A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters

Matthew Orton and William Fitzgerald

Abstract—We present a Bayesian approach to tracking the direction-of-arrival (DOA) of multiple moving targets using a passive sensor array. The prior is a description of the dynamic behavior we expect for the targets which is modeled as constant velocity motion with a Gaussian disturbance acting on the target’s heading direction. The likelihood function is arrived at by defining an uninformative prior for both the signals and noise variance and removing these parameters from the problem by marginalization. Recent advances in sequential Monte Carlo (SMC) techniques, specifically the particle filter algorithm, allow us to model and track the posterior distribution defined by the Bayesian model using a collection of target states that can be viewed as samples from the posterior of interest. We describe two versions of this algorithm and finally present results obtained using synthetic data.

Index Terms—Bayesian, DOA, particle filter, partition, target tracking.

I. INTRODUCTION

O

VER the past 30 years, many techniques have been developed for estimating the direction of arrival (DOA) of signal sources observed with a sensor array. Solutions to this problem are of interest for communications, military, naval, and air traffic control operations. Those based on beamforming and signal-subspace techniques [1]–[3] have poor performance with moving targets, and those based on maximum likelihood principles [4], [5] require heuristics to associate estimates made at different times. Both these problems can be solved if we take a Bayesian approach where we seek the posterior distribution of the targets DOAs given the sensor measurements. Until recently, the only way to solve this problem was to make unreasonable assumptions—chiefly linearity or Gaussianity—which lead to fragile systems. Advances in Monte Carlo techniques allowed us to reliably estimate the unknown parameters from the problem by marginalization. Recent advances in sequential Monte Carlo (SMC) techniques, specifically the particle filter algorithm, allow us to model and track the posterior distribution defined by the Bayesian model using a collection of target states that can be viewed as samples from the posterior of interest. We describe two versions of this algorithm and finally present results obtained using synthetic data.

II. BAYESIAN MODEL

We consider \( K \) targets moving in a plane containing a sensor array. The motion is assumed to be at constant velocity with a Gaussian disturbance on the target’s heading direction, and this will define our prior on the target dynamics. The signals received by the sensors are due to ambient radiation, and not an emitted signal, and in this way, the system can remain covert. The emitted signals are accounted for by describing them with an uninformative prior, which leads to a joint likelihood over the received signals and the emitted signals. This likelihood is then marginalized to give a function with no direct dependence on the emitted signals.
A. Target Dynamics

We assume that the target is moving with constant velocity and is observed every $T$ seconds. We first consider the model for the case when the Gaussian disturbance is zero so that the heading direction is constant. Let $r_k(t) \in [0, \infty)$ and $\theta_k(t) \in (-\pi, \pi]$ (measured ccw wrt y-axis) denote the range and bearing, respectively, of the $k$th target at time $t$. Then, the $k$th target at $(r_k(t) \sin \theta_k(t), r_k(t) \cos \theta_k(t))$ moving with velocity $v_k \in [0, \infty)$ and a heading of $\phi_k \in (-\pi, \pi]$ (measured ccw wrt x-axis) will be at $(r_k(t) \sin \theta_k(t) + v_k T \cos \phi_k, r_k(t) \cos \theta_k(t) + v_k T \sin \phi_k)$ $T$ seconds later. Thus

\begin{align}
\tan(\theta_k(t+T)) &= \frac{r_k(t) \sin \theta_k(t) + v_k T \cos \phi_k}{r_k(t) \cos \theta_k(t) + v_k T \sin \phi_k} \quad (1) \\
r_k(t+T) &= \sqrt{r_k(t)^2 + 2 v_k \sqrt{r_k(t) T \sin(\theta_k(t) + \phi_k)} + v_k^2 T^2}. \quad (2)
\end{align}

