A UNIFIED METHOD FOR PASSIVE MEASUREMENT AND TRACKING OF CONTACTS FROM AN ARRAY OF SENSORS

Robert E. Zarnich, Kristine L. Bell, and Harry L. Van Trees

George Mason University, School of IT & and Engineering, Fairfax, VA 22030-444 USA
rzarnich@gmu.edu, kbell@gmu.edu, hlv@gmu.edu

ABSTRACT

A multiple target track estimation method which operates directly from array data is presented. The Maximum A-Posteriori (MAP) estimator for contact states is derived for temporally uncorrelated signals and uncorrelated contact tracks. This batch estimator is an iterative algorithm employing a nested pair of Expectation Maximization (EM) based algorithms. The hidden data are intermediate direction finding estimates of synthetic signal estimates, each conditioned on prior track distributions. This method eliminates the data association step of traditional multitarget tracking approaches by conditioning the measurement process on individual target state distributions. This approach results in a process similar to the EM algorithm for direction finding by Miller and Fuhrmann, with an additional penalty term imposed by the track distributions. Simulation results for two relevant submarine towed array scenarios are presented and discussed.

1. INTRODUCTION

An important and pervasive problem in the engineering of sensor systems is the detection and tracking of multiple contacts through observations made from an array of sensors. An optimal approach would estimate the tracks of objects directly from the array snapshot data, however the solution to this estimation problem is quite difficult.

Traditional solutions partition the track estimation operation into two isolated processes: direction-of-arrival (DOA) estimation from array snapshot data, followed by track estimation from the DOA estimates. This partitioning results in procedures which are suboptimal and which require data association to match DOA estimates to contacts. Many approaches have been offered to solve the data association problem [1], however it continues to be a significant area of research and can be a major contributor to poor system performance.

In [2], a Maximum A-Posteriori (MAP) solution for estimating the target states directly from the array data was proposed. This approach discretizes the target state space and employs a Viterbi algorithm for determining the optimal state sequences. It provides an elegant but computationally expensive solution.

In this paper we will develop an efficient MAP estimation technique for determining the tracks of multiple objects without a discrete state space approach. We take the approach of introducing ‘hidden data’ as in the Expectation-Maximization (EM) [3] and Space Alternating Generalized EM (SAGE) [4] algorithms. This allows us to develop an iterative procedure for estimating the target states and provides a mechanism to control the trade-off between convergence rate and estimation error.

The paper is organized as follows. In Section 2, the signal and motion model is formulated and the joint pdf for the multitarget tracking problem is specified. In Section 3.1, a solution for the single target case is developed, and in Section 3.2 the concept is extended to multiple targets. Simulation results are presented in Section 4, and a summary is given in Section 5.

2. STATISTICAL MODEL AND ASSUMPTIONS

We consider the multitarget tracking problem where there are \( M \) contacts radiating signals received by an array of sensors. The number of objects \( M \) is assumed known and the trajectories of the objects are uncorrelated with the trajectories of other objects. We assume for simplicity the targets and the array lie in the \( x - y \) plane, and that the array is linear, although we can easily extend the results to other geometries. This is illustrated in Figure 1. The 2-dimensional state is defined as its bearing \( u = \cos(\theta) \) and bearing rate \( \dot{u} \). Thus the state of the \( n th \) contact at snapshot \( k \) is \( x_{k,m} = [u_{k,m}, \dot{u}_{k,m}]^T \). We assume the motion of the objects is described by a first order Gauss-Markov process, \( i.e. \) for the \( n th \) contact,

\[
x_{k,m} = Fx_{k-1,m} + w_{k,m}
\]
The array data depends on the target state. Note that the array data is conditioned on the target states, and the pdf of the array data conditioned on the target states is
\[ p(y_k | x_{k,m}) = \frac{\exp(-\frac{1}{2}(y_k - H x_{k,m})^T \Sigma^{-1}(y_k - H x_{k,m}))}{\sqrt{2\pi} \det[\Sigma]}. \]

At the array, the observations have the form
\[ y_k = \sum_{m=1}^{M} s_{k,m} v(u_{k,m}) + n_k \]
where \( s_{k,m} \) is the frequency domain signal from the \( m \)th object at the \( k \)th snapshot with \( E[s_{k,m} s_{k,m}^H] = \alpha_{k,m} \). The vector \( v(u_{k,m}) \) is the array response vector for the DOA \( u_{k,m} \), and \( n_k \) is a vector of uncorrelated sensor noise samples. The source signals and noise are sample functions of independent zero-mean Gaussian random processes. The signal powers, \( \alpha_{k,m} \), are time varying and the noise covariance matrix is constant with \( E[n_k n_k^H] = \sigma_n^2 I \). Let \( X_k = [x_{k,1}, x_{k,2}, \ldots, x_{k,M}]^T \) and \( \alpha_k = [\alpha_{k,1}, \alpha_{k,2}, \ldots, \alpha_{k,M}]^T \). The array data \( y_k \) is then jointly complex Gaussian with zero mean and covariance
\[ K_{y_k}(X_k, \alpha_k) = \sum_{m=1}^{M} \alpha_{k,m} v(u_{k,m}) v^H(u_{k,m}) + \sigma_n^2 I. \]

