):	Track Coverage	2.929×10^4	1.710×10^4	1.274×10^4
$\mathbf{x}_0$ fixed,	Energy	924.2	0.5443	0
5-days	**Cost**, J	2.836×10^4	1.822×10^4 (+56%)	1.274×10^4 (+123%)
(10,3):	Track Coverage	1.821×10^4	9,731	–
$\mathbf{x}_0$ free,	Energy	164.9	1.508×10^{-4}	–
3-days	**Cost**, J	1.805×10^4	1.217×10^4 (+50%)	–
(15,3):	Track Coverage	1.742×10^4	$8,026$	1,989
$\mathbf{x}_0$ fixed,	Energy	1,307	140.9	0
3-days	Area Coverage	5.213×10^4	5.404×10^4	3.190×10^4
	Cost, J	1.742×10^4	7,880 (+121%)	1,989 (+776%)
(20,4):	Track Coverage	1.818×10^4	6,733	–
$\mathbf{x}_0$ free,	Energy	268.9	9.827×10^{-4}	–
3-days	**Cost**, J	1.771×10^4	6,732 (+166%)	–

6.4.2 Parametric Study of the Multi-Objective Optimal Control Problem

In practice, the weights of the individual cost of energy and track coverage are typically governed by specific mission requirements. For example, missions of longer duration will require a higher energy weight for conservation. The weighted sum approach to multiobjective optimization applied in this research allows for this freedom of weight selection by the user. In order to illustrate the effect of the weights on the solution, i.e., initial deployment scheme and control policy, a parametric study was performed on a sensor network with $n = 10$ sensors and $k = 3$ required detections for the weights $W_E + W_{\mathcal{T}} = K$. Then the multiobjective cost function of the NLP

(6.17) can be restated as

$$J = \max \sum_{\tau=0}^{T} \left[\kappa \cdot \frac{\mathcal{T}_{\mathcal{A}}^{k}(\mathbf{x}(\tau), \mathbf{r}(\mathbf{u}(\tau)))}{sf_{\mathcal{T}}} + (K - \kappa) \cdot \frac{\mathbf{u}(\tau)\mathbf{R}\mathbf{u}(\tau)}{sf_u} \right] \tag{6.44}$$

for the weighting coefficients $W_\phi = 0$, $W_{\mathcal{T}} = \kappa$, $W_u = K - \kappa$, where $\kappa \in [0, K]$, and scaling factors $sf_{\mathcal{T}}$ and sf_u. For this problem these parameters are given as $K = 101$, $sf_{\mathcal{T}} = 100$, $sf_u = 1$. By varying the weights systematically, where each different single objective optimization determines a different Pareto optimal solution, we solve several suboptimization problems obtaining optimal solutions in the objective space, which then leads to an approximation of the Pareto front. The series of Pareto points obtained with weighting increments listed in Table 6.4 are shown in Fig. 6.6(a), where the extremum values of the convex curve are given by the weighting increments $W_E = 0$ and $W_E = K$. Four different solutions are illustrated for their respective weights, three in Fig. 6.6 and one ($W_{\mathcal{T}} = 100$) in Fig. 6.3(a). As expected, the minimal energy ($W_E = K$) provides the least amount of track coverage as the solution is governed by a noncooperative objective. When energy and track-coverage are equally weighted, the solution shows that the sensors more effectively use the natural currents to maneuver to areas of higher track coverage. However, in all cases the solutions recognize the currents in the bottom left hand corner of $\mathcal{A}$ are relatively smaller in magnitude after the first day, which is reflected in the respective initial placements and subsequent trajectories; as $\kappa \to K$, the gliders move to this area of $\mathcal{A}$ much quicker. This natural phenomena of the currents over time shows the importance of including a forecast model of the ocean dynamics into the equations of motion for trajectory optimization.

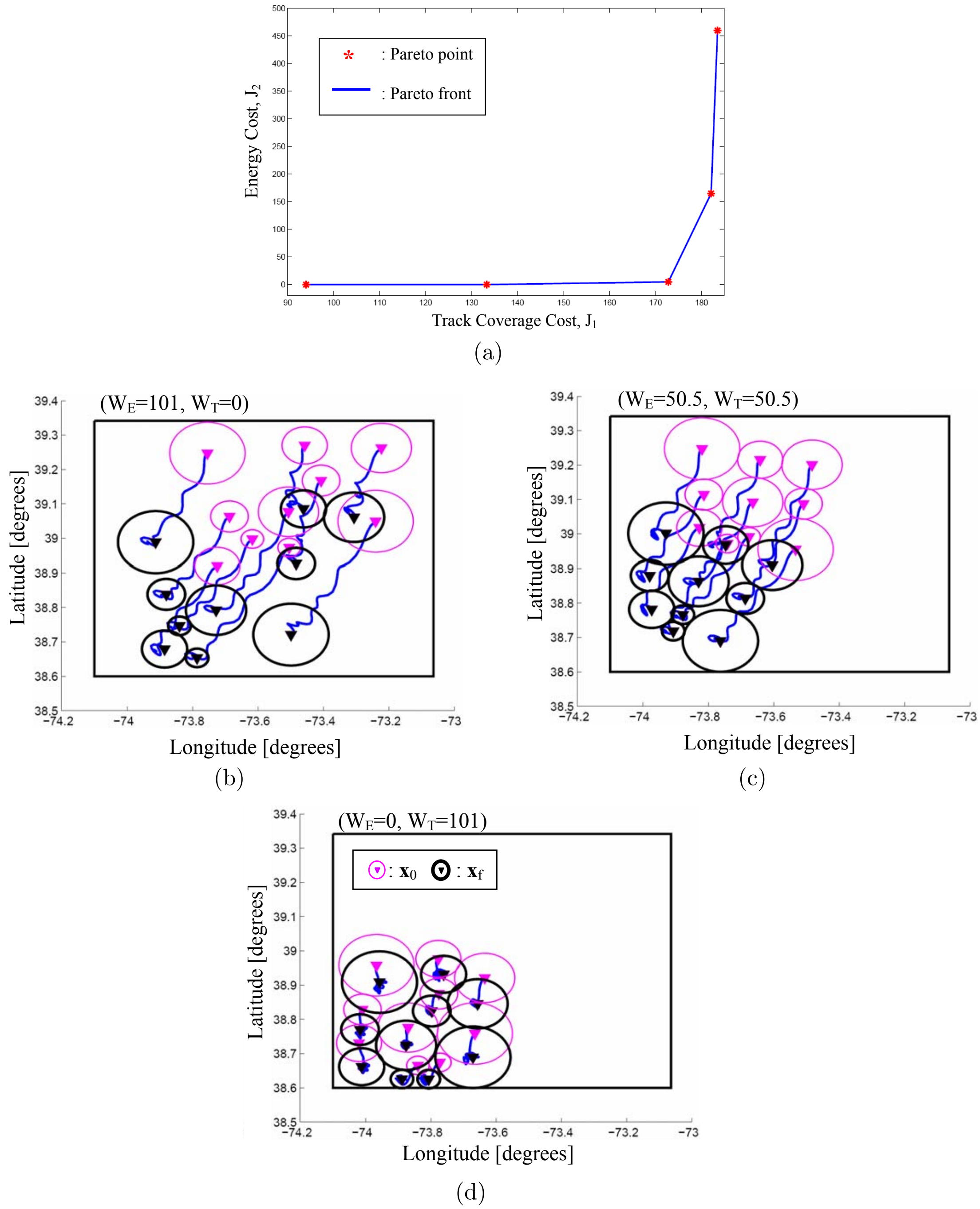


Figure 6.6: (a) Pareto front is a convex curve for the following weights, $(W_E, W_{\mathcal{T}})$:(b) (101,0), (c) (50.5,50.5), (d) (0,101), while (1,100) is in Fig. 6.3(a).

