JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 22, No. 3, May–June 1999

Bayesian Approach to Guidance in the Presence of Glint

Neil Gordon*
Defence Evaluation and Research Agency, Malvern, England WR14 3PS, United Kingdom
and
Angela Whitby†
Defence Evaluation and Research Agency, Farnborough, England GU14 0LX, United Kingdom

When tracking targets with radar, changes in target aspect with respect to the observer can cause the apparent center of radar reflections to wander significantly. The resulting noisy angle errors are called target glint. Glint may severely affect the tracking accuracy, particularly when tracking large targets at short ranges (such as might occur in the final homing phase of a missile engagement). The effect of glint is to produce heavy-tailed, time-correlated non-Gaussian disturbances on the observations. It is well known that the performance of the Kalman filter degrades severely in the presence of such disturbances. We propose a random-sample-based implementation of a Bayesian recursive filter. This filter is based on the Metropolis–Hastings algorithm and the Gaussian sum approach. The key advantage of the filter is that any nonlinear/non-Gaussian system and/or measurement models can be routinely implemented. Tracking performance of the filter is demonstrated in the presence of glint.

I. Introduction

GLINT is defined as the inherent component of error in a radar's measurement of position or Doppler frequency of a complex target due to interference of the reflections from different elements of the target. A complex target, such as an aircraft or tank, may be considered as being made up of a number of independent scattering elements. The radar return from such a target is the vector sum of the radar returns from each of the individual scatterers. As the relative position of the radar and the scatterers changes, the relative phase of the radar returns varies, causing large fluctuations in the combined signal. The change in relative phase of the signal returns causes a change in the apparent target center and, hence, the angular error measured by the radar. This angular noise or scintillation is called glint. Glint can cause the apparent location of the target to fall outside of its physical span (this can be simply demonstrated by analysis of a two-source target). The linear displacement of the apparent target position due to glint varies inversely with range. As a rough guide, the standard deviation of the angular glint from an aircraft target is about kL/R, where L is the wing span of the target, R is the range to the target, and k is a constant taking a value between 0.1 and 0.2 (Ref. 1).

Glint is particularly troublesome for target tracking because it is a low-frequency effect, the bandwidth of the error typically being a few hertz. In Ref. 2, the correlation time t_{gc} of the glint noise is shown to be approximately

$$t_{gc} \approx \frac{\lambda}{3.4\omega_a L} \tag{1}$$

where λ is the wavelength of the radiation, ω_a is the rate of change of the aspect angle (radian per second) and L is the span of the target across the line of sight. For example, with $L = 10.0$ m, $\omega_a = 0.01$ rad/s, and $\lambda = 0.03$ m, we have $t_{gc} \approx 0.1$ s. Because of the low frequency of glint, it may not be possible to distinguish between actual target motion and glint errors.

The probability density function (PDF) of the glint error has been shown to follow a (time-correlated) Student-t distribution. The Student t is a heavier tailed distribution than the Gaussian and has been extensively used as a model for outliers.[3–5]

Bayesian methods provide a rigorous general framework for dynamic state estimation problems. The Bayesian approach is to construct the PDF of the state based on all of the available information. In Sec. II, the general recursive Bayesian filter for discrete time state-space models is described. Under the assumptions of linear-Gaussian measurement and system models, this reduces to the Kalman filter; the required PDF remains Gaussian at every iteration of the filter, and the Kalman filter relations propagate and update the mean and covariance of the distribution. For nonlinear or non-Gaussian problems, e.g., glint, there is no general analytic (closed-form) expression for the required PDF. In Sec. III, approximation techniques for implementing the Bayesian filter with particular reference to the problem of glint are discussed.

In Sec. IV, the new random-sample-based Bayesian–Metropolis filter algorithm is outlined. More complete descriptions of the techniques used are given in Appendix A for the Gaussian mixture approximation algorithm and Appendix B for the Metropolis–Hastings algorithm. Section V gives a two-dimensional target tracking example. Measurement errors are generated according to a glint model derived from physical principles.[6] The filter measurement model is of autocorrelated, heavy-tailed non-Gaussian noise. The performance of the Bayesian–Metropolis filter is compared with that of an interacting multiple model- (IMM-) based filter.[7]

II. Recursive Bayesian Estimation

In this paper we are concerned with the discrete time estimation problem. The state vector, $\boldsymbol{x}_k \in \Re^n$ is assumed to evolve according to the following system model:

$$\boldsymbol{x}_{k+1} = f_k(\boldsymbol{x}_k, w_k) \tag{2}$$

where $f_k : \Re^n \times \Re^m \to \Re^n$ is the system transition function and $w_k \in \Re^m$ is a white noise sequence independent of past and current states. The PDF of w_k is assumed known. At discrete times, measurements $y_k \in \Re^p$ become available. These measurements are related to the state vector via the observation equation

$$y_k = h_k(\boldsymbol{x}_k, v_k) \tag{3}$$

where $h_k : \Re^n \times \Re^r \to \Re^p$ is the measurement function and $v_k \in \Re^r$ is another white noise sequence of known PDF, independent of past and present states and the system noise. It is assumed that the initial PDF $p(\boldsymbol{x}_1 \mid D_0) \equiv p(\boldsymbol{x}_1)$ of the state vector is available together with the functional forms f_i and h_i for $i = 1, \ldots, k$. The

Received Nov. 6, 1997; revision received Oct. 14, 1998; accepted for publication Oct. 23, 1998.

*Senior Scientist, Sensors and Processing Sector, EX30, St. Andrews Road, Worcestershire. E-mail: gordon@signal.dera.gov.uk.

†Senior Engineer, Weapons Sector, A2 Building, Ively Road, Hants. E-mail: ajwhitby@guidance.demon.co.uk.

available information at time step k is the set of measurements $D_k = \{y_i : i = 1, \ldots, k\}$.

The requirement is to construct the PDF of the current state x_k, given all of the available information: $p(x_k \mid D_k)$. In principle, this PDF may be obtained recursively in two stages: prediction and update. Suppose that the required PDF $p(x_{k-1} \mid D_{k-1})$ at time step $k-1$ is available. Then using the system model, it is possible to obtain the prior PDF of the state at time step k:

$$p(\boldsymbol{x}_k \mid D_{k-1}) = \int p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1} \mid D_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1} \tag{4}$$

Here the probabilistic model of the state evolution, $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$, which is a Markov model, is defined by the system equation (2) and the known statistics of w_{k-1}.

