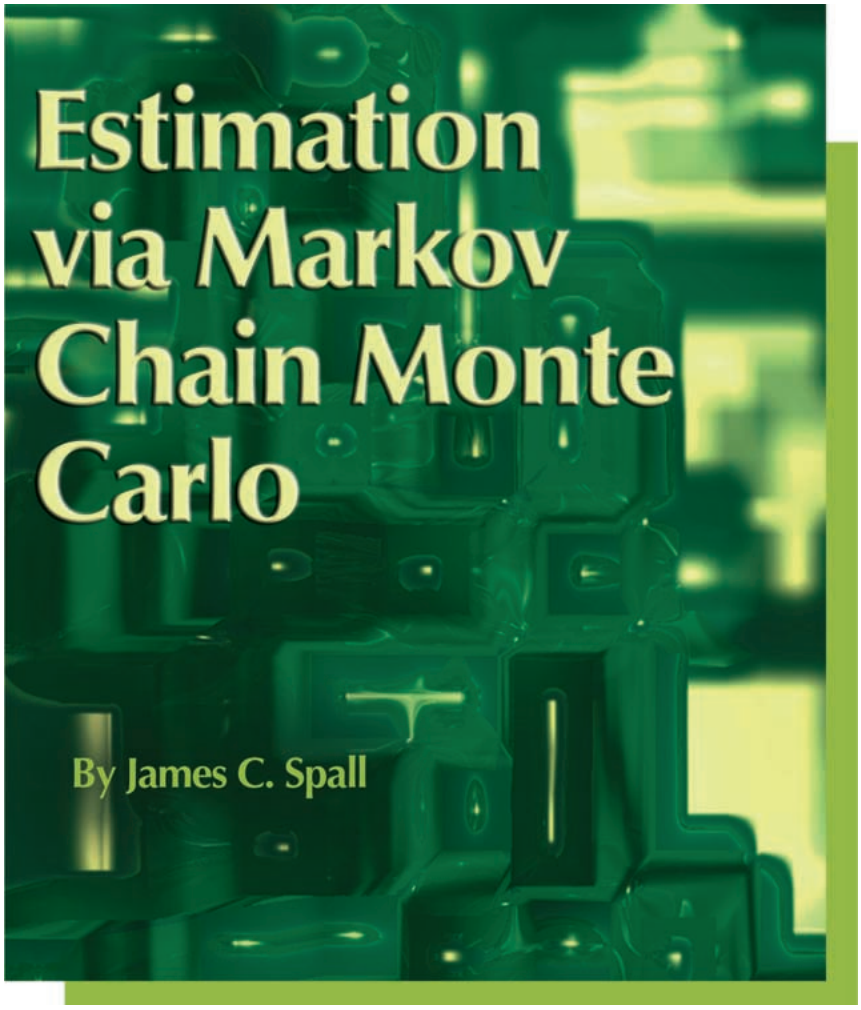

Markov chain Monte Carlo (MCMC) is a powerful means for generating random samples that can be used in computing statistical estimates and marginal and conditional probabilities. MCMC methods rely on *dependent* (Markov) sequences having a limiting distribution corresponding to a distribution of interest. (Note that the use of the term "Markov *chain*" in MCMC is not entirely consistent with standard usage in stochastic processes. The term is generally reserved for use with processes having discrete outcomes. For consistency with standard terminology in the MCMC area, however, we follow suit in this article in using the term Markov chain under the more general application to discrete or continuous outcomes.) The use of dependent sequences contrasts with many classical Monte Carlo methods, which are based on *independent* samples. MCMC methods have the great advantage that they apply to a broader class of multivariate problems than methods based on independent sampling. Over the last 10-15 years, the approach has had a large impact on the theory and practice of statistical modeling. On the other hand, MCMC has had relatively little impact (yet) on estimation problems in control.

Background

This article is a survey of popular implementations of MCMC, focusing particularly on the two most popular specific implementations of MCMC: Metropolis-Hastings (M-H) and Gibbs sampling. Although the results presented have a rigorous basis, the presentation here is relatively informal in the hopes of conveying the main ideas relevant to implementation. Our aim is to provide the reader with some of the central motivation and the rudiments needed for a straightforward application. The cited references provide extensive detail not presented here, including the rigorous justification for many of the results.

Although MCMC has general applicability, one area where it has had a revolutionary impact is Bayesian analysis. MCMC has greatly expanded the range of problems for which Bayesian methods can be applied. Metropolis et al. [1] introduced MCMC-type methods in the area of physics. Following key papers by Hastings [2], Geman and Geman [3], and Tanner and Wong [4], the paper of Gelfand and Smith [5] is largely credited with introducing the applications of MCMC methods in modern statistics, specifically in Bayesian modeling.

Over the last decade, many papers and books have been published displaying the power of MCMC in dealing with realistic problems in a wide variety of areas. Among the excellent review papers in this area are the survey by Besag et al. [6] and the vignettes by Cappé and Robert [7] and Gelfand [8].

The author (james.spall@jhuapl.edu) is with the Johns Hopkins University, Applied Physics Laboratory, 11100 Johns Hopkins Rd., Laurel, MD 20723-6099, U.S.A.

0272-1708/03/$17.00©2003IEEE

The survey by Evans and Swartz [9] puts MCMC in perspective relative to other powerful methods for numerical integration in a statistical setting. The February 2002 issue of *IEEE Transactions on Signal Processing* is devoted to MCMC and closely related methods as used in signal processing and tracking applications. Among the many books devoted to MCMC and its close algorithmic relatives are Gilks et al. [10], Robert and Casella [11], Chen et al. [12], and Liu [13]. Within the field of control there has been a relatively small number of papers on MCMC, including Doucet et al. [14].

The Prototype

Following convention, we use the terms *density* and *distribution* interchangeably when referring to the mechanism for generating a random process; this does not imply, of course, that a density is the same as a distribution *function*. The prototype problem is as follows. Suppose there is a process generating a random vector $\boldsymbol{X}$, and we wish to compute $E[f(\boldsymbol{X})]$ for some function $f(\cdot)$. We are interested in the case where this expectation is not readily available via standard analytical means. For the moment, let us suppose that $\boldsymbol{X}$ is a continuous random vector with an associated probability density function $p(\boldsymbol{x}) = p_{\boldsymbol{X}}(\boldsymbol{x})$. The methods to be described are quite general and are not inherently tied to this assumption of continuity. We usually adopt this restriction for ease of presentation and because such continuous random processes are very common. Hence, we may write

$$E[f(\boldsymbol{X})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}, \tag{1}$$

where the integral is over the domain for $\boldsymbol{X}$. The density $p(\boldsymbol{x})$ is sometimes called the "target density." More generally, the target density (distribution) represents the distribution for the random variables of interest for the analysis. In some cases, for example, the target will pertain to a subset of the elements in $\boldsymbol{X}$ (e.g., it may represent the marginal distribution for only the first component of $\boldsymbol{X}$).

A standard Monte Carlo method for approximating the integral in (1) is to draw n independent identically distributed (i.i.d.) samples of $\boldsymbol{X}$, say $\boldsymbol{X}_k, k = 1,2,...,n$, from the density function $p(\boldsymbol{x})$. We then form the average of these independent samples, leading to

$$E[f(\boldsymbol{X})] \approx \frac{1}{n}\sum_{k=1}^{n} f(\boldsymbol{X}_k).$$

Because the samples $\boldsymbol{X}_k$ are i.i.d., the strong and weak laws of large numbers ensure that the approximation can be made as accurate as desired with increasing n.

Often, however, drawing samples from the density $p(\boldsymbol{x})$ is not feasible. The density may be very complicated and perhaps not even available analytically. Random number generation methods, such as in Rubinstein and Melamed [15, Chap. 2], are generally quite limited in the type of randomness that can be produced. In practice, many distributions are not of the restricted "named" type for which most standard methods apply. The accept-reject method, which is much more general than the inverse transform method, is usually only computationally practical if the structure of $p(\boldsymbol{x})$ is exploited to find a "tight" majorizing function (typically only available after a separate optimization process). A related general method for random number generation is a version of the accept-reject method called adaptive rejection sampling [11, pp. 56-59, 232]. This method is restricted to densities that are log-concave (i.e., the logarithm of the density is a concave function) and known to be very inefficient in high-dimensional problems (computations proportional to the *fifth power* of $\dim(\boldsymbol{X})$; see, e.g., [10, p. 84]). Moreover, this sampling method may not be as efficient as the Markov chain-based methods to be described later [11, p. 241].

Fortunately, the integral approximation above, with its i.i.d. summands and requirement for direct sampling from $p(\boldsymbol{x})$, is more restrictive than necessary to form an estimate of $E[f(\boldsymbol{X})]$. An integral can be approximated by a possibly *dependent* sample $\{\boldsymbol{X}_k\}$ that properly reflects the proportions associated with $p(\boldsymbol{x})$. This less stringent requirement opens up the possibility of efficient Markov chain-based schemes that avoid the need to directly sample from $p(\boldsymbol{x})$. In particular, we can use Markov chain-based Monte Carlo methods to efficiently produce dependent sequences having $p(\boldsymbol{x})$ as a limiting distribution without the difficult or impossible task of sampling directly from $p(\boldsymbol{x})$.

An additional important benefit of MCMC methods is that $p(\boldsymbol{x})$ need only be known to within a scale factor. This is especially relevant in Bayesian applications where the posterior density function is the target density. The posterior depends on an often difficult-to-compute integral (the denominator term in Bayes' rule). With MCMC, it is not necessary to know this integral.