Dividing the numerator and denominator of (1) by $r_k(t)$ and, similarly, both sides of (2) by $r_k(t)$, we see that only the compound variable $v_k/r_k(t)$ appears. We write this as $q_k(t)$ and note that we can only solve the motion equations for this compound variable. In order to model trajectories that are not straight, we apply a random disturbance $\omega_k(t)$ to the heading direction with $\omega_k(t) \sim \mathcal{N}(0, \sigma^2_{\omega_k, t})$. Incorporating these into the target dynamics and writing it as a vector gives (3), shown at the bottom of the page. Note that the disturbed heading direction $\phi_k(t) + \omega_k(t)$ is used when calculating $\theta_k(t)$ and $q_k(t)$. The terms $u_{\phi_k, k}(t+T)$ and $u_{r_k, k}(t+T)$ are included so that the whole state space is reachable with nonzero probability at each transition. $u_{\phi_k, k}(t+T) \sim \mathcal{N}(0, \sigma^2_{\phi_k, t})$, and $\sigma^2_{\phi_k, t}$ is generally very small. Since $q_k(t)$ is related to the target velocity, we can think of it as a scale parameter, that is, to compare different values of $q$, we divide rather than subtract them.

In order to introduce some variance into the $q$ parameter, we must therefore use a positive multiplicative random variable, which accounts for the form of the second row of (3). Again, we use $u_{\phi_k, k}(t+T) \sim \mathcal{N}(0, \sigma^2_{\phi_k, t})$, and $\sigma^2_{\phi_k, t}$ is generally very small. We write $X_k = \{x_{1k}(t), x_{2k}(t), \ldots, x_{Mk}(t)\}$ to denote the states for the $K$ targets at time $t$. In the Appendix, the explicit form of the prior distribution $p_X(x_{1k}(t+T)|x_{Mk}(t))$ is given, along with gradient and Hessian terms, which will be needed later. This prior is based on a deterministic system used in [5].

B. Data Likelihood

The signals reflected by the targets are received by a set of isotropic sensors lying in the same plane as the motion of the targets. The sensor positions are assumed known, and we take them to be placed uniformly around the periphery of a circle of radius $R$. This leads to a likelihood function that is the same shape for all translations of the target DOAs. The time delay of the signal from the $k$th target at the $p$th sensor is $R/\cos(\theta_k - 2\pi p/P)$ relative to the origin of the coordinates, when there are $P$ sensors. The output from the $p$th sensor is then

\[ y_p(t) = \sum_{k=1}^{K} s_k(t + R/\cos(\theta_k - 2\pi p/P)) + w_p(t) \]

where $w_p(t) \sim \mathcal{C}\mathcal{N}(0, \sigma^2_{w_p})$ ($\mathcal{C}\mathcal{N}$ is the isotropic complex normal distribution where the real and imaginary parts of the random variable are independent and taken from identical normal distributions). If we assume that the targets are distant and that the signal is narrowband, then we can replace the time offset with a phase offset of $2\pi R/\lambda_0 \cos(\theta_k - 2\pi p/P)$, where $\lambda_0$ is the target signal wavelength. This can thus be written as a matrix equation

\[ y(t) = A[\theta(t)] s(t) + w(t) \]

where $\theta(t) = [\theta_1(t), \theta_2(t), \ldots, \theta_K(t)]^T$ is the target DOA vector, $A[\theta(t)]$ is the array composite steering matrix, and $s(t)$ is a column vector of complex signals at time $t$ from the $K$ targets. The $p, k$th element of $A$ is then

\[ A_{pk} = \exp(j 2\pi R/\lambda_0 \cos(\theta_k - 2\pi p/P)) \]

where $M$ measurements are taken for each increment $T$ of the dynamic model, and the time interval over which these are gathered is a sufficiently small proportion of $T$ for us to approximate the target as stationary over the measurement interval. We can then consider the data for one step of the dynamic model as

\[ Y_t = \begin{bmatrix} y(t) \\ y(t + \tau) \\ y(t + 2\tau) \\ \vdots \\ y(t + (M-1)\tau) \end{bmatrix} \]

where $\tau$ is the time between measurements, and $\tau M \ll T$. Forming $S_t$ in the same way, we can write

\[ Y_t = A[\theta(t)] S_t + W_t \]