Then at time step k, a measurement y_k becomes available and may be used to update the prior via Bayes rule:

$$p(\boldsymbol{x}_k \mid D_k) = \frac{p(y_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid D_{k-1})}{p(y_k \mid D_{k-1})} \tag{5}$$

where the normalizing denominator is given by

$$p(y_k \mid D_{k-1}) = \int p(y_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid D_{k-1})\, \mathrm{d}\boldsymbol{x}_k \tag{6}$$

The conditional PDF of y_k given x_k, $p(y_k \mid x_k)$, is defined by the measurement model (3) and the known statistics of v_k. In the update equation (5), the measurement y_k is used to modify the predicted prior from the previous time step to obtain the required posterior of the state.

The recurrence relations (4) and (5) constitute the formal solution to the Bayesian recursive estimation problem. Analytic solutions to this problem are only available for a relatively small and restrictive choice of system and measurement models, the most important being the Kalman filter, which assumes f_k and h_k are linear and w_k and v_k are additive Gaussian noise of known variance. Considerations of realism imply that these assumptions are unreasonable for many applications.

The extended Kalman filter is the most popular approximate approach to recursive nonlinear estimation.[8] Here the estimation problem is linearized about the predicted state so that the Kalman filter can be applied. In this case, the required PDF is still approximated by a Gaussian, which may be a gross distortion of the true underlying structure and can lead to filter divergence. Other analytic approximations include the Gaussian sum filter[9] and methods based on approximating the first two moments of the density.[3] A more direct numerical approach is to evaluate the required PDF over a grid in state space.[4,5] The choice of an efficient grid is nontrivial, and in a multidimensional state space a very large number of grid points may be necessary. A significant computation must be performed at each point.

In Ref. 10, a method (called the bootstrap filter) of representing and recursively generating an approximation to the state PDF was proposed. The central idea was to represent the required PDF as a set of random samples, rather than as a function over state space. As the number of samples becomes very large, they effectively provide an exact, equivalent, representation of the required PDF. Estimates of moments (such as mean and covariance) or percentiles of the state vector PDF can be obtained directly from the samples. If necessary, a functional estimate of the PDF could also be constructed from the samples,[11] and from this, estimates of highest posterior density intervals or the mode could be obtained. The bootstrap approach has been successfully applied to problems in nonlinear estimation and target tracking.[12,13] The Bayesian–Metropolis filter described in Sec. IV is an alternative to the bootstrap filter. A random-sample-based technique avoids the need to define a grid in state space, the samples being naturally concentrated in regions of high probability density. It also has the great advantage of being able to handle any functional nonlinearity and system or measurement noise of any distribution. Different, but related approaches utilizing random samples can be seen in Refs. 14–16.

III. Approaches to Tracking with Glint

Many researchers have considered the problem of filtering in the presence of heavy-tailed non-Gaussian measurement errors. Similarly, the problem of estimation with time-correlated (or colored) measurement noise has been investigated. However, previous attempts at dealing with glint measurement noise do not appear to have considered the combined effect of these properties on the measurement noise. Reference 17 considers the application of the score function method[3] to glint tracking. The score function effectively leads to modified Kalman update equations using nonlinear functions of the innovation in the mean and covariance update equations. The exact equations[3] require an analytically intractable convolution operation, and so Ref. 17 proposes a simplifying normal expansion of the density to facilitate the convolution. This is extended in Ref. 18 to incorporate the score method with the IMM algorithm to track maneuvering targets. In Ref. 7, the measurement noise is modeled as a mixture of a Gaussian with a contaminating heavy-tailed distribution, and then the IMM filter is used for estimation.

The effect of time correlation on the measurement errors was first considered by Bryson and Henrickson,[19] who modeled the time correlated errors as the output of an autoregressive model driven by white noise. By differencing successive measurements, they were able to write the filter in standard format and so use the Kalman filter. Time correlated measurements were also considered by Bar-Shalom et al.,[20] who describe the application of the IMM filter to this problem.

IV. Bayesian–Metropolis Filter

Suppose we have a set of independent random samples $\{x_{k-1}(i) : i = 1, \ldots, N\}$ from the PDF $p(\boldsymbol{x}_{k-1} \mid D_{k-1})$. The Bayesian–Metropolis filter is an algorithm for propagating and updating these samples to obtain a set of values $\{x_k(i) : i = 1, \ldots, N\}$, which are approximately distributed as independent random samples from $p(\boldsymbol{x}_k \mid D_k)$. Thus, the filter is an approximate mechanization (simulation) of the relations (4) and (5).

Prediction

Noting the approximate result

$$p(\boldsymbol{x}_k \mid D_{k-1}) = \int p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1} \mid D_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}$$

$$\approx N^{-1} \sum_{i=1}^{N} p[\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1} = \boldsymbol{x}_{k-1}(i)] \tag{7}$$

where $\{x_{k-1}(i) : i = 1, \ldots, N\}$ are N independent, identically distributed (IID) samples from $p(\boldsymbol{x}_{k-1} \mid D_{k-1})$, the following algorithm is suggested.[12]

1) Uniformly resample (with replacement) the values $\{x_{k-1}(i)\}$ N^* times.

2) Pass each resampled value through the system model.