Consider a sequence $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2,...$ such that $\boldsymbol{X}_{k+1}$ is generated from the (conditional) distribution for $\{\boldsymbol{X}_{k+1} | \boldsymbol{X}_k\}$ and $\boldsymbol{X}_0$ represents some initial condition (not needed in the i.i.d. case above). By the form of the conditional distribution, knowledge of $\boldsymbol{X}_k$ provides the information required to probabilistically characterize the behavior of the state $\boldsymbol{X}_{k+1}$. That is, the distribution for $\boldsymbol{X}_{k+1}$ depends only on the most recent state, not on the earlier states $\boldsymbol{X}_0, \boldsymbol{X}_1,..., \boldsymbol{X}_{k-1}$. Hence, $\boldsymbol{X}_0$, $\boldsymbol{X}_1, \boldsymbol{X}_2,...$ is a Markov chain under the proviso that the definition of "chain" includes continuous random processes.

Under standard conditions for Markov chains, the dependence of $\boldsymbol{X}_k$ on any fixed number of early states, say $\boldsymbol{X}_0$, $\boldsymbol{X}_1,..., \boldsymbol{X}_M$, $M < \infty$, disappears as $k \to \infty$ (equivalently, $k - M \to \infty$). Hence, the density (distribution) of $\boldsymbol{X}_k$ will approach a stationary form, say $p^*(\cdot)$. That is, as k gets large, the random vectors in the Markov chain will become a dependent sequence with a common density $p^*(\cdot)$. Ignoring the first M iterations in the chain (called the "burn-in" period), we can form an ergodic average

$$\frac{1}{n-M}\sum_{k=M+1}^{n} f(\boldsymbol{X}_k), \tag{2}$$

so called because it is a practical realization of the famous ergodic theorem of stochastic processes.

The ergodic theorem guarantees that the normalized sum in (2) will approach the mean of $f(\boldsymbol{X})$ (usually in the mean square or almost sure sense) as $n \to \infty$ for any fixed M, where this mean is computed with respect to $p^*(\cdot)$. Necessary and sufficient conditions for the ergodic theorem for mean square convergence to hold are that: i) $\text{cov}[f(\boldsymbol{X}_j), f(\boldsymbol{X}_k)]$ is uniformly bounded in magnitude for all j, k and ii) the correlation between $f(\boldsymbol{X}_n)$ and the sample mean in (2) goes to zero as $n \to \infty$ [16, pp. 72-75]. In other words, the process is ergodic in the mean square sense if there is less correlation between the terminal observation $f(\boldsymbol{X}_n)$ and the sample mean in (2) as n is increased. The key idea in MCMC methods is to design Markov chains that are ergodic and have $p^*(\cdot)$ equaling the target density $p(\cdot)$, as desired. That is, the limit of the ergodic mean in (2) will correspond to the desired value $E[f(\boldsymbol{X})]$ computed with respect to $p(\cdot)$. It is surprisingly easy to construct such chains.

Earlier we outlined a general formulation for generating random samples via the output of a Markov chain. There are, of course, some fundamental elements that need to be specified to justify the approach and to provide the details required for implementation. These elements form the basis for the MCMC approach.

In the remainder of this article, we present an overview of the M-H and Gibbs sampling implementations of MCMC, discuss applications in Bayesian modeling, sketch the theoretical foundations, and present some examples. There are also some important aspects of MCMC that we do not cover here. For example, we do not discuss formal methods for diagnosing convergence, which is an active area of current research, with numerous specialized techniques. Robert and Casella [11, Chap. 8] are one of many references on this subject. As with general stochastic search and optimization methods, there is no universal method for knowing when to stop a chain. We also do not review the many software packages that are available for both simple and sophisticated implementations of MCMC. (One prominent package is worthy of mention, however. BUGS [***B***ayesian inference ***U***sing ***G***ibbs ***S***ampling] is available free on the Web and is one of the most popular "standard" packages. BUGS is available at www.mrc-bsu.cam.ac.uk/bugs.)

Metropolis-Hastings Algorithm

As discussed earlier, the sum of dependent random vectors in (2) converges to the mean of $f(\boldsymbol{X})$ under appropriate conditions. Although this is hopeful, we must show that this is the "right" mean; that is, the mean computed with respect to $p(\boldsymbol{x})$. Fortunately, it is surprisingly easy to produce such a Markov process via a variant of the Metropolis sampling appearing in the popular simulated annealing algorithm for search and optimization [17]. The form given here was introduced by Hastings [2] and builds on the criterion in Metropolis et al. [1].

The M-H algorithm is a mechanism for producing the Markov process $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ for use in (2). Given state $\boldsymbol{X}_k = \boldsymbol{x}$, the next state is chosen by generating a candidate point $\boldsymbol{W}$ from a "proposal distribution" (sometimes called an instrumental distribution or a candidate-generating distribution) $q(\cdot|\boldsymbol{x})$. In principle, this proposal distribution may be chosen arbitrarily, although there may be efficiency advantages to one form over another in some applications. The proposal distribution satisfies the key condition for density functions, namely:

$$\int q(\boldsymbol{w}|\boldsymbol{X} = \boldsymbol{x})d\boldsymbol{w} = \int q(\boldsymbol{x}|\boldsymbol{W} = \boldsymbol{w})d\boldsymbol{x} = 1$$

for any $\boldsymbol{X} = \boldsymbol{x}$ or $\boldsymbol{W} = \boldsymbol{w}$, as appropriate. There are also some very modest regularity conditions, as discussed in Robert and Casella [11, pp. 233-235]. For example, the set of points $\boldsymbol{w}$ where $q(\boldsymbol{w}|\boldsymbol{x}) > 0$, as we let $\boldsymbol{X} = \boldsymbol{x}$ range over the set of points where $f(\boldsymbol{x}) \neq 0$, should be a superset of the set of points $\boldsymbol{x}$ where $f(\boldsymbol{x}) \neq 0$. A common example for $q(\cdot|\boldsymbol{x})$, which satisfies the superset condition, is a uniform distribution centered around $\boldsymbol{x}$. One implication of these conditions is that it is possible to generate candidate points that will "fill up" the support of the target density (i.e., provide an adequate number of points throughout the region where $p(\cdot) > 0$).

Analogous to the key decision step of the simulated annealing algorithm, the candidate point $\boldsymbol{W}$ is accepted with probability $\rho(\boldsymbol{X}_k, \boldsymbol{W})$, where

$$\rho(\boldsymbol{x}, \boldsymbol{w}) = \min\left\{\frac{p(\boldsymbol{w})}{p(\boldsymbol{x})}\frac{q(\boldsymbol{x}|\boldsymbol{w})}{q(\boldsymbol{w}|\boldsymbol{x})}, 1\right\}. \tag{3}$$

If the candidate point is accepted, then $\boldsymbol{X}_{k+1} = \boldsymbol{W}$; otherwise $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k$. Note that $q(\cdot|\cdot)$ appearing in both the numerator and denominator of (3) is the same function with only the conditioning interchanged ($\boldsymbol{x}$ and $\boldsymbol{w}$ have the same dimension). (In general, the functional form for the distribution of one random variable conditioned on another depends on the order of the conditioning.) Likewise $p(\cdot)$ in the numerator and denominator is the same function with only the arguments changed. (The connection of MCMC to simulated annealing is explored further in Robert and Casella [11, p. 281] and Andrieu et al. [18]. General discussion on simulated annealing and the connection to Markov chains appears in Spall [19, chap. 8].)

Because of the ratio form, an important implication of (3) is that one only needs to know $p(\cdot)$ to within a constant because the constant cancels out. Eliminating the need to determine the constant has significant practical advantages by removing the need for a formidable numerical integration. For example, in Bayesian applications, the target density represents a posterior density that is conditioned on some set of data (the data conditioning does not affect the mechanics of the M-H algorithm). It is notoriously difficult to obtain the constant associated with the posterior density function because the constant is the marginal density for

the data appearing in the denominator of Bayes' rule. This marginal density requires difficult numerical integration. We will discuss this further in the section "Applications in Bayesian Analysis."

Table 1 summarizes two common proposal distributions. Suppose that $m = \dim(\boldsymbol{X})$ (so $m = \dim(\boldsymbol{W})$). Let us denote an m-fold uniform distribution by $U_m(\boldsymbol{a},\boldsymbol{b})$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are m-dimensional vectors. This distribution is such that each component of the m-dimensional random vector has an independent uniform distribution with lower and upper endpoints given by the corresponding component of $\boldsymbol{a}$ and $\boldsymbol{b}$ (so the probability is uniform over the hypercube defined by $\boldsymbol{a}$ and $\boldsymbol{b}$). The two candidate-generating processes below are examples of so-called random walk processes because the candidate point may be written as the current point plus noise: $\boldsymbol{W} = \boldsymbol{X}$ + noise, where the noise is a mean $\mathbf{0}$ normal or uniform distribution. Note further that $q(\boldsymbol{w}|\boldsymbol{x}) = q(\boldsymbol{x}|\boldsymbol{w})$, an example of the important special case where the proposal distribution is symmetric. An implication of the symmetric proposal is that the criterion (3) simplifies to

$$\rho(\boldsymbol{x},\boldsymbol{w}) = \min\left\{\frac{p(\boldsymbol{w})}{p(\boldsymbol{x})}, 1\right\}$$

as a result of $q(\boldsymbol{x}|\boldsymbol{w}) / q(\boldsymbol{w}|\boldsymbol{x}) = 1$. For the m-fold uniform distribution in Table 1, let $\mathbf{1}_m$ denote the m-dimensional vector of ones.