(4)

where $A[\theta(t)]$ is a matrix with $M$ copies of $A[\theta(t)]$ down the diagonal and zeros elsewhere (henceforth, $A[\theta(t)]$ will be written...
\( A_t \) and \( W_t \) is a suitably dimensioned noise vector. The full conditional likelihood is, therefore, given by

\[
p(Y_t|A_t, S_t, \sigma^2(t)) = (2\pi \sigma^2(t))^{-MF} \exp \left[ -\frac{(Y_t - A_t S_t)^H(Y_t - A_t S_t)}{2\sigma^2(t)} \right]. \tag{5}
\]

If we specify a prior for the signal vectors and the noise variance, then we can integrate them from the problem. The prior that assumes the least about the signals is a uniform improper prior. Setting this prior to unity and integrating over the space of \( S_t \) leads to an expression with a multiplicative term \( |A_t^H A_t|^{-1} \). If two targets have closely spaced DOAs, then two columns of \( A_t \) become similar. This means that the matrix \( A_t^H A_t \) becomes nearly singular and the inverse of its determinant very large. This feature is avoided if we use the following prior for \( S_t \):

\[
p(S_t|A_t) \propto |A_t^H A_t|.
\]

The prior we use for the noise variance is a Jeffrey’s prior of the form \( 1/\sigma^2 \), which leads, on marginalization over the noise variance and signals, to the likelihood

\[
p(Y_t|A_t) \propto [Y_t^H (I_{MF} - A_t (A_t^H A_t)^{-1} A_t^H) Y_t]^{-MF - K} \tag{6}.
\]

### III. Tracking Algorithm

The distribution of interest is \( p(X_{0:t}|Y_{0:t}) \), where the subscripts denote the collection of parameters from time 0 to time \( t \), and we propose to model and track this using a particle filter algorithm. The output of this allows easy computation of the marginal distribution \( p(\theta(t)|Y_{0:t}) \), which is usually of most interest. First, however, we describe the generic particle filter and then, in two subsections, specific implementations of it that have been applied to the tracking problem we have described in the previous section.

The generic particle filter consists of three stages. First, we use the set of particles from the previous time-step to propose a new set for the new time step. These are then weighted to ensure consistency. Finally, the weighted particles may be resampled to convert them into an unweighted set without changing the distribution they represent. This is described in more detail later.

**Algorithm 1—Generic Particle Filter Algorithm:**

- **At time \( t = 0 \) initialize all particles:**
  - For \( i = 1, \ldots, N \), sample \( x_0^{(i)} \sim \pi(x_0|y_0) \)
  - For \( i = 1, \ldots, N \), evaluate un-normalized weights:
    \[
    w_0^{(i)} = \frac{p(y_0|x_0^{(i)}) p(x_0^{(i)})}{\pi(x_0^{(i)}|y_0)}
    \]
    - For \( i = 1, \ldots, N \), normalize weights:
    \[
    w_0^{(i)} = \frac{w_0^{(i)}}{\sum_{j=1}^{N} w_0^{(j)}}
    \]

- **For times \( t > 0 \):**
  - For \( i = 1, \ldots, N \), sample \( x_t^{(i)} \sim \pi(x_t|x_{t-1}^{(i)}, y_t) \) and set \( x_t^{(i)} = \{x_{t-1}^{(i)}, x_t^{(i)}\} \)
  - For \( i = 1, \ldots, N \), evaluate un-normalized weights:
    \[
    w_t^{(i)} = w_{t-1}^{(i)} p(y_t|x_t^{(i)})
    \]
    - For \( i = 1, \ldots, N \), normalize weights:
    \[
    w_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}
    \]
  - Resample particles:
    - For \( i = 1, \ldots, N \), sample an index \( j^{(i)} \) distributed according to \( \mathbb{P}(j^{(i)} = m) = w_t^{(m)} \)
    - For \( m = 1, \ldots, N \), then set \( x_{(t-1)}^{(i)} = x_{(0)}^{(j^{(i)})} \) and \( w_{(t-1)}^{(i)} = 1/N \).

