BAYESIAN ALGORITHMS FOR THE PASSIVE LOCATION OF A STATIONARY EMITTER BY A MOVING PLATFORM

Michael Liew, Dominic Lee, Nicholas Chia and Kok Ping Cheng

DSO National Laboratories, Singapore

ABSTRACT

We explore three recursive Bayesian algorithms for the passive location of a stationary ground-based emitter using bearing measurements obtained by a flying platform. The passive location of emitters is a frequent requirement in rescue missions. The algorithms that we consider are a particle filter, the unscented Kalman filter (UKF) and the extended Kalman filter (EKF). They require increasingly greater simplification to the models in the passive location problem. The particle filter is a straightforward formulation based on rejection sampling, requiring the measurements to be conditionally independent and the likelihood to be bounded with known bound. Apart from these requirements, which are satisfied for the problem, no further simplification is needed. The UKF preserves the nonlinearities in the models but approximates the posterior distribution at each time step by a Gaussian distribution. The EKF also assumes a Gaussian distribution, but further linearizes the models so that Gaussianity is preserved under the simplified linear models. Our Monte Carlo simulation results show that the recursive particle filter converges more quickly than the UKF and EKF, and performs better in terms of both point estimation and distribution estimation.

1. INTRODUCTION

We consider the problem of passively locating a stationary ground-based emitter using noisy bearing (azimuth and elevation) measurements obtained by a flying platform. We denote the emitter's location, with respect to a fixed reference frame, by $\theta = (x_e, y_e, z_e)^T$. The platform position is assumed to be known and is denoted, with respect to the same reference frame, by $\psi_k = (x_k, y_k, z_k)^T$, at time-step k. The bearing vector of the emitter at time-step k is $\gamma_k = (\alpha_k, \beta_k)^T$, where α is the relative azimuth, and β the relative elevation, of the emitter with respect to the platform, i.e.

$$\gamma_k = \begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} = g(\theta, \psi_k) = \begin{bmatrix} \tan^{-1}\left(\dfrac{y_e - y_k}{x_e - x_k} \right) \\ \sin^{-1}\left(\dfrac{z_e - z_k}{\sqrt{(x_e - x_k)^2 + (y_e - y_k)^2 + (z_e - z_k)^2}} \right) \end{bmatrix}. \tag{1}$$

The measurement vector is

$$\hat{\gamma}_k = \begin{pmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \end{pmatrix} = \gamma_k + w_k , \tag{2}$$

where $w_k \sim N(0, \Sigma_{w_k})$, i.e. the measurement noise is assumed to be independent and distributed according to a normal distribution with mean zero and covariance matrix, Σ_{w_k}. Our goal is to recursively estimate θ using $\hat{\gamma}_1, \hat{\gamma}_2, \ldots$.

2. BAYESIAN SOLUTION AND ALGORITHMS

Using $\hat{\Gamma}_k$ to denote the set of measurements, $\{\hat{\gamma}_1, \ldots, \hat{\gamma}_k\}$, from time-steps 1 to k, the Bayesian solution to our problem at time-step k is the conditional density, $p(\theta \mid \hat{\Gamma}_k)$. Bayes' Theorem tells us how this conditional density can be updated recursively:

$$p(\theta \mid \hat{\Gamma}_k) \propto p(\hat{\gamma}_k \mid \theta) p(\theta \mid \hat{\Gamma}_{k-1}) . \tag{3}$$

Unfortunately, the nonlinear relationship between the emitter's coordinates and its bearings, given in (1), makes it difficult to implement this recursion exactly. This compels us to consider implementable approximations to (3). We explore three such approximations, which require different extents of simplification on the models in (1) and (2). First, preserving the models as they have been defined, we develop a particle-based algorithm that utilizes rejection sampling. Next, we consider the unscented Kalman filter (UKF) [4], which preserves the nonlinearities in (1), but approximates $p(\theta \mid \hat{\Gamma}_k)$ by a Gaussian at every time-step. Finally, we look at the extended Kalman filter (EKF), which linearizes the model in (1) by Taylor expansions, so that the Gaussianity in (2) is preserved under the simplified linear model.