A remarkable result associated with (3) is that although the proposal distribution $q(\cdot|\cdot)$ may have any form (subject to the modest conditions mentioned above), the stationary distribution of the chain will satisfy $p^*(\cdot) = p(\cdot)$ [2]. Chib and Greenberg [20] and Robert and Casella [11, pp. 235-238] elaborate on some of the arguments in Hastings [2]. A summary of the steps for the M-H algorithm is provided in Panel 1. The first several steps pertain to the "burn-in" period and the remaining steps are used to form the ergodic average in (2).

The M-H algorithm can be implemented in many ways. The most obvious variation in implementation is in the choice of the proposal distribution $q(\cdot|\cdot)$. Although almost any choice of $q(\cdot|\cdot)$ will work in the sense that the ergodic average in (2) will converge to $E[f(\boldsymbol{X})]$, there are clear differences in the rate of convergence depending on the nature of the problem. There are also forms of averaging that differ from the standard ergodic averaging. One variation is to run many independent chains, with each chain terminating at $\boldsymbol{X}_{M+1}$. In this way, $E[f(\boldsymbol{X})]$ is estimated by forming a sample mean of independent values $f(\boldsymbol{X}_{M+1})$. Regeneration, as discussed by Rubinstein and Melamed [15, sect. 3.7] and others, may also be used to improve the performance of M-H [21]. This creates independent blocks of iterations, allowing for the proposal distribution to be adapted at each block to improve the sampling.

Panel 1. Metropolis-Hastings Algorithm for Estimating $E[f(\boldsymbol{X})]$.

Step 0. **(Initialization)** Choose the length of the "burn-in" period M and an arbitrary initial state $\boldsymbol{X}_0$. Set $k = 0$.

Step 1. Generate a candidate point $\boldsymbol{W}$ according to the proposal distribution $q(\cdot|\boldsymbol{X}_k)$.

Step 2. Generate a point U from a $U(0,1)$ distribution. Set $\boldsymbol{X}_{k+1} = \boldsymbol{W}$ if $U \le \rho(\boldsymbol{X}_k,\boldsymbol{W})$ from (3). Otherwise, set $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k$.

Step 3. Repeat steps 1 and 2 until $\boldsymbol{X}_M$ is available. Terminate the "burn-in" process and proceed to step 4 with $\boldsymbol{X}_k = \boldsymbol{X}_M$.

Step 4. Carry out step 1.

Step 5. Carry out step 2.

Step 6. Repeat steps 4 and 5 until it is possible to compute the ergodic average of $n - M$ evaluations in (2). (Of course, if desired, this average can be computed recursively without storing all of $f(\boldsymbol{X}_{M+1})$, $f(\boldsymbol{X}_{M+2}),\ldots,f(\boldsymbol{X}_n)$.) This ergodic average is the estimate of $E[f(\boldsymbol{X})]$ under the target density $p(\cdot)$.

The rate at which candidate values $\boldsymbol{W}$ are accepted affects M-H performance. This rate should be neither too small nor too large to enhance the performance of M-H. Roberts and Rosenthal [22] point out that a rate of 23% is a "good"—and sometimes optimal—acceptance rate. The robustness of this value has long been observed both theoretically and empirically. Although it is not always possible to achieve this rate in practice, it is sometimes possible to run "test iterations" of the algorithm, adjusting parameters in the proposal distribution until the acceptance rate is approximately 23%.

Below we present a simple example where the target density is bivariate normal. We use this example to demonstrate the performance of M-H in a setting that is easy to understand. In practice, there are other (likely more) efficient methods of generating samples from a multivariate normal distribution, as discussed in any general reference on pseudorandom number generation (e.g., Appendix D in Spall [19]).

Table 1. Examples of two popular general forms for proposal distributions.

General form of proposal distribution	$q(\boldsymbol{w}\|\boldsymbol{x})$	$q(\boldsymbol{x}\|\boldsymbol{w})$
Normal with covariance matrix $\boldsymbol{\Sigma}$	$N(\boldsymbol{x},\boldsymbol{\Sigma})$	$N(\boldsymbol{w},\boldsymbol{\Sigma})$
Uniform of width 2δ for each component	$U_m(\boldsymbol{x}-\delta\mathbf{1}_m,\boldsymbol{x}+\delta\mathbf{1}_m)$	$U_m(\boldsymbol{w}-\delta\mathbf{1}_m,\boldsymbol{w}+\delta\mathbf{1}_m)$

Example 1: Simulating a Bivariate Normal Distribution

Let us consider a problem where the target density is bivariate normal with the two variables moderately correlated. In particular, suppose that

$$\boldsymbol{X} \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right).$$

Further, let us suppose that the proposal distribution $q(\boldsymbol{w}|\boldsymbol{x})$ is a shifted uniform distribution as in Table 1. In particular, suppose $\boldsymbol{W}$ (conditioned on $\boldsymbol{X} = \boldsymbol{x}$) is generated according to $U_2(\boldsymbol{x} - 3\mathbf{1}_2, \boldsymbol{x} + 3\mathbf{1}_2)$. Because $q(\boldsymbol{w}|\boldsymbol{x}) = q(\boldsymbol{x}|\boldsymbol{w})$, the form of the normal target density function implies that

$$\rho(\boldsymbol{x},\boldsymbol{w}) = \min\left\{\frac{\exp\left(-\frac{1}{2}\boldsymbol{w}^T\begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}^{-1}\boldsymbol{w}\right)}{\exp\left(-\frac{1}{2}\boldsymbol{x}^T\begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}^{-1}\boldsymbol{x}\right)}, 1\right\}$$

As discussed earlier, note that the constant terms in the bivariate normal density function are not needed in constructing $\rho(\cdot)$ (a trivial advantage here, but a major advantage in applications such as Bayesian analysis).

Figure 1 shows the results of a study where M-H was used to estimate $E[f(\boldsymbol{X})]$ where $f(\boldsymbol{X}) = [1,1]\boldsymbol{X}$. Thus, we are estimating the sum of the means for the two elements of $\boldsymbol{X}$. The figure shows the evolution of the ergodic averages for four independent runs. The $M = 500$ burn-in period for each run is initialized at $\boldsymbol{X}_0 = [-1,1]^T$. We see that the four runs are all settling down near the true value $E[f(\boldsymbol{X})] = 0$.

For each of the runs, about 26% of the candidate points $\boldsymbol{W}$ were accepted (i.e., about 26% of the time, $\boldsymbol{X}_{k+1} \neq \boldsymbol{X}_k$ in the M-H steps). Empirically, this rate seemed to provide good results for this problem. Further, this rate is close to the above-mentioned "good" rate of 23%, which is optimal or near optimal in a wide range of problems. (The convergence theory associated with M-H guarantees that the estimate will be reliable if the algorithm is run long enough, even if the acceptance rate is not "optimal" for the particular application.) Other forms for the proposal distribution may also be valuable; as mentioned earlier, the normal distribution is a frequent proposal form.

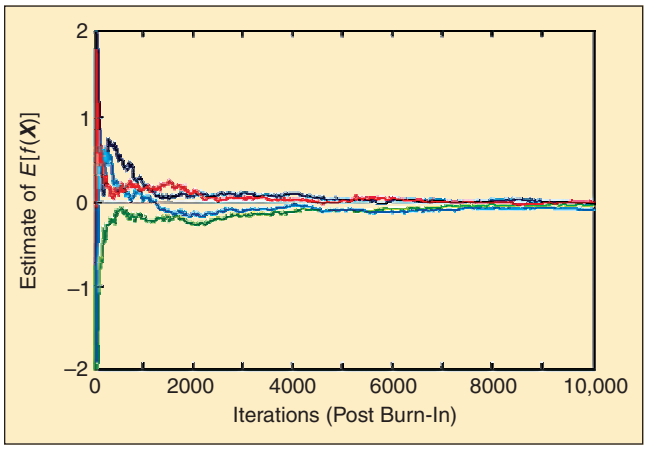

Figure 1. *Traces for four independent runs of M-H sampler in estimating $E[f(\boldsymbol{X})]$ (i.e., estimating the sum of the two components in $E(\boldsymbol{X})$). Target value is zero. Burn-in period (M) is 500 for each run.*

Gibbs Sampling

Gibbs sampling represents an implementation of the M-H algorithm on an element-by-element basis for the components in $\boldsymbol{X}$. The term Gibbs sampling was introduced by Geman and Geman [3] in a specific implementation of a Gibbs distribution for sampling on lattices. (Gibbs sampling derives its name from the physicist Josiah W. Gibbs, 1839-1903, based on the connection to Gibbs random fields identified in [3].) The term is now used more generally (and casually) to refer to the special case where the proposal distribution is built directly from the density of interest $p(\cdot)$. Because of this close connection, the method is more restrictive than M-H; however, these restrictions sometimes lead to advantages in efficiency and ease of implementation via the elimination of the "tuning" needed in M-H. Gibbs sampling is especially important in Bayesian implementations. Gibbs sampling is uniquely designed for multivariate problems. In fact, "the crucial issue is replacement of the sampling of a high-dimensional vector with sampling of lower-dimensional component blocks, thus breaking the so-called curse of dimensionality" [8].

The Gibbs sampling method is based on a concatenation of m M-H algorithms, one for each variable in the random vector of interest, $\boldsymbol{X}$. (Because this version of M-H does not generate a new vector in toto, it is not precisely the same as the standard multivariate form presented earlier.) This concatenation has m target distributions, each representing a conditional distribution for one variable given values for all other variables (called the full conditional distribution, as discussed below). In contrast to the general M-H algorithm, the proposal distributions have a required form: namely, the proposal distribution for the ith element of $\boldsymbol{X}$ is the conditional distribution of that variable given the most recent values for all other variables. Thus, the target and proposal distributions are the same.

