Sensor Management for Multistatics

David W. Krout
Applied Physics Laboratory
University of Washington
1013 NE 40th St.
Seattle, WA 98105-6698
Email: dkrout@apl.washington.edu

Thomas Powers
Dept of Electrical Engineering
University of Washington
Campus Box 352500
Seattle, WA 98195-2500
Email: tcpowers@uw.edu

Abstract—This paper focuses on sensor management for distributed sensor fields in deep water. This paper will investigate the sensor placement problem for the barrier scenario. Field performance is often based solely on sensor coverage or probability of detection evaluations, which provide only a fleeting snapshot of how the field is performing at a particular time. This paper utilizes an efficient time dependent approach to evaluating the performance of distributed sensor fields. Given a priori target probability distributions and sensor measurements distributed in time and space, the approach directly scores entire sensor fields based on their ability to detect threats over time. This formulation is used as an optimization metric for sensor management.

I. INTRODUCTION

This paper is an extension of previous work, where the marine security and defense scenario for distributed sensor fields was examined [1]. This new effort incorporates a greedy optimization algorithm into the threat modeling previously presented in [1] and continues to explore the deep water scenario. This paper will address sensor placement for the barrier problem in deep water. The barrier problem is defined as preventing targets from passing through an area of interest. This paper will also begin to explore the ping sequencing problem, specifically what buoys can transmit simultaneously.

The core of every optimization algorithm is a cost function or metric. In previous work, an accurate and efficient Fokker-Planck (FP) approach was pioneered as an alternative to Monte Carlo simulation (MCS) for modeling temporal and spatial threat probability density function (PDF) evolutions [2], [3]. This approach was applied to managing and optimizing multistatic sonar distributed sensor fields in shallow water. In [4], the FP formulation was refined to advance 3-dimensional PDFs of arbitrary shape with drift (average motion) and diffusion (spreading) in the presence of moving sensor fields. The Probability of Target Presence ($P_T$) for an ocean area-of-interest was represented in three dimensions using voxels (volumetric pixels). The $P_T$ was computed for each voxel from $a$ priori target distributions. $P_T$ values were updated for target motion using an efficient FP equation digital filter method [5].

Target Probability of Detection ($P_D$) was computed for each sensor at each voxel, and Bayes’ rule was used to update the $P_T$ given sensor measurements and their associated $P_D$'s.

This approach allows for a variety of operational goals to be evaluated, including sensor deployment effectiveness and sensor field leakage, through simulation, in very reasonable computational time. The simulation in this paper assumes a fixed ping interval, however this does not have to be the case. For large sensor fields in deep water, multiple sources can be pinged at one time. This will be discussed later in this paper and will be an emphasis in future work.

A similar approach to evaluating field performance is presented by Incze [6], which uses a Bayesian method and Monte Carlo simulations to evaluate field performance. Monte Carlo simulations are formed by sending many targets with random velocities and bearings through a field of sensors and collecting statistics on detections. This type of Monte Carlo analysis is computationally demanding. One of the motivations behind the work in this paper is to present an alternate method to Monte Carlo that is less computationally demanding.

Section 2 will summarize the ocean model description. Section 3 will discuss the simulation and application to the barrier problem. Section 4 will summarize the paper and discuss areas of continued work.

II. MODEL DESCRIPTION

The management of distributed sensor fields in an ocean environment requires a model that can keep track of position and sensor performance. Directing sensor placement and activation for efficient target detection and tracking requires an updated prediction of potential target locations. In this work the probability model is implemented on a two-dimensional grid. Multiple layers of the two dimensional grid can be used if there is more than one threat model in the scenario. There is only one layer used in the work. Ocean environmental information is included in the sensor performance predictions. Real time feedback from sensor returns based on probabilistic propagation models are used to update $P_T$ values. The basis for the propagation model was described in [3], and is briefly described below.

A. Probability of Target Presence

Here, a metric called “Probability of Target Presence,” $P_T(x, y)$ is defined. This should be interpreted as the probability that a target is present at discrete geographic coordinate $(x, y)$. $P_T$ equal to one means that there is a good chance a target exists and there is no evidence to the contrary. A $P_T$ of zero indicates that a target does not exist at the given location due to sensor measurements or $a$ priori information.
$P_T$ equal to 0.5 would then correspond to an unknown state or equally likely target state. The search area is discretized into cells of equal size, which are assigned an *a priori* probability of target presence. This formulation accounts for the sensor $P_D$ against the target presence model, assuming that there are no detections. Using Bayes theorem, the probability of target presence given no detections is defined as

$$P_{T|ND} = \frac{P_{ND|T}P_T}{P_{ND|T}P_T + P_{ND|NT}P_{NT}},$$  

(1)

where $ND$ denotes no detection, $NT$ denotes no target present, and $T$ denotes target presence. Substituting the probability of detection and the probability of false alarm into Eq. 1 gives the following:

$$P_{T|ND} = \frac{(1 - P_D) P_T}{(1 - P_D) P_T + (1 - P_{fa}) (1 - P_T)},$$

(2)

$P_D$ is the probability that we detect a target when a target actually is present, and $P_{fa}$ is the probability that we decide a target is present when there is no actual target (false alarm).

