MONTE CARLO METHODS FOR SIGNAL PROCESSING: RECENT ADVANCES

Petar M. Djurić

Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794, USA
phone: +1 631-632-8423, fax: +1 631-632-8494, email: djuric@ece.sunysb.edu
web: www.ece.sunysb.edu/djuric.html

ABSTRACT

In many areas of signal processing, the trend of addressing problems with increased complexity continues. This is best reflected by the forms of the models used for describing phenomena of interest. Typically, in these models the number of unknowns that have to be estimated is large and the assumptions about noise distributions are often non-tractable for analytical derivations. One major reason that allows researchers to resolve such difficult problems and delve into uncharted territories is the advancement of methods based on Monte Carlo simulations including Markov chain Monte Carlo sampling and particle filtering. In this paper, the objective is to provide a brief review of the basics of these methods and then elaborate on the most recent advances in the field.

1. INTRODUCTION

A large number of statistical signal processing applications including filtering, estimation, and detection require evaluation of integrals, optimization and simulation of stochastic systems. These methods have not only played a prominent role in the field of signal processing but also in physics, econometrics, statistics, and computer science.

A critical step in many signal processing algorithms that require stochastic simulations is the generation of samples from a multivariate probability distribution $p(\mathbf{x}|\mathbf{y})$, where $\mathbf{x}$ is a vector of unknown states or parameters, and $\mathbf{y}$ represents a vector of observations. Take for example the need to compute the expectation of a function $g(X)$, where X is a continuous random variable with distribution $p(x)$ (here we simplify matters by assuming that X is one dimensional). Thus, we need to find

$$\mu_g = \int g(x)p(x)dx. \tag{1}$$

In many signal processing problems $g(x) = x$, $p(x)$ represents the a posterior density of x, that is $p(x|\mathbf{y})$, and the integral is the minimum mean-square error (MMSE) estimate of x. Suppose now that the integration in (1) is analytically intractable. Then we could resort to numerical integration applying the Monte Carlo method [67]. The idea is to draw samples from $p(x)$, $x^{(m)}$, and evaluate the integral by

$$\hat{\mu}_g = \frac{1}{M}\sum_{m=1}^{M} g(x^{(m)}) \tag{2}$$

where M is the total number of generated samples. Note that the approximation (2) of the integral in (1) implies that the density $p(x)$ is approximated according to

$$p(x) = \frac{1}{M}\sum_{m=1}^{M} \delta(x - x^{(m)}) \tag{3}$$

where $\delta(\cdot)$ is the Dirac delta function. It can be shown that the estimate $\hat{\mu}_g$ converges to μ_g almost surely as $M \to \infty$ [68]. In addition, if the variance of $g(X)$ is finite and equal to σ_g^2, then the variance of $\hat{\mu}_g$ is σ_g^2/M. By invoking central limit theorem, one can also show that the distribution of $\hat{\mu}_g$ is Gaussian and centered at μ_g.

This simple example shows that drawing samples from a distribution can be quite advantageous. In fact, if we draw samples from a distribution $p(\mathbf{x})$, we can use them for obtaining various types of estimates, not just the MMSE estimate, and we can carry out other important signal processing operations such as model selection and detection. However, the operation of drawing samples from $p(\mathbf{x})$ is in general not easy. The generation is especially difficult if the dimension of $\mathbf{x}$ is large and $p(\mathbf{x})$ is not a standard distribution. In such cases, we can use methods known as rejection, MCMC, and importance sampling. In this paper we focus on the latter two, first by providing their basic descriptions, and then by elaborating on some of their more recent advances. It should be noted that these methods are considered primarily as tools of the Bayesian methodology. However, they are also applied in other optimization techniques. For example, MCMC sampling is the basis of the optimization technique known as simulated annealing [48].

The bibliography on this subject have grown significantly. Therefore, it is impossible to cite all the relevant literature, and as a result many good papers have been omitted. We primarily focused on journal articles that have been published after 2002 and that have signal processing contents.