The pdf of the array data conditioned on the target states is
\[ p(y_k | x_{k,m}) = \frac{\exp(-\frac{1}{2}(y_k - H x_{k,m})^T \Sigma^{-1}(y_k - H x_{k,m}))}{\sqrt{2\pi} \det[\Sigma]} \]

where \( y_k \) is a vector of complex samples.

We will use both \( x_{k,m} \) and \( u_{k,m} \) to denote bearing as necessary in the subsequent derivation in order to reduce notational complexity.

The single scan joint pdf conditioned on the previous contact state is
\[ p(y_k | x_{k,m}) = \prod_{m=1}^{M} \varphi(x_{k,m} | x_{k-1,m}). \]

There are \( K \) snapshots in an observation batch. No data is available at \( k = 0 \) so we assume a Gaussian prior distribution on object states with mean \( \bar{x}_{0,m} \) and covariance \( \bar{P}_{0,m} \).

The joint pdf over the batch is
\[ p(y, \bar{x} : \alpha) = \prod_{\nu=1}^{M} \varphi(\bar{x}_{0,\nu}) \cdot \prod_{k=1}^{K} p(y_k | x_{k,m}) \prod_{m=1}^{M} \varphi(x_{k,m} | x_{k-1,m}) \]
where \( Y = \{y_1, \ldots, y_K\}, \bar{X} = [\bar{X}_1, \ldots, \bar{X}_K] \), and \( \alpha = [\alpha_1, \ldots, \alpha_K] \).

### 3. MAP ESTIMATOR DEVELOPMENT

Traditional calculus based optimization is not a tractable approach and a brute force Newton type algorithm for the maximization of Eq. (7) has difficulties from both an analytic and an implementation perspective as discussed in [5]. An exact solution under a discrete state space model is provided by [2], however it is computationally very expensive. What is desired is a solution which provides a tractable and efficient algorithm.

Nonlinear programming methods [6] such as relaxation and auxiliary penalty functions solve optimization problems iteratively by finding the solution to an approximated problem and successively force the approximation and the original problem to converge with each iteration. We use a similar approach. Assume we have an ‘observation error’ at the array. We can model the ‘observed’ DOA as
\[ u_{k,m} = u_{k,m} + e_{k,m} \]

where \( p(e_{k,m}) = \mathcal{N}(0, \sigma_{e_{k,m}}^2) \),
\[ p(\mu_{k,m} | x_{k,m}) = \mathcal{N}(u_{k,m}, \sigma_{\epsilon_{k,m}}^2). \]

Accounting for this error term, the observed array data is then given by (3) with \( u_{k,m} \) replaced by \( \mu_{k,m} \) and the pdf of
the array data conditioned on the corrupted observation of target states is
\[
p(y_k | \mu_k : \alpha_k) = \frac{\exp \left(-y_k^H K^{-1}_{y \alpha} (\mu_k, \alpha_k) y_k \right)}{\pi^N \det \left[ K_{y \alpha}(\mu_k, \alpha_k) \right]}, \quad (11)
\]
where \( \mu_k = [\mu_{k,1}, \ldots, \mu_{k,M}]^T \). The modified single scan joint pdf is
\[
\mathcal{Y}(y_k, \mu_k, x_k | x_{k-1} : \alpha_k) = p(y_k | \mu_k, x_k, x_{k-1} : \alpha_k)p(\mu_k, x_k | x_{k-1}) \\
= p(y_k | \mu_k, x_k, x_{k-1} : \alpha_k) \prod_{m=1}^M p(\mu_{k,m} | x_{k,m}) \varphi(x_{k,m} | x_{k-1,m}). \quad (12)
\]
and the modified batch joint pdf is
\[
\mathcal{Y}(Y, M, X : \alpha) = \prod_{\mu=1}^M \varphi(x_{0,\mu}) \cdot \prod_{k=1}^K \left\{ \prod_{m=1}^M p(y_k | \mu_k, x_k, x_{k-1} : \alpha_k) \prod_{m=1}^M p(\mu_{k,m} | x_{k,m}) \varphi(x_{k,m} | x_{k-1,m}) \right\}. \quad (13)
\]
While on the surface it appears we have complicated the problem, this formulation will allow us to develop an iterative procedure for estimating \( M \) and \( X \).

