SENSOR SCHEDULING FOR TARGET TRACKING IN LARGE MULTISTATIC SONOBUOY FIELDS

D. Angley, S. Suvorova, B. Ristic, W. Moran

RMIT University,
School of Engineering,
Melbourne, VIC 3000

F. Fletcher, H. Gaetjens, S. Simakov

Maritime Division,
Defence Science Technology Group,
Edinburgh, SA 5111

ABSTRACT

Sonobuoy fields, consisting of many distributed emitter and receiver sonar sensors on buoys, are used to seek and track underwater targets in a defined search area. A sensor scheduling algorithm is required in order to optimise tracking performance by selecting which emitter sonobuoy should transmit in each time interval, and which waveform it should use. In this paper we describe a new long term sensor scheduling algorithm for sonobuoy fields, called the continuous probability states algorithm. This algorithm reduces the scheduling search space by keeping track of the probability that a target is undetected, rather than modelling all possible detection outcomes, which reduces the computation complexity of the algorithm. It is shown that this approach results in high quality tracking for multiple targets in a simulated sonobuoy field.

Index Terms— Multi-static sonar, Sensor scheduling, Resource management, Target tracking

1. INTRODUCTION

Sonobuoys are expendable acoustic sensors that are deployed in a body of water to search for and track subsurface targets. A multistatic sonobuoy field consists of a number of these sonobuoys distributed across a large search area. The field operates by transmitting a signal, or "ping", from an emitter sonobuoy and receiving the signal, possibly reflected from a target, at all nearby receivers. The fusion of detections from these sensors dramatically improves the performance of the overall detection process for targets in the typical context where the signal-to-noise ratio of the returned signal is low.

Figure 1 shows an example of such a sonobuoy field, consisting of 16 emitter and 25 receiver sonobuoys, where each type of sonobuoy is laid out in a grid with a spacing of 10 km. The receiver grid is offset from the emitter grid by 5 km in each direction. This field is similar to that considered in our previous work [1, 2, 3]. Drift due to ocean currents can be compensated for using the GPS devices on the sonobuoys, and as such it is ignored in this paper.

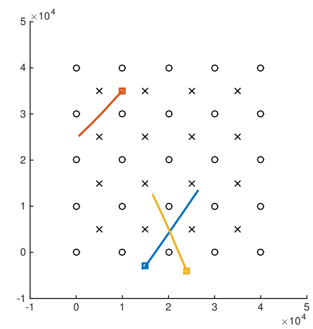


Fig. 1. Sonobuoy field and target trajectories. The crosses are emitter sonobuoys and the circles are receiver sonobuoys. Three target trajectories are shown as lines that start at squares. Units in the diagram are in metres.

Sensor scheduling in such a sonobuoy field means deciding, in each time interval, which sonobuoy should transmit, and which waveform it should use, in order to detect and track targets with high accuracy. The need for intelligent scheduling for multistatic sonar systems was first identified by Krout et al [4], who developed a framework for greedy scheduling for search and tracking using separate metrics. Significant work has since been undertaken to define new metrics for sensor scheduling using greedy algorithms [1, 5, 6, 7, 8, 9, 10].

Sensor scheduling algorithms that optimise over longer time horizons than just the next transmission, such as the one presented in this paper, have been less explored in the literature. Saksena [11] formulates the search and tracking problem as a partially observed Markov decision process and solves it over a long time horizon using policy rollout, taking into account the energy usage at each source. Wakayama [12] formulates the long term ping scheduling problem as a constrained optimisation problem that can be solved using integer-linear programming to maximise a sonar performance metric with constraints to penalise over- or under-usage of the sources. These works only consider scheduling the choice of source to ping, resulting in an action space with size defined by the number of sources, whereas our work uses a larger action space, choosing both a source to ping and which waveform to use, each action resulting in differing detection performance.

A key limiting factor of long term sensor scheduling is its computational complexity. As an example, consider sensor scheduling for the sonobuoy field in Figure 1, with 16 emitter sonobuoys, each with a library of 8 different waveforms that it can transmit. Scheduling a single step ahead in this case requires considering $N_S = 128$ possible actions. Additionally, a detection probability for each target that depends on its state—depth, speed, orientation, location, etc—needs to be calculated for each potential action considered. For three targets there are 8 possible binary detection outcomes for each action, resulting in 1024 detection outcomes for every interval. Considering all possible outcomes n steps ahead would require considering 1024^n paths through a search tree. To make such problems computationally feasible, some approximation of this search tree is required.