2. BASICS

2.1 Markov chain Monte Carlo sampling

MCMC sampling is considered to be the most successful and influential Monte Carlo method in computational science [10]. We usually apply MCMC sampling when we cannot generate samples directly from a multivariate distribution $p(\mathbf{x}|\mathbf{y})$ but can evaluate $p(\mathbf{x}|\mathbf{y})$ up to a proportionality constant. The key idea of MCMC methods is to generate samples by running an ergodic chain whose equilibrium distribution is the desired distribution [17, 33, 55, 68]. An aperiodic and irreducible Markov chain has the property of converging to a unique stationary distribution which does not depend on the initial sample or the iteration number j. If $P(\mathbf{x}^{(j)}|\mathbf{x}^{(j-1)})$

This work was supported by the National Science Foundation under Award CCR-0082607.

is the transition kernel of the chain and $p(\mathbf{x})$ is the desired stationary distribution, and if the chain satisfies the *detailed balance* equation

$$p(\mathbf{x}^{(j-1)})P(\mathbf{x}^{(j)}|\mathbf{x}^{(j-1)}) = p(\mathbf{x}^{(j)})P(\mathbf{x}^{(j-1)}|\mathbf{x}^{(j)}) \quad (4)$$

it will produce samples, which after convergence, are from the stationary distribution $p(\mathbf{x})$ [33].

It turns out that the generation of samples from the required distribution is remarkably easy. If a candidate sample $\mathbf{x}^*$ is proposed by the density $\pi(\cdot|\mathbf{x}^{(j-1)})$ where $\mathbf{x}^{(j-1)}$ is the last generated sample, then the probability of accepting it is

$$\alpha(\mathbf{x}^{(j-1)}, \mathbf{x}^*) = \min\left(1, \alpha_0\right) \quad (5)$$

where

$$\alpha_0 = \frac{p(\mathbf{x}^*)\pi(\mathbf{x}^{(j-1)}|\mathbf{x}^*)}{p(\mathbf{x}^{(j-1)})\pi(\mathbf{x}^*|\mathbf{x}^{(j-1)})}. \quad (6)$$

The new sample is

$$\mathbf{x}^{(j)} = \phi(\mathbf{x}^{(j-1)}, U)$$

where $\phi(\cdot,\cdot)$ is called the updating function that satisfies

$$\text{Prob}\left(\phi(\mathbf{x}^{(j-1)}, U) = \mathbf{x}^{(j)}\right) = P(\mathbf{x}^{(j)}|\mathbf{x}^{(j-1)}) \quad (7)$$

and U is a random variable defined on an arbitrary probability space. Typically U is drawn from the uniform density function defined on the interval (0,1) and

$$\phi(\mathbf{x}^{(j-1)}, U) = \mathbf{x}^{(j)} = \begin{cases} \mathbf{x}^*, & \text{if } U \leq \alpha(\mathbf{x}^{(j-1)}, \mathbf{x}^*) \\ \mathbf{x}^{(j-1)}, & \text{otherwise} \end{cases} \quad (8)$$

where $\alpha(\cdot)$ is given by (5).

Three standard MCMC schemes are the Metropolis [59], the Metropolis-Hastings [39] and the Gibbs sampling methods [30]. If the proposal density is symmetric, then the method is known as the Metroplis sampler, and if it is non-symmetric, as the Metropolis-Hastings algorithm. Finally, the Gibbs sampler is a special case of the Metropolis-Hastings algorithm that generates samples from conditional densities which make the probability of acceptance equal to one.

There are several important issues related to the use of these three algorithms. The algorithms are iterative in nature and therefore their convergence must be addressed. All the schemes usually discard the first M samples, a period called burn-in, and to determine M, convergence diagnostics are applied [15, 29, 68]. Of practical importance, too, is the stopping time of the chain. One would like to run the chain long enough to obtain the desired accuracy, so its characterization is also necessary. Finally, another concern is the correlation of the generated samples. Since it is often important to get independent and identically distributed samples, either several chains are run simultaneously or only every k-th sample from the chain is used, where the value of k is carefully determined.

A nontrivial extension of MCMC sampling is the reversible jump MCMC (RJMCMC) sampler [36]. The RJMCMC sampler can jump between parameter spaces with different dimensions. Typically the parameters with different dimensions correspond to different models. Thus, the generalization of the MCMC scheme amounts to allowing the chain not only to explore the parameter space of one model but spaces of many more models. This implies that RJMCMC can be applied to problems where, besides estimation of unknown parameters of a model, the selection of model is also of interest [6]. The latter problem includes tasks like detection of signals, multiple changepoint problems, object recognition, and variable selection.