2.1. Recursive Particle Filter

The particle approximation to (3) represents $p(\theta \mid \hat{\Gamma}_{k-1})$ by a collection of n particles, $\Theta_{k-1,n} = \{\theta_{k-1,1},...,\theta_{k-1,n}\}$, and formulates a recursion that gives a new set of particles, $\Theta_{k,n} = \{\theta_{k,1},...,\theta_{k,n}\}$, which represents $p(\theta \mid \hat{\Gamma}_k)$ (see [1], [3]). Here, we develop a recursion based on rejection sampling, that requires the noise sequence, $w_1, w_2, ...$, to be independent, and the likelihood, $p(\hat{\gamma}_k \mid \theta)$, to be bounded, with known bound, λ_k. Let $\tilde{p}(\theta \mid \hat{\Gamma}_k) = p(\hat{\gamma}_k \mid \theta) p(\theta \mid \hat{\Gamma}_{k-1})$, which is $p(\theta \mid \hat{\Gamma}_k)$ known up to normalizing constant. Then since

$$\frac{\tilde{p}(\theta \mid \hat{\Gamma}_k)}{p(\theta \mid \hat{\Gamma}_{k-1})} \propto p(\hat{\gamma}_k \mid \theta) \le \lambda_k, \tag{4}$$

one way to get a particle from $p(\theta \mid \hat{\Gamma}_k)$ is to use rejection sampling is to generate a particle from $p(\theta \mid \hat{\Gamma}_{k-1})$ and then accept it with probability

$$\pi_k(\theta) = \frac{p(\hat{\gamma}_k \mid \theta)}{\lambda_k}. \tag{5}$$

However, we cannot generate directly from $p(\theta \mid \hat{\Gamma}_{k-1})$ and must therefore approximate it by a density that we can generate from. We use the Gaussian kernel density estimator with shrinkage of kernel locations [5], constructed from $\Theta_{k-1,n}$:

$$\hat{p}(\theta \mid \hat{\Gamma}_{k-1}) = \frac{1}{n} \sum_{j=1}^{n} \phi(\theta \mid \bar{\theta}_{k-1,j}, n^{-2/7} \hat{\Sigma}_{k-1}), \tag{6}$$

where $\phi(\cdot \mid \mu, \Sigma)$ denotes the Gaussian density with mean vector, μ, and covariance matrix, Σ, and $\hat{\Sigma}_{k-1}$ is the sample covariance matrix for $\Theta_{k-1,n}$, and where the Scott bandwidth matrix, $n^{-2/7} \hat{\Sigma}_{k-1}$, has been used for the kernel. The kernel locations are

$$\bar{\theta}_{k-1,j} = \delta \theta_{k-1,j} + (1-\delta) \hat{\mu}_{k-1}, \tag{7}$$

where

$$\delta = \sqrt{1 - n^{-2/7}}, \tag{8}$$

and $\hat{\mu}_{k-1}$ is the sample mean for $\Theta_{k-1,n}$, so that the mean and covariance of the fitted kernel density estimator match those of $\Theta_{k-1,n}$.

Suppose, at time-step k, we have $\Theta_{k-1,n}$ approximately distributed according to $p(\theta \mid \hat{\Gamma}_{k-1})$. Our recursive particle algorithm proceeds as follows:

Set $j = 1$
While $j \le n$
    Generate $\theta \sim \hat{p}(\theta \mid \hat{\Gamma}_{k-1})$ and U~U(0,1)
    If $U < \pi_k$ then $\theta_{k,j} = \theta$ and $j = j + 1$

In the case of the measurement model in (2), where the measurement noise is additive and Gaussian with mean zero and known covariance matrix, Σ_{w_k},

$$\lambda_k = (2\pi)^{-1} |\Sigma_{w_k}|^{-1/2}, \tag{9a}$$

and so the acceptance probability is simply

$$\pi_k(\theta) = \exp\left(-\frac{1}{2} [\hat{\gamma}_k - g(\theta, \psi_k)]^T \Sigma_{w_k}^{-1} [\hat{\gamma}_k - g(\theta, \psi_k)] \right). \tag{9b}$$

2.2. Unscented Kalman Filter

Whilst a particle filter uses a large number of random particles to represent a distribution, the UKF can be thought of as using a small number of deterministic particles, known as sigma points, to obtain an appropriate Gaussian approximation for the distribution. Let μ_{k-1} and Σ_{k-1} be the mean vector and covariance matrix, respectively, for $p(\theta \mid \hat{\Gamma}_{k-1})$, and let $\hat{\mu}_{k-1}$ and $\hat{\Sigma}_{k-1}$ be the UKF's estimates of these quantities. At time-step k, the UKF generates a set of ($2d$ + 1) sigma points from $\hat{\mu}_{k-1}$ and $\hat{\Sigma}_{k-1}$, and uses them to estimate μ_k and Σ_k for $p(\theta \mid \hat{\Gamma}_k)$, where d is the dimension of θ.