The function \( \pi(x_t|x_{t-1}, y_t) \) is known as the importance function, and it is this that allows us to modify the algorithm for various applications. It can be shown [13] that the optimal importance function, in the sense that it minimizes the variance of the weights, is \( p(x_t|x_{t-1}, y_t) \). Computing various marginal distributions derived from the posterior is straightforward. The distribution \( p(X_{t-1}|Y_{0:t}) \) for \( 0 \leq \tau \leq t \) is modeled by \( \{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N} \). Special cases of this are the filtering distribution when \( \tau = t \) and the fixed-interval smoothing distribution when \( 0 \leq \tau < t \). We will be interested exclusively with the filtering distribution, which means that there is no need to store in memory the whole trajectory for each particle.

In the current implementation, the following functions are used in the above algorithm:

\[
p(y_t|x_t^{(i)}) \equiv p(Y_t|X_t^{(i)})
\]

\[
p(x_t^{(i)}|x_{t-1}^{(i)}) \equiv p(X_t^{(i)}|X_{t-1}^{(i)})
\]

**A. Choice of Importance Function**

The simplest importance function is the prior \( p(x_t|x_{t-1}) \), which leads to a very simple algorithm since the particle weights are then given by the likelihood only. The disadvantage of this is that we take no account of the current measurements when proposing new target states. If the likelihood is very narrow with respect to the prior, only a few particles will receive significant weights, the rest being essentially redundant. The optimal importance function relieves this problem since points are proposed conditional on both the previous state and the current measurement. The model we have proposed does not allow us to sample directly from the optimal importance function, but we propose instead to sample from a Gaussian approximation to the optimal importance function. If this approximation is good, then we expect to decrease the number of redundant particles at each step so that fewer particles are required overall.
The optimal importance function is proportional to $p(y_t|x_t)p(x_t|x_{t-1})$, the proportionality being independent of $x_t$. The algebraic property that defines a Gaussian is that the Taylor series of the logarithm of its density function is zero after the quadratic term. We can thus approximate the optimal importance function locally by computing the gradient and Hessian at a point close to its mode. These terms are then used to define a Gaussian from which we can easily sample.

Let $L_{yi}(x_t) = \log p(y_t|x_t)$ and $L_{xi}(x_t) = \log p(x_t|x_{t-1})$; then, the log of the optimal importance function is $L(x_t) = L_{yi}(x_t) + L_{xi}(x_t)$. We use a second-order Taylor expansion about the sensibly chosen point $\mathbf{x}$ to give

$$L(x_t) \approx L_{yi}(x) + L_{xi}(x) + \nabla L_{yi}(x)T(x_t - x) + \frac{1}{2}(x_t - x)^T\nabla^2 L_{yi}(x)(x_t - x)$$

which defines a Gaussian distribution with the following covariance and mean:

$$\Sigma(x) = -\left(\nabla^2 L_{yi}(x) + \nabla^2 L_{xi}(x)\right)^{-1}$$

$$m(x) = x + \Sigma(x)\nabla L_{yi}(x) + \nabla L_{xi}(x)$$

which we use as the importance function. The point $\mathbf{x}$ should be as close as possible to the mode of the optimal importance function so that the approximation models the true density as closely as possible. A convenient method for finding suitable values of $\mathbf{x}$ is to generate them deterministically by propagating the previous state through the dynamic model with all the variances set to zero. Details of the calculation of $\nabla L_{xi}(x)$ and $\nabla^2 L_{xi}(x)$ are given in the Appendix. For the terms $\nabla L_{yi}(x)$ and $\nabla^2 L_{yi}(x)$, we rely on an existing calculation to be found in [5], except that we must make a few modifications. In [5], the gradient and Hessian of a term $p(x)$ is computed, and $p(x)$ is related to $p(y)$ by

