NON-STATIONARY BAYESIAN DIRECTION OF ARRIVAL ESTIMATION WITH DRIFTING SENSOR LOCATIONS

W D Addison and M D MacLeod

QinetiQ Ltd
St Andrew's Road, Malvern, WR14 3PS, UK
phone: + (44) 1684 543802, fax: + (44) 896592 , email: {wdaddison,mmacleod}@qinetiq.com

ABSTRACT

We describe a new tracking algorithm for the direction of arrival estimation problem where both the locations of the sensors in the array and the directions of arrival are non-stationary. The approach taken is Bayesian. The algorithm assumes that the filtering distribution is approximately Gaussian and maintains the mean and covariance of this approximation by fitting a quadratic surface to the log posterior around the location where the log posterior is maximized. In the case where the sensor locations are stationary, the algorithm is shown to have similar performance to particle filter based algorithms but at a reduced computational cost. In the case where the sensor locations are non-stationary particle filtering is unsuccessful and the new algorithm performs significantly better than currently existing algorithms.

1. INTRODUCTION

Direction of arrival (DOA) estimation is the use of measurements from a sensor array to estimate the direction of arrival of incoming signals. It is a widely employed technique used in sonar, acoustic localization, electronic surveillance, and phased array radar, among others. However, most algorithms assume that the directions of arrival are static and that the locations of the sensors within the array are known. This requires the array to be accurately calibrated, which may be expensive and is not always possible. Even if an array has been well calibrated it is possible for the sensors to drift over time causing the calibration to lose accuracy. In this paper we tackle the problem of online narrow band DOA estimation, where the DOAs are non-stationary, and where the sensor locations are also non-stationary and unknown.

The rest of this section gives an overview of the current state of the art with regard to the non-stationary DOA estimation problem. Section 2 describes the model we work with and introduces the notation for the rest of the paper. Section 3 then describes a new filtering algorithm that has been developed for the problem. This algorithm maintains a Gaussian approximation to the filtering distribution. Section 4 provides results from applying the algorithm to some simulated data. Conclusions about the new algorithm are drawn in section 5.

1.1 Background

Two commonly used tools for online estimation of dynamic parameters are the Kalman filter and the particle filter. Both have been employed to tackle the DOA estimation problem. Because the Kalman filter relies on the measurement model being linear and Gaussian, all the approaches using the Kalman filter require the data to be broken into windows in which the DOAs are assumed to be stationary. In the context of the Kalman filter, the data from each window is then assumed to form a single 'measurement'. This allows information from multiple data points to be combined, which yields a much more Gaussian likelihood for the DOAs, allowing the Kalman filter to be successfully applied. The approaches vary in how they calculate the mean of the 'measurement' formed by each window. In [6] an iterative algorithm for updating the signal subspace is used. The mean of the measurement is then found by using an optimization routine to find the DOAs which best fit the signal subspace. In [3] a maximum likelihood approach is taken to finding the mean of each 'measurement'. For the covariance of the error, both approaches use the Cramer Rao bound which is a theoretic lower bound on the actual covariance of the error. In order to track rapidly varying parameters a short window will be needed in order for the assumption of stationarity within the window to be valid. However, this will mean that the Gaussian approximation to the measurement error and the Cramer Rao bound approximation to its covariance will be poorer. Indeed, the Cramer Rao bound can be highly over-optimistic about the error covariance even with quite large windows, which leads to the Kalman filter placing too much credence on the unreliable 'measurements'. This limits the performance of these approaches.

In order to avoid the need to make the approximations required to run the Kalman filter, several particle filter approaches have been proposed [4], [2]. Because of the ability of particle filters to cope with non-linear non-Gaussian measurement models such as the one involved in the DOA model, only [4] takes the step of breaking the data into windows. However, these approaches are very computationally expensive. Furthermore, particle filters are known to struggle when tracking slowly varying parameters, such as the slowly drifting sensor locations.

There are very few approaches in the literature to simultaneous tracking of DOAs and non-stationary sensor locations. Of the approaches mentioned above only [3] performs tracking of the sensor locations.

2. DIRECTION OF ARRIVAL MODEL

In this section we define a model for how the data is generated in the DOA estimation problem. There are two components to the model. The first is the observation model, which describes how the observed data is generated at a single time instant. The second is the transition model which describes how the parameters of the model evolve with time.