Let the visited model and its parameters at the $(j-1)-$th iteration of the chain be $\mathrm{M}_l^{(j-1)}$ and $\mathbf{x}_l^{(j-1)}$, respectively. Suppose that at the $j-$th iteration there is a proposal to move to the model $\mathrm{M}_k^{(j)}$. One way to construct an RJMCMC is to allow a transition which is accepted with probability

$$\alpha(\mathrm{M}_l^{(j-1)}, \mathrm{M}_k^{(j)}) = \min\left(1, \alpha_0\right). \quad (9)$$

where

$$\begin{aligned} \alpha_0 &= \frac{p(\mathbf{y}|\mathbf{x}^{(j)}, \mathrm{M}_k^{(j)})p(\mathbf{x}^{(j)}, \mathrm{M}_k^{(j)})}{p(\mathbf{y}|\mathbf{x}^{(j-1)}, \mathrm{M}_l^{(j-1)})p(\mathbf{x}^{(j-1)}, \mathrm{M}_l^{(j-1)})} \\ &\times \frac{\pi(\mathbf{x}^{(j-1)}, \mathrm{M}_l^{(j-1)}|\mathbf{x}^{(j)}, \mathrm{M}_k^{(j)})}{\pi(\mathbf{x}^{(j)}, \mathrm{M}_k^{(j)}|\mathbf{x}^{(j-1)}, \mathrm{M}_l^{(j-1)})} \end{aligned} \quad (10)$$

and $\pi(\cdot|\cdot)$ represents proposal distribution. Thus, the nature of the method is preserved except that it is much more complex. One has to explore not just one parameter space but often many more.

The importance of the RJMCMC sampling is in that it allows inference in situations of uncertainties about both, the model parameters and the used models. From the frequency of the visits of the sampler to the various models, we can estimate the posterior probabilities of these models. Thus, the method can be used for model selection, model evaluation, and model averaging.

2.2 Particle filtering

Particle filtering is a sequential Monte Carlo method intended for use in sequential signal processing. Besides Bayes theory, it exploits the principle of importance sampling. The interest in particle filtering stems from its potential for coping with difficult nonlinear and/or non-Gaussian problems.

The basic idea in particle filtering is the recursive approximation of relevant probability distributions with discrete random measures. The earliest applications of sequential Monte Carlo methods were in the field of growing polymers [38, 70]. Later they expanded to other disciplines including physics and engineering. Sequential Monte Carlo methods found limited use in the past, except for the last decade, mostly due to their very high computational complexity and the lack of adequate computing resources. The rapid advancement of computers in the last decade and the outstanding potential of the particle filters have made them a very attractive research topic. The current interest in particle filtering for signal processing applications was brought on by [35]. Recent reviews and accounts of new developments on the subject can be found in [8, 26, 27].

Standard description of problems addressed by particle filters is given by a dynamic state-space model represented by state-space and observation equations, i.e.,

$$\begin{array}{rcl} \mathbf{x}_t & = & f_t(\mathbf{x}_{t-1}, \mathbf{u}_t) \\ \mathbf{y}_t & = & g_t(\mathbf{x}_t, \mathbf{v}_t) \end{array} \tag{11}$$

where $\mathbf{x}_t$ is a state vector, $f_t(\cdot)$ is a system transition function, $\mathbf{y}_t$ is a vector of observations, $g_t(\cdot)$ is a measurement function, $\mathbf{u}_t$ and $\mathbf{v}_t$ are noise vectors, and the subscript t denotes a discrete time index. The first equation is known as state equation, and the second, as measurement equation. The standard assumptions are that the analytical forms of the functions and the distributions of the two noises are known. Based on the observations $\mathbf{y}_t$ and the assumptions, the objective is to estimate $\mathbf{x}_t$ recursively.

The method that has been investigated the most and that has been most frequently applied for estimating $\mathbf{x}_t$ in practice is the Kalman filter [3], which is optimal in the important case when the equations are linear and the noises are independent, additive and Gaussian. In this situation, the densities of interest (filtering, predictive, or smoothing) are also Gaussian and the Kalman filter can compute them exactly without approximations. For scenarios where the models are nonlinear or the noise is non-Gaussian, various approximate methods have been proposed of which the extended Kalman filter is perhaps the most prominent [3].

Particle filtering has become an important alternative to the extended Kalman filter. With particle filtering the approximation is not in the linearizations around current estimates but rather approximations in the representation of the desired distributions by discrete random measures.