Thus

$$\nabla L_{yi}(x) = -N(P - K)\frac{\nabla J}{J}$$

and

$$\nabla^2 L_{yi}(x) = -N(P - K)\frac{J\nabla^2 J - \nabla J \nabla J^T}{J^2}.$$
2) For times $t > 0$:
- Decide on a partitioning such that
  \[ x_t \equiv \{x_1(1) \mid \cdots \mid x_{N_t}(K_t)\}. \]
- For $i = 1, \ldots, N$, and $k = 1, \ldots, K_t$
  - Sample $x^* \sim \pi_k(x_t(k)|x_{t-1}(k), y_t)$
  - Set $x^{\mu_{k}(i)}_t(k) = \{x^* \mid x_{t-1}(k)\}$
  - Compute $\rho^{\mu_{k}(i)}_t(k) = q_k(x^*)$, where $q_k$ is the
    partition weighting function (see below for description).
- For $k = 1, \ldots, K_t$, and $i = 1, \ldots, N$
  Normalize partition weights:
  \[ \rho^{\mu_{k}(i)}_t(k) = \frac{\rho^{\mu_{k}(i)}_t(k)}{\sum_{j=1}^{N} \rho^{\mu_{k}(j)}_t(k)}. \]
- Sample an index $j_k(i)$ from the distribution
  formed by $\{\rho^{\mu_{k}(i)}_t(k)\}_{i=1}^{N}$
- Set $x_{t+1}(i) \equiv \{x_{t+1}(1) \mid \cdots \mid x_{t+1}(K_t)\}$
- Compute particle weight:
  \[ w^{(i)}_t = \frac{p(y_t|\pi^{(i)}_t)}{\pi(\pi^{(i)}_t|y_t) \prod_{k=1}^{K} \rho^{\mu_{k}(i)}_t(k)}. \]
- For $i = 1, \ldots, N$, normalize weights:
  \[ w^{(i)}_t = \frac{w^{(i)}_t}{\sum_{j=1}^{N} w^{(j)}_t}. \]
- Resample particles: For $i = 1, \ldots, N$, sample an index $j(i)$ distributed according
  to $P(j(i) = m) = w^{(m)}_t$ for $m = 1, \ldots, N$.
  Then set $x^{(i)}_t = x^{j(i)}_t$ and $w^{(i)}_t = 1/N$.

The function $\pi_k(\cdot)$ is the importance function for the $k$th partition so that it is assumed that the importance function may be written as $\pi(\cdot) = \prod_{k=1}^{K} \pi_k(\cdot)$. We must still specify the weighted resampling functions $q_k(\cdot)$. In fact, any function is asymptotically acceptable, but we wish to choose a function that has advantages for our application. If we can find a function for each partition that is peaked where the likelihood is peaked for that partition, then at the selection step, we are more likely to choose partition states that will contribute to particles with high likelihoods and thus survive the final resampling step. A set of functions that achieve this are cross-sections through the likelihood parallel to the axis for each partition, going through the maximum of the likelihood. We therefore need to know the maximum likelihood state at each time step. We propose using a smaller secondary particle filter to find the maximum likelihood state by propagating every $n$th particle using just the dynamic prior and computing the likelihood for each of these particles. The particle with the highest likelihood is used as an estimate of the maximum likelihood state. The important property of this method is that as the number of particles tends to infinity the estimate of the maximum likelihood state will tend to the true value with probability 1. If required the estimate can be refined using a Newton descent algorithm based on the following recursion, although care must be taken to ensure that any divergence is detected and the step length $\mu$ decreased appropriately:

\[ x^{(i)}_{t+1} = x^{(i)}_t - \mu \nabla^2 L_y(x^{(i)}_t)^{-1} \nabla L_y(x^{(i)}_t). \]

The relevant gradients and Hessians have already been calculated for use with the approximated optimal importance function. The Hessian at the maximum likelihood estimate gives an estimate of the covariance at the same point. This is used to deduce an appropriate partitioning for that time step by seeing if there are any strong correlations indicated by the covariance.

It should be noted that this algorithm gives an unbiased description of the posterior distribution $p(x_t|y_{t:T})$ only if the partitioning is the same for all $t$. In reality, this is not usually the case, and we will introduce some bias to our estimates, in particular, lagged estimates such as $p(x_{t-1}|y_{t:T})$. However, in practice, this bias has been found to be small.

C. Optional MCMC Step

After each resampling step, it is almost inevitable that some particles will be exact copies of other particles. The limit of this is that all the particles are the same with the result that the sample variance falls to zero. If we apply one or more MCMC transitions whose invariant distribution is the posterior of interest, then we can be almost certain that every particle is distinct. Moreover, it can be shown that the distribution represented by the particles can only get closer to the true posterior using this method.