2.1 Notation

We introduce the following notation.

- M is the number of sensors.
- $\mathbf{y}_t$ is the vector of complex valued observations made at each of the sensors in the array at time t. In other words $(\mathbf{y}_t)_j$ is the value observed by the j^{th} sensor at time t.
- X_t is an $M \times 2$ matrix containing the positions of the sensors at time t. So $(X_t)_{ji}$ is the location of the j^{th} sensor in the i^{th} dimension.
- $\mathbf{b}_t$ is a complex vector containing the amplitude and phase offsets for each of the sensors at time t.
- $\boldsymbol{\theta}_t$ is a vector containing the angles of arrival for each of the sources at time t.
- $\dot{\boldsymbol{\theta}}_t$ is a vector containing the rate of change of $\boldsymbol{\theta}_t$.
- $\mathbf{s}_t$ is the vector containing the complex amplitudes of each of the sources at time t.
- $\boldsymbol{\delta}_t$ is a vector containing the standard deviations of the distributions of the source amplitudes at time t.
- $\boldsymbol{\alpha}_t = \log \boldsymbol{\delta}_t$. This will be needed for defining the transition model.
- ν is the wave number of the narrowband signals that are being measured. In other words $\nu = (2\pi\omega_0)/c$ where ω_0 is the central frequency of the signals and c is their propagation velocity.
- σ is the standard deviation of the sensor noise.
- $\boldsymbol{\phi}_t$ is a vector containing the concatenation of the parameters that we wish to track. This will always contain $\boldsymbol{\theta}_t$, $\dot{\boldsymbol{\theta}}_t$ and $\boldsymbol{\alpha}_t$, and in the case where we wish to track the sensor locations it will contain X_t as well.

2.2 Observation model

We use the standard DOA estimation model. This models the values of the narrow-band signals as complex values which represent the phase and the amplitude of the signals. This means that the observed signal is modelled as

$$\mathbf{y}_t = D_t \mathbf{s}_t + \boldsymbol{\xi}_t$$

where $\boldsymbol{\xi}_t$ is the additive noise assumed to be zero-mean complex Gaussian with covariance $\sigma^2 I$. D_t is a matrix of phase shifts which represent the time delays across the sensors due to the angle of arrival of the incoming wave. It is given by

$$\begin{aligned} (D_t)_{jk} &= (\mathbf{b}_t)_j \exp(i 2\pi\omega_0 \Psi_{jk}) \\ \Psi_{jk} &= (X_{j1} - X_{11}) sin\theta_k - (X_{j2} - X_{12}) cos\theta_k. \end{aligned}$$

The subscript t has been omitted from the last equation for notational simplicity.

2.2.1 Priors

We place a complex Gaussian prior distribution on the values of the source amplitudes:

$$p(\mathbf{s}_t) \sim \mathscr{CG}(0, K_t)$$

where $\mathscr{CG}(\mu, \Sigma)$ is used to denote a complex Gaussian distribution with mean μ and covariance Σ. We take K_t to have the form:

$$K_t = \mathrm{diag}(\boldsymbol{\delta}_t^2)$$

where $\mathrm{diag}(\boldsymbol{\nu})$ is the diagonal matrix formed by placing the elements of $\boldsymbol{\nu}$ along the diagonal.

Since $\mathbf{s}_t$ is Gaussian it can easily be marginalized out of the likelihood for the observation model leaving

$$\mathbf{y_t} \sim \mathscr{CG}(0, W_t), \tag{1}$$

where $W_t = D_t K_t D_t^\dagger + \sigma^2 I$.

2.3 Transition model

Both the parameters relating to the sources and to the sensors evolve over time. Here we define transition models for how the parameters evolve from one time instant to the next.

The angles of arrival are modelled by a constant velocity model. So

$$\left[\begin{array}{c} \boldsymbol{\theta}_{t+1} \\ \dot{\boldsymbol{\theta}}_{t+1} \end{array}\right] \sim N\left(\left[\begin{array}{c} \boldsymbol{\theta}_t + \dot{\boldsymbol{\theta}}_t \\ \dot{\boldsymbol{\theta}}_t \end{array}\right], q_\theta \Sigma_\theta \right).$$

In our experiments we use

$$\Sigma_\theta = \left(\begin{array}{cc} \frac{1}{3} I & \frac{1}{2} I \\ \frac{1}{2} I & I \end{array} \right),$$

which is based on the standard structure for the innovation covariance for a constant velocity model. q_θ is a parameter that must be chosen so that the model most accurately reflects the expected dynamics of the DOAs.