In this paper we build on our previous work [3] on long term scheduling, described in Section 4.1, by introducing a new sensor scheduling algorithm to reduce the search space by keeping track of the probability that a target is undetected, and neglecting other aspects of the state of the system. We demonstrate that this algorithm achieves high quality tracking in simulations of the scenario shown in Figure 1.

2. MEASUREMENT SIMULATION

In order to evaluate our sensor scheduling algorithm, a representative model which takes into account some of the features of acoustic propagation was used for simulating multistatic sonobuoy measurements. The target detection process implemented uses precomputed signal excess component data, stored in look up tables that depend on waveforms, source, receiver and target depths and source-receiver separation distance to calculate a realisation of the signal-to-noise ratio (SNR) for each target at each receiver to use in the detector. Computation of the signal excess component data was carried out offline using the Gaussian ray bundle eigenray propagation model [13]. When the detector's threshold is exceeded, a detection is calculated as the bistatic range (from source to target to receiver) and bearing of the target from the receiver (and Doppler for a continuous wave transmission), with additive Gaussian measurement noise. False alarms are also generated using a Poisson distribution for the number of measurements with parameter dependent on the waveform and spatially uniformly distributed in bearing and bistatic range (and bistatic Doppler for continuous wave transmissions). This measurement process is described in greater detail in [1, 14].

3. TRACKER

A multi-target tracking algorithm, to process the measurements and track targets, is also required. The algorithm that we use combines the multi-sensor Bernoulli filter [15, 16] with the linear-multitarget paradigm [17], and employs the target amplitude as a useful feature in track initiation. The tracker, implemented using both the sequential Monte Carlo method and the Gaussian mixture model, produces a probability of target existence for each track. It has been tested numerically in a realistic multistatic sonobuoy environment, indicating robust performance even with unfavourable ping scheduling [14].

4. SENSOR SCHEDULING ALGORITHMS

For a sonobuoy field to operate effectively, a sensor scheduling algorithm is required to optimise tracking performance by selecting which emitter sonobuoy should transmit, and which waveform it should use, in each time interval. An accurate model of this problem would require a partially observable Markov decision process (POMDP) [18], modelling the joint distribution of all the signals received at the receiver sonobuoys given the action taken and the target locations. Solving this POMDP is not currently computationally tractable, so simplifications need to be made.

The primary simplification we use here and in previous work [3] is to only consider the probability of detection for each target, rather than all detection outcomes at each receiver. This reduces a very high dimensional detection model down to a single probability per target. The probability of detection for the targets is calculated using BRISE as described in Section 2.

In order to maintain high quality tracking of every target in a scenario, we build a search tree where the branches at each node are the possible actions in that time period. The simplified state that we keep track of in the nodes of our search tree is a vector containing the probability that each target is undetected over the scheduling horizon considered. Given the actions $\mathbf{a_n} = a_1, a_2, \ldots, a_n$ were chosen in the search tree in the previous n time intervals, the state is

$$\mathbf{s}_{n|\mathbf{a_n}} = (P^1_{\text{undetected}|\mathbf{a_n}}, P^2_{\text{undetected}|\mathbf{a_n}}, \ldots, P^m_{\text{undetected}|\mathbf{a_n}}), \quad (1)$$

where there are m targets and $P^i_{\text{undetected}|\mathbf{a_n}}$ is the probability that the ith target is undetected. The initial state of the search tree is all ones, i.e. none of the targets have been detected yet.

The continuous probability states sensor scheduling algorithm described here is an evolution of our previous long term scheduling algorithm, the cookie cutter algorithm [3]. We first describe the cookie cutter algorithm before introducing the continuous probability states algorithm.

4.1. Cookie Cutter

Our initial method of simplifying the search space for sensor scheduling was the cookie cutter algorithm, which considers a target detected if the probability of detection exceeds the threshold, D, and otherwise considers it undetected [3]. This

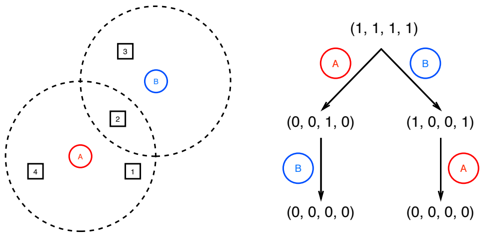


Fig. 2. An example cookie cutter scenario (left) and search tree (right). There are two emitter sonobuoys, A and B, each with a single waveform it can transmit. The circles shows simplified examples of areas where a target will be detected by a sonobuoy with probability $> D$. If A transmits then targets 1, 2 and 4 are considered detected. If B transmits then targets 2 and 3 are considered detected. The search tree on the right shows the evolution of the state in the scheduling algorithm as a vector with four dimensions, one for each target. The circled letters represent the chosen action at each branch.