To make the above concepts more concrete, consider a trivariate ($m = 3$) problem based on density functions, with the three variables being X, Y, and Z. The three target densities are $p_{X|Y,Z}(\cdot|\cdot)$, $p_{Y|X,Z}(\cdot|\cdot)$, and $p_{Z|X,Y}(\cdot|\cdot)$. As with the generic M-H, let $\boldsymbol{W}$ represent a candidate random variable generated according to the candidate density. For the first element, $\boldsymbol{X}$, the candidate-generating density is then

$$q(w|y,z) = p_{X|Y,Z}(w|y,z). \tag{4}$$

Substituting (4) into the probability of acceptance for the M-H algorithm as given in (3), we have

$$\rho(x,w)=\min\left\{\frac{p_{X|Y,Z}(w|y,z)}{p_{X|Y,Z}(x|y,z)}\frac{q(x|y,z)}{q(w|y,z)},1\right\}=\min\{1,1\}=1. \tag{5}$$

Relationship (5) implies that, unlike the general M-H algorithm, the new point W is always accepted as a representation for X. Hence, from (5), given any previous values for Y and Z, the candidate value W generated from $q(w|y,z) = p_{X|Y,Z}(w|y,z)$ is guaranteed to be the new value for X. Identical arguments apply for the other two variables Y and Z. In fact, because of the automatic acceptance of a candidate point, there is no need for a separate (W) candidate process for each variable; one simply generates a new X, Y, or Z as appropriate. The general multivariate relationship between Gibbs sampling and M-H is discussed in Robert and Casella [11, pp. 296-297] and Gilks et al. [10, pp. 10-12]. This relationship is an obvious extension of the trivariate one above.

As a consequence of the theory of Markov processes, the values X, Y, or Z generated via the Gibbs sampler will, in the limit, represent observations from the joint density $p_{X,Y,Z}(x,y,z)$. Likewise, X,Y, and Z individually have distributions that approach their respective marginals $p_X(x)$, $p_Y(y)$, and $p_Z(z)$.

The specific implementation of the Gibbs sampler for the trivariate problem of generating samples from $p_{X,Y,Z}(\cdot)$ is as follows. Suppose that we begin the sampler with an initial guess at Y and Z, say Y_0 and Z_0. Using the full conditional $p_{X|Y,Z}(x|Y_0,Z_0)$, we can then generate a sample point X_1 by Monte Carlo. We next use the full conditional $p_{Y|X,Z}(y|X_1,Z_0)$ to generate a sample point Y_1. Likewise, we use $p_{Z|X,Y}(z|X_1,Y_1)$ to generate Z_1. At this point, we have completed one iteration of the Gibbs sampler, producing X_1, Y_1, and Z_1. We now repeat the process, using Y_1 and Z_1 to initiate the conditioning and producing a sample X_2, Y_2, and Z_2. This Monte Carlo sampling forms a sequence

$$Y_0,Z_0;X_1,Y_1,Z_1;X_2,Y_2,Z_2;\ldots;X_n,Y_n,Z_n. \tag{6}$$

The sequence in (6) is called the Gibbs sequence. The sequential sampling approach above extends in an obvious way to the general m case.

The Gibbs sequence above can be used in several ways to estimate $E[f(X,Y,Z)]$. For example, let M denote the burn-in period, as with the M-H algorithm. Under modest conditions, the later observations, X_{M+1}, Y_{M+1}, $Z_{M+1};\ldots;X_n,Y_n,Z_n$, in the Gibbs sequence will represent measurements from $p_{X,Y,Z}(\cdot)$. These can then be substituted in the averaging of (2) to produce an estimate of $E[f(X,Y,Z)]$. A variation on standard ergodic averaging is to pick off every (say) ℓth value in the chain and average only these values. If ℓ is reasonably large, this will be roughly equivalent to averaging independent samples. Yet another way of estimating $E[f(X,Y,Z)]$ is to generate N independent Gibbs sequences and use only the final output in each sequence. Now, $E[f(X,Y,Z)]$ will be estimated by an average of N samples, where the summands are i.i.d.

We emphasize that the sampling may be done according to continuous, discrete, or hybrid random variables. The relevant distributions need not be represented as probability density functions. One simple illustration of this point is the example associated with a Bernoulli (binary) probability distribution that is given in Casella and George [23]. In that example, the relevant bivariate random vector $[X,Y]^T$ has a joint probability mass function. The next section of this article sketches the theoretical foundation for Gibbs sampling.

This discussion serves to motivate the general multivariate setting. In this more general setting, we continue to be interested in estimating quantities of the form $E[f(\boldsymbol{X})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$ (or its discrete analog), where $\boldsymbol{X}$ is a collection of m (say) univariate or multivariate components. In the common case where the m components are univariate, the kth sample $\boldsymbol{X}$ from the Gibbs sampling algorithm is

$$\boldsymbol{X}_k=\begin{bmatrix}X_{k1}\\X_{k2}\\\vdots\\X_{km}\end{bmatrix},$$

where X_{ki} denotes the ith component for the kth replicate of $\boldsymbol{X}$ generated via the sampling algorithm. In Gibbs sampling, it is not necessary to introduce a separate candidate process $\boldsymbol{W}$ (as in the general M-H approach) because—as we saw above—the candidate point is always accepted. Although each X_{ki} is usually a scalar element, there are problems where it is useful to have at least some X_{ki} being multivariate (see "Applications in State Estimation for Dynamic Models"). The idea of the proposal distribution for M-H can be used to update each component of $\boldsymbol{X}$. This updating is done on a component-by-component basis. Central to the updating is the conditional random variable $\{\boldsymbol{X}_{k+1,i}|\boldsymbol{X}_{k\backslash i}\}$ where

$$\boldsymbol{X}_{k\backslash i}\equiv\{X_{k+1,1},X_{k+1,2},\ldots,X_{k+1,i-1},X_{k,i+1},\ldots,X_{km}\},$$

$i=1,2,\ldots,m$. To avoid very cumbersome subscript notation associated with the relevant random variables and the associated conditioning, let $p_i(\cdot)$ represent the sampling density for the above conditional random variable:

$$\{X_{k+1,i}|\boldsymbol{X}_{k\backslash i}\}\sim p_i\left(x|\boldsymbol{X}_{k\backslash i}\right),\quad i=1,2,\ldots,m.$$

That is, $p_i(\cdot)$ represents the sampling density for the random variable $X_{k+1,i}$ conditioned on $\boldsymbol{X}_{k\backslash i}$, where the first $i-1$ elements of $\boldsymbol{X}_{k\backslash i}$ represent sample points at the same $(k+1)$st iteration, whereas the remaining $m-i$ elements are points available from the kth iteration. This strange-looking conditioning follows naturally from the sequential component-wise processing in the Gibbs sampling procedure, as given below. The conditioning represents the most recent information available when generating the ith component of $\boldsymbol{X}$.

The density $p_i(\cdot)$ is the generalization of the full conditional densities above from $m=3$ to an arbitrary m. Note

Panel 2. Gibbs sampling algorithm for estimating $E[f(\boldsymbol{X})]$.

Step 0. **(Initialization)** Choose the length of the "burn-in" period M and an arbitrary initial state $\boldsymbol{X}_0$. Set $k = 0$.

Step 1. Generate $\boldsymbol{X}_{k+1}$ according to the following m steps:
1. Generate $X_{k+1,1} \sim p_1(x|\boldsymbol{X}_{k\backslash 1})$.
2. Generate $X_{k+1,2} \sim p_2(x|\boldsymbol{X}_{k\backslash 2})$.
$\vdots$
m. Generate $X_{k+1,m} \sim p_m(x|\boldsymbol{X}_{k\backslash m})$.

Step 2. Repeat step 1 until $\boldsymbol{X}_M$ is available. Terminate "burn-in" process and proceed to step 3 with $\boldsymbol{X}_k = \boldsymbol{X}_M$.

Step 3. Carry out step 1.

Step 4. Repeat step 3 until it is possible to compute the ergodic average of $n - M$ evaluations in (2). (Of course, if desired, this average can be computed recursively without storing all of $f(\boldsymbol{X}_{M+1})$, $f(\boldsymbol{X}_{M+2})$,..., $f(\boldsymbol{X}_n)$.) This ergodic average is the estimate of $E[f(\boldsymbol{X})]$ under the target distribution $p(\cdot)$.

that the full conditional is a univariate sampling density in the common case where each X_{ki} is univariate. Thus, even when $\boldsymbol{X}$ is high dimensional, the sampling is univariate. This has significant potential advantages. The derivation of the full conditional follows from basic laws of probability:

$$p_i(x|\boldsymbol{X}_{k\backslash i}) = \frac{p_{\boldsymbol{X}}(x, \boldsymbol{X}_{k\backslash i})}{\int p_{\boldsymbol{X}}(x, \boldsymbol{X}_{k\backslash i})dx}, \tag{7}$$

where the denominator integral is over the domain for $X_{k+1,i} = x$ (i.e., over the domain of the random variable to be sampled).