This generates the required expanded sample set $\{x_k^*(i) : i = 1, \ldots, N^*\}$. Note that $N^* > N$, and in general it is suggested that $N^* > 10N$. These are independent random samples from the PDF $p(\boldsymbol{x}_k \mid D_{k-1})$. These samples are used to generate a Gaussian mixture approximation to the prior PDF at time step k:

$$\hat{p}(\boldsymbol{x}_k \mid D_{k-1}) = \sum_{i=1}^{C(k)} \pi_i N(\boldsymbol{x}_k; m_i, V_i) \tag{8}$$

where $C(k)$ is the number of components required to approximate the N^* point discrete sample set, m_i the mean of the ith component, V_i the covariance of the ith component, and π_i the weight attached to ith component with

$$\sum_{i=1}^{C(k)} \pi_i = 1 \tag{9}$$

By using a mixture approximation to the discrete sample set, we effectively fill in the gaps left between the samples in a way consistent with the disposition of probability mass indicated by the relative

location and importance of the samples. The cluster algorithm used to generate the mixture is outlined in Appendix A. Estimates of the mean and covariance of the prior at time step k are available:

$$E(\boldsymbol{x}_k \mid D_{k-1}) = \bar{m}_k = \sum_{i=1}^{C(k)} \pi_i m_i \tag{10}$$

$$V(\boldsymbol{x}_k \mid D_{k-1}) = P_k = \sum_{i=1}^{C(k)} \pi_i \left[C_i + (m_i - \bar{m}_k)(m_i - \bar{m}_k)^T \right] \tag{11}$$

Update

On receipt of the measurement y_k, the posterior PDF at time step k is available (up to proportionality) as

$$p(\boldsymbol{x}_k \mid D_k) \propto p(y_k \mid \boldsymbol{x}_k) \sum_{i=1}^{C(k)} \pi_i N(\boldsymbol{x}_k, m_i, C_i) = s_k(\boldsymbol{x}_k) \tag{12}$$

The Metropolis–Hastings algorithm (see Appendix B) is applied to Eq. (12) to give N random samples, which are approximately distributed as $p(\boldsymbol{x}_k \mid D_k)$. Also obtained by ergodic averaging of the chain are approximations to the posterior mean and covariance, m_k and C_k. These samples are then passed to the prediction phase of the algorithm for recursive implementation.

V. Tracking and Guidance with Glint Noise

Consider a simple single-plane model of a typical air-to-ground engagement. The airframe is simply represented by a critically damped second-order response, with latax limiting. Initially the airframe is in level flight at 100 m/s, 800 m downrange of the target and 300 m above the target. It is assumed that range and range rate are available from the seeker. Three different target conditions are examined: stationary, approaching at 10 m/s and receding at 10 m/s. The measured sightline is corrupted by glint simulated according to the model of Ref. 6, with the variance of the error increasing as range decreases. The airframe is guided using a standard proportional navigation guidance law. An example glint error history over a flight is shown in Fig. 1. The presence of positive time correlation, large error spikes, and increasing variance with decreasing range are all easily observed. The performance of the Bayesian–Metropolis filter will be compared with that of the IMM-based method proposed in Ref. 7.

For both filters under consideration, the system model is taken to be

$$\boldsymbol{x}_k = (\theta \quad \dot{\theta} \quad v)_k^T = \begin{pmatrix} 1 & T & 0 \\ 0 & 1 - 2T\,\mu & 0 \\ 0 & 0 & \beta \end{pmatrix} \boldsymbol{x}_{k-1} + \begin{pmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} w_k \\ u_k \end{pmatrix} = F_k x_k + \Gamma_k \boldsymbol{w}_k \tag{13}$$

where T is the sample period of 0.05 s with a data rate of 20 Hz, where $p(w_k) = N(0, q)$ and $p(u_k) = t(0, s, d)$, and where $t(a, b, c)$ is a Student-t density of mean a, scale b, and degrees of freedom c, and $\mu = \dot{r}/r$. The variables θ and $\dot{\theta}$ represent sightline angle and sightline angular rate, respectively, with respect to inertial, nonrotating axes. The sequence $\{v_k\}$ represents the time-correlated component of the measurement error, and the Student-t driving noise on this sequence leads to heavy-tailed non-Gaussian errors. The parameter β must have absolute value less than unity for stationarity of the noise sequence. A large positive value for β, e.g., $\beta = 0.9$, leads to positively correlated errors; large negative values of β, e.g., $\beta = -0.9$, cause negative correlation in the error sequence, and if $\beta = 0$, then the sequence is uncorrelated. Thus, a positive value for β is appropriate for a glint representation. The model used for the received measurement depends on the filter.

The IMM filter was set up as in Ref. 7. Two models are considered: with and without correlated measurement errors. Thus, the models have the same dynamic model but model i ($i = 1, 2$) has measurement equation

$$y_k = H_k^i x_k + e_k \tag{14}$$

where

$$H_k^1 = (1 \quad 0 \quad 1) \tag{15}$$

$$H_k^2 = (1 \quad 0 \quad 0) \tag{16}$$

and $p(e_k) = N(0, r)$. The model Markov switching probability matrix is taken to be

$$P = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} \tag{17}$$

The initial model probabilities are taken to be equal.

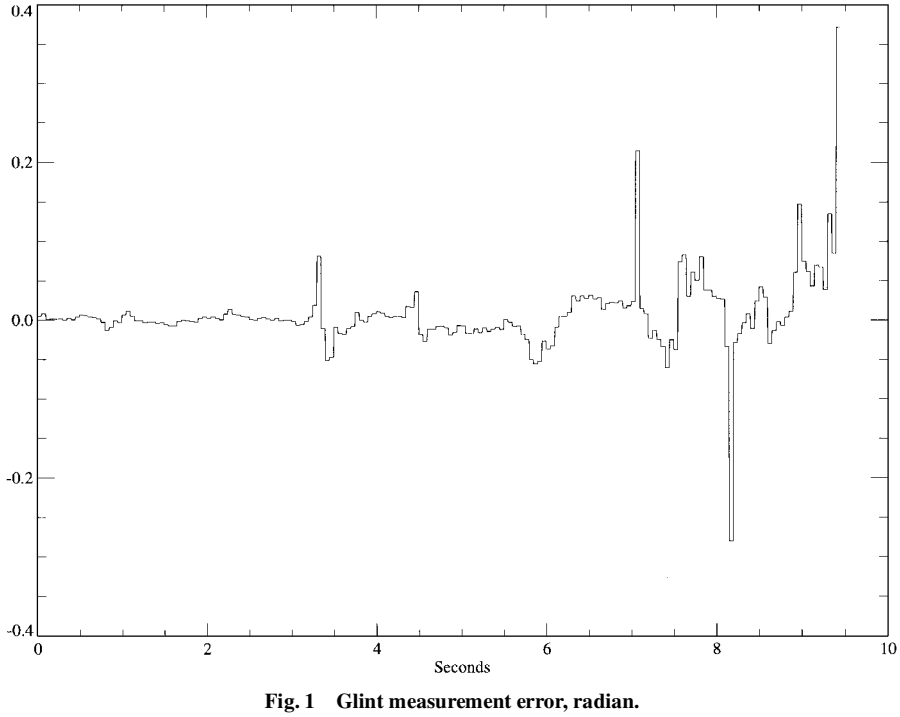

Fig. 1 Glint measurement error, radian.