If the distribution of interest is $p(\mathbf{x})$, we approximate it by a random measure

$$\chi = \left\{ \mathbf{x}^{(m)}, w^{(m)} \right\}_{m=1}^{M} \tag{12}$$

where $\mathbf{x}^{(m)}$ are the particles, $w^{(m)}$ are their weights, and M is the number of particles used in the approximation. The measure χ approximates the density $p(\mathbf{x})$ by

$$p(\mathbf{x}) \approx \sum_{m=1}^{M} w^{(m)} \delta(\mathbf{x} - \mathbf{x}^{(m)}) \tag{13}$$

where $\delta(\cdot)$ is the Dirac delta function. With this approximation, similarly as in (2) and (3), the computations of expectations are simplified to summations.

The next important concept used in particle filtering is the principle of importance sampling. Suppose we want to approximate a density $p(\mathbf{x})$ with a discrete random measure. If we can generate the particles from $p(\mathbf{x})$, each of them will be assigned a weight equal to $1/M$. When direct sampling from $p(\mathbf{x})$ is intractable, one can generate particles $\mathbf{x}^{(m)}$ from a density $\pi(\mathbf{x})$, known also as importance function, and assign normalized weights according to

$$w^{*(m)} = \frac{p(\mathbf{x})}{\pi(\mathbf{x})}, \quad \text{and} \quad w^{(m)} = \frac{w^{*(m)}}{\sum_{i=1}^{M} w^{*(i)}}. \tag{14}$$

Let $\mathbf{x}_{0:t} \equiv \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_t\}$ represent a state trajectory from time 0 to t, and let $\mathbf{y}_{1:t}$ have analogous meaning. Suppose now that the posterior distribution $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$ is approximated by the discrete random measure $\chi_{t-1} = \{\mathbf{x}_{0:t-1}^{(m)}, w_{t-1}^{(m)}\}_{m=1}^{M}$. The discrete random measure χ_{t-1} is modified to χ_t after $\mathbf{y}_t$ is observed by exploiting the principle of importance sampling. First, particles $\mathbf{x}_t^{(m)}$ are generated and $\mathbf{x}_{0:t}^{(m)}$ is formed, and then the weights are updated to $w_t^{(m)}$.

More specifically, the first step amounts to drawing samples from the *importance* function $\pi(\cdot)$, i.e.,

$$\mathbf{x}_t^{(m)} \sim \pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)}, \mathbf{y}_{1:t}) \tag{15}$$

and the second step to computing the weights according to

$$w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})\, p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{\pi(\mathbf{x}_t^{(m)}|\mathbf{x}_{0:t-1}^{(m)}, \mathbf{y}_{1:t})} w_{t-1}^{(m)}. \tag{16}$$

A major problem with particle filtering is that the discrete random measure degenerates quickly. After only several samples, all the particles except for a very few are assigned negligible weights. The degeneracy implies that the performance of the particle filter degrades considerably. Degeneracy, however, can be reduced by using good importance sampling functions and *resampling*.

Resampling is a scheme that eliminates particles with small weights and replicates particles with large weights. In principle, it is implemented as follows:

1. Draw M particles, $\mathbf{x}_t^{*(m)}$ from the discrete distribution χ_t.
2. Let $\mathbf{x}_t^{(m)} = \mathbf{x}_t^{*(m)}$, and assign equal weights $(1/M)$ to the particles.

3. RECENT ADVANCES IN MARKOV CHAIN MONTE CARLO SAMPLING

In the past decade, the MCMC methodology has continued to attract the attention and important advances have been made both in theory and in its applications. Even though this method is computationally intensive and has issues that need very careful consideration before it is applied, its adoption by the scientific community has been outstanding.

3.1 A generalized Markov sampler

An important recent development in the theory of MCMC sampling has been the emergence of a new MCMC sampler which generalizes not only the Metropolis-Hastings and the Gibbs samplers, but also the RJMCMC sampler [47]. In fact, the proposed sampler is a natural extension of the Gibbs sampler, which is somewhat surprising if we recall the previous beliefs that the Gibbs sampler is inadequate for use in a general setting of model switching.