In some practical applications, this definition can be used directly to obtain the full conditionals required for the steps of the Gibbs sampling procedure. In other cases, standard numerical methods for random number generation can be used. A good discussion of methods for obtaining samples from full conditionals is included in Gilks et al. [10, chap. 5]. There are, however, many applications for which sampling from full conditionals is not convenient. A way of coping with this problem is to combine elements of M-H and Gibbs sampling. In one popular form, sometimes called *Metropolis within Gibbs*, M-H is used within the Gibbs sampling steps described earlier to produce samples from the full conditionals. This is useful when it is difficult to produce such samples directly (see, e.g., [10, pp. 84-85] and [11, pp. 322-326]).

We are now in a position to present a "standard" implementation of the Gibbs sampling algorithm for estimating $E[f(\boldsymbol{X})]$ in (1). See Panel 2.

Sketch of Theoretical Foundation for Gibbs Sampling

As it is not immediately obvious why the above Markovian sampling procedure works, let us sketch the rationale. The focus here is Gibbs sampling, but given the close connection between Gibbs sampling and the M-H algorithm introduced earlier, the arguments also provide some flavor of the basis for M-H, although the detailed arguments are somewhat different (see, e.g., Robert and Casella [11, sect. 6.2.2] for some theory behind M-H convergence). This relatively informal discussion is a simplified version of the discussion in Gelfand and Smith [5] and Robert and Casella [11, sect. 7.1.3].

Let us consider the three-variable case, X, Y, and Z, and show that sampling from the full conditionals as above provides the information necessary to obtain samples from the density $p_{X,Y,Z}(x,y,z)$ (or from the marginals for any of the three variables). The ideas for three variables given below extend immediately to an arbitrary number of variables (m as above). From basic rules of conditional probability,

$$p_X(x) = \int p_{X|Y,Z}(x|y,z)p_{Y,Z}(y,z)dydz,$$
$$p_Y(y) = \int p_{Y|X,Z}(y|x,z)p_{X,Z}(x,z)dxdz,$$
$$p_Z(z) = \int p_{Z|X,Y}(z|x,y)p_{X,Y}(x,y)dxdy,$$

where the integrals are over the relevant domains in $\mathbb{R}^2$. Note the presence of a full conditional in each of the integrands above. The full conditionals form the basis for the Markov aspect of the sampling because the next random variate is generated based on only the most recent conditioning.

Suppose that we begin the sampler with the first of the three expressions above and make an initial guess at Y and Z, say Y_0 and Z_0. Using the full conditional $p_{X|Y,Z}(x|Y_0,Z_0)$, we can then generate a sample point X_1 by Monte Carlo. Proceeding downward through the expressions above, we next use the full conditionals $p_{Y|X,Z}(y|X_1,Z_0)$ and $p_{Z|X,Y}(z|X_1,Y_1)$ to generate sample points Y_1 and Z_1, respectively. At this point, we have completed one iteration of the Gibbs sampler, producing a sample X_1, Y_1, and Z_1. We now repeat the process, using Y_1 and Z_1 to initiate the conditioning and producing a sample X_2, Y_2, and Z_2. Continuing this process, it can be shown under modest conditions that X_k, Y_k, and Z_k jointly converge in distribution to $p_{X,Y,Z}(x,y,z)$ as $k \to \infty$. Likewise, X_k, Y_k, and Z_k individually converge in distribution to their respective marginals $p_X(x)$, $p_Y(y)$, and $p_Z(z)$ (see [11, Sect. 7.1.3]).

The formal basis for convergence follows from a so-called fixed-point integral equation that establishes a relationship between marginal and conditional distributions. For instance, if the marginal $p_X(x)$ in the three-variable problem above is of interest, one can show that the Gibbs sampling routine provides a sample having limiting density $p_X(x)$, where $\phi(\cdot) = p_X(\cdot)$ is the unique solution to the integral equation

$$\phi(x) = \int\left[\int p_{X|Y,Z}(x|y,z)p_{Y,Z|X}(y,z|\tau)dydz\right]\phi(\tau)d\tau$$
$$\equiv \int K(x|\tau)\phi(\tau)d\tau,$$

and where the integral inside the [ ] in the top expression, represented by $K(\cdot)$, is over the appropriate subspace of $\mathbb{R}^2$ [4], [5]. The term $K(\cdot)$ is often called the transition kernel. From the expression above, the Gibbs recursion can be written in Markov transition form as

$$\phi_{k+1}(x) = \int K(x|\tau)\phi_k(\tau)d\tau,$$

where $\phi_k(\cdot)$ denotes the true density for X_k. The fundamental result in Gibbs sampling is that $\phi_k(\cdot)$ converges to $p_X(\cdot)$ as $k \to \infty$. Similar ideas apply, with analogous transition kernels, for other target densities and dimensions $m \neq 3$.

Analogous situations apply when the random variables have a discrete distribution. Here the kernel-based form described above is replaced with a Markov transition matrix. Let $\boldsymbol{p}_k$ represent the vector of probabilities associated with the possible outcomes for X_k (so, e.g., if X is a random variable having ten possible outcomes, there are ten nonnegative elements in $\boldsymbol{p}_k$ for all k, with the elements summing to one). Let $\boldsymbol{P}$ represent the transition matrix governing the probability of going from X_k to X_{k+1} (so $\boldsymbol{P}$ has dimension $\dim(\boldsymbol{p}_k) \times \dim(\boldsymbol{p}_k)$). The elements of this transition matrix are directly available through the conditional probabilities of the variables in the problem. So, in the three-variable setting above, an individual element of $\boldsymbol{P}$ is available by applying the total probability theorem to first determine the probabilities of going from X_k to Y_{k+1}, then from Y_{k+1} to Z_{k+1}, and finally from Z_{k+1} to X_{k+1}. By standard Markov chain theory,

$$\boldsymbol{p}_{k+1}^T = \boldsymbol{p}_k^T\boldsymbol{P} = \boldsymbol{p}_0^T\boldsymbol{P}^{k+1}.$$

Then, if all elements of $\boldsymbol{P}$ are strictly positive, $\boldsymbol{p}_k$ converges to the limiting $\overline{\boldsymbol{p}}$ that is the solution to the balance equation

$$\overline{\boldsymbol{p}}^T = \overline{\boldsymbol{p}}^T\boldsymbol{P}. \tag{8}$$

In particular, the $\overline{\boldsymbol{p}}$ that satisfies this stationarity condition must be the marginal distribution for X. Thus, the Gibbs sampler converges to the marginal distribution, as desired.

Some Examples of Gibbs Sampling

We now present two examples of Gibbs sampling. The first is for a (multivariate) normal target distribution, which leads to conditional distributions that are also normal. The second is for a truncated exponential distribution. One point illustrated in the second example is that the target $p(\boldsymbol{x}) = p_X(\boldsymbol{x})$ does not automatically exist even when the full conditionals do exist. The Gibbs sampler only produces meaningful results (of course!) if the target distribution exists. General regularity conditions for the existence of the target distribution (which often corresponds to the joint density for the elements in $\boldsymbol{X}$) are beyond the scope of the treatment here, but may, for example, be found in Robert and Casella [11, Section 7.1.5]. More generally, one should be aware that the relative ease of implementing the Gibbs sampler in some problems does not obviate the need for careful mathematical analysis of the problem structure and the results.

Example 2: Gibbs Sampling for a Normal Distribution

Suppose that $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for some mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Note that, as discussed for Example 1, the Gibbs sampler may not be the most efficient method of generating samples from a multivariate normal distribution. This example serves to illustrate more general principles, where Gibbs sampling is used to generate samples from nonstandard distributions.

A standard result from multivariate normality is that the distribution of any selection of components within $\boldsymbol{X}$ conditioned on the remaining components is also normal (e.g., Mardia et al. [24, pp. 62-63]). Specifically, the distribution of the ith component conditioned on the remaining components provides the sampling distribution

$$\left\{X_{ki}|\boldsymbol{X}_{k\setminus i}\right\} \sim N\left(\mu_i + \boldsymbol{\Sigma}_{i\setminus i}^T\boldsymbol{\Sigma}_{\setminus i,\setminus i}^{-1}(\boldsymbol{X}_{k\setminus i} - \boldsymbol{\mu}_{\setminus i}), \sigma_i^2 - \boldsymbol{\Sigma}_{i\setminus i}^T\boldsymbol{\Sigma}_{\setminus i,\setminus i}^{-1}\boldsymbol{\Sigma}_{i,\setminus i}\right), \tag{9}$$

where μ_i and σ_i^2 are, respectively, the ith component of $\boldsymbol{\mu}$ and ith diagonal component of $\boldsymbol{\Sigma}$; $\boldsymbol{\mu}_{\setminus i}$ is the vector containing all components of $\boldsymbol{\mu}$ except μ_i; $\boldsymbol{\Sigma}_{i,\setminus i}$ is the column vector of the elements of $\boldsymbol{\Sigma}$ corresponding to the covariances between the ith component of $\boldsymbol{X}$ and all *other* components of $\boldsymbol{X}$; and $\boldsymbol{\Sigma}_{\setminus i,\setminus i}$ contains the elements of $\boldsymbol{\Sigma}$ with the row and column corresponding to the ith component of $\boldsymbol{X}$ removed. (Note that $\boldsymbol{\mu}_{\setminus i}$ and $\boldsymbol{\Sigma}_{i,\setminus i}$ are $m-1$ dimensional and $\boldsymbol{\Sigma}_{\setminus i,\setminus i}$ is $(m-1)\times(m-1)$ dimensional.) Hence, to generate sample values of $\boldsymbol{X}$ via the Gibbs sampler, the densities $p_i(x|\boldsymbol{X}_{k\setminus i})$, $i = 1,2,\ldots,m$, in steps 1 and 3 of the procedure outlined above are equal to the right-hand side of (9).