Table 6.4: Maximal solutions from the weighted sum approach

$(W_E,\ W_{\mathcal{T}})$	$1/sf_u \cdot \int \mathbf{u}(t)^T \mathbf{R}\mathbf{u}(t)dt$	$1/sf_{\mathcal{T}} \cdot \int \mathcal{T}_{\mathcal{A}}^k(\,\cdot\,)dt$	J
(101, 0)	2.065×10^{-5}	94.036	-2.086×10^{-3}
(100, 1)	0.0172	133.3	131.6
(50.5, 50.5)	4.844	172.8	8,482
(1,100)	164.9	182.1	1.805×10^4
(0,101)	459.9	183.5	1.853×10^4

6.4.3 Inequality Constraints for Maintaining Track Coverage Above a Minimum Threshol

In certain missions, it may be necessary to maintain a minimum amount of track-coverage. Then, the problem is reformulated as a minimum energy problem (i.e., $W_{\mathcal{T}} = 0$) that includes an additional constraint in $\mathbf{C}(\,\cdot\,)$, as

$$\mathbf{c}(\mathcal{T}_{\mathcal{A}}^k[\mathbf{x}(t), \mathbf{r}(\mathbf{u}(t))]) = T_{\min} - \mathcal{T}_{\mathcal{A}}^k[\mathbf{x}(t), \mathbf{r}(\mathbf{u}(t))] \leq 0 \tag{6.45}$$

where $T_{\min}$ is the minimum allowable track coverage for the mission. These results for $(n, k) = (20, 3)$ show how the minimum energy solution is affected when a minimum constraint on energy is included in the problem formulation. Figure 6.7(a) shows that the amount of track coverage remains close to its lower bounds over the mission time, in this case a three-day period.

Slight changes in the minimum energy constraint results in significant differences in total energy consumption. For example, even though two values of $T_{\min}$ are only 3% different, the gliders' trajectories are significantly different. When $T_{\min} = 315$, the total energy is increased nearly 320 times over $T_{\min} = 300$. When $T_{\min} = 325$, sensors are placed very close to their final positions in a corner (Fig. 6.8(b)), as a clustered corner configuration provides the most instantaneous track-coverage. It is the left hand corner due to the natural phenomena of smaller currents after the first day, and any other corner would have both used more energy to remain there, and

$T_{\min}$ would have been violated by traveling from one corner to the next. Therefore, most of the control is applied in the first day until the currents become smaller, as evident from Fig. 6.7(b). The solution to $T_{\min} = 300$, which is illustrated in Fig. 6.8(a), shows how important the initial positions are to minimizing the overall energy, as very little control is needed to achieve relatively high track coverage over the entire mission Fig. 6.7).

Table 6.5: Total amounts of energy and track coverage for different values of $T_{\min}$.

$\mathbf{T_{\min}}$	**Energy**	**Track Coverage**
300	0.591	2.191×10^4
315	188.8	2.260×10^4
325	254.0	2.346×10^4

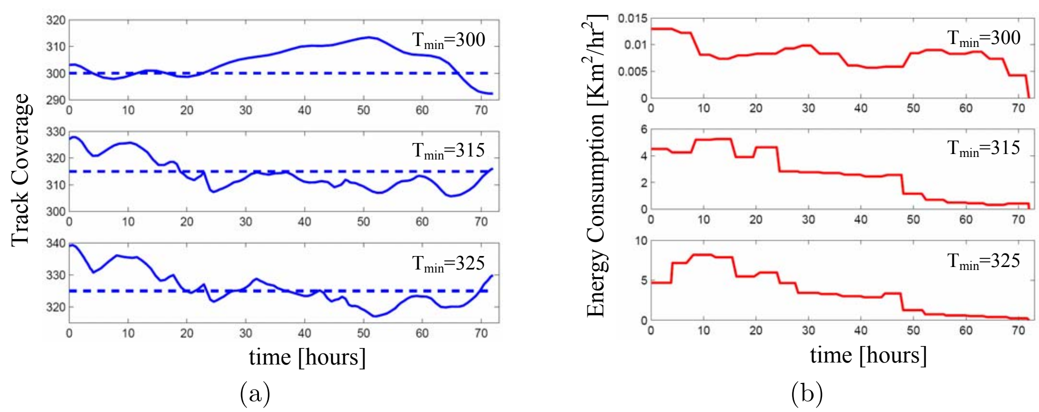


Figure 6.7: The (a) track coverage and (b) energy for the three different values of $T_{\min}$.

6.5 Summary and Conclusions

A new optimal control problem has been formulated here that seeks to control a group of independent gliders in a cooperative manner for the purpose of track coverage. By incorporating a forecast model into the dynamic constraints, the glider trajectories can effectively maneuver to areas that require less energy long-term. Prediction of

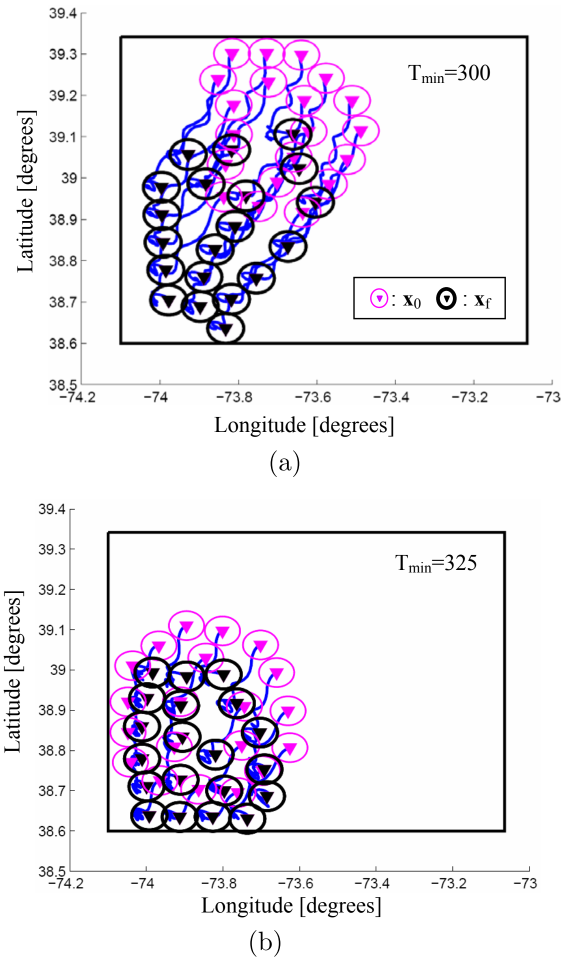


Figure 6.8: The solution of the gliders positions and trajectories for (a) $T_{\min} = 300$ and (b) $T_{\min} = 325$.

the temporal evolution of the oceanographic conditions can be generated by forecast models, but also updated as the vehicles navigate by incorporating new measurements and updated forecasts. Hence the trajectory optimization problem can be applied in mixed pregenerative/reactive planning situations. A central challenge in guidance and control of autonomous vehicles is the difficulty of efficiently computing trajectories that exploit the domains of the vehicle's nonlinear behavior as a result of environmental conditions. It is shown here that the direct shooting, parametric zero order hold control numerical solution converged to a better local solution in a much shorter time compared to GPM. It was also shown that the track-coverage solution provided nearly maximum area coverage, while the maximum area coverage solution provided much less track coverage. When compared to the independent repositioning problem, the improvement of the cooperative track coverage solution was less significant than the improvement over area coverage due to the incorporation of the static local optimums into the initial and final conditions of the problem formulation. This approach could provide the basis for future trajectory optimization algorithms. A parametric study performed on the weights of the objective function provided an approximation to the Pareto-front. In conclusion, the optimal control problem was shown to be easily modified according to specific mission requirements, e.g., longer missions vs. shorter missions, achieve maximal area coverage, and maintain a minimum amount of track coverage over time.

Chapter 7

Conclusion

The primary objective of this dissertation is to optimize the quality of service of sensor networks that cooperatively detect targets traversing a region of interest. A novel approach is presented for defining and formulating the track coverage in a sensor network that tracks a moving target through limited measurements, such as CPA detections. Central to this approach is the formulation that is based on planar geometry and the introduction of a k-coverage cone, which quantifies the amount of tracks detected by k sensors in terms of opening angles along the boundaries of an area of interest. A coverage function is derived analytically to express the k-coverage in terms of sensor locations and ranges. Consequently, the track coverage function can be optimized using a nonlinear program (NLP) in order to compute the optimal network placement over a region of interest for known and constant sensors range. The numerical results show that optimal sensor placement significantly increases track coverage compared to existing grid and random deployment schemes. In scenarios where a deterministic deployment is not feasible or when sensor networks have been displaced over time by winds or oceanic currents, this methodology is easily modified to reposition sensors and significantly improve track coverage. This method enables a very practical and cost-efficient alternative to replenishing sensor networks.