In the limit as the ‘observation error’ variance goes to zero we have \( p(\mu_k | x_k) \) taking on the behavior of a Test Sequence. By initially assuming a large error variance and successively reducing it with every iteration cycle, we recover the original problem. This provides a mechanism to control the trade off between convergence rate and estimation error.

This auxiliary observation error has a real physical counterpart. In all real systems we employ physical devices which are subject to error as well as observe energy through a propagation channel with random fluctuation. In these systems, particularly towed hydro-acoustic arrays this error manifests itself as minor propagation fluctuation and random array orientation error. By treating this error term up front, we get the same structure as in Eq. (13).

### 3.1. Single Signal MAP Estimator

Considering only a single signal we drop the notation on \( m \) and Eq. (13) becomes
\[
p(Y, \mu, X : \alpha) = \varphi(x_0) \prod_{k=1}^K \left\{ \prod_{m=1}^M p(y_k | \mu_k : \alpha_k) p(\mu_k | x_k) \varphi(x_k | x_{k-1}) \right\}. \quad (14)
\]
For a fixed \( \mu \) and \( \alpha \), the MAP estimator of \( X \) is now the the fixed interval Kalman smoother. On the other hand, for a fixed \( X \), we can solve for both \( \mu \) and \( \alpha \) using Maximum Likelihood Estimation (MLE) techniques from array process where an additional ‘penalty’ term is imposed due to \( p(\mu_k | x_k) \).

Eq. (14) is of an identical form used by the SAGE algorithm [4]. It is shown in [4] that if the observed data \( y \), the hidden data \( (\mu) \), and the non-random parameters of interest \( (\theta) \) are related as follows,
\[
f(y, \mu; \theta) = f(y | \mu)f(\mu; \theta), \quad (15)
\]
then \( (y, \mu) \) is an Admissible Complete-Data Space for the estimation of \( \theta \). Since we are interested in MAP estimation, \( \theta \) is a random parameter. The requirement for MAP estimation can be restated as
\[
f(y, \mu, \theta) = f(y | \mu)f(\mu | \theta)f(\theta). \quad (16)
\]
Eq. (14) is of the form of (16) for an admissible hidden data space, with the noisy bearing vector \( \mu \) behaving as the hidden data. Since our form and the required SAGE form are equivalent, we will borrow from the SAGE theory for its convergence proof as well as the optimization technique to construct a MAP estimation algorithm.

A detailed development of the algorithm is presented in [5] and the results are summarized here. The algorithm is initialized by defining an initial track and signal power in the vicinity of their true values. The observation error variance is initially set to the Cramér-Rao Bound and reduced at every iteration. This ensures that the estimation error introduced by the observation error is significantly less than the inherent resolution of the array. The Algorithm consists of the Expectation (E) Step and the Maximization (M) step as follows:

- **E-Step:** Estimate \( \mu^p_k \) for each \( k = 1, \ldots, K \) by finding
\[
\mu^p_k = \arg\max_{\mu} \left\{ -\ln \left( \frac{V^H(\mu) C_{y_\mu} v(\mu) }{N \sigma^2_n} \right) + \frac{V^H(\mu) C_{y_\mu} v(\mu) }{N \sigma^2_n} - \frac{(\mu - u^p_{k-1})^2}{2\sigma^2_{e_{k,m}}} \right\}. \quad (17)
\]
where \( C_{y_\mu} = y_k Y_k^H \). The error variance is set to \( \sigma^2_{e_{k,m}} = e^p \cdot \sigma^2_{C_{RRB_k}} \), where \( e^p \) is a scaling parameter chosen for convergence properties and \( \sigma^2_{C_{RRB_k}} \) is the Cramér-Rao Bound evaluated at the estimated bearing \( u^p_{k-1} \) and power \( \sigma^2_{k} \) from the previous iteration. The E-Step is in fact a Penalized Maximum Likelihood Estimate (PMLE) of the noisy bearing \( \mu_k \), where the penalty term is based on the relationship of \( \mu_k \) to the true bearing \( u_k \) through the observation error.
- **M-Step:**
- Part I: Compute $\alpha_k$ by

$$
\alpha_k^p = \max \left[ \frac{1}{N} \left( \frac{v^H(\mu_k^p)}{N} C_{Y_k} v(\mu_k^p) - \sigma_n^2 \right), 0 \right].
$$

(18)

- Part II: Compute $X^p$ using the $\mu_k^p$ as the ‘state measurements’ and $e^p \cdot \sigma_{CRB}^2$ as their variances in a fixed interval Kalman smoother.