We briefly describe the generalized Gibbs sampler. This sampler generates a chain in a space $\mathrm{I} \times \mathrm{X}$, where I is an index set and X is the target set (the parameter space from which we sample). Suppose that $G := \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots\}$ is a Markov chain obtained by the standard Gibbs sampler which operates in a $d-$dimensional space. Then, each element of $\mathbf{x} \in \mathbb{R}^d$ is updated to form $\{x_1^{(j)}, x_2^{(j)}, \cdots, x_d^{(j)}\}$. The process $G' = \{x_1^{(1)}, x_2^{(1)}, \cdots, x_d^{(1)}, x_1^{(2)}, x_2^{(2)}, \cdots\}$ has the same limiting distribution as G but is not a Markov chain. We can form the process $G'' = \{(1, x_1^{(1)}), (2, x_2^{(2)}), \cdots, (d, x_d^{(1)}), (1, x_1^{(2)}), \cdots\}$ which is a Markov chain in $\mathrm{I} \times \mathrm{X}$. So the Gibbs sampler is a Markov chain in the space $I \times X$. The projection of its limiting distribution onto X is the required limiting distribution.

When we want to run the Gibbs sampler over parameter spaces with different dimensions, we need to introduce four new entities. The first is the set $\Phi(\mathbf{x})$ which is a catalogue of *types* of transitions that one may make from the element $\mathbf{x}$. In the case of the standard Gibbs sampler, it is $\Phi(\mathbf{x}) = \{(1,\mathbf{x}),(2,\mathbf{x}),\cdots,(d,\mathbf{x})\}$. The transition from $\mathbf{x}$ at a given step is defined by a transition distribution $Q((i,\mathbf{x}),(m,\mathbf{y}))$. Once the type of transition is selected, we need a set of transitions for that type. We denote it by $\Psi((i,\mathbf{x}))$. On this set, we define a distribution $R((i,\mathbf{x}),(m,\mathbf{y}))$, which is the fourth quantity that is needed. With these notions, we can summarize the algorithm as follows:

1. **Intialization.** Start with an arbitrary $\mathbf{U}^{(0)} = (i,x)$.
2. **Q step:** Given $\mathbf{U}^{(j)} = (i,\mathbf{x})$, generate $\mathbf{V} \in \Phi(\mathbf{x})$ from the distribution $Q((i,\mathbf{x}),\cdot)$.
3. **R step:** Given $\mathbf{V} = (m,\mathbf{y})$, generate $\mathbf{W} \in \Psi(m,\mathbf{y})$ from the distribution $R((m,\mathbf{y}),\cdot)$.
4. **Prepare for the next iteration**: Set $\mathbf{U}^{(j+1)} = \mathbf{W}$.

The algorithm generates a Markov chain $\{\mathbf{U}^{(0)},\mathbf{U}^{(1)}, \mathbf{U}^{(2)},\cdots\} = \{(i^{(0)},\mathbf{x}^{(0)}),(i^{(1)},\mathbf{x}^{(1)}),(i^{(2)},\mathbf{x}^{(2)}),\cdots\}$. The limiting distribution of $\mathbf{x}^{(j)}$ as $j \to \infty$ is the stationary distribution of the chain provided it is irreducible and aperiodic.

The above scheme is a generalized Gibbs sampler. With minor modifications one can obtain the generalized Metropolis and Hastings samplers. For more details, see [47].

3.2 Other developments

One difficulty with the MCMC methods is the impediment of finding a good proposal transition kernel. The implications of inadequate kernel are slow convergence of the chain and poor exploration of the parameter space. Novel methods have been proposed which generalize the Metropolis algorithm in that more than one proposals are made for moving the chain, for example, see [57].

As already mentioned, the convergence of the Markov chains is an important practical issue in applications. In [19], the convergence of several MCMC algorithms is studied for problems in digital communications. In [74], a novel method for visualizing convergence is proposed.

MCMC sampling has continued to find applications in a large number of areas. One of them is image and video retrieval. For example, in [58] an MCMC stochastic gradient algorithm is proposed for finding representations that have optimal retrieval performance. MCMC sampling is used in [49] for automated restoration of archived sequences. There, the problem is to simultaneously treat missing data and motion in degraded video sequences.

Medical imaging is another field where MCMC and RJMCMC have been applied. In [78], MCMC sampling is used for obtaining the full posterior distribution of the models of functional magnetic resonance image data and in [24] RJMCMC simulation is exploited to detection of brain lesions from magnetic resonance image data.

Other areas that have benefited recently from MCMC methods are machine learning [5], pattern analysis [72], neural networks [23, 62], behavioral studies [46], phylogenetic analysis [1, 63], multiuser detection in CDMA [75, 79], communication networks [37], seismology [69], and synthetic aperture radar imagery [66, 73].