The methodology developed for measuring the fixed (instantaneous) track coverage of a sensor network is extended to measure track coverage of a moving sensor network with respect to an area of interest over a period of time. This problem is relevant to a sensor network of sonobuoys floating and drifting according to the oceanic-induced velocity field. This system is initially considered non-maneuverable, and therefore the track coverage optimization problem is formulated as the optimal initial positions of the sonarbuoys that, over a fixed period of time, provide maximum cumulative track coverage. To solve this problem, we developed a closed-form system of differentiable equations describing the cumulative track coverage with respect to the natural trajectories of the sensors, and their location-dependent range over a fixed period of time. A state-space representation of the motions of the individual sensors subject to the ocean current vector fields can be derived from sonobuoy drift models, with specific values obtained by CODAR measurements in tabular form. In order to incorporate the drift dynamics into the system dynamics for optimization, a closed form function describing the nonlinear, time-varying ocean velocity field is approximated by a neural network. A BN was shown to provide a realistic model of the location-dependent ranges by incorporating both known range models and the passive sonar equation.

Then, the closed-form models of the current-velocities and sensor ranges are incorporated implicitly into an objective function that is the weighted sum of the cumulative track and area coverage. The inclusion of area coverage is necessary to counteract the trivial solution to the track-coverage function. The optimal initial positions significantly improve upon other deployment methods proposed in the literature (i.e., grid and random), as well as the solution to the track coverage problem that does not include the ocean current-velocities. Most significant to the optimal initial deployment solution is that when uncertainty is incorporated into the system and incorporated over the entire trajectory, cumulative track coverage is not significantly affected. Even when a bias error is included into the system, the outcome is nearly the same as including no bias error. This entire methodology demonstrates the importance of a deterministic deployment scheme, even if the deployment must be done quickly and in less than ideal weather conditions.

The methodology was then extended to a controllable platform of underwater gliders with onboard acoustic sensors. This formulation led to a new optimal control problem, as none of the current control formulations for mobile sensor networks address the optimization of the trajectories of a group of underwater vehicles for cooperative coverage in the presence of ocean dynamics. As solving for a large-scale, highly nonlinear control problem is very difficult, several methods of solution were implemented. A computationally less-expensive direct-shooting method of solution was shown to provide superior results compared to GPM. Although this particularly method of solution may not always be applicable (as discussed in Section 7.1), it does provide an efficient approximation that can either be used as a main solution or as a comparison to a more effective solution in future work.

The contributions described in this dissertation lead to a successful implementation of the different types of sensor networks, i.e., fixed, moving, and mobile. The improvements of all key design stages compared to existing methods lead to a highly effective method for coverage that is both cost efficient and practical. The formulations developed here greatly increases the potential for real life applications of simple and inexpensive proximity sensors for advanced surveilling of a region of interest.

7.1 Recommendations

The main recommendation of future work is to expand upon the different nonlinear models used in this research and address these problems in a three-dimensional Euclidean space. In this case, the two-dimensional disk becomes a three-dimensional spherical sensor field-of-view and the system dynamics include the z-direction of the ocean current-velocities. However, CODAR data is limited to measuring two-dimensional surface current velocities over a period of time. Current projects, such as the large-scale Autonomous Ocean Surveying Network [110], employ a group of underwater gliders to survey a region of the ocean. Gliders naturally provide data of a three-dimensional region of interest over a period of time and may provide a sample of the ocean current velocities for purposes of training a NN to map a four-dimensional input (x, y, z, and t) to a three dimensional output ($\boldsymbol{v}_x$, $\boldsymbol{v}_y$, $\boldsymbol{v}_z$). In addition to the three-dimensional ocean current profile, it would be interesting and provide more validation to the effectiveness of this coverage formulation and method of solution to investigate a different area of the ocean. One potential area of interest is off the coast of Monterey Bay, CA, where particularly strong surface currents are known to occur just outside the bay.

One of the most difficult components of optimizing a highly nonlinear cost function is to solve for the global optimum. In this dissertation, we either introduced many random initializations or compared the solutions from to several other independent methods. The GPM actually provided, at least in theory, an analytical test in which to show the goodness of solution, as it provides estimation procedures which can be used to verify the optimality of the resulting solution. However, it was shown here that GPM does not yet converge to solution in a reasonable amount of time due to the large-scale, highly nonlinear model.

Appendix A

Proof of Remark 3.1.1

Let $\mathcal{R}_\alpha(b_y)$ denote a ray that intersects $C_i = C_i(\mathbf{x}_i, r_i)$ in $\mathbb{R}^2_+$. Consider any two points that lie on $\mathcal{R}_\alpha(b_y)$ and inside $C_i(\mathbf{x}_i, r_i)$, and let $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^2_+$ denote their positions relative to the origin $\mathbf{y}_0$ of the coverage cone $K(C_i, \mathbf{y}_0)$. By construction, $\mathbf{u}_1, \mathbf{u}_2 \in C_i(\mathbf{x}_i, r_i)$ and a vector $\mathbf{z}$ joining the two points will lie on the ray $\mathcal{R}_\alpha(b_y)$. Let c_1 and c_2 denote any two positive constants. By definition of vector sum and subtraction [59], if $\mathbf{z} = c_1\mathbf{u_1} + c_2\mathbf{u}_2$ then $\mathbf{z}$ has the same origin as $\mathbf{u}_1$ and $\mathbf{u}_2$. Thus, since $\mathbf{z}$ lies on $\mathcal{R}_\alpha(b_y)$, $\mathcal{R}_\alpha(b_y)$ intercepts the y-axis at the cone origin $\mathbf{y}_0$. If $\mathbf{z} = \pm c_1\mathbf{u}_1 \mp c_2\mathbf{u}_2$, $\mathbf{z}$ does not have the same origin as $\mathbf{u}_1$ and $\mathbf{u}_2$ and, thus, $\mathcal{R}_\alpha(b_y)$ does not intercept the y-axis at $\mathbf{y}_0$. By definition, $K(C_i, \mathbf{y}_0)$ is the set of all nonnegative combinations of the elements in C_i. Since $\mathbf{u}_1$ and $\mathbf{u}_2$ are two elements in C_i, and any nonnegative combination of these two elements can be written as $c_1\mathbf{u}_1 + c_2\mathbf{u}_2$, with $c_1, c_2 > 0$, it follows that $\mathbf{z} = c_1\mathbf{u}_1 + c_2\mathbf{u}_2 \in K(C_i, \mathbf{y}_0)$. Finally, since $\mathcal{R}_\alpha(b_y)$ denotes any ray with intercept b_y that intersects $C_i = C_i(\mathbf{x}_i, r_i)$ in $\mathbb{R}^2_+$, and $z = c_1u_1 + c_2u_2$ provided $\mathcal{R}_\alpha(b_y)$ intercepts the y-axis at $\mathbf{y}_0$, it also follows that any $\mathcal{R}_\alpha(b_y)$ that intersects C_i and the y-axis at $\mathbf{y}_0$ is contained by $K(C_i, \mathbf{y}_0)$.

Appendix B

Proof of Proposition 3.1.2

This proof considers a family of $k = 3$ non-translates $S_3 = \{C_i, C_j, C_l\}$ with index set $I_{S_3} = \{i, j, l\}$. The results can be extended to higher k by induction. From Remark 3.1.1, a coverage cone $K(C_\ell, \mathbf{y}_0)$ contains the set of all tracks $\mathcal{R}(b_y)$ that intersect C_ℓ in $\mathbb{R}^2_+$, where $\ell \in I_{S_3}$. Then, from set theory, the set of tracks intersecting all disks in the family S_3 is given by the following intersection:

$$K_3(S_3, \mathbf{y}_0) = \bigcap_{\ell \in I_{S_3}} K(C_\ell, \mathbf{y}_0) = K(C_i, \mathbf{y}_0) \cap K(C_j, \mathbf{y}_0) \cap K(C_l, \mathbf{y}_0) \tag{B.1}$$

From the properties of cones [58, pg. 70], the intersection of a collection of cones is also a cone. Thus, $K_3(S_3, \mathbf{y}_0)$ is a cone. A vector $\mathbf{z}$ representing a ray $\mathcal{R}$ lies in a cone K if and only if $\mathcal{R}$ lies in K, since any point on $\mathcal{R}$ can be written as $c\mathbf{z}$, with $c > 0$.