Example 3: Gibbs Sampling for Truncated Exponential Distributions

Following Casella and George [23], let X and Y have conditional exponential probability density functions on an interval $(0,B)$:

$$p_{X|Y}(x|y) = \frac{ye^{-yx}}{1-e^{-By}}, \qquad 0 < x < B,$$
$$p_{Y|X}(y|x) = \frac{xe^{-xy}}{1-e^{-Bx}}, \qquad 0 < y < B.$$

With the sampling densities $p_1(\cdot) = p_{X|Y}(\cdot)$ and $p_2(\cdot) = p_{Y|X}(\cdot)$, the Gibbs algorithm can be used to produce samples from the joint density $p_{X,Y}(x,y)$. Suppose for the application dis-

cussed here that the interest is in the marginal density $p_X(x)$ rather than the joint density.

As noted in Casella and George [23], the density $p_X(x)$ does not exist for $B = \infty$. That is, at $B = \infty$, the marginal "density" that results is improper in the sense that $\int_0^\infty \left[\int_0^\infty p_{X,Y}(x,y)dy\right]dx = \infty$ even though both conditional densities above exist and are proper. From Robert and Casella [11, sect. 7.1.5], the existence of the joint density in this problem (from which the marginal for X can be determined) requires $\int_0^B p_{Y|X}(y|x) / p_{X|Y}(x|y)dy < \infty$. This condition is violated for $B = \infty$.

Nontrivial calculations show that for the truncated $B < \infty$ case, the marginal density exists and satisfies

$$p_X(x) = c\frac{1 - e^{-Bx}}{x},$$

where c is the normalizing constant [23]. Using the property $\int_0^B p_X(x)dx = 1$, and letting $B = 4$, it is found that $c \approx 0.2985$. This known marginal density can be used for comparisons with the output of the Gibbs sampler. (In practice, of course, the marginal or joint densities are usually unknown.)

Figure 2 shows a histogram of output for a Gibbs sampler based on $n = 500$. The histogram is constructed from the terminal output of the chain using 2,000 independent replications. The histogram closely matches the marginal density, indicating that the chain output has a distribution close to the desired distribution. Note that the overall run length (n) for this problem is shorter than needed in many other problems.

Applications in Bayesian Analysis

The above discussion of MCMC has been for general problems where the desire is to obtain quantities related to a target distribution for a random vector $\boldsymbol{X}$. More specifically,

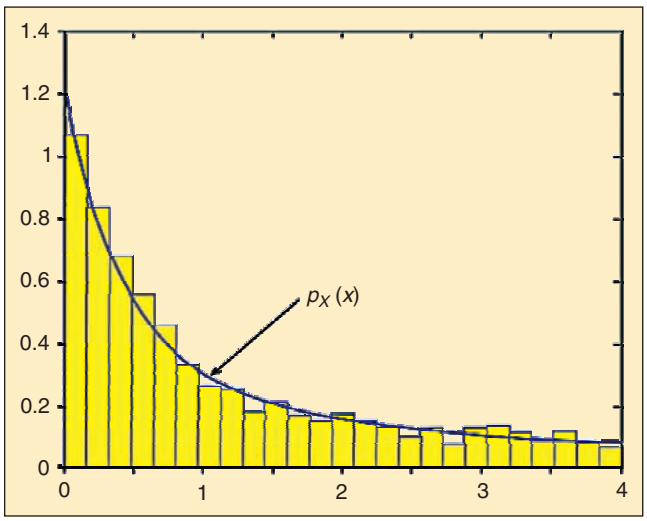

Figure 2. *Histogram for terminal X from Gibbs sampling process with truncated exponential sampling. Histogram constructed from 2000 independent replications. The desired marginal density is shown in the solid line.*

MCMC—particularly Gibbs sampling—has had an especially profound effect in *Bayesian* methods of analysis. There seem to be at least two reasons for this: 1) the structure of Bayes' rule is well matched to the requirements of MCMC for drawing samples from appropriate conditional densities, and 2) MCMC fills a long-standing need for a general-purpose method for constructing quantities related to posterior distributions that does not require cumbersome numerical integration. The seminal paper related to Bayesian applications of MCMC is Gelfand and Smith [5].

Let us review the Bayesian framework. Suppose that $\boldsymbol{\Pi}$ represents a vector of terms important to the analysis of some system and that data $\boldsymbol{Z}$ can be collected on that system. For example, $\boldsymbol{\Pi}$ might be the parameters $\boldsymbol{\Theta}$ of a model that are to be estimated. Another example is in the state-space modeling context, where $\boldsymbol{\Pi}$ represents both model parameters and the state vector (see the next section for a discussion and references). In the Bayesian approach, $\boldsymbol{\Pi}$ is treated as a random vector instead of a fixed constant. Let us suppose that $\boldsymbol{\Pi}$ has a prior probability density, say $p_{\boldsymbol{\Pi}}(\boldsymbol{\pi})$, reflecting a priori information available about $\boldsymbol{\Pi}$. (The general Bayesian approach can work with discrete or hybrid $\boldsymbol{\Pi}$ as well.) There are many issues associated with the prior, philosophical and mathematical, but we will not delve into those here. Further, there are Bayesian MCMC methods for determining the number of terms in a model (e.g., the number of lagged terms in an autoregressive process); this represents the model selection problem (e.g., Green [25]).

From Bayes' rule one can take the prior information on $\boldsymbol{\Pi}$, expressed via the prior density function, combine it with the conditional density function for the data $p_{\boldsymbol{Z}|\boldsymbol{\Pi}}(\boldsymbol{z}|\boldsymbol{\pi})$ (sometimes called the likelihood function), and form the *posterior* density function $p_{\boldsymbol{\Pi}|\boldsymbol{Z}}(\boldsymbol{\pi}|\boldsymbol{z})$ according to

$$p_{\boldsymbol{\Pi}|\boldsymbol{Z}}(\boldsymbol{\pi}|\boldsymbol{z}) = \frac{p_{\boldsymbol{Z}|\boldsymbol{\Pi}}(\boldsymbol{z}|\boldsymbol{\pi})p_{\boldsymbol{\Pi}}(\boldsymbol{\pi})}{\int p_{\boldsymbol{Z}|\boldsymbol{\Pi}}(\boldsymbol{z}|\boldsymbol{\pi})p_{\boldsymbol{\Pi}}(\boldsymbol{\pi})d\boldsymbol{\pi}} = \frac{p_{\boldsymbol{\Pi},\boldsymbol{Z}}(\boldsymbol{\pi},\boldsymbol{z})}{p_{\boldsymbol{Z}}(\boldsymbol{z})}, \tag{10}$$

where the integral is over the domain for $\boldsymbol{\Pi}$. This simple-looking formula has profound implications and applications relative to modern system identification, estimation, and statistical analysis. The posterior density provides a fundamental means of characterizing the system parameters (or other quantities) $\boldsymbol{\Pi}$ by combining data ($\boldsymbol{Z}$) with prior information.

In almost all practical applications, numerical methods will be needed to calculate the integrals required for forming and using the posterior density. Rarely will this integration be feasible in closed form. One area requiring integration is the computation of the conditional expectation, $E[f(\boldsymbol{\Pi})|\boldsymbol{Z}] = \int f(\boldsymbol{\pi})p_{\boldsymbol{\Pi}|\boldsymbol{Z}}(\boldsymbol{\pi}|\boldsymbol{Z})d\boldsymbol{\pi}$, particularly the special case of the conditional mean $E[\boldsymbol{\Pi}|\boldsymbol{Z}]$. Further, in using the posterior density function in (10) to construct posterior *probabilities* associated with $\boldsymbol{\Pi}$, multifold integration of the left side of (10) is required. In particular, we might be interested

in probabilities of the form $P(\boldsymbol{\Pi} \in S|\boldsymbol{Z}) = \int_S p_{\boldsymbol{\Pi}|\boldsymbol{Z}}(\boldsymbol{\pi}|\boldsymbol{Z})d\boldsymbol{\pi}$, where S is some subset of the domain for $\boldsymbol{\Pi}$. (This type of integration arises, for instance, in computing credible regions for $\boldsymbol{\Pi}$, the Bayesian analog of confidence regions.) A final aspect involving multivariate integration is the computation of the marginal density for $\boldsymbol{Z}$ (i.e., the denominator of (10)). This marginal is needed in some applications (e.g., [26]), but (happily!) not in the Gibbs sampling for producing samples from the posterior, as we discuss below.

As mentioned earlier, MCMC, particularly Gibbs sampling, is especially well suited to Bayesian analysis. Recall that the Gibbs sampling procedure can be implemented if one can construct full conditional densities for each element of the vector of interest and samples can be generated from the full conditionals. Following the discussion of the section "Gibbs Sampling," $\{\Pi_{k+1,i}|\boldsymbol{\Pi}_{k\setminus i},\boldsymbol{Z}\} \sim p_i(\pi|\boldsymbol{\Pi}_{k\setminus i},\boldsymbol{Z})$, $i = 1,2,\ldots,m$, where we are using notation analogous to the previous generic notation. In particular, $\boldsymbol{\Pi}_k \equiv [\Pi_{k1},\Pi_{k2},\ldots,\Pi_{km}]^T$ and the set of $m-1$ components without the ith component is $\boldsymbol{\Pi}_{k\setminus i} \equiv \{\Pi_{k+1,1}, \Pi_{k+1,2},\ldots,\Pi_{k+1,i-1}, \Pi_{k,i+1},\ldots,\Pi_{km}\}$. In the special case where $\boldsymbol{\Pi} = \boldsymbol{\Theta}$, then $m = \dim(\boldsymbol{\Theta})$ and the ith component of $\boldsymbol{\Pi}$ corresponds to the ith element of $\boldsymbol{\Theta}$. As mentioned earlier, the individual components Π_{ki} may be scalar or multivariate. Note the extra conditioning ($\boldsymbol{Z}$) due to the data, reflecting the posterior aspect of the Bayesian formulation. This conditioning, although critical to the Bayesian analysis, may be treated as a constant relative to the Gibbs sampling process. We emphasize this treatment as a constant by writing $\boldsymbol{Z} = \boldsymbol{z}$ in the conditioning arguments below.