We shall consider the UKF with two design parameters [4]: $a \in (0,1)$ determines the spread of the sigma points about the mean, and b is a parameter for incorporating prior knowledge of the distribution – for a Gaussian distribution, b = 2 is optimal. Let

$$c = (a^2 - 1)d, \tag{10}$$

$$\eta = \sqrt{c + d}, \tag{11}$$

$$\omega_0^{(\mu)} = \frac{c}{\eta^2}, \tag{12}$$

$$\omega_0^{(\Sigma)} = \frac{c}{\eta^2} + \left(1 - a^2 + b\right), \tag{13}$$

and

$$\omega_j^{(\mu)} = \omega_j^{(\Sigma)} = \frac{1}{2\eta^2}, \tag{14}$$

for $j = 1, \ldots, 2d$. In the context of our problem, the UKF recursion at time-step k is

1. Calculate the sigma points:

$$\theta_{k,0}^* = \hat{\mu}_{k-1}, \tag{15}$$

$$\theta_{k,j}^* = \begin{cases} \hat{\mu}_{k-1} + \eta (\hat{\Sigma}_{k-1}^{1/2})_j, & j = 1, ..., d, \\ \hat{\mu}_{k-1} - \eta (\hat{\Sigma}_{k-1}^{1/2})_{j-d}, & j = d+1, ..., 2d. \end{cases} \tag{16}$$

where $(\hat{\Sigma}_{k-1}^{1/2})_j$ denotes the j-th column of the matrix square-root of $\hat{\Sigma}_{k-1}$.

2. Calculate the weighted mean and covariance of the sigma points:

$$\mu_{\theta_k^*} = \sum_{j=0}^{2d} \omega_j^{(\mu)} \theta_{k,j}^* \ , \tag{17}$$

$$\Sigma_{\theta_k^*} = \sum_{j=0}^{2d} \omega_j^{(\Sigma)} (\theta_{k,j}^* - \mu_{\theta_k^*})(\theta_{k,j}^* - \mu_{\theta_k^*})^T \ . \tag{18}$$

3. Calculate the predicted measurements from the sigma points:

$$\gamma_{k,j}^* = g(\theta_{k,j}^*, \psi_k) \ , j = 1, \ldots, 2d. \tag{19}$$

4. Calculate the weighted mean and covariance of the predicted measurements:

$$\mu_{\gamma_k^*} = \sum_{j=0}^{2d} \omega_j^{(\mu)} \gamma_{k,j}^* \ , \tag{20}$$

$$\Sigma_{\gamma_k^*} = \sum_{j=0}^{2d} \omega_j^{(\Sigma)} (\gamma_{k,j}^* - \mu_{\gamma_k^*})(\gamma_{k,j}^* - \mu_{\gamma_k^*})^T + \Sigma_{w_k} \ . \tag{21}$$

5. Calculate the weighted covariance between the sigma points and the predicted measurements:

$$\Sigma_{\theta_k^* \gamma_k^*} = \sum_{j=0}^{2d} \omega_j^{(\Sigma)} (\theta_{k,j}^* - \mu_{\theta_k^*})(\gamma_{k,j}^* - \mu_{\gamma_k^*})^T \ . \tag{22}$$

6. Calculate the Kalman gain:

$$K_k = \Sigma_{\theta_k^* \gamma_k^*} \Sigma_{\gamma_k^*}^{-1} . \tag{23}$$

7. Estimate the mean of $p(\theta \mid \hat{\Gamma}_k)$:

$$\hat{\mu}_k = \mu_{\theta_k^*} + K_k (\hat{\gamma}_k - \mu_{\gamma_k^*}) \tag{24}$$

8. Estimate the covariance of $p(\theta \mid \hat{\Gamma}_k)$:

$$\hat{\Sigma}_k = \Sigma_{\theta_k^*} - K_k \Sigma_{\gamma_k^*} K_k^T \ . \tag{25}$$