Consider any ray $\mathcal{R}^\ell \in K(C_\ell, \mathbf{y}_0)$, where $K(C_\ell, \mathbf{y}_0) = \text{cone}(\hat{\mathbf{l}}_\ell, \hat{\mathbf{h}}_\ell)$, and thus can be represented by a vector $\mathbf{z}_\ell = c_1\hat{\mathbf{l}}_\ell + c_2\hat{\mathbf{h}}_\ell$ with constants $c_1, c_2 > 0$. Then, $\mathbf{z}_\ell \in K(C_\ell, \mathbf{y}_0)$ and, by the properties of vector sum, $\hat{\mathbf{l}}_\ell \prec \mathbf{z}_\ell \prec \hat{\mathbf{h}}_\ell$. Next, consider a cone $K^* = \text{cone}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*)$ that is finitely generated by two unit vectors $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath$ and $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\imath$ with $\jmath, \imath \in I_{S_3}$, and assume $\hat{\mathbf{l}}_\imath \prec \hat{\mathbf{h}}_\jmath$. By the properties of finitely generated cones [58], any vector $\mathbf{z}^* = b_1\hat{\mathbf{l}}^* + b_2\hat{\mathbf{h}}^*$ with constants $b_1, b_2 > 0$ must lie in K^*. It follows that a ray $\mathcal{R}^*$ with the same slope and origin as $\mathbf{z}^*$ must also lie in K^*, since any point on $\mathcal{R}^*$ can be written as $c\mathbf{z}^*$ with $c > 0$. Since $\mathbf{z}^*$ is a positive combination of $\hat{\mathbf{l}}^*$ and $\hat{\mathbf{h}}^*$, it also follows that $\hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^*$.

According to Proposition 3.1.2, choose $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath \preceq \hat{\mathbf{h}}_\ell$ and $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{l}}_\ell$ for $\forall \ell \in I_{S_3}$. Suppose the unit vectors of S_3 can be ordered as $\hat{\mathbf{h}}_l \prec \hat{\mathbf{h}}_j \prec \hat{\mathbf{h}}_i$ and $\hat{\mathbf{l}}_i \prec \hat{\mathbf{l}}_l \prec \hat{\mathbf{l}}_j$. Then, the unit vectors and $\mathbf{z}^*$ can be ordered as follows,

$$\hat{\mathbf{l}}_\ell \preceq \hat{\mathbf{l}}_j = \hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^* = \hat{\mathbf{h}}_l \preceq \hat{\mathbf{h}}_\ell \quad \text{for} \quad \forall \ell \in \{i, j, l\} = I_{S_3} \tag{B.2}$$

or, more explicitly:

$$\hat{\mathbf{l}}_i \prec \hat{\mathbf{l}}_l \prec \hat{\mathbf{l}}_j = \hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^* = \hat{\mathbf{h}}_l \prec \hat{\mathbf{h}}_j \prec \hat{\mathbf{h}}_i \tag{B.3}$$

Since the above order also implies $\hat{\mathbf{l}}_\ell \prec \mathbf{z}^* \prec \hat{\mathbf{h}}_\ell$ for $\forall \ell \in I_{S_3}$, then $\mathbf{z}^*, \mathcal{R}^* \in K(C_\ell, \mathbf{y}_0)$ for $\forall \ell \in I_{S_3}$. Thus, from (B.1), $\mathbf{z}^*, \mathcal{R}^* \in K_3(S_3, \mathbf{y}_0) = K^* = \text{cone}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*)$, provided $\hat{\mathbf{l}}^*$ and $\hat{\mathbf{h}}^*$ are chosen subject to (B.2).

So far it was assumed that $\hat{\mathbf{l}}_\imath \prec \hat{\mathbf{h}}_\jmath$. If the unit vectors in $\Omega(S_3, \mathbf{y}_0)$ and $\Lambda(S_3, \mathbf{y}_0)$ are such that $\hat{\mathbf{l}}_\imath \succ \hat{\mathbf{h}}_\jmath$, then there are no vectors that can satisfy the order $\hat{\mathbf{l}}_\imath = \hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath$, and $K_3(S_3, \mathbf{y}_0) = K^* = \emptyset$.

Appendix C

Linear Operations for Ordering Unit Vectors According to a Frame of Reference

This Appendix illustrates a methodology for efficiently ordering sets of unit vectors according to a fixed frame of reference. Consider a set of unit vectors $\{\hat{\mathbf{u}}_1, ..., \hat{\mathbf{u}}_n\}$ with index set I. Any unit vector can be written in terms of its direction sine and cosine, namely $\hat{\mathbf{u}}_i = [\cos\gamma_i \quad \sin\gamma_i]^T$, for $\forall i \in I$. We seek to order the unit vectors according to the xy-frame, therefore γ_i can also be viewed as the angle that $\hat{\mathbf{u}}_i$ makes with the x-axis. Then, for any two unit vectors $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}_j$ in the first and fourth quadrant, $\hat{\mathbf{u}}_i \prec \hat{\mathbf{u}}_j$ if and only if $\sin\gamma_i < \sin\gamma_j$. From Proposition 3.1.2, it is of interest to obtain the first or last element of a list comprised of these unit vectors in ascending order: $\{\hat{\mathbf{u}}_\jmath, \hat{\mathbf{u}}_l, \ldots, \hat{\mathbf{u}}_\imath\}$, with $\hat{\mathbf{u}}_\jmath \preceq \hat{\mathbf{u}}_l \preceq \ldots \preceq \hat{\mathbf{u}}_\imath$. The first and last elements, $\hat{\mathbf{u}}_\jmath$ and $\hat{\mathbf{u}}_\imath$, can be obtained without ordering the entire set, using the following pair-wise linear operations on the direction sines of the unit vectors:

$$\sin\gamma_\jmath = \frac{1}{2}\left[\sin\gamma_i + \sin\gamma_j - |\sin\gamma_i - \sin\gamma_j|\right] \tag{C.1}$$

$$\sin\gamma_\imath = \frac{1}{2}\left[\sin\gamma_i + \sin\gamma_j + |\sin\gamma_i - \sin\gamma_j|\right] \tag{C.2}$$

$$i \neq j, \quad \forall i, j \in I$$

It can be easily shown that the unit vectors generating cones with origin $\mathbf{y}_0$ on the y-axis always lie in the first or fourth quadrant. Thus, the k-coverage cone $K_k(S_k, \mathbf{y}_0)$ can be obtained by applying (C.1) to $\Omega(S_k, \mathbf{y}_0)$, and by applying (C.2) to $\Lambda(S_k, \mathbf{y}_0)$, as shown in Proposition 3.1.2.

In Section 3.1.1 , k-coverage cones are also defined with respect to the x, y', and x' axes. These cones can also be obtained by applying (C.1) and (C.2) to the corresponding sets of unit vectors provided they first undergo a constant rotation. Let the following rotation matrices denote clockwise rotations of 90°, 180°, and 270°:

$$Q^{90} \equiv \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad Q^{180} \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q^{270} \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{C.3}$$

Then, (C.1) and (C.2) are applied to the rotated unit vector sets, Ω^R and Λ^R, obtained by the following linear operations:

$$\Omega^R(S_k, \mathbf{x}_0) \equiv \{\hat{\mathbf{h}}_i^R \mid \hat{\mathbf{h}}_i^R = Q^{90}\hat{\mathbf{h}}_i, \ \forall \hat{\mathbf{h}}_i \in \Omega(S_k, \mathbf{x}_0)\} \tag{C.4}$$

$$\Omega^R(S_k, \mathbf{y}_0') \equiv \{\hat{\mathbf{h}}_i^R \mid \hat{\mathbf{h}}_i^R = Q^{180}\hat{\mathbf{h}}_i, \ \forall \hat{\mathbf{h}}_i \in \Omega(S_k, \mathbf{y}_0')\} \tag{C.5}$$

$$\Omega^R(S_k, \mathbf{x}_0') \equiv \{\hat{\mathbf{h}}_i^R \mid \hat{\mathbf{h}}_i^R = Q^{270}\hat{\mathbf{h}}_i, \ \forall \hat{\mathbf{h}}_i \in \Omega(S_k, \mathbf{x}_0')\} \tag{C.6}$$

And, the sets $\Lambda^R(S_k, \cdot)$ are defined by substituting Ω with Λ in the above three equations. The rotated unit vector sets are only used to determine the indices ($\jmath, \imath \in I_{S_k}$) of the unit vectors generating a k-coverage cone (as indicated by Proposition 3.1.2). Once the indices are determined, the original unit vectors ($\hat{\mathbf{h}}_\jmath$ and $\hat{\mathbf{l}}_\imath$) generate the actual cone $K_k(S_k, \cdot)$ and are used in all subsequent operations (Section 3.1.2).