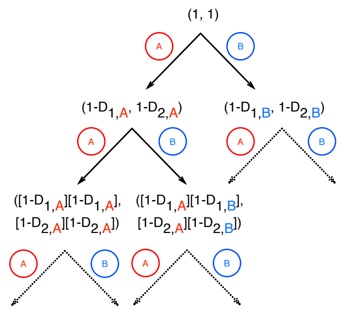


Fig. 3. Continuous probability states search tree for two targets, labelled 1 and 2, and two actions, labelled A and B. $D_{i,j}$ here is the probability that target i is detected when action j is taken. Only a partial tree is shown, and the dotted arrows show where the tree has been truncated.

results in a binary vector for the search states, $\mathbf{s}_{n|\mathbf{a_n}}$. The algorithm terminates a search path when all targets have been detected, $\mathbf{s}_{n|\mathbf{a_n}} = (0, 0, \ldots, 0)$, or if there is no change to the state when an action is taken. This algorithm is computationally efficient, as it only has to consider a single detection outcome per action, instead of modelling all possible outcomes with different probabilities of occurring.

The reward used in the cookie cutter algorithm is the sum of detection probabilities for all targets that were detected for the first time in the given time interval, or

$$R_{n|\mathbf{a_n}} = \text{sum}((\mathbf{s}_{n|\mathbf{a_n}} \oplus \mathbf{s}_{n-1|\mathbf{a_{n-1}}}) \odot \mathbf{D_{a_n}}), \tag{2}$$

where sum is the element-wise sum, $\oplus$ is element-wise exclusive-or, $\odot$ is element-wise multiplication and $\mathbf{D_{a_n}}$ is a vector of the probabilities that each target is detected in time n. The sequence of actions $\mathbf{a}_n$ is chosen to maximise the sum of the reward over all n.

The cookie cutter algorithm can be modelled as a Markov decision process (MDP) and solved using the Gauss-Seidel policy iteration method. An example search tree for the cookie cutter algorithm is shown in Figure 2.

4.2. Continuous Probability States

The cookie cutter algorithm shows the benefit of long term scheduling, but it has some downsides. Firstly, it requires careful tuning of the threshold parameter, D, which should take different values for different sonobuoy deployments. Secondly, thresholding the detection probability means that it is unable to consider paths through the search tree that result in a high detection probability for a target due to multiple transmissions with lower detection probabilities. Instead, it requires a high probability of detection for a target from a single transmission.

In order to fix this flaw, in this paper we introduce the continuous probability states algorithm. Instead of modelling all possible detection outcomes, or thresholding the detection probability, this algorithm calculates the probability that a target is still undetected at a given point in the search tree. The initial state is all ones, and the new state can be calculated as

$$\mathbf{s}_{n|\mathbf{a_n}} = \mathbf{s}_{n-1|\mathbf{a_{n-1}}} \odot (\mathbf{1} - \mathbf{D_{a_n}}). \tag{3}$$

Search is done to a given depth, optimising the reward over a finite number of future actions.

The reward used in the continuous probability states algorithm is the sum of the improvement in the probabilities that each target has been detected in that time interval, or

$$R_{n|\mathbf{a_n}} = \text{sum}(\mathbf{s}_{n-1} - \mathbf{s}_n). \tag{4}$$

The sequence of actions $\mathbf{a}_n$ is chosen to maximise the total reward.

The states in this search tree are generically unique, so there is no benefit to modelling it as an MDP. Instead, a depth-first search of the tree is performed, generating branches on the fly in order to conserve memory. An example search tree for the continuous probability states algorithm is shown in Figure 3.

5. RESULTS

The performance of the sensor scheduling algorithms was tested using 200 Monte Carlo runs of the scenario shown in Figure 1, a three target scenario with 16 emitter sonobuoys and 25 receiver sonobuoys. The two different types of sonobuoy were each placed in a square grid with 10 km spacing, with the two grids offset from each other by 5 km. Each emitter can emit 8 different waveforms, those being all combinations of 1 kHz or 2 kHz in frequency, 2 s or 8 s in duration, and frequency modulated or continuous wave. Each simulation was performed for 50 transmissions, with 40 s between consecutive transmissions. In each time interval, the sensor scheduling algorithm was run to select an emitter and waveform, the measurements were simulated as described in Section 2, and the resulting measurements were provided to the tracking algorithm described in Section 3.