In order to form a transition model for the source amplitude standard deviations we take the same approach as [1]. We let $\boldsymbol{\alpha}_t$ follow a Gaussian random walk rather than defining a transition model on $\boldsymbol{\delta}_t$ itself. So

$$\boldsymbol{\alpha}_{t+1} \sim N(\boldsymbol{\alpha}_t, q_\alpha I),$$

where q_α is a parameter that can be chosen according to the expected rate of change of $\boldsymbol{\alpha}$.

The sensor locations are also modelled as following a Gaussian random walk. So

$$(X_{t+1})_{ji} \sim N\left((X_t)_{ji}, q_X\right),$$

where q_X is a parameter that can be chosen according to the expected rate of change of the sensor locations.

Note that the transition model for the parameters is entirely linear-Gaussian. For notational convenience we write

$$\boldsymbol{\phi}_{t+1} \sim N(A\boldsymbol{\phi}_t, Q), \tag{2}$$

where A and Q can be derived from the transition models for the individual components of $\boldsymbol{\phi}_t$.

3. ALGORITHM

In this section we propose a new filtering algorithm for use with the model described above. The Extended Kalman Filter (EKF) is often applied to the problem of estimating non-stationary parameters, when the observation model is non-linear. The EKF operates by maintaining a Gaussian approximation to the filtering distribution even though the true filtering distribution is non-Gaussian. Due to the nature of the non-linearity of our model, the EKF cannot be applied. So in this section we present a new algorithmic approach for maintaining a Gaussian approximation to the filtering distribution. The approach taken to obtain the approximation is the same as that taken to obtain the Gaussian approximation that is used by the Laplacian approximation of integrals. That is, the peak of the distribution to be approximated is found and taken as the mean of the Gaussian approximation. Then the Hessian of the negative log of the distribution is calculated at its peak and this is taken as the inverse covariance of the approximation.

3.1 Recursive Approximation of Filtering Distribution

Like the EKF, our algorithm recursively calculates an approximation to the filtering distribution, $p(\boldsymbol{\phi}_t|\mathbf{y}_{1:t})$, where $\mathbf{y}_{1:t}$ denotes all the data observed up to and including time t. The approximation at time t is based on an update of the approximation at time $t-1$. So here we present an approximation to $p(\boldsymbol{\phi}_t|\mathbf{y}_{1:t})$ in terms of the approximation to $p(\boldsymbol{\phi}_{t-1}|\mathbf{y}_{1:t-1})$. Let $\mathbf{m}_t$ and C_t be the mean and covariance of the Gaussian approximation to $p(\boldsymbol{\phi}_t|\mathbf{y}_{1:t})$, which we refer to as the filtering parameters. Let $\tilde{p}(\boldsymbol{\phi}_t|\mathbf{m}_t, C_t)$ represent the Gaussian approximation to $p(\boldsymbol{\phi}_t|\mathbf{y}_{1:t})$. Then

$$\begin{aligned}
& p(\boldsymbol{\phi}_t|\mathbf{y}_{1:t}) \\
\propto\ & p(\mathbf{y}_t|\boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t|\mathbf{y}_{1:t-1}) \\
=\ & p(\mathbf{y}_t|\boldsymbol{\phi}_t)\int p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})p(\boldsymbol{\phi}_{t-1}|\mathbf{y}_{1:t-1})d\boldsymbol{\phi}_{t-1} \\
\approx\ & p(\mathbf{y}_t|\boldsymbol{\phi}_t)\int p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})\tilde{p}(\boldsymbol{\phi}_{t-1}|\mathbf{m}_{t-1}, C_{t-1})d\boldsymbol{\phi}_{t-1}
\end{aligned}$$