Appendix D

Proof of Equation (3.14)

Consider the CPA triangle formed by joining the y-intercept $\mathbf{y}_0$, the CPA point, and the i^{th} sensor position $\mathbf{x}_i = [x_i \quad y_i]^T$. This is always a right triangle, and the side opposite to the right angle (located at the CPA point) is the relative-position vector $\mathbf{v}_i = (\mathbf{x}_i - \mathbf{y}_0)$, with $\mathbf{y}_0 = [0 \quad b_y]^T$. Let $\mathbf{z} = [-b_y/a_y \quad -b_y]^T$ be a vector parallel to a track $\mathcal{R}_\alpha(b_y)$ that is detected by the i^{th} sensor. Then, the angle ϕ_i that is opposite to the right angle can be obtained from the following dot product,

$$\mathbf{v}_i \cdot \mathbf{z} = \|\mathbf{v}_i\|\|\mathbf{z}\| \cos\phi_i = \frac{-b_y x_i}{a_y} - b_y(y_i - b_y) \tag{D.1}$$

and the distance between the CPA point and the sensor is given by

$$d_i = \|\mathbf{v}_i\| \sin\phi_i \tag{D.2}$$

The angle ϕ_i is eliminated by dividing (D.2) by (D.1),

$$\frac{d_i}{-b_y(\frac{x_i}{a_y} + y_i - b_y)} = \frac{\tan\phi_i}{\|z\|} = \frac{\tan\phi_i}{|\frac{b_y}{a_y}|\sqrt{a_y^2 + 1}} \tag{D.3}$$

and by using the trigonometric identity,

$$\tan(\phi_i + \alpha) = \frac{\tan\phi_i + \tan\alpha}{1 - \tan\phi_i \tan\alpha} = \frac{(y_i - b_y)}{x_i} \tag{D.4}$$

Then, since $a_y = \tan\alpha$, an equation for $\tan\phi_i$ is found solely with respect to the track parameters a_y and b_y:

$$\tan\theta_i = \frac{(y_i - b_y - a_y x_i)}{(x_i + a_y y_i - a_y b_y)} \tag{D.5}$$

Hence, by combining (D.5) with (D.3) and simplifying the result, an equation can be obtained expressing the CPA distance in terms of the track parameters,

$$d_i = \left| \frac{(b_y + a_y x_i - y_i)}{\sqrt{a_y^2 + 1}} \right| \leq r_i$$

since the CPA must occur within the sensor range in order for the track to be detected.

Appendix E

Proof of Theorem 3.1.3

We seek a measure μ on the set of tracks $\mathcal{K}_k(S, \mathbf{y}_0)$ given by (3.9). Since $\mathcal{K}_k(S, \mathbf{y}_0)$ is the union of m sets that may or may not be disjoint, we apply the principle of inclusion-exclusion [21, 62]

$$\begin{aligned}
\mu(\mathcal{K}_k(S, \mathbf{y}_0)) &= \mu\left(\bigcup_{j=1}^{m} K_k(S_k^j, \mathbf{y}_0)\right) \qquad \text{(E.1)} \\
&= \sum_{j=1}^{m}(-1)^{j+1} \sum_{1 \le i_1 < \ldots < i_j \le m} \mu(K_k(S_k^{i_1}, \mathbf{y}_0) \cap \ldots \cap K_k(S_k^{i_j}, \mathbf{y}_0))
\end{aligned}$$

where,

$$m = \left(\begin{array}{c} n \\ k \end{array}\right) = \frac{n!}{(n-k)!\, k!}, \quad \text{and} \quad \sum_{1 \le i_1 < \ldots < i_j \le m}$$

is a sum over all the $[m!/(m-j)!\, j!]$ distinct integer j-tuples $(i_1, \ldots, i_j)$ satisfying $1 \le i_1 < \ldots < i_j \le m$. Also, $\mu(\,\cdot\,)$ denotes a measure on the set. Since the right-hand side of (E.1) is an intersection of cones, it also is a cone on which we can impose the Lebesgue measure μ.

Now, consider the intersection of cones $K_k(S_k^{i_1}, \mathbf{y}_0) \cap \ldots \cap K_k(S_k^{i_j}, \mathbf{y}_0)$ inside the inner summation in (E.1). Where, $S_k^{i_l}$ denotes the i_l^{th} k-subset of S, i_l is a positive integer between 1 and $i_j \le m$, and m is the total number of k-subsets in S. By the properties of cones, this intersection is also a cone. Also, this intersection is the set of tracks through y_0 that intersect all sensors in the set $S_p = \{S_k^{i_1} \cup \ldots \cup S_k^{i_j}\}$. Based on the properties of k-subsets, this set must contain $k \le p \le n$ elements of S and, thus, is a p-subset of S. Based on the properties of k-coverage cones

(Proposition 3.1.2), the set of tracks intersecting all p sensors in S_p (also referred to as line transversals of S_p) through $\mathbf{y}_0$ can be represented by the p-coverage cone $K_p(S_p, \mathbf{y}_0) = K_p(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, \mathbf{y}_0)$. Therefore, (E.1) can be written as,

$$\mu(\mathcal{K}_k(S, \mathbf{y}_0)) = \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} \mu(K_p(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, \mathbf{y}_0)), \tag{E.2}$$

where p is the number of elements in the union of j k-subsets of S. Finally, since a Lebesgue measure on a k-coverage cone is its opening angle, a Lebesgue measure on $\mathcal{K}_k(S, \mathbf{y}_0)$ is

$$\mathcal{T}_{\mathbf{y}_0}^k = \mu(\mathcal{K}_k(S, \mathbf{y}_0)) = \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} \psi(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, \mathbf{y}_0) \tag{E.3}$$

The opening angles in the above summation are given by (3.5)-(3.6) and, thus, $\mathcal{T}_{\mathbf{y}_0}^k$ is a function of the sensors positions $X_S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $R_S = \{r_1, \ldots, r_n\}$ as $C_i = C(\mathbf{x}_i, r_i)$.

Appendix F

Opening Angles Equations

Let $\psi = \psi(S_k, \mathbf{y}_0)$ denote the opening angle of the k-coverage cone $K_k(S_k, \mathbf{y}_0)$ for $\forall k$, $1 \leq k \leq n$ and $\mathbf{y}_0 \equiv [0 \quad b_y]^T$. Then, according to Section 3.1.1, the cone is finitely generated by two unit vectors $\hat{\mathbf{l}}_\imath$ and $\hat{\mathbf{h}}_\jmath$ obtained from Λ and Ω, such that $\imath, \jmath \in I_{S_k}$ and $\hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{l}}_i \in \Lambda(S_k, \mathbf{y}_0)$ and $\hat{\mathbf{h}}_\jmath \preceq \hat{\mathbf{h}}_i \in \Omega(S_k, \mathbf{y}_0)$ for $\forall i \in I_{S_k}$ (as shown in Appendix C). Letting $\imath$ and $\jmath$ denote the indices of these unit vectors and using (3.17), the opening angle can be be written explicitly as a function of X_S:

$$\begin{cases} \psi = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \\ \det(M_{\imath\jmath}) = \frac{1}{w_\imath^2 w_\jmath^2}\{[x_\imath q_\imath + (y_\imath - b_y)r_\imath][x_\jmath r_\jmath + (y_\jmath - b_y)q_\jmath] \\ \qquad - [x_\jmath q_\jmath - (y_\jmath - b_y)r_\jmath][(y_\imath - b_y)q_\imath - x_\imath r_\imath]\} \\ w_i \equiv \|\mathbf{v}_i(\mathbf{y}_0)\| = \sqrt{x_i^2 + (y_i - b_y)^2}, \quad q_i \equiv \sqrt{w_i^2 - r_i^2}, \quad i = \imath, \jmath \end{cases} \tag{F.1}$$

Where, $X_S = \{\mathbf{x}_i \mid i \in I_S\}$, $I_{S_k} \subset I_S$, and $\mathbf{x}_i \equiv [x_i \quad y_i]^T$ for $\forall i$.