Four sensor scheduling algorithms were compared, namely:

1. Random: The emitter and waveform are chosen at random. This gives a baseline performance of the tracker in this scenario without intelligent sensor scheduling.

2. Greedy: The emitter and waveform that maximise the sum of detection probabilities for all known targets in the current time interval is chosen.

3. Cookie Cutter: As described in Section 4.1. The threshold for the cookie cutter method, determined experimentally to give good results in this scenario, was $D = 0.2$.

4. Continuous Probability States: As described in Section 4.2. The search tree was evaluated to depth 3, as deeper search trees were not found to give better tracking performance.

Tracking performance was measured using the optimal subpattern assignment (OSPA) metric [19]. The OSPA distance between two sets $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ and $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, $n \geq m$, is defined as

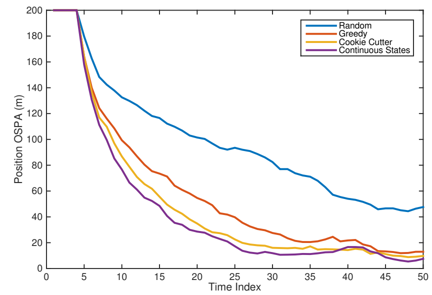
$$d(X,Y) = \Big(1/n \min_{\pi \in \Pi_n} \sum_{i=1}^{m} \min\left\{\|\boldsymbol{x}_i - \boldsymbol{y}_{\pi(i)}\|, c\right\}^p + c^p(n-m)\Big)^{1/p}$$

where Π_n is the set of permutations of the integers $1, \ldots, n$ and c and p are pre-defined constants. We calculate both the position and the velocity OSPA for the targets separately. For this evaluation we used $p = 2$, corresponding to the Euclidean distance, $c = 200$ m for the position OSPA cutoff parameter, and $c = 8$ m/s for the velocity OSPA cutoff parameter.

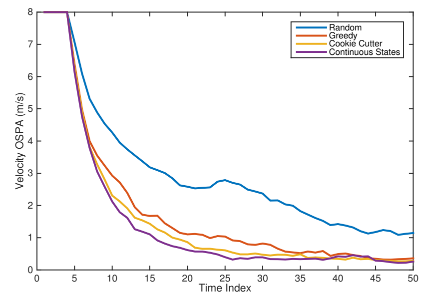
Figure 4 shows the mean position and velocity OSPA over time for the different sensor scheduling algorithms. All intelligent scheduling algorithms outperformed the random algorithm, as expected. The long term scheduling algorithms, cookie cutter and continuous probability states, outperformed the greedy algorithm, demonstrating the benefit of long term scheduling for multiple target scenarios.

The continuous probability states algorithm introduced in this paper outperformed, in most time intervals, the cookie cutter algorithm, especially in the initial stages before the target tracks were fully established. As there is 40 s between pings, this results in information being acquired about targets a few minutes earlier than the cookie cutter algorithm. This speed and accuracy improvement can make a significant difference in some applications. As explained in Section 4.2, this tracking improvement is likely due to the algorithm considering paths that build up a high detection probability for a target over multiple transmissions, a situation that is ignored due to the thresholding in the cookie cutter algorithm.

To understand the computational cost of the different scheduling algorithms, 500 Monte Carlo runs of the scheduling algorithm were performed on a set of data taken from the scenario in Figure 1. Each algorithm was implemented in MATLAB and tested on a 2.3GHz Intel Core i7. The mean time for a single run of greedy algorithm was 1.30 ± 0.07 s, for the cookie cutter algorithm it was 2.75 ± 0.09 s, and for the continuous probability states algorithm it was 4.38 ± 0.06, with a 95% confidence interval. The algorithms with higher tracking performance are therefore associated with a higher computation cost.



(a)



(b)

Fig. 4. (a) Mean Position OSPA with $p = 2$ and $c = 200$ m. (b) Mean Velocity OSPA with $p = 2$ and $c = 8$ m/s.

6. CONCLUSION

When tracking multiple targets in large multistatic sonobuoy fields it is beneficial, but computationally complex, to look multiple steps ahead when scheduling. We introduced the continuous probability states algorithm, a sensor scheduling algorithm for sonobuoy fields. This algorithm reduces the computation complexity of long term scheduling and outperforms our previous long term scheduling algorithm, the cookie cutter algorithm [3].