Expression (7) provides the fundamental form for the full conditionals. When substituting the right-hand side of (10) into (7), we get the full conditionals for the random variables

$$p_i(\pi|\boldsymbol{\Pi}_{k\setminus i},\boldsymbol{Z} = \boldsymbol{z}) = \frac{p_{\boldsymbol{\Pi}|\boldsymbol{Z}}(\pi,\boldsymbol{\Pi}_{k\setminus i}|\boldsymbol{z})}{\int p_{\boldsymbol{\Pi}|\boldsymbol{Z}}(\pi,\boldsymbol{\Pi}_{k\setminus i}|\boldsymbol{z})d\pi} = \frac{p_{\boldsymbol{\Pi},\boldsymbol{Z}}(\pi,\boldsymbol{\Pi}_{k\setminus i},\boldsymbol{z})}{\int p_{\boldsymbol{\Pi},\boldsymbol{Z}}(\pi,\boldsymbol{\Pi}_{k\setminus i},\boldsymbol{z})d\pi}, \quad (11)$$

where the second equality follows by the cancellation of the marginal density $p_{\boldsymbol{Z}}(\boldsymbol{z})$ in the numerator and denominator. Hence, to create samples from the posterior density function via the Gibbs sampler, it is not necessary to compute the denominator integral in Bayes' rule (10).

Applications in State Estimation for Dynamic Models

A popular area for the application of MCMC ideas is in state estimation in nonstandard situations where the Kalman filter may perform poorly. Although Monte Carlo methods were proposed for state estimation several decades ago (e.g., [27]), the advent of MCMC combined with modern computational power has been invaluable in making Monte Carlo approaches practical for realistic systems. Several MCMC methods have been introduced to address the shortcomings of the Kalman filter in the presence of nonlinearity in the state-space model and/or nonnormality for the noise distributions. Some of these methods also simultaneously address the system identification problem of estimating model parameters. Carlin et al. [28], Gordon et al. [29], Carter and Kohn [30], Doucet et al. [14], and Geweke and Tanizaki [31] are several of the numerous references in this area. In this brief treatment, we restrict ourselves to state estimation with linear models having known model parameters but *non-Gaussian* distributed noise terms.

For purposes here, consider the linear state-space model

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k\boldsymbol{x}_k + \boldsymbol{w}_k$$
$$\boldsymbol{z}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k,$$

where $\boldsymbol{x}_k$ is the state vector, $\boldsymbol{z}_k$ is the measurement, $\boldsymbol{\Phi}_k$ and $\boldsymbol{H}_k$ are the transition and measurement matrices, and $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are noise terms that are assumed mutually independent over all k. Suppose that the initial state and noise terms may have nonnormal probability distributions. Because of the nonnormality, the Kalman filter for state estimation is not the optimal estimator (although it remains the optimal linear estimator). (Some discussion on the use of the Kalman filter in non-Gaussian situations is given in Spall [32] and Maryak et al. [33]. These references recognize that distributions are often unknown in non-Gaussian situations. As with any Monte Carlo technique, MCMC requires enough information about the distributions to make possible the computer generation of the random terms.) Our aim is to develop the optimal estimator in a given non-Gaussian situation. Because we assume that the state-space model parameters are known, our focus is solely state estimation, not parameter identification.

The aim is to develop an online procedure for state estimation that accounts for the nonnormality of the noise terms. Below we outline a Gibbs sampling approach that achieves this goal. The Gibbs sampler can be used to produce samples from the posterior distributions for $\{\boldsymbol{x}_n|\boldsymbol{Z}_n\}$, $\{\boldsymbol{x}_n|\boldsymbol{Z}_k\}$, and $\{\boldsymbol{x}_n|\boldsymbol{Z}_N\}$, where $k < n < N$ and $\boldsymbol{Z}_n = [\boldsymbol{z}_1^T,\boldsymbol{z}_2^T,\ldots,\boldsymbol{z}_n^T]^T$ is the stacked vector of measurements. These three posterior distributions correspond to the filtering, prediction, and smoothing distributions. Note that $\boldsymbol{x}_n$ corresponds to Π in the general Bayesian notation above. Computing the mean of these conditional distributions produces the corresponding filter, prediction, or smoother estimate.

The aim is to calculate the conditional mean $E(\boldsymbol{x}_k|\boldsymbol{Z}_n)$, $k = 1,2,\ldots,n$. Let $\boldsymbol{X}_n = [\boldsymbol{x}_0^T,\boldsymbol{x}_1^T,\ldots,\boldsymbol{x}_n^T]^T$. Assuming all relevant random terms have density functions, the joint density for all states and measurements is

$$p(\boldsymbol{X}_n,\boldsymbol{Z}_n) = p(\boldsymbol{Z}_n|\boldsymbol{X}_n)p(\boldsymbol{X}_n),$$

where

$$p(\boldsymbol{X}_n) = p_{\boldsymbol{x};0}(\boldsymbol{x}_0|E(\boldsymbol{x}_0))\prod_{k=1}^{n} p_{\boldsymbol{x};k}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}),$$

$$p(\boldsymbol{Z}_n|\boldsymbol{X}_n) = \prod_{k=1}^{n} p_{\boldsymbol{z};k}(\boldsymbol{z}_k|\boldsymbol{x}_k)$$

for some densities $p_{\boldsymbol{x};k}(\cdot)$ and $p_{\boldsymbol{z};k}(\cdot)$. We now introduce two nuisance parameters, $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, whose purpose will be to serve as dummy variables in a process of forming Gaussian mixture densities that approximate the non-Gaussian densities $p_{\boldsymbol{x};k}(\cdot)$ and $p_{\boldsymbol{z};k}(\cdot)$. It is well known that continuous or discrete mixtures of Gaussian distributions can be used to approximate a wide variety of non-Gaussian distributions (as used, e.g., in Spall and Hill [34] to approximate otherwise intractable "noninformative" prior distributions; a filtering application of Gaussian mixtures is described in Alspach and Sorenson [35]). Let us model $p_{\boldsymbol{x};k}(\cdot)$ and $p_{\boldsymbol{z};k}(\cdot)$ by the continuous mixtures

$$p_{\boldsymbol{x};k}\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right) = \int N(\boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1},\lambda_k\boldsymbol{\Sigma}_x)p_\Lambda(\lambda_k)d\lambda_k \text{ and} \tag{12a}$$

$$p_{\boldsymbol{z};k}\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right) = \int N(\boldsymbol{H}_k\boldsymbol{x}_k,\omega_k\boldsymbol{\Sigma}_z)p_\Omega(\omega_k)d\omega_k, \tag{12b}$$

where the $N(\cdot,\cdot)$ terms appear in the integrands representing normal densities with the indicated mean vector and covariance matrix, $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_z$ are two covariance matrices for use in forming the mixture distributions for the state and measurement, and $p_\Lambda(\cdot)$ and $p_\Omega(\cdot)$ represent user-specified densities governing the mixtures. By varying $p_\Lambda(\cdot)$ and $p_\Omega(\cdot)$ one can produce many non-Gaussian densities, including logistic, double exponential, student *t*, and hyberbolic densities (see Carlin et al. [28] for a brief discussion and references).

Given the above structure, we are now in a position to implement the Gibbs sampler for producing the Monte Carlo samples from the distributions for $\{\boldsymbol{x}_k|\boldsymbol{Z}_n\}$. The mean of the samples at each k represents an estimate of $E(\boldsymbol{x}_k|\boldsymbol{Z}_n)$. In the notation of the section "Gibbs Sampling," $\boldsymbol{X}$ represents the collection of states and nuisance parameters: $\boldsymbol{X}_n,\boldsymbol{\lambda} \equiv \{\lambda_1,\lambda_2,\ldots,\lambda_n\}$, and $\boldsymbol{\omega} \equiv \{\omega_1,\omega_2,\ldots,\omega_n\}$. Recall that the fundamental quantities needed are the full conditional distributions. In the state estimation context here, therefore, we need the distributions for the following:

$$\left\{\boldsymbol{x}_k\middle|\boldsymbol{x}_{j\neq k},\boldsymbol{\lambda},\boldsymbol{\omega},\boldsymbol{Z}_n\right\}, \tag{13a}$$

$$\left\{\lambda_k\middle|\lambda_{j\neq k},\boldsymbol{\omega},\boldsymbol{X}_n,\boldsymbol{Z}_n\right\}, \tag{13b}$$

$$\left\{\omega_k\middle|\omega_{j\neq k},\boldsymbol{\lambda},\boldsymbol{X}_n,\boldsymbol{Z}_n\right\}, \tag{13c}$$

$k = 1,2,\ldots,n$. For a given k, one iteration of the Gibbs sampler is one cycle through the full conditionals corresponding to the random variables in (13a), (13b), and (13c). The required full conditionals are determined in Carlin et al. [28]. Let us summarize them here.