The opening angles of the k-coverage cones defined with respect to the other axes are obtained by the redefining the relative position vector $\mathbf{v}_i$. The opening angle of k-coverage cones with intercept $\mathbf{x}_0 = [b_x \quad 0]^T$ on the x-axis is given by:

$$\begin{cases} \zeta = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \\ \det(M_{\imath\jmath}) = \frac{1}{w_\imath^2 w_\jmath^2}\{[(x_\jmath - b_x)q_\jmath + y_\jmath r_\jmath][(x_\imath - b_x)r_\imath + y_\imath q_\imath] \\ \qquad - [(x_\imath - b_x)q_\imath - y_\imath r_\imath][y_\jmath q_\jmath - (x_\jmath - b_x)r_\jmath]\} \\ w_i \equiv \|\mathbf{v}_i(\mathbf{x}_0)\| = \sqrt{(x_i - b_x)^2 + y_i^2}, \quad q_i \equiv \sqrt{w_i^2 - r_i^2}, \quad i = \imath, \jmath \end{cases} \tag{F.2}$$

The opening angles of the k-coverage cones with intercepts on the remaining axes, x'

and y', are given by,

$$\begin{cases} \xi = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \\ \det(M_{\imath\jmath}) = \frac{1}{w_\imath^2 w_\jmath^2}\{[(L_1 - x_\imath)q_\imath + (L_2 - y_\imath - b_{y'})r_\imath] \cdot [(L_1 - x_\jmath)r_\jmath - (L_2 - y_\jmath - b_{y'})q_\jmath] \\ -[(L_1 - x_\jmath)q_\jmath - (L_2 - y_\jmath - b_{y'})r_\jmath][(x_\imath - L_1)r_\imath + (L_2 - y_\imath - b_{y'})q_\imath]\} \\ w_i \equiv \|\mathbf{v}_i(\mathbf{y}_0')\| = \sqrt{(L_1 - x_i)^2 + (y_i + b_{y'})^2}, \quad q_i \equiv \sqrt{w_i^2 - r_i^2}, \quad i = \imath, \jmath \end{cases} \tag{F.3}$$

and,

$$\begin{cases} \rho = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \\ \det(M_{\imath\jmath}) = \frac{1}{w_\imath^2 w_\jmath^2}\{[(L_2 - y_\imath)q_\imath + (L_1 - x_\imath - b_{x'})r_\imath] \cdot [(L_2 - y_\jmath)r_\jmath + (L_1 - x_\jmath - b_{x'})q_\jmath] \\ -[(L_2 - y_\jmath)q_\jmath - (L_1 - x_\jmath - b_{x'})r_\jmath][-(L_2 - y_\imath)r_\imath + (L_1 - x_\imath - b_{x'})q_\imath]\}, \\ w_i \equiv \|\mathbf{v}_i(\mathbf{x}_0')\| = \sqrt{(L_1 - b_{x'} - x_i)^2 + y_i^2}, \quad q_i \equiv \sqrt{w_i^2 - r_i^2}, \quad i = \imath, \jmath \end{cases} \tag{F.4}$$

respectively. Where, the indices $\imath$ and $\jmath$ are always determined by ordering unit vectors according to Appendix C.

Appendix G

Total Track-Coverage

Consider the union of k-coverage cones $\mathcal{K}_k(S, \mathbf{y}_0)$, representing the set of tracks through $\mathbf{y}_0$ that are detected by at least k sensors in S, and given by (3.9). Since all cones in this union are generated by objects in $\mathbb{R}^2_+$, $\mathcal{K}_k(S, \mathbf{y}_0)$ only contains cones in the first and fourth quadrant of the xy-reference frame, and its measure $\mathcal{T}^k_{\mathbf{y}_0}$ in (3.19) is bounded from above by π. This upper bound corresponds to the case in which $\mathcal{K}_k(S, \mathbf{y}_0)$ is a non-convex cone and is a half-space with $x \geq 0$. By induction, the measures $\mathcal{T}^k_{\mathbf{x}_0}$, $\mathcal{T}^k_{\mathbf{y}'_0}$, and $\mathcal{T}^k_{\mathbf{x}'_0}$ are all bounded from above by π, for any value of the intercept. Thus, the upper bound on the track-coverage function is obtained by substituting $\mathcal{T}^k_{\mathbf{y}^\ell_0} = \mathcal{T}^k_{\mathbf{x}^\ell_0} = \mathcal{T}^k_{\mathbf{y}'^\ell_0} = \mathcal{T}^k_{\mathbf{x}'^\ell_0} = \pi$ for any value of ℓ in (3.23):

$$\begin{aligned} \mathcal{T}^{\max}_{\mathcal{A}} &= \frac{1}{2}\sum_{\ell=0}^{N_2}(\pi+\pi) + \frac{1}{2}\sum_{\ell=0}^{N_1}(\pi+\pi) = (N_2+1)\pi + (N_1+1)\pi \\ &= \left(\frac{L_1+L_2}{\delta b} + 2\right)\pi \end{aligned} \tag{G.1}$$

Appendix H

Probability of Detection of Unobserved Tracks

Let the ray $\mathcal{R}_\alpha(b_y^\ell)$ denote a track with intercept value $b_y^\ell \in \mathcal{I}_y \equiv [0, \quad L_2]$, and let D_k denote a cooperative detection event, such that $D_k = 1$ if a target is detected by at least k sensors, and $D_k = 0$ otherwise. Also, let $\Pr(b_y^\ell)$ denote the prior probability that a target enters $\mathcal{A}$ at $\mathbf{y}_0^\ell = [0 \quad b_y^\ell]^T$. Then, the probability that a target enters $\mathcal{A}$ at $\mathbf{y}_0^\ell$, and is detected by k sensors is,

$$\Pr\{\mathcal{R}_\alpha(b_y^\ell), D_k = 1\} = \Pr(b_y) \cdot \Pr(\mathcal{K}_k(S, \mathbf{y}_0^\ell)) \tag{H.1}$$

where, $\Pr(\mathcal{K}_k(S, \mathbf{y}_0^\ell))$ denotes the probability that the track lies inside the set $\mathcal{K}_k(S, \mathbf{y}_0^\ell)$. Now, assuming that all y-intercepts are equally likely, $\Pr(b_y^\ell) = \delta b/(L_2 + \delta b)$. Also, assuming that all directions $\alpha \in (-\pi/2, \quad +\pi/2)$ are equally likely, (H.1) can be written as,

$$\Pr\{\mathcal{R}_\alpha(b_y^\ell), D_k = 1\} = \frac{\delta b}{(L_2 + \delta b)} \cdot \frac{1}{\pi} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \le i_1 < \ldots < i_j \le m} \psi({S_p}^{i_{1,j}}, \mathbf{y}_0^\ell) \tag{H.2}$$

where $\psi({S_p}^{i_{1,j}}, \mathbf{y}_0^\ell)$ is the opening angle of the p-coverage cone of the p-subset of S that is defined as the union $\{S_k^{i_1} \cup \ldots \cup S_k^{i_j}\}$, for every tuple $(i_1, \ldots, i_j)$ in the inner summation (as shown in Appendix E).

Since sets $\mathcal{K}_k(S, \mathbf{y}_0^\ell)$ with different values of $\mathbf{y}_0^\ell$ are always disjoint, the probability that a target enters $\mathcal{A}$ through $\mathcal{I}_y$ and is detected by k sensors is:

$$\begin{aligned}\Pr\{\mathcal{R}\cap\mathcal{I}_y \neq \emptyset, D_k = 1\} &= \sum_{\ell=0}^{N_2} \Pr\{\mathcal{R}_\alpha(b_y^\ell), D_k = 1\} \\ &= \frac{\delta b}{\pi(L_2+\delta b)} \cdot \sum_{\ell=0}^{N_2}\sum_{j=1}^{m}(-1)^{j+1} \sum_{1\leq i_1<\ldots<i_j\leq m} \psi({S_p}^{i_{1,j}}, \mathbf{y}_0^\ell)\end{aligned} \tag{H.3}$$

Similarly, the probability that $D_k = 1$ and the target intersects the sides $\mathcal{I}_x$, $\mathcal{I}_{y'}$, and $\mathcal{I}_{x'}$, can be obtained in terms of the opening angles. Then, the probability that a target traverses $\mathcal{A}$ and $D_k = 1$ is obtained by considering the probability of the union of intersecting sets [111]. The set of tracks that traverse $\mathcal{A}$ and are detected by at least k sensors is given by the union $\mathcal{K}_k(S,\mathcal{A})$ in (3.12). Since every track in this union intersects two sides of $\mathcal{A}$ and belongs to two k-coverage cones, the intersection of these cones is equal to its complement and, thus:

$$\begin{aligned} Pr_{\mathcal{A}}^k(X_S) &\equiv \Pr\{\mathcal{R}\cap\mathcal{A}\neq\emptyset, D_k = 1\} \qquad \text{(H.4)} \\ &= \frac{\delta b}{2\pi(L_2+\delta b)} \sum_{\ell=0}^{N_2}\sum_{j=1}^{m}(-1)^{j+1} \sum_{1\leq i_1<\ldots<i_j\leq m} [\psi({S_p}^{i_{1,j}}, \mathbf{y}_0^\ell) + \xi({S_p}^{i_{1,j}}, \mathbf{y}_0'^\ell)] \\ &+ \frac{\delta b}{2\pi(L_1+\delta b)} \sum_{\ell=0}^{N_1}\sum_{j=1}^{m}(-1)^{j+1} \sum_{1\leq i_1<\ldots<i_j\leq m} [\zeta({S_p}^{i_{1,j}}, \mathbf{y}_0^\ell) + \rho({S_p}^{i_{1,j}}, \mathbf{y}_0'^\ell)]\end{aligned}$$

It can be easily shown by substituting the same upper bounds used in Appendix G in the above equation that when the sensor network provides total track coverage $Pr_{\mathcal{A}}^k = 1$.