The M-Step produces a PMLE of $\alpha_k$ and a MAP estimate of the states $x_k$ via the Kalman smoother.

- Increment $p = p + 1$ and iterate.

We carry out these steps for a prescribed number of iterations and accept the final value as the MAP estimate.

3.2. Multiple Signal MAP Estimator

To extend this idea we need only to consider the idea of nesting an EM Algorithm within the single signal MAP estimator algorithm. If we could separate the mixed observed signals in $Y$ into $M$ distinct signals $Z_m$ where $Y = \sum_{m=1}^M Z_m$, we would have all we need to solve the multiple contact problem. Since we have assumed contacts with uncorrelated signals we can employ the technique of Miller and Fuhrmann in [7] to synthesize $M$ single signals from the mixed data in $Y$. At each scan we then have the standard EM Algorithm for direction finding, modified with the quadratic penalty term. There are then $M \times K$ estimates to form and $M$ fixed interval Kalman smoothers in parallel for every iteration. The extension of the single signal algorithm to multiple signals results in the following algorithm.

- E-Steps:

$$
R_{z_{k,m}}^{p,q} \equiv E \left[ R_{z_{k,m}} \mid C_{Y_k}, K_{y_k}^{p,q} \right]
$$

\begin{align*}
&= K_{y_k}^{p,q} (K_{y_k}^{p,q})^{-1} C_{Y_k} (K_{y_k}^{p,q})^{-1} K_{y_k}^{p,q} - K_{y_k}^{p,q} (K_y^{p,q})^{-1} K_{y_k}^{p,q},
\end{align*}

(19)

where

$$
K_{y_k}^{p,q} = \alpha_k^{p,q-1} v(\mu_k^{p,q-1}) v^H(\mu_k^{p,q-1}) + \frac{\sigma_n^2}{M} I,
$$

(20)

and

$$
K_{y_k}^{p,q} = \sum_{m=1}^M K_{z_{k,m}}^{p,q}.
$$

(21)

- M-Step, Part I:

$$
\alpha_k^{p,q} = \max \left[ \frac{1}{N} \left( \frac{v^H(\mu_k^{p,q})}{N} R_{z_{k,m}}^{p,q} v(\mu_k^{p,q}) - \sigma_n^2 \right), 0 \right].
$$

(23)

Increment $q = q + 1$ and iterate.

- M-Step, Part II: Compute $X^p$ with $M$ independent fixed interval Kalman smoothers using $M$ and the associated variances. Increment $p = p + 1$ and iterate.

The iterations may be performed for fixed number of cycles or until some convergence criterion is reached. The trade-off is algorithm complexity vs. estimation accuracy. Similarly, the choice of the reduction schedule for $e^p$ represents a tradeoff between convergence speed and estimation accuracy. One possibility is to set $e^p = \eta e^{p-1}$ where $0 < \eta < 1$. In explicit pseudo-code form, the algorithm is:

Initialize $\alpha_k^0, x_k^0, \forall k, m$

for $p = 1, \ldots, p_{\text{max}}$

$e^p = \eta^{(p-1)}$

for $k = 1, \ldots, K$

Initialize $\mu_k^{p,0} \equiv u_{k,m}^{p-1}$

for $q = 1, \ldots, q_{\text{max}}$

Compute $K_{y_k}^{p,q} \forall m$ \{Eq. (20)\}

Compute $K_{y_k}^{p,q} \forall m$ \{Eq. (21)\}

for $m = 1, \ldots, M$

Compute $R_{z_{k,m}}^{p,q} \forall m$ \{Eq. (19)\}

Find $\mu_k^{p,q}$ \{Eq. (22)\}

Compute $e_k^{p,q}$ \{Eq. (23)\}

end $\{m\}$

end $\{q\}$

end $\{k\}$

Update State Estimates $H = [1 \ 0]$

for $m = 1, \ldots, M$

Initialize $b_{0,0} \equiv \bar{x}_{0,m}, P_{0,0} \equiv \bar{P}_{0,m}$

Forward Kalman Filter

for $k = 1, \ldots, K$

$P_{k|k-1} = F P_{k-1|k-1} F^T + Q$

$W_k = P_{k|k-1} H^T \left\{ HP_{k|k-1} H^T + \sigma_{e_k}^2 \right\}^{-1}$

$P_{k|k} = P_{k|k-1} - W_k H P_{k|k-1}$

$b_{k|k} = F d_{k-1,k-1} + W_k \left\{ \mu_k^{p,q} - H F b_{k-1|k-1} \right\}$

end $\{k\}$

Backward Kalman Smoother
Set $x^p_{k,m} = b_{k|c}$ for $k = K - 1, \ldots , 1$

$G = P_{d|k}F^T P_{k-1}^{-1}$

$x^p_{k,m} = b_{k|k} + G(x^p_{k+1,m} - Fb_{k|k})$

end \{k\}

end \{p\}

The algorithm has a great intuitive appeal in that given an assumed track for each contact, we decompose the array data into the $M$ synthetic signal covariances for each snapshot, find their directions constrained to the neighborhood of the current track, and then adjust the current track estimate to satisfy a weighted least squares criterion (the fixed interval Kalman smoother). We repeat the process and at every iteration we enforce a stricter relationship between the ‘observed’ DOA and the estimated track.

The outstanding feature of this approach is that the association of the measurements to tracks is implicit since each measurement is conditioned, via the Gaussian penalty term, on its track. This provides a near optimal measurement for each track.

4. RESULTS

Two simulated scenarios were used to develop and test the algorithm. An array of 10 elements is used. Contact parameters are summarized for both scenarios in Tables 1 and 2. In scenario 1, the ‘Target’ starts as the left most trace in Figure 2(a), Interferer 3 is the next trace to the right, then Interferer 2 and then Interferer 1. In scenario 2, Interferer 3 starts as the left most trace in Figure 3(a), then the Target is the next trace to the right, then Interferer 2 and then Interferer 1. In the simulations, a fixed number of iterations was used with $p_{max} = 10$ and $q_{max} = 5$. The error reduction parameter was set to $\eta = 0.1$. Figures 2(a) and 3(a) show the true paths of the contacts summarized in Tables 1 and 2 as they move through the observation space. Figures 2(b) and 3(b) show the true tracks (marked with the tick marks) and the estimated tracks (lines) after 10 iterations overlaid on the data.

The results were compared to the discrete state space (DSS) technique proposed in [2]. This required developing a straightforward extension of their algorithm to the stochastic signal model used here. Table 3 shows the estimation accuracy performance of the two algorithms. Both algorithms were initialized with the same information and great care was taken to tune the algorithm of [2] to the data and to make it as computationally efficient as possible. Our continuous state space (CSS) technique out-performed the discrete state space algorithm of [2] from both an RMS Error perspective and in computational complexity. The execution time for the CSS algorithm was about 4 minutes, while the execution time for the DSS algorithm was approximately 20 hours. These times were on a 550MHz Dell PC using MATLAB® v5.3.

5. CONCLUSIONS

We have developed an efficient algorithm for the MAP estimation of multiple contact tracks. To construct this algorithm we introduced an observation error to decouple the Gauss-Markov motion model and the array data model. Convergence of the artificial problem to the original problem was enforced through the reduction of the observed error variance with each iteration. The single target solution was extended to multiple targets by using the decomposition capabilities of the EM algorithm developed in [7] to break the problem into $M$ single target problems.

The algorithm performed very well. In addition to accuracy, the convergence rate was quite good. The algorithm provides a substantial reduction in computation to a discrete state space solution while also decreasing the RMS error. A sequential version for real-time estimates, and a beamspace formulation for computational considerations, are developed and discussed in [5]. The sequential version is presented in [9].

Table 1: Parameters for Scenario 1

<table>
<thead>
<tr>
<th>Contact</th>
<th>SNR</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>-7dB</td>
<td>2.56·10^{-2}</td>
</tr>
<tr>
<td>Interferer 1</td>
<td>0dB</td>
<td>6.10·10^{-3}</td>
</tr>
<tr>
<td>Interferer 2</td>
<td>-7dB</td>
<td>8.99·10^{-3}</td>
</tr>
<tr>
<td>Interferer 3</td>
<td>0dB</td>
<td>4.85·10^{-3}</td>
</tr>
</tbody>
</table>

Table 2: Parameters for Scenario 2

<table>
<thead>
<tr>
<th>Contact</th>
<th>SNR</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>-10dB</td>
<td>2.45·10^{-2}</td>
</tr>
<tr>
<td>Interferer 1</td>
<td>0dB</td>
<td>5.62·10^{-3}</td>
</tr>
<tr>
<td>Interferer 2</td>
<td>-10dB</td>
<td>2.94·10^{-2}</td>
</tr>
<tr>
<td>Interferer 3</td>
<td>0dB</td>
<td>3.76·10^{-3}</td>
</tr>
</tbody>
</table>
6. REFERENCES