It is useful to define

$$\begin{aligned}
& \tilde{p}(\boldsymbol{\phi}_t|\mathbf{m}_{t-1}, C_{t-1}) \\
=\ & \int p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})\tilde{p}(\boldsymbol{\phi}_{t-1}|\mathbf{m}_{t-1}, C_{t-1})d\boldsymbol{\phi}_{t-1} \\
=\ & \frac{\exp(-\frac{1}{2}(\boldsymbol{\phi}_t - A\mathbf{m}_{t-1})^{\mathsf{T}} R_{t-1}^{-1}(\boldsymbol{\phi}_t - A\mathbf{m}_{t-1}))}{(2\pi)^{\frac{n}{2}}|R_{t-1}|^{\frac{1}{2}}}
\end{aligned}$$

Where $R_{t-1} = AC_{t-1}A^{\mathsf{T}} + Q$ and n is the dimension of $\boldsymbol{\phi}_t$. We now define $f_t(\boldsymbol{\phi}_t)$ to be the negative log of our approximation to the filtering distribution. So

$$\begin{aligned}
& f_t(\boldsymbol{\phi}_t) \\
=\ & -\log p(\mathbf{y}_t|\boldsymbol{\phi}_t) - \log \tilde{p}(\boldsymbol{\phi}_t|\mathbf{m}_{t-1}, C_{t-1}) \\
=\ & \log(\pi^M|W_t|) + \mathbf{y}_t^{\mathsf{T}} W_t^{-1}\mathbf{y}_t + \log((2\pi)^{\frac{n}{2}}|R_{t-1}|^{\frac{1}{2}}) \\
& + \tfrac{1}{2}(\boldsymbol{\phi}_t - A\mathbf{m}_{t-1})^{\mathsf{T}} R_{t-1}^{-1}(\boldsymbol{\phi}_t - A\mathbf{m}_{t-1}).
\end{aligned}$$

Note that while such a recursive approximation to the filtering distribution has the potential for errors to accumulate in such as way that the algorithm diverges, this is no more the case than for the EKF.

3.2 Finding the Filtering Parameters

Now that we can evaluate an approximation to the filtering distribution we are in a position to calculate the parameters of a Gaussian approximation to the filtering distribution. The mean, $\mathbf{m}_t$, is taken as the minimum of f_t. We apply Newton's method in order to find this minimum. That is, we use the iterative scheme

$$\boldsymbol{\phi}_t^{i+1} = \boldsymbol{\phi}_t^i - \gamma H_{\boldsymbol{\phi}_t^i}^{-1}\nabla f(\boldsymbol{\phi}_t^i) \tag{3}$$

where $\boldsymbol{\phi}_t^i$ is the value at the i^{th} iteration towards the minimum. $H_{\boldsymbol{\phi}_t^i}$ is the Hessian of f_t at $\boldsymbol{\phi}_t^i$. The values of the derivative of f_t are given in appendix A. A value of 1 is used for γ provided that it yields a reduction in f_t. If this is not the case then γ is reduced until such a reduction in f_t is achieved. This can be done simply by repeatedly halving γ, or more efficiently by using the line search algorithm described in [5]. We are guaranteed to find some $\gamma > 0$ such that a reduction in f_t is achieved, because the gradient of f_t in the direction of search is initially negative.

While this sounds like a potentially computationally expensive procedure, in practice this turns out not to be the case. First, the procedure is initialized from $A\mathbf{m}_{t-1}$, which is usually a very good initial estimate of the minimum. This also makes it unlikely that the algorithm will fall into a local minimum. Second, the filtering distribution is in practice close to being Gaussian. (If this were not the case then it would not be appropriate to apply this algorithm.) Therefore, the log of the filtering distribution is close to being quadratic, which means that the optimization procedure converges very rapidly to the solution. In the experiments reported below, the procedure generally converged after just three steps.

Once this procedure has converged we set $\mathbf{m}_t$ equal to the converged value and C_t equal to $H_{\mathbf{m}_t}^{-1}$. The algorithm then proceeds to the next observation. A pseudocode outline of the complete algorithm is given in table 1.

3.3 Discussion of the Proposed Algorithm

In general it is not easy to tell in advance whether the Gaussian approximation to the filtering distribution will be valid. However, our experiments suggest that for the problem being addressed in this paper, the approximation is reasonable. Note that the proposed algorithm needs to be initialized with both the number of sources that are present, and a good estimate of the initial directions of arrival of the sources. Without a good initialization, the assumption that the filtering distribution is approximately Gaussian for the initial stages, before much data has been observed, will not be valid. While it is important to be able to address both these aspects of the problem, it is beyond the scope of this paper to do so.