2.3. Extended Kalman Filter

Since the emitter is stationary, the state evolution matrix is simply an identity matrix and the process noise is zero. By letting

$$G_k = \left. \frac{\partial g(\theta, \psi_k)}{\partial \theta} \right|_{\theta = \hat{\mu}_{k-1}} \tag{26}$$

be the Jacobian matrix of the measurement function, evaluated at $\theta = \hat{\mu}_{k-1}$, the EKF recursion for our problem, at time-step k, is

1. Compute the Kalman gain:

$$K_k = \hat{\Sigma}_{k-1} G_k^T (\Sigma_{w_k} + G_k \hat{\Sigma}_{k-1} G_k^T)^{-1} \tag{27}$$

2. Estimate the mean of $p(\theta \mid \hat{\Gamma}_k)$:

$$\hat{\mu}_k = \hat{\mu}_{k-1} + K_k [\hat{\gamma}_k - g(\hat{\mu}_{k-1}, \psi_k)] \ . \tag{28}$$

3. Estimate the covariance of $p(\theta \mid \hat{\Gamma}_k)$:

$$\hat{\Sigma}_k = (I - K_k G_k) \hat{\Sigma}_{k-1} \ . \tag{29}$$

3. RESULTS

Monte Carlo simulation results for the Bayesian algorithms are shown in Figures 1 and 2. The results are for 100 Monte Carlo realizations, with 30 time-steps per realization. In Figure 1, the conditional mean, $\hat{\theta}$, is taken as the point estimate of emitter location, θ, at each time-step and for each of the three algorithms. The root-mean-squared (RMS) error is the square root of the mean (over 100 realizations) of the squared Euclidean distance between $\hat{\theta}$ and θ. For each time-step, bootstrap bias-corrected and accelerated (BCA) intervals [2] with 90% confidence level are constructed for the RMS errors. Each bootstrap interval is represented by an error bar, with a dot indicating the RMS error. At each time-step, the error bars are drawn slightly displaced in time to enhance visibility.

Figure 1(a) shows the bootstrap intervals for time-steps 2-15, and Figure 1(b) for time-steps 16-30. The RMS error is shown as a percentage of the range between the platform and the emitter. All three algorithms were initialized by a common procedure using the measurement from the first time step. The smaller RMS error of the recursive particle filter (RPF), compared to both the UKF and EKF, is clearly evident.

Figure 2 shows the mean probability mass, over the 100 realizations, that each algorithm places in a rectangular box centered at the true emitter position. Once again, we show the bootstrap BCA intervals with 90% confidence level in the form of error bars, with the mean probability mass marked by a dot. Whilst the RMS error indicates the point estimate performance of the algorithms, the mean probability mass provides an indication of their distribution performance. The shape and size of the box are completely arbitrary and a different choice can be used depending on operational considerations. The results show that the RPF places significantly more mass close to the true emitter location than either the UKF or the EKF.

Taken together, the results indicate quicker convergence of the RPF and better performance in terms of both point estimation and distribution estimation. The results show the UKF performing only slightly better than the EKF, suggesting that the nonlinearities for the scenario considered must have been mild. Nevertheless, the RPF still managed to exhibit quicker convergence and better performance.

Although the UKF and EKF appear to perform quite well, there are theoretical and practical difficulties arising from their use. For instance, by considering θ in natural space, its support there may not be compatible for Gaussian distributions. This means that the UKF and EKF can, and do, put probability mass in regions where it is physically impossible for the emitter to be located. We considered transforming θ to a space where the support was compatible but found the Gaussian fit in the transformed space to be worse than in the natural space. Thus, with the UKF and the EKF, it is difficult to impose constraints on the support of θ, which is something that can be easily done with a particle filter. Such constraints can be exploited to further improve the estimation of emitter location.

The computational complexity of the UKF is comparable to the EKF, but the RPF requires several orders of magnitude more computations than the UKF. Nevertheless, with the computing power that is available today, real-time implementation of the RPF is not impossible if the data rate is not too high.

4. CONCLUSION

We have formulated a recursive particle filter based on rejection sampling and have obtained simulation results showing that it converges more quickly than the UKF and EKF, and performs better in terms of both point estimation and distribution estimation. We are conducting more extensive studies with the algorithm, and are exploring how it can exploit constraints on the support of θ as well as other forms of prior information.