Based on the mixture assumptions in (12a) and (12b), $\{\boldsymbol{x}_k|\boldsymbol{x}_{j\neq k},\boldsymbol{\lambda},\boldsymbol{\omega},\boldsymbol{Z}_n\}$ has a normal distribution with mean vector and covariance matrix dependent on the parameters of the state-space model, the nuisance parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, the states $\boldsymbol{x}_{k-1}$ and $\boldsymbol{x}_{k+1}$, and the current measurement $\boldsymbol{z}_k$. The second and third conditional expressions above ((13b) and (13c)) have distributions that are a direct consequence of Bayes' rule. For example, with the expression in (13c), note that not all of the indicated full conditioning is relevant in the sense that $\{\omega_k|\omega_{j\neq k},\boldsymbol{\lambda},\boldsymbol{X}_n,\boldsymbol{Z}_n\}$ and $\{\omega_k|\boldsymbol{x}_k,\boldsymbol{z}_k\}$ have the same distribution. By Bayes' rule, the density for $\{\omega_k|\boldsymbol{x}_k,\boldsymbol{z}_k\}$ is proportional to the product of the densities for $\{\boldsymbol{z}_k|\boldsymbol{x}_k,\omega_k\}$ and the prior $p_\Omega(\omega_k)$, as appear in (12b). Analogous arguments may be used in generating the sampling distribution for λ_k and ω_k, corresponding to (13b) and (13c). Carlin et al. [28] present examples where the non-Gaussian noise terms in the state-space model are assumed to have a double exponential distribution. The resulting distributions for $\{\lambda_k|\boldsymbol{x}_k,\boldsymbol{z}_k\}$ and $\{\omega_k|\boldsymbol{x}_k,\boldsymbol{z}_k\}$ (corresponding to (13b) and (13c)) are generalized inverse Gaussian distributions, which can easily be used to generate random samples. Hence, by using the Gibbs sampler, one can generate optimal (conditional mean) state estimates at each k for state-space models with particular non-Gaussian noise distributions.

Conclusions

This discussion summarized the motivation, theory, implementation, and connection to Bayesian analysis of MCMC, focusing on the M-H and Gibbs sampling versions. Many large-scale practical implementations of MCMC borrow aspects from both M-H and Gibbs sampling.

Although we saw that Gibbs may, in a certain sense, be considered a special case of M-H, the techniques have developed largely independent of each other. Recognizing this, we discussed M-H and Gibbs as separate approaches. No one approach is universally preferred. One strong aspect of both M-H and Gibbs is the theory supporting the methods and guaranteeing convergence under modest conditions.

Because the structure of Gibbs sampling (with its requirement that the full conditional distributions be available) is more restrictive than M-H, Gibbs has the advantage of not needing the "tuning" that is required in M-H. A serious application of M-H will usually require some experimentation with the proposal distribution, not only in the specific parameters of the distribution, but perhaps in the general form of the distribution (e.g., gamma or uniform?). The restrictions in Gibbs sampling are analogous to certain search methods (e.g., Newton-Raphson algorithm), where the tuning is eliminated because the structure of the algorithm is sufficiently proscribed.

One of the areas of application for MCMC techniques, especially Gibbs sampling, is state and parameter estimation in dynamic models having forms different from the commonly assumed linear Gaussian models. Although the MCMC implementation can be computationally intensive in these (and other) applications, the approach allows for the treatment of models that were effectively impossible before.

Of course, as with any Monte Carlo technique, one has to make the necessary assumptions about distributions to allow the computer to generate the "proper" samples. That is, MCMC does not magically obviate the need for understanding and careful analysis of the problem at hand. Further, MCMC is generally computationally demanding in large-scale applications. Among MCMC techniques, it is not possible to say whether M-H or Gibbs is computationally more efficient in general. Much depends on the specifics of the implementation. Examples are available in the literature illustrating both efficient and inefficient results for either or both approaches.

The ultimate value of M-H or Gibbs sampling, of course, is their worth in solving practical problems. Both approaches have proved to be important tools in the modern analyst's toolbox. Although MCMC has revolutionized applied statistics in the last decade or so, the impact in control and related fields has thus far been much more modest. This is expected to change in the coming years as researchers and practitioners in control systems recognize the value of MCMC in many "nonstandard" problems of estimation and identification.

Acknowledgments

This work was partially funded through the U.S. Navy (contract N00024-98-D-8124), the JHU/APL Janney Fellowship and IRAD programs, and the State of Maryland MAITI program. Selected parts of this article (primarily the step-by-step algorithm descriptions and supporting discussion) have been reprinted, by permission, from J.C. Spall, *Introduction to Stochastic Search and Optimization* (Chapter 16), copyright 2003 by John Wiley and Sons, Inc. The author appreciates the helpful comments of Dr. John Maryak and two anonymous reviewers.

References

[1] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, 1953.

[2] W.K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97-109, 1970.

[3] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721-741, 1984.

[4] M. Tanner and W. Wong, "The calculation of posterior distributions by data augmentation (with discussion)," *J. Amer. Statist. Assoc.*, vol. 82, pp. 528-550, 1987.

[5] A.E. Gelfand and A.F.M. Smith, "Sampling-based approaches to calculating marginal densities," *J. Amer. Statist. Assoc.*, vol. 85, pp. 399-409, 1990.

[6] J. Besag, P. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Statist. Sci.*, vol. 10, pp. 3-66, 1995.

[7] O. Cappé and C.P. Robert, "Markov chain Monte Carlo: 10 years and still running!,"*J. Amer. Statist. Assoc.*, vol. 95, pp. 1282-1286, 2000.

[8] A.E. Gelfand, "Gibbs sampling," *J. Amer. Statist. Assoc.*, vol. 95, pp. 1300-1304, 2000.

[9] M. Evans and T. Swartz, "Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems," *Statist. Sci.*, vol. 10, pp. 254-272, 1995.

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

[11] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[12] M.-H. Chen, Q.-M. Shao, and J.G. Ibrahim, *Monte Carlo Methods in Bayesian Computation*. New York: Springer-Verlag, 2000.

[13] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2001.

[14] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation in jump Markov linear systems," *IEEE Trans. Automat. Contr.*, vol. 45, no. 2, pp. 188-202, 2000.

[15] R.Y. Rubinstein and B. Melamed, *Modern Simulation and Modeling*. New York: Wiley, 1998.

[16] E. Parzen, *Stochastic Processes*. New York: Holden-Day, 1962.

[17] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

[18] C. Andrieu, L. Breyer, and A. Doucet, "Convergence of simulated annealing using Foster-Lyapunov criteria," *J. Appl. Prob.*, vol. 38, no. 4, pp. 975-994, 2001.

[19] J.C. Spall, *Introduction to Stochastic Search and Optimization*. Hoboken, NJ: Wiley, 2003.

[20] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *Amer. Statist.*, vol. 49, pp. 327-335, 1995.

[21] W.R. Gilks, G.O. Roberts, and S.K. Sahu, "Adaptive Markov chain Monte Carlo through regeneration," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1045-1054, 1998.

[22] G.O. Roberts and J.S. Rosenthal, "Optimal scaling for various Metropolis-Hastings algorithms," *Statist. Sci.*, vol. 16, no. 4, pp. 351-367, 2001.

[23] G. Casella and E.I. George, "Explaining the Gibbs sampler," *Amer. Statist.*, vol. 46, pp. 167-174, 1992.

[24] K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis*. New York: Academic, 1979.

[25] P.J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711-732, 1995.

[26] S. Chib, "Marginal likelihood from the Gibbs sampler," *J. Amer. Statist. Assoc.*, vol. 90, pp. 1313-1321, 1995.

[27] J.E. Handschin and D.Q. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multistage, nonlinear filtering," *Int. J. Contr.*, vol. 9, pp. 547-559, 1969.

[28] B.P. Carlin, N.G. Polson, and D.S. Stoffer, "A Monte Carlo approach to nonnormal and nonlinear state-space modeling," *J. Amer. Statist. Assoc.*, vol. 87, pp. 493-500, 1992.

[29] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elec. Eng.*, vol. 140, pt. F, 1993, pp. 107-113.

[30] C.K. Carter and R. Kohn, "On Gibbs sampling for state-space models," *Biometrika*, vol. 81, pp. 541-553, 1994.

[31] J. Geweke and H. Tanizaki, "Bayesian estimation of state-space models using the Metropolis-Hastings algorithm within Gibbs sampling," *Comput. Statisti. Data Anal.*, vol. 37, pp. 151-170, 2001.

[32] J.C. Spall, "The Kantorovich inequality for error analysis of the Kalman filter with unknown noise distributions," *Automatica*, vol. 31, pp. 1513-1517, 1995.

[33] J.L. Maryak, J.C. Spall, and B.D. Heydon, "Use of the Kalman filter for inference in state-space models with unknown noise distributions," in *Proc. American Control Conf.*, Albuquerque, NM, 1997, pp. 2127-2132.

[34] J.C. Spall and S.D. Hill, "Least-informative Bayesian prior distributions for finite samples based on information theory," *IEEE Trans. Automatic Contr.*, vol. 35, pp. 580-583, 1990.

[35] D.L. Alspach and H.W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automat. Contr.*, vol. AC-17, pp. 439-448, 1972.

James C. Spall is a member of the principal professional staff at the Johns Hopkins University Applied Physics Laboratory and is the chair of the Applied and Computational Mathematics Program within the Johns Hopkins School of Engineering. He has published extensively in the areas of control and statistics and holds two U.S. patents. Among other appointments, he is an associate editor at large for the *IEEE Transactions on Automatic Control* and a contributing editor for the *Current Index to Statistics*. He has received numerous research and publications awards and is a Fellow of the IEEE.