D. Initialization

The particle filter is essentially a method of updating a set of samples from $p(x_0|y_{0:T})$ to become samples from $p(x_{t+1}|y_{t+1})$, and this relies on the accuracy of the input sample set. Usually, the filter would be initialized by sampling from the prior or from some initial importance function and weighting in the usual manner, but this is only feasible if the sampling function can be constructed to be reasonably informative over the whole state space. If it is not, then the filter would still operate, but rejection rates in the early stages would be very high. In this application, we can only sample informatively from the whole state space if we initialize the filter with $p(x_0|y_{0:s})$ with $s > 0$. This is because the velocity and heading parameters cannot be inferred directly from the data so that a sequence of data is required for their estimation. MCMC techniques allow us to easily sample from arbitrary distributions, and a Gibbs sampler is used to generate the initial samples. Each Gibbs step is for a single target with Metropolis–Hastings steps used to propose points within each Gibbs step.

IV. RESULTS OF SIMULATIONS

In order to ascertain the effectiveness of the algorithm, data is simulated according to the models given and the algorithm output checked against the known trajectories. The robustness
of the algorithm is indicated by the survival rate of the resampling step. If the resampling generates many copies of only a few particles, we are less likely to have particles that will predict the future well; therefore, we run the risk of the filter diverging from the true posterior. A survival rate of between 30–80% is generally seen as acceptable. The tracking algorithm was tested in three different circumstances to demonstrate the effectiveness of the choice of importance function and use of the independent partition particle filter (IPPF) algorithm. The filters were initialized with the MCMC algorithm described above.

### A. Tracking With Matched Likelihood and Prior

If the width of the prior and likelihood distributions are similar, then particles tend to be weighted relatively uniformly. In this case, survival rates on resampling are expected to be stable and within acceptable limits. With the following parameter set, the likelihood and prior have similar widths with respect to the DOA, with typical deviations of 0.13° and 0.15°, respectively, as shown in Table I.

The parameter $R/\lambda_0$ is chosen so that the spacing between elements is just under half a wavelength (0.45) when there are 20 elements. The standard algorithm of Section III was used with both the approximation to the optimal importance function and the prior density (bootstrap filter) as importance function. We generated 100 randomly initialized simulations of 150 time steps with a single target. Both algorithms were used to track the posterior density, and the estimated posterior mean DOA was verified to be within ±3 standard deviations of the true value. The particle survival rate was measured at each step, and a summary of this is shown in Fig. 1. The mean survival rate is very similar for both choices of importance function, but the spread of values is significantly greater for the bootstrap filter.

### B. Tracking With Unmatched Likelihood and Prior

If either the prior variance becomes large, the likelihood becomes narrow, or both, the bootstrap filter loses robustness, whereas the approximation of the optimal importance function does not. We consider a scenario where the prior deviation on the heading direction is large, and the width of the likelihood is narrow: approximately 0.3° and 0.03°, respectively. This latter condition is achieved by collecting more measurements per estimate. The alterations to the parameter set of Section IV-A was used, as shown in Table I.

Again, we show results of particle survival rates over 100 simulations using the bootstrap filter and the approximate optimal importance function; see Fig. 2. It is clear that the approximate importance function continues to perform well, whereas the bootstrap filter suffers quite badly with frequent near-zero survival rates.

### C. Tracking Multiple Targets

The independent partition particle filter (IPPF) is intended to improve particle survival rates when there are multiple targets, meaning that we can use fewer particles while maintaining robustness. To test this, we compare particle survival rates when tracking four targets using the standard filter with the approximate optimal importance function and the IPPF with the optimal importance function as well. The alterations to the parameter set of Section IV-A were used, as shown in Table III. Again, we tracked 100 simulations with both filters and the particle survival rates recorded. Fig. 3 shows the IPPF output for a typical
TABLE III
ALTERATIONS TO THE PARAMETER SET

| number of measurements per estimate, N | 50 |
| number of particles | 400 |
| $\theta$ process noise | $(0.25\sigma^2)^2$ |
| $\phi$ process noise | $(2\sigma^2)^2$ |

Fig. 3. Filter output using IPPF with four targets and 400 particles. Dark lines are posterior mean DOA estimates, and faint lines are 95% confidence limits. Dots are ground truth.