Bibliography

[1] C. Cai and S. Ferrari. Bayesian network modeling of acoustic sensor measurements. In *Proceedings of the IEEE Sensors Conference*, Atlanta, GA, 2007.

[2] J. E. Goodman, R. Pollack, and R. Wenger. Geometric transversal theory. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 163–198. Springer Verlag, 1991.

[3] T. A. Wettergren, R. L. Streit, and J. R. Short. Tracking with distributed sets of proximity sensors using geometric invariants. *IEEE Transactions on Aerospace and Electronic Systems*, 40(4):1366–1374, October 2004.

[4] D. S. Hammond. Sonobuoy field drift prediction. *Technical Report: Naval Air Warfare Center Aircraft Division*, 13 January 2005.

[5] A. K. Liu, Y. Zhao, and M.-K. Hsu. Ocean surface drift revealed by synthetic aperture radar images. *EOS*, 87(24):233–239, 13 June 2006.

[6] Alaska Native Technologies. Acoustic Glider Eyak 02, available at http://www.alaskanativetech.com/gallery/, 2006.

[7] T. G. Bell. Sonar and submarine detection. Report 545, United States Navy Underwater Sound Laboratory, 1962.

[8] V. Isler, S. Khanna, J. Spletzer, and C. J. Taylor. Target tracking with distributed sensors: The focus of attention problem. *Computer Vision and Image Understanding*, 100:225–247, 2005.

[9] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho. Grid coverage of surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51(12):1448–1453, 2002.

[10] Jing Ai and Alhussein A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11:21–41, 2006.

[11] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing*, 4(2):84–92, January-February 2005.

[12] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Scroff. A minimum cost heterogenous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing*, 4(1):4–15, 2005.

[13] C.-F. Huang and Y.-C. Tseng. The coverage problem in wireless sensor network. *Mobile Networks and Applications*, 10:519–528, 2005.

[14] N. Heo and P. K. Varshney. Energy-efficient deployment of intelligent mobile sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans*, 35(1):78–92, January 2005.

[15] A. Howard, M. J. Mataric, and G. S. Sukhatme. An incremental self-deployment alfgorithm for mobile sensor networks. *Autonomous Robots*, 13(2):113–126, 2002.

[16] J. A. George, J. M. George, and B. W. Lamer. Packing different-sized circles into a rectangular container. *European Journal of Operational Research*, 84:693–712, 1995.

[17] W. Q. Huang, Y. Li, and C. M. Li. Greedy aglrotihms for packing unequal circles into a rectangular container. *Journal of the Operational Research Society*, 56:539–548, 2005.

[18] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, 1987.

[19] T. Shermer. Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399, 1992.

[20] J. Urritia. Art gallery and illumination problems. In J.R. Sack and J. Urritia, editors, *Handbook on Computational Geometry*, pages 387–434. Elsevier Science Publishers, 1992.

[21] S.M. Ross. *Introduction to Stochastic Dynamic Programming.* Academic Press, 1983.

[22] L. D. Stone. *Theory of Optimal Search.* Academic Press, 1975.

[23] R. Ahlswede and I. Wenger. *Search Problems.* John Wiley & Sons, 1987.

[24] S. S. Blackman. Multitarget tracking with an agule beam radar. In Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications, Vol. II.* Artech House, 1992.

[25] B.O. Koopman. *Search and Screening: General Principles with Historical Applicaitons*. Pergamon Press, 1980.

[26] P. M. Morse and G. E. Kimball. *Methods of Operations Research*. Dover Publications, 2003.

[27] J. M. Dobbie. Search theory: a sequential approach. *Naval Research Logistics Quarterly*, 4:323–334, 1963.

[28] T. J. Stewart. A search for a moving target when the searcher motion is restricted. *Computers and Operations Research*, 6:129–140, 1979.

[29] D.A. Castañon. Optimal search strategies in dynamic hypothesis testing. *IEEE transactions on systems, man, and cybernetics*, 25(7):1130–1138, 1995.

[30] L. Yao, W. A. Sethares, and D. C. Kammer. Sensor placement for on-orbit modal classification via a genetic algorithm. *AIAA Journal*, 31:1167–1169, 1993.

[31] K. Worden and A. P. Burrows. Optimal sensor placement for fault detection. *Engineering Structures*, 23:885–901, 2001.

[32] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[33] Y. Bar-Shalom. *Multitarget-Multisensor Tracking: Applications and Advances, Vol. III*. Artech House, 2000.

[34] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction*. J. Wiley and Sons, 2001.

[35] C. L. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 22(3):302–312, 1977.

[36] S. S. Blackman. *Multi-Target Tracking with Radar Applications*. Artech House, 1986.

[37] A. B. Poore and N. Rijavec. A numerical study of some data association problems arising in multitarget tracking. *Computational Optimization and Applications*, 3(1):27–57, 1994.

[38] S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias. Optimal distributed decision fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 25(5):761–765, 1989.

[39] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors: Part i- fundamentals. *Proceedins IEEE*, 85(1):54–63, 1997.

[40] R. Blum, S. A. Kassam, and H. V. Poor. Distributed detection with multiple sensors: Part ii- advanced topics. *IEEE Proceedings*, 85(1):64–79, 1997.

[41] M. Xiang and J. Zhao. On the performance of distributed Neyman-Pearson detection systems. *IEEE Transactions in Systems, Man, and Cybernetics, Part A*, 31(1):78–83, January 2001.

[42] Q. Yan and R. S. Blum. Distributed signal detection under the Neyman-Pearson criterion. *IEEE Transactions on Information Theory*, 47(4):1368–1377, May 2001.

[43] Q. Yan and R. S. Blum. On some unresolved issues in finding optimum distributed detection schemes. *IEEE Transactions on Information Theory*, 48(12):3280–3288, May 2000.

[44] N. S. V. Rao. Distributed decision fusion using empirical estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1106–1114, 1997.

[45] J. Liu, M. Chu, J. Liu, and F. Zhao. Distibuted state representation for tracking problems in sensor networks. *Proceedings IPSN04*, pages 234–242, 2004.

[46] N. Ansari, J-G. Chen, and Y-Z. Zhang. Adaptive decision fusion for unequiprobable sources. *IEEE Proceedings of Radar, Sonar, and Navigation*, 144(3):105–111, June 1997.

[47] R.E. Kalman. A new approach to linear filtering and prediction theory. *Journal of Basic Engineering*, 1960.

[48] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. *Computer Journal*, 47(4):448–60, 2004.

[49] M. Marengoni, B. A. Draper, A. Hanson, and R. A. Sitaraman. System to place observers on a polyhedral terrain in polynomial time. *Image and Vision Computing*, 18:773–780, 1996.

[50] E. Helly. über mengen konvexer körper mit gemeinschaftlichen punkten. *Jber. Deutsch. Math. verein*, 32:175–176, 1923.

[51] N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. Assoc. Comput. Mach*, 31:114–127, 1984.

[52] E. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood. Stabbing line segments. *BIT*, 22:274–281, 1982.

[53] M. Atallah and C. Bajaj. Efficient algorithms for common transversals. *Information Processing Letters*, 25:87–91, 1987.

[54] E. Edelsbrunner. Finding transversals for sets of simple geometric figures. *Theoretical Computer Science*, 35:55–69, 1985.

[55] E. Edelsbrunner, L. J. Guibas, and M. Sharir. The upper envelope of piecewise linear functions: Algorithms and applications. *Discrete Computational Geometry*, 4:311–336, 1989.

[56] D. Avis, J. M. Robert, and R. Wenger. Lower bounds for line stabbing. *Information Processing Letters*, 33:59–62, 1989.

[57] P. Egyed and R. Wenger. Stabbing pairwise disjoint translates in linear time. *Proceedings of the Fifth Annual ACM Symposium on Computational Geometry*, pages 364–369, 1989.

[58] D. P. Bertsekas. *Convex Analysis and Optimization.* Athena Scientific, Belmont, MA, 2003.

[59] H. F. Davis and A. D. Snider. *Vector Analysis.* William C Brown Publisher, 1987.

[60] S. Skiena. Generating k-subsets. In *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, pages 44–46. Addison-Wesley, Reading, MA, 1990.

[61] S. Ferrari. Track coverage in sensor networks. *Proceedings of the ACC 2006*, pages 1–10, 2006.