4. RECENT ADVANCES IN PARTICLE FILTERING

4.1 Gaussian particle filters

Recently, a new type of particle filters called Gaussian particle filters (GPFs) has been proposed [51]. These particle filters approximate the predictive, $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$, and filtering, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, densities by Gaussians whose mean vectors and covariance matrices are computed from the particles. If at time $t-1$ we approximate the filtering density by $\mathrm{N}(\mu_{t-1},\mathbf{\Sigma}_{t-1})$ and at time t the predictive density by $\mathrm{N}(\bar{\mu}_t,\bar{\mathbf{\Sigma}}_t)$, then the steps of a simple implementation of the GPF are as follows:

1. Draw particles according to $\mathbf{x}_{t-1}^{(m)} \sim \mathrm{N}(\mu_{t-1},\mathbf{\Sigma}_{t-1})$.
2. Draw particles according to $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$.
3. Compute the weights of the particles by
$$w_t^{*(m)} = p(\mathbf{y}_t|\mathbf{x}_t^{(m)}).$$
4. Normalize the weights by
$$w_t^{(m)} = \frac{w_t^{*(m)}}{\sum_{j=1}^{M} w_t^{*(j)}}.$$
5. Estimate μ_t and $\mathbf{\Sigma}_t$ by
$$\begin{aligned}\mu_t &= \sum_{m=1}^{M} w_t^{(m)}\mathbf{x}_t^{(m)}\\ \mathbf{\Sigma}_t &= \sum_{m=1}^{M} w_t^{(m)}\left(\mathbf{x}_t^{(m)}-\mu_t\right)\left(\mathbf{x}_t^{(m)}-\mu_t\right)^\top\end{aligned}$$
which are the parameters of the Gaussian density that approximates $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

A distinctive feature of GPFs is that they do not require resampling, which is important in hardware implementation because this operation complicates hardware architectures. The resampling in GPFs is replaced by sampling from a Gaussian, which as a procedure is much simpler (in the outlined scheme, this is the second step). In a companion paper, [52], the approximating densities are modeled as mixture Gaussians, which provide more flexibility in capturing the shapes of the predictive and filtering densities but require more intense computations. In general, as noted in [51], the approximating densities can be any appropriate parametric densities.

Recall that resampling entails a problem that is referred to as particle attrition. It is particularly emphasized when the used models have *fixed* parameters. As the recursions progress with time, unless special steps are undertaken, the size of the particle set of the fixed parameters decreases and very quickly is depleted. This deficiency of the PFs has been recognized and addressed in the past, for example in [35] and [77], and more recently in [54]. In these approaches, the idea is to introduce artificial evolution of the particles and thereby treat them in more or less the same way as the dynamic states of the model. GPFs do not share the problem of standard PFs regarding constant parameters. This is because in the first step when the particles are drawn from the normal (or any other approximating) distribution, there is a natural evolution of the particles that are drawn from the space of the constant parameters.

4.2 Cost-reference particle filters

Another interesting development in the theory of particle filtering represents the introduction of the class of cost-reference particle filters (CRPFs) [60]. The idea behind them is the desire to have particle filters that can process data modeled by the standard state-space dynamic models but without the usual probabilistic assumptions about the noise in the state and observation equations. In other words, we want to use the same mechanism of exploring the space of unknowns by particles with assigned weights but without knowing the noise distributions of the model.

Key concepts of the new filters are the cost and risk functions. The cost function has a recursive structure and is defined by

$$C(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\lambda) = \lambda C(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1},\lambda) + \triangle C(\mathbf{x}_t|\mathbf{y}_t), \quad (17)$$

where $0 < \lambda < 1$ is a forgetting factor, $\triangle C : \mathbb{R}^{L_x} \times \mathbb{R}^{L_y} \to \mathbb{R}$ is the incremental cost function with L_x and L_y being the dimensions of $\mathbf{x}_t$ and $\mathbf{y}_t$, and $C(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\lambda)$ is defined by

$$C : \mathbb{R}^{(t+1)L_x} \times \mathbb{R}^{tL_y} \times \mathbb{R} \to \mathbb{R}.$$

The forgetting factor λ avoids attributing an excessive weight to old observations when a long series of data are collected, hence allowing for potential adaptivity. Note that we could have adopted a different recursion from the one in (17).

When $C(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\lambda)$ has a large value, the corresponding state sequence $\mathbf{x}_{0:t}$ is not a good estimate given the sequence of observations $\mathbf{y}_{1:t}$, and if $C(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\lambda)$ has small value, the associated trajectory $\mathbf{x}_{0:t}$ is close to the true state signal.