There are many other ways that the search tree for this sensor scheduling algorithm can be simplified or pruned that could be explored in future work. For example, the continuous probability states algorithm could prune search paths than have a low probability of detection, instead of searching to a set depth. Also, branch and bound methods [20] could be used to prune branches of the search tree that can be proven not to result in the optimal solution.

7. REFERENCES

[1] S. Suvorova, M. Morelande, B. Moran, S. Simakov, and F. Fletcher, "Ping scheduling for multistatic sonar systems," in *Proceedings of the 17th International Conference on Information Fusion*, 2014.

[2] S. Suvorova, M. Morelande, B. Moran, S. Simakov, and F. Fletcher, "Multi-target tracking for multistatic sonobuoy systems," in *Proceedings of the 18th International Conference on Information Fusion*, 2015.

[3] S. Suvorova, F. Fletcher, D. Angley, H. Gaetjens, S. Simakov, M. Morelande, and B. Moran, "Markov decision process for sonobuoy transmission scheduling," in *Proceedings of the 19th International Conference on Information Fusion*. IEEE, 2016, pp. 2155–2162.

[4] D. W. Krout, M. A. El-Sharkawi, W. L. Fox, and M. U. Hazen, "Intelligent ping sequencing for multistatic sonar systems," in *Proceedings of the 9th International Conference on Information Fusion*. IEEE, 2006, pp. 1–6.

[5] S. Simakov and F. Fletcher, "GPU acceleration of threat map computation and application to selection of sonar field controls," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1827–1831.

[6] B Imre Incze and Steven B Dasinger, "A Bayesian method for managing uncertainties relating to distributed multistatic sensor search," in *9th International Conference on Information Fusion*, 2006, pp. 1–7.

[7] D. W. Krout, W. L. Fox, and M. A. El-Sharkawi, "Probability of target presence for multistatic sonar ping sequencing," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 4, pp. 603–609, 2009.

[8] C. Y. Wakayama and D. J. Grimmett, "Adaptive ping control for track-holding in multistatic active sonar networks," in *Proceedings of the 13th International Conference on Information Fusion*. IEEE, 2010, pp. 1–8.

[9] C. Y. Wakayama, D. J. Grimmett, and Z. B. Zabinsky, "Forecasting probability of target presence for ping control in multstatic sonar networks using detection and tracking models," in *Proceedings of the 14th International Conference on Information Fusion*, Chicago, USA, 2011, pp. 1–8.

[10] C. Y. Wakayama and Z. B. Zabinsky, "Simulation-driven task prioritization using a restless bandit model for active sonar missions," in *Proceedings of the 2015 Winter Simulation Conference*. IEEE Press, 2015, pp. 3725–3736.

[11] A. Saksena and I-J. Wang, "Dynamic ping optimization for surveillance in multistatic sonar buoy networks with energy constraints," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 1109–1114.

[12] C. Y. Wakayama, Z. B. Zabinsky, and D. J. Grimmett, "Linear optimization models with integer solutions for ping control problems in multistatic active acoustic networks," in *Proceedings of the 15th International Conference on Information Fusion*. IEEE, 2012, pp. 2354–2361.

[13] H. Weinberg and R. E. Keenan, "Gaussian ray bundles for modeling high-frequency propagation loss under shallow-water conditions," *The Journal of the Acoustical Society of America*, vol. 100, no. 3, pp. 1421–1431, 1996.

[14] B. Ristic, D. Angley, F. Fletcher, S. Simakov, H. Gaetjens, S. Suvorova, and B. Moran, "Bayesian multitarget tracker for multistatic sonobuoy systems," in *Proceedings of the 19th International Conference on Information Fusion*. IEEE, 2016, pp. 2171–2178.

[15] B.-T. Vo, C.-M. See, N. Ma, and W. T. Ng, "Multi-sensor joint detection and tracking with the Bernoulli filter," *IEEE Trans. Aerospace and Electronic Systems*, 2012.

[16] B. Ristic and A. Farina, "Target tracking via multi-static doppler shifts," *IET Radar, Sonar & Navigation*, vol. 7, no. 5, pp. 508–516, 2013.

[17] D. Musicki and B. La Scala, "Multi-target tracking in clutter without measurement assignment," *IEEE Trans. Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 877–896, July 2008.

[18] O. Sigaud and O. Buffet, *Markov decision processes in artificial intelligence*, John Wiley & Sons, 2013.

[19] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *Signal Processing, IEEE Transactions on*, vol. 56, no. 8, pp. 3447–3457, 2008.

[20] J. Clausen, "Branch and bound algorithms-principles and examples," *Department of Computer Science, University of Copenhagen [Online]*, pp. 1–30, 1999.