[62] J. Pitman. *Probability.* Springer-Verlag, 1993.

[63] H. T. Croft, K. J. Falconer, and R. K. Guy. *Unsolved Problems in Geometry.* Springer-Verlag, New York, NY, 1991.

[64] E. W. Weisstein. Circular segment, 2003.

[65] G. Gomez, W.S. Koon, M.W. Lo, J.E. Marsden, J. Masdemont, and S.D. Ross. Invariant manifolds, the spatial three-body problem and space mission design. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Meeting*, Quebec City, Quebec, Canada, 2001.

[66] D. E. Kirk. *Optimal Control Theory: An Introduction.* Prentice-Hall, 1970.

[67] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic, London, 1981.

[68] Mathworks. *Matlab Optimization Toolbox.* [Online]. Avaliable: http://www.mathworks.com, 2004. function: fmincon.

[69] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. Saluja. Sensor deployment strategy for target detection. The First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), September 2002.

[70] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. Saluja. Sensor deployment for detection of targets traversing a region. *Mobile Networks and Applications*, 8:453–461, August 2003.

[71] H. Freeland, P. Rhines, and T. Rossby. Eddy resolution versus eddydiffusion in a double gyre GCM. part II: Mixing of passive tracers. *Journal of Physical Oceanography*, 24:387–402, 1975.

[72] W. Krauss and C. Boning. Lagrangian properties of eddy fields in the Northern Atlantic as deduced from satellite-tracked buoys. *Journal of Marine Research*, 45:259–291, 1987.

[73] V. Rupola, B. Hua, A. Provenzale, and V. Artale. Lagrangian velocity spectra at 700 m in the western North Atlantic. *Journal of Physical Oceanography*, 26:1591–1607, 1996.

[74] R. E. Davis. Lagrangian ocean studies. *Annual Reviews of Fluid Mechanics*, 23:43–64, 1991.

[75] W. B. Wilson. Numerical modeling of drifting buoys and its relevence to lagrangian tracking. *Engineering Structures*, 23:885–901, 2001.

[76] S. E. Pazan. Intercomparison of drogued and undrogued drift buoys. *IEEE*, 3(1):864–872, 1996.

[77] COOL. *Rutgers University*. [Online]. Avaliable: http://marine.rutgers.edu, 2002.

[78] Mathworks. *Matlab*. [Online]. Avaliable: http://www.mathworks.com, 2004.

[79] F. D. Foresee and M. T. Hagan. Gauss-Newton approximation to Bayesian regularization. In *Proceedings of the 1997 International Joint Conference on Neural Networks*, pages 1930–1935, 1997.

[80] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[81] R. J. Urick. *Principles of Underwater Sound*. McGraw-Hill Book Company, 3 edition, 1983.

[82] M. D. Collins. A split-step Pade solution for the parabolic equation method. *The Journal of the Acoustical Society of America*, 93:1736–1742, 1993.

[83] M. D. Collins. Generalization of the split-step Pade solution. *The Journal of the Acoustical Society of America*, 96:382–385, 1994.

[84] F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, 2001.

[85] D. Heckerman. A Bayesian approach to learning causal networks. *Technical Report MSR-TR-95-04*, pages 1–23, May 1995.

[86] H. Roarty, J. Kohut, and S. Glenn. Intercomparison of an ADCP, ADP, standard and long-range HF radar: influence of horizontal and vertical shear. In *Proceedings of the IEEE/OES Seventh Working Conference on Current Measurement Technology*, pages 75–78, March 13-15 2003.

[87] R. F. Stengel and L. R. Ray. Stochastic robustness of linear time-invariabt control systems. *IEEE Transactions on Automatic Control*, 36(1):82–87, 1991.

[88] Mathworks. *Matlab Optimization Toolbox*. [Online]. Avaliable: http://www.mathworks.com, 2004. function: ode45.

[89] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3, 2005.

[90] P. Ögren, E. Fiorelli, and N. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Autonomous Control*, 49(8):1292–1302, 2004.

[91] J. Cortés, S. Martínez, T. Karataş, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[92] P. W. J. van de Ven, C. Flanagan, and D. Toal. Neural network control of underwater vehicles. *Detection and Remediation Technologies for Mines and Minelike Targets VIII, Proc. of the SPIE*, 18(5):533–547, 2005.

[93] I. Kamon and E. Rivlin. Sensory based motion planning with global proofs. In *Proceedings of IEEE/RSJ International Conference of Intelligent Robots and System Human Interaction Cooperation*, pages 435–440, Durham, New Hampshire, 1995.

[94] T. Inanc, S. C. Shadden, and J. E. Marsden. Optimal trajectory generation in ocean flows. In *Proceedings of the 2005 American Control Conference*, Portland, OR.

[95] A. Alvarez, A. Caiti, and R. Onken. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering*, 29(2):418–429, 2004.

[96] I. Spangelo and O. Egeland. Trajectory planning and collision avoidance for underwater vehicles using optimal control. *IEEE Journal of Oceanic Engineering*, 19(4):502–511, 1994.

[97] L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mischenko. *The Mathematical Theory of Optimal Processes*. Interscience, New York, 1962.

[98] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, NJ, 1962.

[99] R. F. Stengel. *Optimal Control and Estimation*. Dover Publications, Inc., 1986.

[100] M. Athans and P. L. Falb. *Optimal Control*. McGraw-Hill, New York, 1966.

[101] S. Jenkins, D. Humphreys, J. Sherman, J. Osse, C. Jones, N. Leonard, J. Graver, and R. Bachmayer. Alternatives for enhancement of transport economy iin underwater gliders. *Proceedings of Oceans 2003*, 2:948– 950, September 2003.

[102] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[103] D. A. Benson. *A Gauss Pseudospectral Transcription for Optimal Control*. PhD thesis, MIT, 2004.

[104] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao. Direct trajectory optimization and costate estimation via an orthogonal collocation method. *Journal of Guidance, Control, and Dynamics*, 29(6):1435–1440, 2006.

[105] TOMLAB. *TOMLAB*. [Online]. Avaliable: http://www.tomopt.com, 2007. GPOCS User Manual, Author: Anil Rao.

[106] P. Davis. *Interpolation and Approximation*. Dover Publications, 1975.

[107] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2007.

[108] V. Isler, S. Khanna, and K. Daniilidis. Sampling based sensor-network deployment. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 100:1780–1785, 2004.

[109] L. Benyuan and D. Towsley. A study of the coverage of large-scale sensor networks. *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 475–83, 2004.

[110] N. Leonard and A. Robinson. Autonomous Ocean Sampling Network II, available at http://www.princeton.edu/ dcsl/aosn/, 2006.

[111] R. V. Hogg, J. W. McKean, and A. T. Craig. *Introduction to Mathematical Statistics*. Prentice Hall, Upper Saddle River, NJ, 2005.

Biography

Kelli Baumgartner received a B.S. degree (2003) in Aerospace Engineering with minors in Mathematics, Physics, and International Studies from Embry-Riddle Aeronautical University (Prescott, AZ). She received M.S. (2005) and Ph.D. (2007) degrees in Mechanical Engineering and Materials Science from Duke University (Durham, NC). In addition to her graduate studies at Duke, her research experience includes a summer internship (2001) at Fermi National Accelerator Laboratory (Batavia, IL) for the MiniBooster Neutrino Experiment. She designed and conducted a Photo-Multiplier Tube (PMT) specialty test that studied the average quantum efficiency over the face of PMT for the MiniBooster Neutrino Experiment. Her results are summarized in the technical note, "*A PhotoMultiplier Tube Specialty Test: Quantum Efficiency as a Function of Angle*". Her study abroad experience includes EPF Ecole d'Ingénieurs in Paris, FR (2000) and participating in the International Space Workshop in Moscow, RU (2002). Her teaching assistant experience includes two semesters of the undergraduate course "Introduction to Mechanical Systems Engineering", one semester of the undergraduate course "Introduction to Statistics", and two semesters of the graduate course "Intelligent Systems". In Summer 2004, she was the instructor of the course "Introduction to Aerospace Engineering" for the Talented Identification Program.

Dr. Baumgartner is a member of IEEE, AIAA, ASME, and a Ronald E. McNair Scholar. To date, she has co-authored three accepted journal papers, two conference papers, and currently has three journal papers under review. She won the "Best Paper in Session Award" at the American Controls Conference 2007. Currently, she is a private pilot, where she holds more than a 100+ flight hours and an Instrument Flight Rating. She is employed as an aerospace engineer in launch vehicle flight design and control at Analex Corporation, a subsidiary of QinetiQ North America.