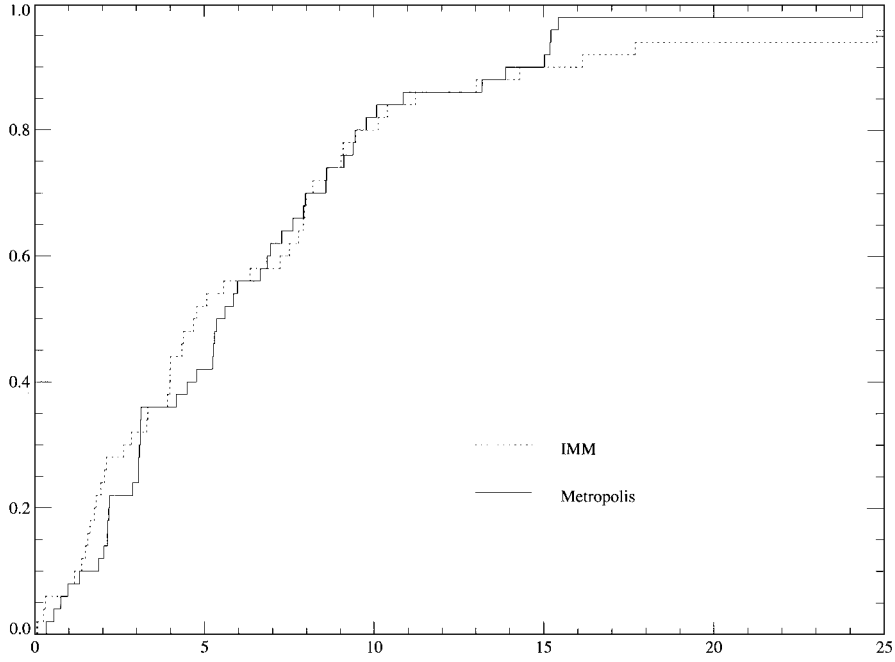

Fig. 2 Cumulative miss distances (meters): stationary target.

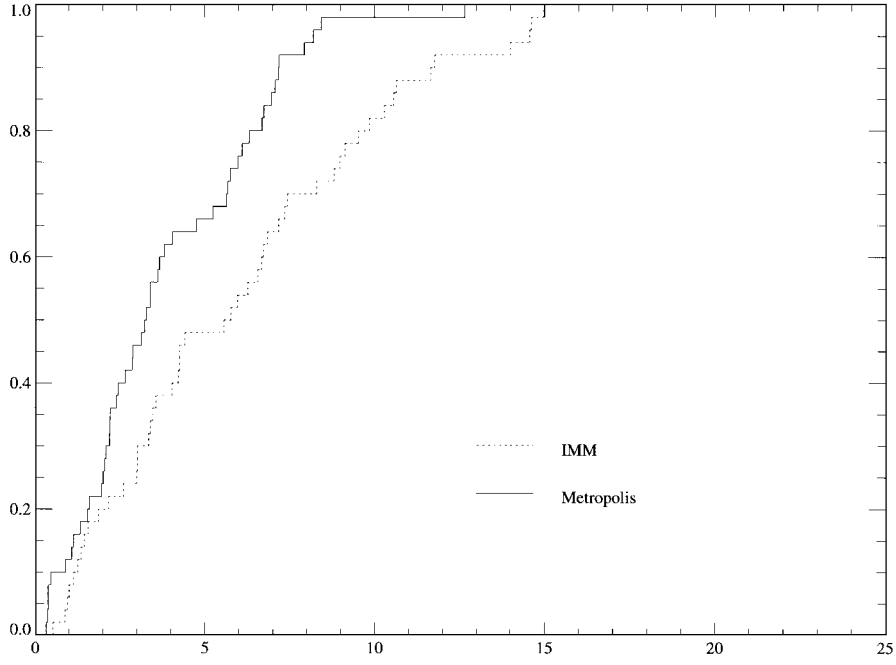

Fig. 3 Cumulative miss distances (meters): target approaching.

For the Bayesian–Metropolis filter (see Appendix B for explanation of the specified chain parameters), the initial state of the Markov chain is set at the prior mean $\bar{m}_k$ [where $\bar{m}_k = E(\boldsymbol{x}_k \mid D_{k-1})$]. The chain is generated by using an 80:20 mixture of random walk and autoregressive choices for the candidate generating density $q(\boldsymbol{x}, \boldsymbol{x}')$. The initial scale of the random walk chain is set to $P_k = V(\boldsymbol{x}_k \mid D_{k-1})$ and scale adaptation used with $\delta = 1\%$. The current value of the ergodic approximation to the posterior mean is used for the reflection point in the autoregressive chain. The burn-in section of the chain was set at 1000 iterations and the decorrelation gap to 10. The sample sizes were set to $N = 200$ and $N^* = 2000$, i.e., a boosting factor of 10 for the predictive step. The maximum allowed number of components in the mixture reduction process is set to 20. The measurement model is taken to be

$$y_k = (1 \quad 0 \quad 1)\boldsymbol{x}_k + e_k \tag{18}$$

again, with $p(e_k) = N(0, r)$.