While a particle filter could be applied to the problem it would be computationally more costly. Also particle filters are known to struggle with slowly varying parameters like the sensor locations in our problem.

4. EXPERIMENTS

In this section we report the results of running the proposed algorithm on some simulated data, in order to demonstrate its performance. All the experiments are conducted with the same ground truth for the directions of arrival, θ, and the standard deviations of the source amplitudes, δ. The values that these variables took are given in the equations below.

$$\begin{aligned}
\theta_1(i) &= -20 + 25\cos\left(\frac{(\pi/8)(i-500)}{2000}\right). \\
\theta_2(i) &= 5.5 - \pi\frac{2000-i}{2000}. \\
\delta_1(i) &= 0.03 + 0.02\cos\left(\frac{0.88(i+1000)\pi}{1500}\right). \\
\delta_2(i) &= 0.02 - 0.01\sin\left(\frac{(i+1000)\pi}{1500}\right).
\end{aligned}$$

Set $\mathbf{m}_0$ and C_0 equal to the mean and covariance of the prior on $\boldsymbol{\phi}_0$;
for $t = 1$ *to* T **do**
  Set $\boldsymbol{\phi}_t^{new} = A\mathbf{m}_{t-1}$;
  repeat
    Set $\gamma = 1$;
    Set $\boldsymbol{\phi}_t^{old} = \boldsymbol{\phi}_t^{new}$;
    repeat
      Set $\boldsymbol{\phi}_t^{new} = \boldsymbol{\phi}_t^{old} - \gamma H^{-1}_{\boldsymbol{\phi}_t^{old}} \nabla f(\boldsymbol{\phi}_t^{old})$;
      Set $\gamma = \gamma/2$;
    until $f_t(\boldsymbol{\phi}_t^{new}) - f_t(\boldsymbol{\phi}_t^{old}) < -10^{-4}$;
  until $|\boldsymbol{\phi}_t^{new} - \boldsymbol{\phi}_t^{old}| < 10^{-4}$;
  Set $\mathbf{m}_t = \boldsymbol{\phi}_t^{new}$;
  Set $C_t = H^{-1}_{\mathbf{m}_t}$;
end

Table 1: Pseudocode outlining the proposed new filtering algorithm.

The data was generated for an array of five sensors arranged to form a cross. The coordinates of the sensors were

$$\begin{pmatrix} -0.3 & -0.3 \\ 0.3 & -0.3 \\ -0.3 & 0.3 \\ 0.3 & 0.3 \\ 0 & 0 \end{pmatrix}.$$

The propagation velocity of the signals was taken to be $340ms^{-1}$ and the carrier frequency was 224Hz.

The values of the parameters used by the algorithm were $q_\theta = 10^{-10}$ and $q_\alpha = 10^{-3}$. The value of σ used by the algorithm was always taken to be the value that was used when generating the data. The prior for the parameters to be tracked was always centered on their true initial values with a standard deviation of 10^{-3}.

4.1 Stationary Sensor Locations

We begin by running some experiments in a scenario with stationary sensor locations, where we assume that the sensor locations are known. We investigate the effect that different levels of sensor noise have on the performance of the algorithm. We also ran a particle filter as a benchmark for the performance of the algorithm. The particle filter does not make the approximation that the posterior distribution is Gaussian and therefore, provided enough particles are used, is the most accurate algorithm that is available. This accuracy is achieved at the expense of computational cost. The particle filter therefore serves as an upper bound on performance, and can be used to assess how much impact the approximations made by the new algorithm have on its performance.

Table 2 presents the accuracy achieved by the two algorithms. It can be seen that the performance of the new algorithm is not much below that of the particle filter, which demonstrates that it is working well. The new algorithm ran approximately eight times faster than the particle filter.