5. REFERENCES

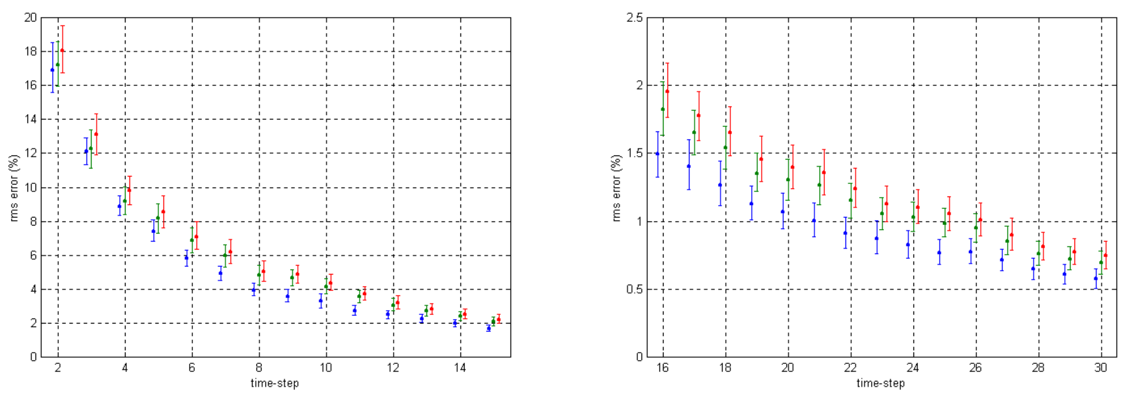
[1] Doucet, A., deFreitas, N. and Gordon, N. (editors), *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[2] Efron, B. and Tibshirani, R.J., *An Introduction to the Bootstrap*, Chapman and Hall, 1993.

[3] Liu, J.S., *Monte Carlo Strategies in Scientific Computing*, Springer, 2001.

[4] van der Merwe, R. and Wan, E.A., “The square-root unscented Kalman filter for state and parameter estimation”, International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, Utah, May 2001.

[5] West, M., “Approximating posterior distributions by mixtures”, Journal of the Royal Statistical Society 55, 409-422, 1993.



(a) (b)

Figure 1. 90%-Bootstrap BCA intervals for RMS error with 100 realizations and 5000 bootstrap replications, for time-steps (a) 2-15, and (b) 16-30. RPF – left error bar; UKF – middle error bar; EKF – right error bar.

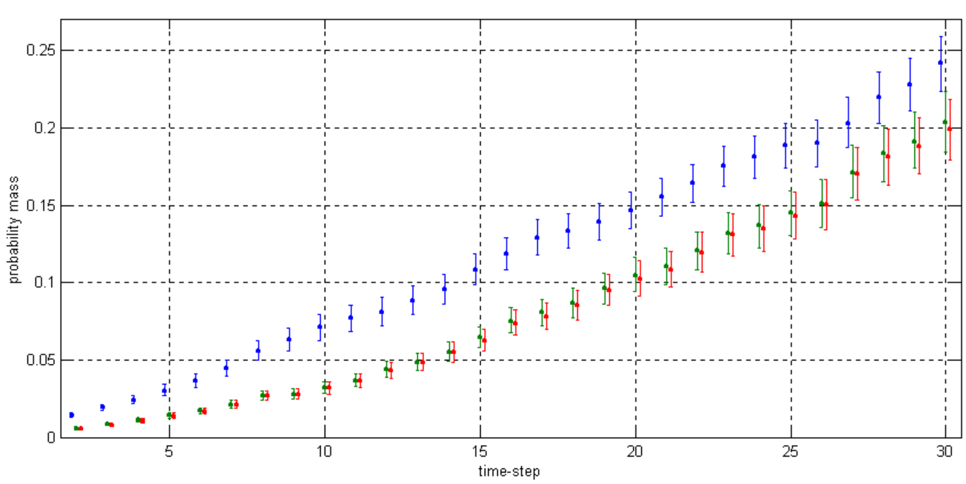


Figure 2. 90%-Bootstrap BCA intervals for mean probability mass in a rectangular box centered at the true emitter position, with 100 realizations and 5000 bootstrap replications: RPF – left error bar; UKF – middle error bar; EKF – right error bar.