For both filters, the model parameters are taken to be $q = E[w_k^2] = 5$ deg^2/s^2, $r = E[e_k^2] = 0.8$ deg^2, $\beta = 0.9$, and $d = 2$. Filter performance is summarized by the terminal miss distance, and 50 Monte Carlo runs were performed for each of the three target motions. Figures 2–4 show the results in the form of a cumulative miss-distance curve. For the stationary target, there is practically zero sightline rate throughout the flight, and hence, very little information is obtained on target location. The IMM and Bayesian–Metropolis filters give practically identical performance. However, when there is information in the measurements (due to target motion inducing a relative sightline rate), the Bayesian–Metropolis filter is able to make better use of the available information, and so, as is clear from Figs. 3 and 4, it yields improved miss-distance performance compared to the IMM. Figure 5 shows the rms estimation errors of angle for the case where the target is moving away. The nonstationary variance of the measurement errors is immediately evident. At long range the glint effect is negligible, and both IMM

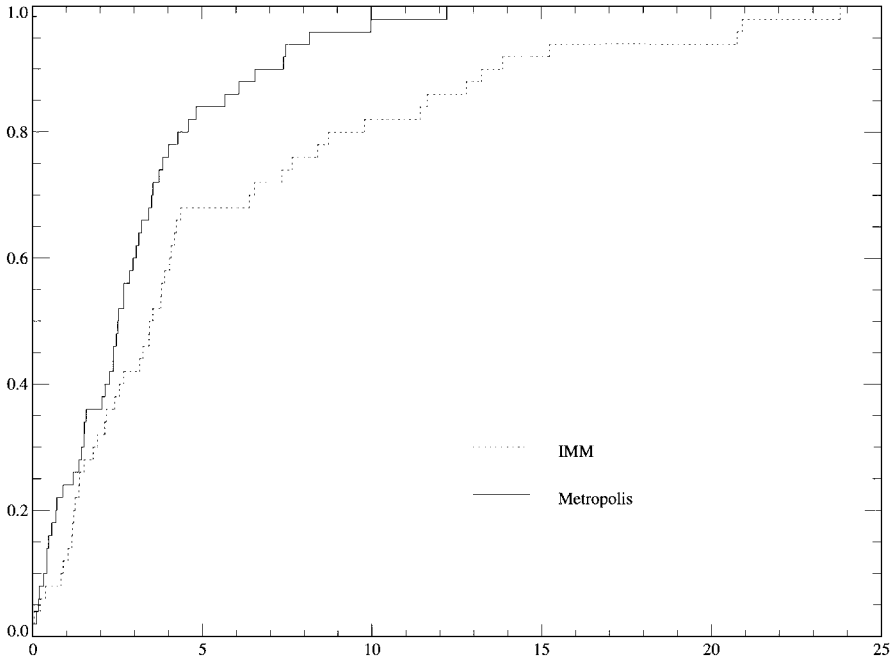

Fig. 4 Cumulative miss distances (meters): target moving away.

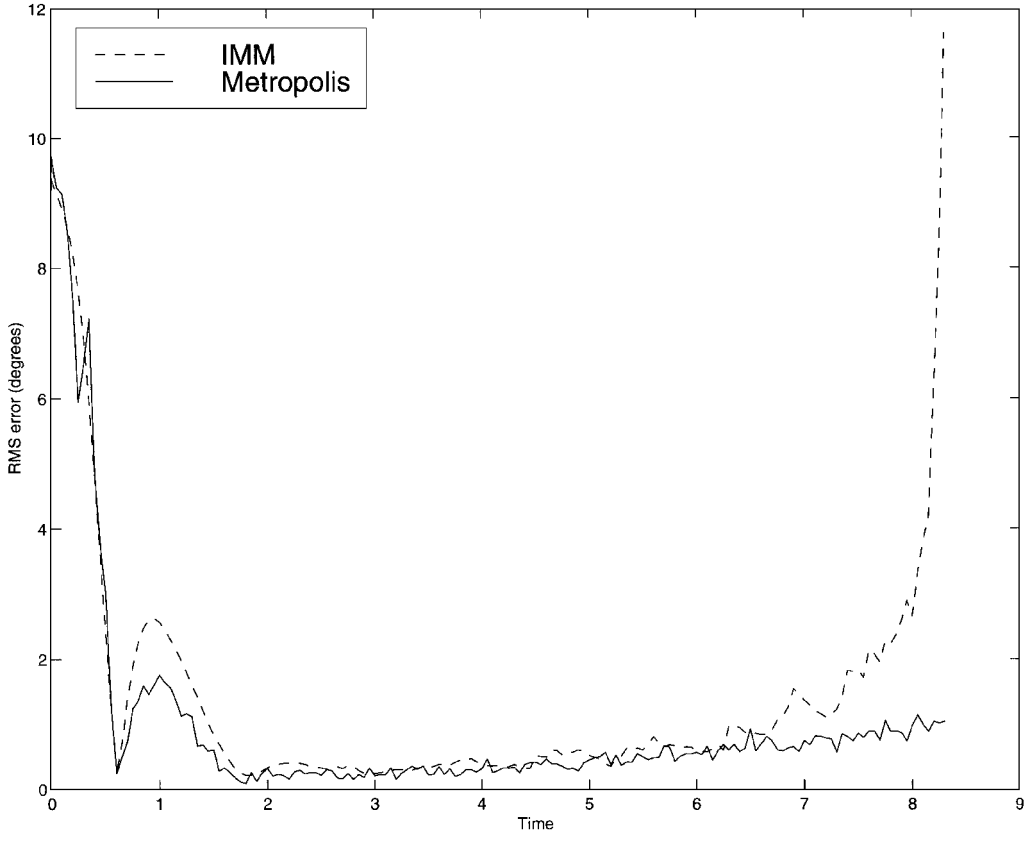

Fig. 5 Root mean square estimation errors (degree): target moving away.

and Bayesian–Metropolis filters exhibit similar error characteristics. However, during the terminal phase of the engagement, the improved implementation of the non-Gaussian model lowers the error in the Bayesian–Metropolis filter, and this is the key factor leading to reduced miss distance.

VI. Concluding Remarks

We have presented a new random-sample-based approach to the implementation of general nonlinear/non-Gaussian filters. The key Bayesian update stage of the filter is implemented via the Metropolis–Hastings algorithm. This is a method for generating random samples from an unnormalized PDF by the process of simulating an associated Markov chain. The required posterior is obtained up to proportionality by the use of a Gaussian sum approximation to the predictive distribution. The mixture approximation is created via the clustering algorithm of Ref. 21. Use of the Metropolis–Hastings algorithm allows us to obtain good approximations to moments of the posterior distribution (via ergodic averaging

of the Markov chain) in addition to random samples of the posterior PDF.