Fig. 4. Same as Fig. 1 but with four targets, using IPPF (dark) and standard filter (faint).

example where only 50 of the usual 150 are shown for clarity. Fig. 4 shows survival rates for the two filters, where it can be seen that the IPPF maintains survival rates close to the single target case and has a relatively small spread, whereas the standard filter has reduced survival rates but, more importantly, a very large spread in their values, regularly falling to quite low levels. For the standard filter, about 50% of attempted simulations diverged from the correct posterior, whereas only 7% of the IPPF simulations suffered the same fate.

V. CONCLUSIONS

The Bayesian model proposed allows us to robustly track objects moving with realistic trajectories while automatically maintaining track association. Particle filters allow us to track the posterior, and by approximating the optimal importance function with a Gaussian, we can improve particle survival rates so that fewer particles are needed to ensure robustness. The independent partition filter allows the tracking of multiple targets with little deterioration in the particle survival rates and very few occurrences of dangerously low survival rates. The extended Kalman filter (EKF) has been applied to a similar scenario with the same prior and a likelihood where we assume the target signals are known. Results using the EKF are inferior to those using the algorithm described here, despite the extra assumption of known signals, although there is no room to show the results here.

APPENDIX
DERIVATION OF THE PRIOR DENSITY

We give here a derivation of the explicit form of the prior for one target and the gradient and Hessian of the logarithm of this. For notational efficiency, we write $\theta_t(t)$ as $\theta_t$ and similarly for all other terms. Let $Q_t = \log q_t$ so that

$$
\theta_{t+T} = \arctan \left( \frac{\sin \theta_t + e^{Q_t} T \cos (\phi_{t+T})}{\cos \theta_t + e^{Q_t} T \sin (\phi_{t+T})} \right) + u_{\theta, t+T}
$$

(8)

$$
Q_{t+T} = Q_t - \frac{1}{2} \log \left( 1 + 2e^{Q_t} T \sin (\theta_t + \phi_{t+T}) + (e^{Q_t} T)^2 \right) + u_{Q, t+T}
$$

(9)

$$
\phi_{t+T} = \phi_t + u_{\phi, t+T}.
$$

(10)

Thus, the new state is a deterministic function of the old state and a random vector $u_t = [u_{\theta, t} \ u_{\phi, t} \ u_{Q, t}]^T$. That is, $x_{t+T} = f(x_t, u_{t+T})$, and therefore

$$
p(x_{t+T} | x_t) = \frac{p(u_{t+T})}{\left[ \frac{\partial f}{\partial u} \right]_{u = f^{-1}(x_{t+T}, x_t)}}.
$$

However

$$
\left[ \frac{\partial f}{\partial u} \right] = \begin{bmatrix}
1 & 0 & \frac{\partial \theta_{t+T}}{\partial u_{\theta, t+T}} \\
0 & 1 & \frac{\partial \phi_{t+T}}{\partial u_{\phi, t+T}} \\
0 & 0 & 1
\end{bmatrix} = 1
$$

so that $p(x_{t+T} | x_t) = p(u_{t+T})$. The components of $u_{t+T}$ are assumed to be independent so that the logarithm of the prior, neglecting a constant term, is

$$
L_x(x_t) = \frac{1}{2} \left( \frac{u_{\theta, t}^2}{\sigma_{\theta}^2} + \frac{u_{\phi, t}^2}{\sigma_{\phi}^2} + \frac{u_{Q, t}^2}{\sigma_{Q}^2} \right).
$$

Equations (8)–(10) give $u_{\theta, t}$, $u_{\phi, t}$, and $u_{Q, t}$, directly. The first and second derivative terms are given, in (11)–(19), at the top of the next page.