4.2 Non-stationary Sensor Locations

We now run some experiments to investigate the performance of the algorithm in tracking non-stationary positions of the

	σ			
	0.0006	0.00125	0.0025	0.005
New Filter	0.0021	0.0039	0.0072	0.0128
Particle Filter	0.0020	0.0038	0.0064	0.0109

Table 2: The average of the median absolute error for the estimate of the first DOA obtained across the length of the data set at the different noise levels. The results are the average over 50 data sets.

	q_X			
σ	static	10^{-6}	10^{-5}	10^{-4}
0.0025	0.0072	0.0152	0.0202	0.0204
0.02	0.0575	0.0628	0.0844	0.3941

Table 3: The average of the median absolute error for the estimate of the first DOA, obtained across the length of the data set, with different values of q_X and σ. The results are the average over 50 data sets. The performance of the static algorithm with stationary sensor locations is included for comparison.

sensors, and we compare the performance with that of the algorithm proposed in [3], which is the only approach to the problem we are aware of in the literature. It should be noted that in this case a particle filter cannot be successfully run on the problem because the sensor locations evolve much more slowly than the DOAs. Such a scenario causes particle filters to fail. When generating the data all but the first and last sensors were simulated according to the transition model used by the algorithm. Arbitrary translations of the array make no difference to the observed data, and arbitrary rotations of the array can be accounted for by a rotation of all the directions of arrival. Therefore, there is a fundamental translational and rotational ambiguity when the positions of the sensors are unknown. Fixing the locations of two of the sensors removes this ambiguity, which is why this is done for our experiments.

Figure 1 plots an example of the true and estimated values of the x-coordinates for the three moving sensors. This demonstrates that when the noise is reasonably low the algorithm is capable of tracking the sensor locations even when they are changing quite rapidly. Table 3 presents the accuracy obtained by the algorithm.

The behaviour of the algorithm is generally as one would expect, with the performance degrading as the rate at which the sensors drift or as the sensor noise increases. In general the algorithm works well and the effect of the moving sensor locations is well mitigated so that its impact on the estimate of the directions of arrival is not too severe. However, in the hardest case with $\sigma = 0.02$ and $q_X = 10^{-4}$ the algorithm fails to provide reliable information about the directions of arrival.

4.2.1 Comparison with Goldberg's Algorithm

The algorithm proposed by Goldberg in [3] was run on the same data as the new algorithm in order to benchmark its performance. It was run with window lengths 10, 50 and 100. The same transition model was used but it now describes the transition from one window to the next, rather than one time instant to the next; therefore the covariance matrix for the transition model was multiplied by the window length.

For the most benign case with $\sigma = 0.0025$ and $q_X = 10^{-6}$, the median value of the median absolute error in the es-

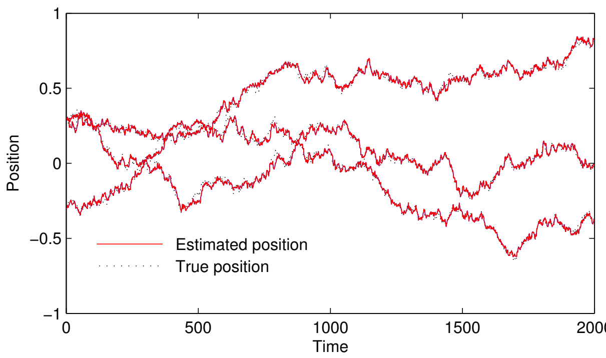

Figure 1: An example of the true and estimated x-coordinates of the three moving sensors. It is difficult to distinguish the true and estimated values on the plot as the two are so close. This plot is taken from the experiment that had the median RMSE of the 50 experiments performed with $\sigma = 0.0025$ and $q_X = 10^{-4}$.

timate of the first DOA was 0.10, 0.17 and 0.15 respectively for the three different window lengths. The corresponding value for our new algorithm was 0.01, so it can be seen that it performs significantly better.

5. CONCLUSIONS

We have proposed a new tracking algorithm for the problem of non-stationary narrow band direction of arrival estimation. The algorithm takes a fully Bayesian approach to the non-linear problem without the need for employing a particle filter. It is significantly computationally less expensive than a particle filter but results in comparable performance.

The algorithm has also been applied to the case where the locations of the sensors in the array are unknown and non-stationary. As long as the noise level is not too high and the sensors are not moving too quickly, the algorithm performs well, successfully tracking the sensor locations and yielding good estimates of the directions of arrival. The new algorithm has been shown to significantly outperform the only known approach to this problem in the literature.

The algorithm that has been developed has a very general form and has the potential to be applied to any non-linear non-Gaussian online estimation problem where the EKF or UKF cannot be successfully applied. An example of such a problem would be that of online non-stationary independent component analysis.