The other new concept is the one-step *risk* function, which is defined by

$$\begin{array}{cccccc} R: & \mathbb{R}^{L_x} & \times & \mathbb{R}^{L_y} & \to & \mathbb{R} \\ & \mathbf{x}_{t-1} & , & \mathbf{y}_t & \rightsquigarrow & R(\mathbf{x}_{t-1}|\mathbf{y}_t). \end{array} \quad (18)$$

The risk measures the quality of the state estimate $\mathbf{x}_{t-1}$ given the new observation $\mathbf{y}_t$. We can also view the risk function as a prediction of the cost increment, $\triangle C(\mathbf{x}_t|\mathbf{y}_t)$, before $\mathbf{x}_t$ is actually propagated. Thus, one choice of the risk function can be

$$R(\mathbf{x}_{t-1}|\mathbf{y}_t) = \triangle C\left(f_x(\mathbf{x}_{t-1})|\mathbf{y}_t\right). \quad (19)$$

The operation of the new filter proceeds sequentially in a similar way as the conventional particle filter. Given a set of M particle and costs (instead of weights) up to time $t-1$,

$$\Xi_{t-1} = \left\{\mathbf{x}_{t-1}^{(m)}, C_{t-1}^{(m)}\right\}_{m=1}^{M}$$

where $C_{t-1}^{(m)} = C(\mathbf{x}_{0:t-1}^{(m)}|\mathbf{y}_{1:t-1},\lambda)$, with the new observation $\mathbf{y}_t$, the set of state trajectories is extended to Ξ_{t+1}. The state-space model is the same as given by (11) and there are no assumptions about the forms of the noise probability distributions. We only add the following mild requirements:

1. The initial state is known to lie in a bounded interval $I_{\mathbf{x}_0} \subset \mathbb{R}^{L_x}$.

2. The system and observation noise processes are both zero-mean.

The resulting CRPF is implemented as follows:

1. **Time $t = 0$: initialization.**
   Generate M particles from the uniform distribution in the interval $I_{\mathbf{x}_0}$,
   $$\mathbf{x}_0^{(m)} \sim U(I_{\mathbf{x}_0}),$$
   and assign them zero costs. The initial weighted particle set is
   $$\Xi_0 = \{\mathbf{x}_0^{(m)}, C_0^{(m)} = 0\}_{m=1}^{M}$$
   is obtained.
2. **Time t: selection of the most promising trajectories.**
   The goal of the selection step is the same as that of resampling. We want to replicate those particles with a low cost and remove the ones with high-cost. With these filters, unlike in conventional particle filters, after resampling the particles preserve their costs.
   For $m = 1,2,\ldots,M$, compute the one-step risk of particle m and let
   $$R_t^{(m)} = \lambda C_t^{(m)} + R(\mathbf{x}_{t-1}^{(m)}|\mathbf{y}_t)$$
   which yields a predictive cost of the trajectory $\mathbf{x}_{0:t-1}$ according to the new observation, $\mathbf{y}_t$. We now define a probability mass function (pmf) of the form
   $$\hat{\pi}_{t+1}^{(m)} \propto \varphi(R_t^{(m)}) \quad (20)$$
   where $\varphi : \mathbb{R} \to [0,+\infty)$ is a monotonically decreasing function. We use this function to obtain an intermediate weighted particle set $\hat{\Xi}_t = \left\{\hat{\mathbf{x}}_{t-1}^{(m)}, \hat{C}_{t-1}^{(m)}\right\}_{m=1}^{M}$.
3. **Time $t+1$: random particle propagation.** We exploit an arbitrary conditional probability distribution function $\pi_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ (which must satisfy some very mild condition), to draw new particles $\mathbf{x}_t^{(m)} \sim \pi_t(\mathbf{x}_t|\hat{\mathbf{x}}_{t-1}^{(m)})$ and update the associated costs,
   $$C_t^{(m)} = \lambda \hat{C}_{t-1}^{(m)} + \triangle C_t^{(m)}$$
   where
   $$\triangle C_t^{(m)} = \triangle C(\mathbf{x}_t^{(m)}|\mathbf{y}_t)$$
   for $m = 1,2,\ldots,M$. The obtained set of weighted particles is $\Xi_t = \left\{\mathbf{x}_t^{(m)}, C_t^{(m)}\right\}_{m=1}^{M}$.
4. **Time $t+1$: estimation of the state.** To obtain estimates of the states, we assign probability masses to the particles of Ξ_t by using their costs, for example by
   $$\pi_t^{(m)} \propto \varphi(C_t^{(m)}) \quad (21)$$
   where φ is a monotonically decreasing function. Once the pmf is defined, the estimation of the states is straightforward.