ACKNOWLEDGMENT

The authors would like to thank Dr. C. Andrieu and Dr. A. Doucet for many helpful discussions.
\[
\begin{align*}
\frac{\partial L_x}{\partial \theta_{t+T}} &= \frac{1}{\sigma^2} \left[ \theta_{t+T} - \arctan \left( \frac{\sin \theta_t + e^{\omega T} \cos (\phi_{t+T})}{\cos \theta_t + e^{\omega T} \sin (\phi_{t+T})} \right) \right] \\
\frac{\partial^2 L_x}{\partial Q_{t+T} \partial \theta_{t+T}} &= \frac{1}{\sigma^2} \left[ Q_{t+T} - Q_t + \frac{1}{2} \log \left( 1 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T}) + (e^{\omega T})^2 \right) \right] \\
\frac{\partial L_x}{\partial \theta_{t+T}} &= \frac{\partial L_x}{\partial Q_{t+T}} \left[ \frac{(e^{\omega T})^2 + e^{\omega T} \sin (\theta_t + \phi_{t+T})}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \\
&\quad + \frac{\partial L_x}{\partial Q_{t+T}} \left[ \frac{e^{\omega T} \cos (\theta_t + \phi_{t+T})}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \left[ \frac{1}{\sigma^2} (\phi_{t+T} - \phi_t) \right] \\
\frac{\partial^2 L_x}{\partial Q_{t+T}^2} &= \frac{1}{\sigma^2} \\
\frac{\partial^2 L_x}{\partial Q_{t+T} \partial \theta_{t+T}} &= \frac{1}{\sigma^2} \left[ \frac{(e^{\omega T})^2 + e^{\omega T} \sin (\theta_t + \phi_{t+T})}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \\
\frac{\partial^2 L_x}{\partial \theta_{t+T}^2} &= \frac{1}{\sigma^2} \left[ \frac{e^{\omega T} \cos (\theta_t + \phi_{t+T})}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \\
\frac{\partial L_x}{\partial Q_{t+T}} &= \frac{\partial L_x}{\partial \theta_{t+T}} \left[ \frac{(e^{\omega T})^2 + e^{\omega T} \sin (\theta_t + \phi_{t+T})}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \\
&\quad + \frac{\partial L_x}{\partial \theta_{t+T}} \left[ \frac{e^{\omega T} \cos (\theta_t + \phi_{t+T})(1 - (e^{\omega T})^2)}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \\
&+ \frac{\partial^2 L_x}{\partial \theta_{t+T}^2} \left[ \frac{e^{\omega T} \cos (\theta_t + \phi_{t+T})}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})} \right] \\
&\quad - \frac{\partial L_x}{\partial Q_{t+T}} \left[ \frac{e^{\omega T} \sin (\theta_t + \phi_{t+T})(1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T}))^2}{1 + (e^{\omega T})^2 + 2 e^{\omega T} \sin (\theta_t + \phi_{t+T})^2} \right] \left[ \frac{1}{\sigma^2} \right] \\
\end{align*}
\]

References


Matthew Orton was born in Chelmsford, U.K., on May 23, 1977. He received the B.A. and M.Eng. degrees in electrical and information sciences from the University of Cambridge, Cambridge, U.K., in 1999. He is currently pursuing the Ph.D. degree with the Signal Processing Group, Department of Engineering, Cambridge University.

His current research interests include Bayesian target tracking, particle filters, tracking with out-of-sequence measurements, and on-line changepoint detection.

William Fitzgerald received the B.Sc., M.Sc., and Ph.D. degrees in physics from the University of Birmingham, Birmingham, U.K.

He is the Head of the Statistical Modeling Group within the Signal Processing Laboratory, Department of Engineering, University of Cambridge, Cambridge, U.K., where he is also University Reader in Statistics and Signal Processing. Prior to joining Cambridge, he worked at the Institut Laue Langevin, Grenoble, France for seven years and was then a Professor of Physics at the Swiss federal Institute of Technology (ETH), Zurich, for five years. He is a Fellow of Christ’s College, Cambridge, where he is Director of Studies in Engineering. He works on Bayesian inference applied to signal and data modeling. He is also interested in nonlinear signal processing, image restoration and extreme value statistics.