Sonar performance predictions are typically calculated in terms of signal excess (SE) via manipulation of the sonar equation [7]. Signal excess is defined as the signal-to-noise ratio (SNR) minus a detection threshold (DT), where DT is set based on design values of $P_D$ (usually set to 0.5) and $P_{fa}$ (usually set to a very low number, like $10^{-5}$). Signal excess equal to 0 dB corresponds to the design value of $P_D$. Analytic representations are available to calculate $P_D$ for other values of SE. See [8] [9] [10] for further discussion on this topic. The SE data used in these simulations are generated by a standard acoustic model.

The *a posteriori* $P_{T|ND}$ for the given sensor coverage becomes the *a priori* value of $P_T$ for the next time step. The time step can be defined as a discrete change in time (such as every 60 seconds) or whenever there is a sensor measurement. This approach keeps track of search efforts, and can be used for optimizing future search.

B. Moving Targets

The formulation for $P_{T|ND}$ as so far discussed does not account for targets moving into an area that has already been searched or any other target motion. Moving targets are accounted for based on an approach by Kanchanavally *et al.* in [11], where a diffusion process is modeled using the Fokker-Planck equation and Brownian motion. In [3], the approach was extended to account for average target motion. The Fokker-Planck (FP) filter, derived in [3], is applied to more complex threat types in the current work. Various threats can be injected into the $P_T$ surface based on scenario requirements.

In previous work, [1], three different threats models were used (Gaussian, line, and background). The line threat is used if there is information about targets coming from a particular area. This is analogous to the barrier problem where the line threat represents targets trying to penetrate a barrier. In this work, the barrier problem will be the focus so a line threat will be used.

III. Application Description

The $P_T$ and FP filter algorithms are applied to a distributed sensor field where there is a line threat that represents targets penetrating a barrier. The goal of this work is to place sensors so that targets will be unable to penetrate through the barrier. A greedy optimization approach was used to place four colocated sensors pairs (transmitter and receiver) along a line forming a barrier. The problem was constrained to searching over a line for computational concerns and implementation. A more sophisticated approach, such as Particle Swarm Optimization [12], will be investigated in future work.

The greedy optimization approach is formulated by placing one sensor (co-located transmitter and receiver) at a time. The objective function used to evaluate the best placement is the sum of the probabilities in the $P_T$ surface. The first iteration looks for the best placement of a single sensor along a line. This sensor location minimizes the sum of the $P_T$ surface. When this location is found, it is fixed for the second iteration where the second sensor is placed. This process is repeated until minimal improvement is found (four sensors for this scenario). A typical coverage plot for a single transmission (ping) can be seen in Fig. 1, where the gray scale in the plot represents the probability of detection derived from the sensors performance calculation. This is a typical coverage plot for deep water. The Convergence Zone (CZ) ring is not a complete circle in this case due to environmental variables, most likely bottom interaction. In a perfect case, where this is no bottom interaction, the CZ ring tends to be a complete circle of high probability of detection.

The final solution for this scenario has four sensors with direct path and CZ detection regions. The sensors are drifting at 1 knot at a compass heading of 0 degrees. There is a line threat coming from the west which represents the barrier problem. The line threat has a speed of 5 knots at a heading of 45 degrees. The simulation is 24 hours long.

The first step in the greedy algorithm is to place a single sensor in the best possible position. To illustrate the simulation, four snapshots for the single sensor simulation will be presented. Figures 1 through 4 are four snapshots in time of the simulation for the the first sensor that is placed. The gray scale regions represent the Probability of Detection ($P_D$), where lighter gray is the better coverage. The red/blue area represents the $P_T$ map. Blue represents low $P_T$ ($P_T = 0$) and red represents high $P_T$ ($P_T = 1$). The simulation area is approximately 5 degrees by 6 degrees, represented by a grid of 673x673 cells.

In Fig. 1, the scenario is just starting so the line threat has not moved yet. The line threat is propagated using the FP filter and the Bayes updates process described above. As time progresses, shown in Fig. 2, the line threat moves east. As time progresses even further in Fig. 3 and Fig. 4, the line threat has moved all the way through the barrier and a large swath of the probability of target presence has been cleared.

Although the single sensor cleared a large swath of the line threat, there is still a lot of leakage. The next greedy step
Fig. 1. Probability of Target Presence \((P_T)\) Map at the first ping (assumes all sensors ping at the same time at regular intervals). The color bar shows the level of \(P_T\) from zero to one. There is one sensor and the coverage is shown in gray scale, the lighter gray the better the coverage.

is to place another co-located transmitter and receiver in the best possible position given the position already found for the first sensor. The second, third, and fourth sensor placement solutions are shown in figures 5 through 7. It should be clear from these plots that the greedy approach is suboptimal in the sense there is leakage, especially between the sensors. The placement of the fourth sensor (Fig. 7) improves the performance of the barrier by a small amount, showing diminishing returns. In future work, better optimization algorithms will be investigated to solve some of these issues as well as remove some of the constraints on sensor locations.