The Bayesian–Metropolis filter is valid for (practically) any nonlinear/non-Gaussian measurement and system model choices. We have demonstrated the efficacy of the method against time-correlated non-Gaussian measurement errors, such as would be encountered when tracking in the presence of glint noise. Filter performance is seen to be superior to the IMM. This performance improvement is gained at the expense of a significantly increased computational overhead, which increases run time in our simulations by a factor of around 1000. However, it should be noted that no attempt has been made to optimize the efficiency of the code, and in addition, the Bayesian–Metropolis algorithm is particularly well suited to parallel implementation.

Appendix A: Mixture Approximation by Component Merging

Given a set of discrete independent random samples, it is required to generate a finite mixture approximation to the prior PDF at time step k:

$$\hat{p}(\boldsymbol{x}_k \mid D_{k-1}) = \sum_{i=1}^{C(k)} \pi_i d(\boldsymbol{x}_k \mid m_i, V_i) \tag{A1}$$

where $d(\boldsymbol{x} \mid m, V)$ is a density function of random variable $\boldsymbol{x}$, centered at mode m and scaled by some positive definite matrix V. A key example[11] has $d(x \mid m, V) = N(\boldsymbol{x}; m, V)$, the density of a multivariate Gaussian with mean m and covariance matrix V. Other possibilities include choosing d to be a multivariate Student-t distribution.[16] In this paper, we only consider Gaussian mixture approximations. This gives

$$\hat{p}(\boldsymbol{x}_k \mid D_{k-1}) = \sum_{i=1}^{C(k)} \pi_i N(\boldsymbol{x}_k; m_i, V_i) \tag{A2}$$

where $C(k)$ is the number of components required to approximate the N point discrete sample set, m_i the mean of the ith component, V_i the covariance of the ith component, and π_i the weight attached to ith component with $\pi_1 + \cdots + \pi_N = 1$.

To generate the mixture approximation (A2) we consider the clustering method of Ref. 21. This consists of a way of merging together the individual points to create the mixture. The aim is to take the sample points $\{x_k^*(i) : i = 1, \ldots, N^*\}$ and produce a mixture of the form of Eq. (A2). The cluster algorithm is designed to reduce a Gaussian mixture with N_1 components to a Gaussian mixture with N_2 components ($N_2 < N_1$). If we set

$$\hat{p}_1(\boldsymbol{x}_k \mid D_{k-1}) = \sum_{i=1}^{N^*} q_i N\left(\boldsymbol{x}_k; \boldsymbol{x}_k^*(i), hV\right) \tag{A3}$$

where V is the sample covariance matrix (of the sample set) and $h \in \Re^+$, then we can use this method for our problem: We aim to merge Eq. (A3) into a Gaussian mixture with $C(k)$ components. The cluster algorithm has the following properties.

1) Reduction in mixture size should continue as long as the underlying structure of the distribution is not significantly altered. Conversely, to avoid retaining unnecessary components, reduction should continue until this limit is reached. The measure of structure is derived from a decomposition of the overall sample covariance matrix.

2) The approximation should preserve the mean and covariance of the sample set. This is done for the pragmatic reason that these are the most commonly required summary statistics, and so it makes sense to ensure that these are not altered under the mixture approximation procedure.

3) The maximum number of mixture components allowed after approximation should be specified. Clearly the maximum that can be allowed depends on the available computational power and storage capacity.

The cluster algorithm is based on the proposition that the mixture components with the largest weightings carry the most important information. Thus, starting with the largest component, this algorithm gathers in all surrounding components that are in some sense close to the principal component. Subsequently, the largest component of the remainder is selected, and the process is repeated until all of the components have been clustered.

Appendix B: Metropolis–Hastings Algorithm

Suppose we wish to sample from the posterior distribution $p(\boldsymbol{x} \mid D)$, but cannot do this directly. However, suppose that it is possible to construct a Markov chain, which is straightforward to simulate from, whose equilibrium distribution is $p(\boldsymbol{x} \mid D)$. If this chain is then run for a long time, simulated values from the chain can be used as a basis for summarizing features of the posterior $p(\boldsymbol{x} \mid D)$. To implement this strategy, we need algorithms for constructing chains with specified equilibrium distributions.

If $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, x^t, \ldots$ is a realization from an appropriate chain, then it can be shown[22] that, as $t \to \infty$

$$\boldsymbol{x}^t \to \boldsymbol{x} \sim p(\boldsymbol{x} \mid D)$$

in distribution

$$t^{-1} \sum_{i=1}^{t} g(x^i) \to E_{x \mid D}[g(x)]$$

almost surely. Clearly, successive x^t will be correlated, so that, if the first of these asymptotic results is to be exploited to mimic a random sample from $p(\boldsymbol{x} \mid D)$, suitable spacings will be required between realizations used to form the sample, or parallel independent realizations of the chain might be considered. The second of the asymptotic results implies that ergodic averaging of a function of interest over realizations from a single run of the chain provides a consistent estimator of its expectation. Simple examples of functions of interest (for scalar $\boldsymbol{x}$) are $g(\boldsymbol{x}) = \boldsymbol{x}$, which yields $E(\boldsymbol{x} \mid D)$ and $g(\boldsymbol{x}) = [\boldsymbol{x} - E(x \mid D)]^2$, which gives $V(\boldsymbol{x} \mid D)$. We now outline a particular form of Markov chain scheme that has proved popular for a range of applications. This algorithm is due to Ref. 23 who adapted the work of Ref. 24. For further details of Markov chain Monte Carlo procedures and their application see Refs. 25 and 26.