Acknowledgements The authors would like to thank K Weekes and S Maskell for their input and ideas. The research presented in this paper is based on research that was supported by the United Kingdom's MOD Corporate Research Program.

A. DERIVATIVES

In this appendix we provide the derivatives of f_t:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\phi}_t} f_t(\boldsymbol{\phi}_t) &= -\frac{\partial}{\partial \boldsymbol{\phi}_t} \log p(\mathbf{y}_t|\boldsymbol{\phi}_t) \\ &\quad -\frac{\partial}{\partial \boldsymbol{\phi}_t} \log \tilde{p}(\boldsymbol{\phi}_t|\mathbf{m}_{t-1}, C_{t-1}).\end{aligned}$$

Now

$$\frac{\partial}{\partial \boldsymbol{\phi}_t} \log \tilde{p}(\boldsymbol{\phi}_t|\mathbf{m}_{t-1}, C_{t-1}) = -R_t^{-1} A \mathbf{m}_{t-1}.$$

$$\begin{aligned}\left[\frac{\partial}{\partial \boldsymbol{\theta}_t} \log p(\mathbf{y}_t|\boldsymbol{\phi}_t)\right]_p &= 2\left[K_t D_t^\dagger W_t^{-1}(\mathbf{y}_t\mathbf{y}_t^\dagger W_t^{-1} - I) M_t\right]_{pp}. \\ \left[\frac{\partial}{\partial \boldsymbol{\alpha}_t} \log p(\mathbf{y}_t|\boldsymbol{\phi}_t)\right]_p &= 2\left[G_t D_t^\dagger W_t^{-1}(\mathbf{y}_t\mathbf{y}_t^\dagger W_t^{-1} - I) D_t G_t\right]_{pp}. \\ \left[\frac{\partial}{\partial X_t} \log p(\mathbf{y}_t|\boldsymbol{\phi}_t)\right]_{p1} &= 2\left[(P_t^1 K_t D_t^\dagger) W_t^{-1}(\mathbf{y}_t\mathbf{y}_t^\dagger W_t^{-1} - I)\right]_{pp}. \\ \left[\frac{\partial}{\partial X_t} \log p(\mathbf{y}_t|\boldsymbol{\phi}_t)\right]_{p2} &= 2\left[(P_t^2 K_t D_t^\dagger) W_t^{-1}(\mathbf{y}_t\mathbf{y}_t^\dagger W_t^{-1} - I)\right]_{pp}.\end{aligned}$$

where

$$\begin{aligned}P_{ij}^1 &= \mathbf{b}_i \exp(i\nu\Psi_{ij}) i\nu \sin\theta_j. \\ P_{ij}^2 &= -\mathbf{b}_i \exp(i\nu\Psi_{ij}) i\nu \cos\theta_j. \\ M_{ip} &= \mathbf{b}_i \exp(i\nu\Psi_{ip}) i\nu g(X_i, \theta_p). \\ g(X_i, \theta_p) &= (X_{i1} - X_{11})\cos\theta_p + (X_{i2} - X_{12})\sin\theta_p. \\ G &= \mathrm{diag}(\boldsymbol{\alpha}).\end{aligned}$$

Note that in the definitions above the subscript t has been dropped to simplify the notation.

Together this provides everything needed to calculate the gradient of f_t. The Hessian can be calculated numerically using these calculations for the first derivative.

REFERENCES

[1] Christophe Andrieu. Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing*, 51(7):1762–1770, July 2003.

[2] Volkan Cevhar and James H McClellan. General direction-of-arrival tracking with accoustic nodes. *IEEE Transactions on Signal Processing*, 53(1):1–12, 2005.

[3] J. M. Goldberg. Joint direction-of-arrival and array-shape tracking for multiple moving targets. *IEEE Journal of Oceanic Engineering*, 23(2):118–126, 1998.

[4] Matthew Orton and William Fitzgerald. A Bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, 50(2):216–223, February 2002.

[5] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes in C++*, chapter 9, pages 388–939. Cambridge University Press, second edition, 2002.

[6] Javier Sanchez-Araujo and Sylvie Marcos. An efficient PASTd-algorithm implementation for multiple direction of arrival tracking. *IEEE Transactions on Signal Processing*, 47(8):2321–2324, August 1999.