A. Computational Considerations

In order for this approach to be viable in a real system, the model will need to be computationally efficient enough to run at real time. The simulation for the scenario described above runs anywhere from 40x to 100x real time on a single Intel Xeon 2.8 GHz processor. Some aspects of this approach have been parallelized for further improvement above 100x real time but are not reported here. The variability between 40x and 100x real time is due to the computational load of the sensor performance prediction. For the above greedy algorithms, each iteration (placement of one sensor) solution took about 20 minutes to compute (25 objective function calls).

B. Ping Sequence Optimization

A second aspect to sensor management that was explored is ping sequence optimization. Specifically, which transmitters can be pinged at the same time without interference. Consider an arbitrary arrangement of buoys. Let the speed of sound through water be 0.8 nmi/s, and let the ping duration for a buoy be 5 seconds. Due to the propagation, there will be an interference region around a buoy when it pings. This interference is due to the propagating wave interfering with the listening windows of the other buoys in the field. This region is

Fig. 2. Probability of Target Presence \((P_T)\) Map after 8 hours and several pings. The line threat has propagated according to its velocity, and the target probability has been cleared according to the sensor coverage. There is quite a bit of target probability that has penetrated passed the first part of the sensors coverage ring.

Fig. 3. Probability of Target Presence \((P_T)\) Map after 16 hours. The sensor coverage is performing well, because the right side of the sensors coverage is clearing the remaining line threat.
between 56 and 72 nmi (assuming one CZ detection regions). Any buoy closer than 56 nmi and further than 72 nmi will not be interfered with and can therefore ping simultaneously. Given these constraints, we can construct a graph, where the nodes are the buoys and an edge between a pair of nodes signifies that they can ping simultaneously.

An example of a buoy arrangement where some of the buoys interfere can be found in figure 8. The four buoys are arranged in a diamond pattern with locations represented by black dots. The blue rings denote the coverage regions for each buoy and the red rings denote the regions where another buoy will interfere with a given buoy. In this arrangement, the buoys across from each other, i.e. the top and bottom pair and left and right pair, will interfere with each other, since the buoys in
each pair are in the red interference region of the other buoy. However, any other pair of buoys can ping simultaneously.

In order to find out the maximum number of buoys that can ping simultaneously, the largest set of nodes is picked such that all the nodes in the set are connected to every node in the set. Note that self-loops are required, since a buoy does not interfere with itself. The problem of finding the largest subset of fully connected nodes is a well known problem in computer science referred to as the largest clique in a graph [13]. Exact methods for solving this problem run in exponential time. However, if the graph meets certain conditions, i.e. if the graph is “planar” or “perfect,” finding the largest clique can be solved in polynomial time [14]. For the arrangement in figure 8, there is a four-way tie for largest clique, which are the adjacent pairs (top and left buoys, left and bottom buoys, bottom and right buoys, and right and top buoys). In a real scenario, the detection regions will not be perfect rings, so one of the pairs might have better coverage than the others. A more complicated interference pattern will emerge as the size of the buoy field is increased.

In analysis of combinatorial systems such as graphs, submodular functions often arise. Submodularity is a property that describes set functions similar to how convexity describes functions in a continuous space. For ping sequence optimization, submodular functions can be used to find optimal subsets of buoys to achieve objectives like maximizing coverage of non-interfering buoys, or maximizing probability of target detection in a tracking scenario. Rather than exhaustively searching over all combinations of subsets, submodular functions provide a fast and tractable framework to compute a solution [15]. This topic will be explored further in future work.

IV. CONCLUSIONS

In this work the $P_T$ map and FP filter approach showed the effectiveness of the sensor field as well as places where targets could slip through for a barrier problem. A greedy optimization approach was used to place the sensors. Although the greedy approach is known to be suboptimal, it works reasonably well. Other optimization methods will be investigated to further improve the sensor field layout. One candidate is Particle Swarm Optimization, [12], which was shown to be effective in the shallow water scenarios.

The computational efficiency of the filter approximation is much faster than a MC approach, however there is continued interest in speeding up this process even more. One approach we will look at is running these algorithms on a GPU where convolution (at the core of the FP filter) is highly efficient. Speeding up the algorithms combined with better optimization algorithms should lead to better solutions for the sensor placement problem.

Ultimately the goal of this work is to include ping sequencing into the $P_T$ map and FP filter formulation. Future work will include further investigation into the simultaneous ping solution as well as including variable ping schedules within the FP filter framework.

ACKNOWLEDGMENTS

This work was funded in part by the U.S. Office of Naval Research, Contract Number N00014-12-G-0078, Delivery Order #0024.

REFERENCES