A. Metropolis–Hastings Algorithm

This algorithm constructs a Markov chain $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^t, \ldots$, with equilibrium distribution $p(\boldsymbol{x} \mid D)$ by defining the transition from $\boldsymbol{x}^t = \boldsymbol{x}$ to the next realized state $\boldsymbol{x}^{t+1}$ as follows. Let $q(\boldsymbol{x}, \boldsymbol{x}')$ be an arbitrary transition probability function such that, if $\boldsymbol{x}^t = \boldsymbol{x}$, the vector $\boldsymbol{x}'$ drawn from $q(\boldsymbol{x}, \boldsymbol{x}')$ is considered as a candidate value for $\boldsymbol{x}^{t+1}$. With probability $\alpha(\boldsymbol{x}, \boldsymbol{x}')$, the candidate value is accepted, and $\boldsymbol{x}^{t+1} = \boldsymbol{x}'$; otherwise, the candidate value is rejected, and $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t$. The acceptance probability is defined as

$$\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{\frac{p(\boldsymbol{x}' \mid D) q(\boldsymbol{x}', \boldsymbol{x})}{p(\boldsymbol{x} \mid D) q(\boldsymbol{x}, \boldsymbol{x}')}, 1\right\} \quad \text{if} \quad p(\boldsymbol{x} \mid D) q(\boldsymbol{x}, \boldsymbol{x}') > 0 \tag{B1}$$

and $\alpha(\boldsymbol{x}, \boldsymbol{x}') = 1$ if $p(\boldsymbol{x} \mid D) q(\boldsymbol{x}, \boldsymbol{x}') = 0$. Then if $q(\boldsymbol{x}, \boldsymbol{x}')$ is chosen to be irreducible and aperiodic, this can be shown to be a sufficient condition for $p(\boldsymbol{x} \mid D)$ to be the equilibrium distribution of the chain.[22] It is important to note that the distribution of interest $p(\boldsymbol{x} \mid D)$ only enters $\alpha(\boldsymbol{x}, \boldsymbol{x}')$ through the ratio $p(\boldsymbol{x}' \mid D)/p(\boldsymbol{x} \mid D)$. This is crucial because the distribution only needs to be specified up to proportionality, i.e., likelihood times prior, for algorithm implementation, thus circumventing the need to obtain a normalization constant for the posterior distribution [see Eq. 12].

Because the Markov chain has the posterior distribution as its stationary distribution in the limit, it is usual to run the chain for an initial transient (or burn-in) period and then to discard this portion of the chain. The values obtained from then on are considered to be approximately distributed as the distribution of interest. However, neighboring states in the chain exhibit dependence, so that to obtain independent random samples it is necessary to leave a gap (called the decorrelation gap) between states. It is also necessary to carry out numerical checks as the chain progresses (particularly when the chain accesses regions of low posterior probability). An example of

a possible check is to ensure that the ergodic mean estimate satisfies the modal condition.[27]

B. Choices of Candidate Generating Function: $q(\boldsymbol{x}, \boldsymbol{x}')$

We now consider various possible choices of candidate generating density. For a more complete discussion see Ref. 22.

Random Walk Chains

The candidate state is generated by taking the current state and adding to it a random vector drawn from some fixed distribution, e.g., multivariate normal or Student-t distribution. In this case $q(\boldsymbol{x}, \boldsymbol{x}') = q(\boldsymbol{x}', \boldsymbol{x})$ and the acceptance criterion reduces to

$$\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{\frac{p(x' \mid D)}{p(x \mid D)}, 1\right\} \tag{B2}$$

In choosing the scale of the fixed distribution, a tradeoff must be made between the higher acceptance probabilities arising from smaller jumps in the random walk and the reduced correlation in the chain obtained by making the distribution of candidate values as diffuse as possible. Two possible methods of setting the scale matrix are as follows.

1) Use a constant c times the inverse information at the posterior mode. Suggested values[22] are $c = 0.5$ or 1, although no firm guidelines are available.

2) During the initial transient period, adapt the scale by inflating by $\delta\%$ whenever a sample is accepted and shrinking by $100/(100+\delta)\%$ whenever a candidate is rejected. This should give a final acceptance rate of about 50%. Typical values for δ are 1, 5, or 10.

Independence Chain

Here candidate states are drawn from a fixed density f. In this case $q(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x}')$ and the acceptance probability is

$$\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{\frac{w(x')}{w(x)}, 1\right\} \tag{B3}$$

where $w(\boldsymbol{x}) = p(\boldsymbol{x} \mid D)/f(\boldsymbol{x})$. The independence chain Metropolis–Hastings algorithm is closely related to the corresponding importance sampling process. Candidate states with low weights are rarely accepted, whereas candidate states with high weights are usually accepted, and the process will usually remain at these points for several steps, thus allowing repetition to build up weight on these points within the chain. If some points have very high weight, the chain may get stuck at these points for a long time. Thus, it is useful to choose f to produce a weight function that is bounded and nearly constant. This occurs when f very closely resembles $p(\boldsymbol{x} \mid D)$, but has thicker tails. If the weight function is constant, then candidate states are never rejected, and the chain produces an IID sample from $p(\boldsymbol{x} \mid D)$. A popular choice for f is a multivariate Student-t density with a low degree of freedom.

Autoregressive Chain

This is an intermediate strategy between random walk and independence chains. Candidate states are obtained by generating ϵ from a fixed density f and setting $\boldsymbol{x}' = a + b(\boldsymbol{x} - a) + \epsilon$, where a and b are fixed constants. Thus, $q(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x}' - a - b(\boldsymbol{x} - a))$, where $b = 1$ reduces this to a random walk chain and $a = b = 0$ gives an independence chain. If $b = -1$, then the candidate state is obtained by reflecting the current state about a and adding ϵ. In this case $q(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x} + \boldsymbol{x}' - 2a) = q(\boldsymbol{x}', \boldsymbol{x})$. This was suggested by Ref. 23 as a method of inducing negative correlation between successive states in the chain. This is useful because it should reduce the variance of estimates of expectations of linear functions. The reflection strategy is most useful when the density $p(\boldsymbol{x} \mid D)$ is approximately symmetric about a.

Combined Strategy Chain

These chain strategies can be used in their simple forms just described, or they can be combined to form a compound strategy (or hybrid strategy[22]). For example, the chain could be formed as a repeating cycle of random walk and independence steps. Alternatively, the current choice of strategy can be randomly selected according to mixing probabilities. The advantage of using mixed strategies is that it can help to avoid the problem of the chain getting stuck in a small portion of state space.

C. Determining the Accuracy of Moment Estimates

For an IID sample of size n from the posterior $p(\boldsymbol{x} \mid D)$, the variance of the sample mean is σ^2/n, where σ^2 is the posterior variance. However, when using a realization of a Markov chain, there exists dependence between terms in the chain. If the correlation between terms separated by a lag j on the chain is ρ_j, then it can be shown[28] that, for a scalar state,

$$V(\hat{\mu}) = \frac{\sigma^2}{n}\left[1 + 2\sum_{j=1}^{n-1}\left(1 - \frac{j}{n}\right)\rho_j\right] \tag{B4}$$

where $\hat{\mu}$ is the sample mean. Because ρ_j is often positive, it can be seen that the standard error for dependent observations is greater than that for IID observations, and hence a longer chain will be required for a given accuracy. If the chain can be modeled as a first-order autoregressive process of parameter α (where $-1 \le \alpha \le 1$), then $\rho_j = \alpha^{|j|}$, and it follows that

$$V(\hat{\mu}) = \frac{\sigma^2}{n}\left[\left(\frac{1+\alpha}{1-\alpha}\right) - \frac{2\alpha(1-\alpha^n)}{n(1-\alpha)^2}\right] \tag{B5}$$

which asymptotically (as $n \to \infty$) becomes

$$V(\hat{\mu}) = (\sigma^2/n)[(1+\alpha)/(1-\alpha)] \tag{B6}$$

Hence, a rough estimate of α can be used to adjust the standard error and allow for chain dependence. Note that if negative correlation can be induced in the chain then the standard error will be reduced relative to IID observations, and hence a shorter chain can be used for a given accuracy. This is the reason for considering the autoregressive chain structure described earlier. It is also possible to use the estimated value of α (and, hence, ρ_j) to set the required decorrelation gap.

References

[1]Skolnik, M. I., *Introduction to Radar Systems*, 2nd ed., McGraw–Hill, New York, 1981, p. 169.

[2]Barton, D. K., *Modern Radar Systems Analysis*, Artech House, Boston, 1988, p. 120.

[3]Masreliez, C. J., "Approximate Non-Gaussian Filtering with Linear State and Observation Relations," *IEEE Transactions on Automatic Control*, Vol. 20, Feb. 1975, pp. 107–110.

[4]Kitagawa, G., "Non-Gaussian State Space Modelling of Non-Stationary Time Series (with Discussion)," *Journal of the American Statistical Association*, Vol. 82, No. 400, 1987, pp. 1032–1063.

[5]Pole, A., and West, M., "Efficient Bayesian Learning in Dynamic Models," *Journal of Forecasting*, Vol. 9, No. 2, 1990, pp. 119–136.

[6]Borden, B. H., and Mumford, M. L., "A Statistical Glint/Radar Cross Section Target Model," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 19, No. 5, 1983, pp. 781–785.

[7]Daeipour, E., and Bar-Shalom, Y., "An Interacting Multiple Model Approach for Target Tracking with Glint Noise," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 31, No. 2, 1995, pp. 706–715.

[8]Jazwinski, A. H., *Stochastic Processes and Filtering Theory*, Academic, New York, 1970, p. 278.

[9]Alspach, D. L., and Sorenson, H. W., "Nonlinear Bayesian Estimation Using Gaussian Sum Approximation," *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, 1972, pp. 439–447.

[10]Gordon, N. J., Salmond, D. J., and Smith, A. F. M., "A Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proceedings on Radar and Signal Processing*, Vol. 140, No. 2, 1993, pp. 107–113.

[11]Silverman, B. W., *Density Estimation for Statistics and Data Analysis*, Champan and Hall, London, 1984.

[12]Gordon, N. J., Salmond, D. J., and Ewing, C. M., "Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, 1995, pp. 1434–1443.

[13]Gordon, N. J., "A Hybrid Bootstrap Filter for Tracking in Clutter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 1, 1997, pp. 353–358.

[14]Muller, P., "Monte Carlo Integration in General Dynamic Models," *Contemporary Mathematics*, Vol. 115, American Mathematical Society, Providence, RI, 1991, pp. 145–163.

[15]Carlin, B. P., Polson, N. G., and Stoffer, D. S., "A Monte-Carlo Approach to Nonnormal and Nonlinear State Space Modelling," *Journal of the American Statistical Association*, Vol. 87, No. 418, 1992, pp. 493–500.

[16]West, M., "Modelling with Mixtures (with Discussion)," *Bayesian Statistics*, Vol. 4, Oxford Univ. Press, Oxford, England, UK, 1992, pp. 503–524.

[17]Wu, W.-R., "Target Tracking with Glint Noise," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 1, 1993, pp. 174–185.

[18]Wu, W.-R., and Cheng, P.-P., "A Nonlinear IMM Algorithm for Manoeuvring Target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 30, No. 3, 1994, pp. 875–885.

[19]Bryson, A. E., and Henrikson, L. J., "Estimation Using Sampled Data Containing Sequentially Correlated Noise," *Journal of Spacecraft*, Vol. 5, No. 6, 1968, pp. 662–665.

[20]Bar-Shalom, Y., Kumar, A., Blair, W. D., and Groves, G. W., "Tracking Low Elevation Targets in the Presence of Multipath Propagation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 30, No. 3, 1994, pp. 973–979.

[21]Salmond, D. J., "Mixture Reduction Algorithms for Target Tracking in Clutter," *SPIE Signal and Data Processing of Small Targets 1990*, edited by O. E. Drummond, Vol. SPIE 1305, Society of Photo–Optical Instrumentation Engineers, Orlando, FL, 1990, pp. 434–445.

[22]Tierney, L., "Markov Chains for Exploring Posterior Distributions (with Discussion)," *Annals of Statistics*, Vol. 22, No. 4, 1994, pp. 1701–1762.

[23]Hastings, W. K., "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, Vol. 57, No. 1, 1970, pp. 97–109.

[24]Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equations of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21, No. 6, 1953, pp. 1087–1092.

[25]Smith, A. F. M., "Bayesian Computational Methods," *Philosophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 337, No. 1647, 1991, pp. 369–386.

[26]Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.

[27]Gordon, N. J., and Smith, A. F. M., "Approximate Non-Gaussian Bayesian Estimation and Modal Consistency," *Journal of the Royal Statistical Society, Series B*, Vol. 55, No. 4, 1993, pp. 913–918.

[28]Chen, M.-H., and Schmeiser, B., "Performance of the Gibbs, Hit-and-Run and Metropolis Samplers," *Journal of Computational and Graphical Statistics*, Vol. 2, No. 3, 1993, pp. 251–272.