The details of the filter which includes discussion on choosing the cost and risk functions as well as the φ function can be found in [60]. In addition, that paper provides convergence results of the proposed filter. Simulation results show that the CRPFs are very promising and that they yield much improved performance over conventional particle filters in cases where the noise distributions are unknown.

4.3 Implementation of particle filters

It is well known that the particle filters are computationally intensive but that they are also parallelizable. The latter opens up an interesting area of research related to the hardware design of particle filters. Some progress in the developing of architectures for particle filters has already been made and the research related to these activities has produced some very interesting results.

The main design objective is the achievement of high speed, that is, the minimization of the input sampling period for a given number of particles. The sampling period of the particle filter is the time interval that is necessary to process one input observation. A high speed implementation requires that all the operations are spatially mapped, i.e., each operation has its own hardware block and all the blocks perform their operations concurrently. Moreover, the particle filter operations have to be overlapped in time and each block has to be pipelined [41]. The only platforms that allow for application of temporal and spatial concurrency are ASIC [43] and FPGA.

The minimum sampling period that can be obtained after implementing a standard particle filter is $(2M+L)\cdot T_{clk}$ where L is the constant hardware latency defined by the depth of pipelining of particle filtering operations, T_{clk} is the clock period and M is the number of particles [11, 12, 14, 42]. The sampling period can be decreased by introducing parallelism [12, 13] so that there are K processing elements which perform the same particle filtering operations on the particles that are distributed to them. The minimum sampling period is then

$$T_{\min} = (2M/K + L + L_c)\cdot T_{clk} \tag{22}$$

where L_c is the latency due to communication between parallel elements. We point out that increases in speed can be accomplished not only by using better technology but also by combining the outputs of particle filters that share the data processing load.

A prototype of a particle filter has recently been built using an FPGA platform. The filter achieves processing speeds that are 32 times higher than the speed of a state-of-the art DSP processor which implements the same particle filter.

4.4 Other developments

Much of the new developments is related to novel applications of particle filters in a wide range of areas. One of them is wireless communications where particle filters are used for blind equalization in time-invariant channels, [56, 61], time-varying channels [32], additive Gaussian and non-Gaussian channels [65], and in OFDM systems [80]. Blind detection over flat fading channels [18, 44, 50, 65], multiuser detection [45], space-time coding [81], syncronization [31], phase tracking [2], and tracking the statistical variations of channel matrices in MIMO wireless channels [40] is also addressed. More on the application of particle filtering in wireless communications can be found in [25].

An area where particle filters are frequently applied is target tracking. In [34], a particle filter is applied to tracking a target that is occasionally hidden in blind Doppler regions. Real-time speaker tracking by applying sensor fusion is described in [20], and similarly, tracking of acoustics source in a reverberant environment in [76]. Additional work on distributed signal processing by particle filtering in sensor networks is reported in [9, 21, 71].

Many problems in computer vision such as probabilistic tracking of objects in image sequences are addressed by particle filters. For example, in [64] visual tracking by fusion of the three cues with particle filtering is achieved, and results on real teleconference and surveillance data are provided. In [82], a particle filter is proposed for face recognition from a probe video and compared with a gallery of still templates, and in [16], target tracking in cluttered image sequences of infrared airborne radar is performed.

We conclude this list with a few more applications of particle filtering. In [28], a new particle filter is proposed for online inference of hidden-Markov models, and in [4] a particle filter is developed for jump Markov processes, which have a hierarchical structure and consist of a mixture of heterogeneous models. A survey of applications of particle filtering to change detection, system identification and control is provided in [7]. Recent applications of particle filters in mobile robots are presented in [22, 53].

5. CONCLUSIONS

Monte Carlo methods will find increasing use in many signal processing problems. With the continued advances in the theory of these methods, the proliferation of computer technology, and the development of special purpose hardware, the role of Monte Carlo methods in resolving highly complex problems in science and engineering will only grow.

REFERENCES

[1] G. Altekar, S. Dwarkadas, J. P. Huelsenbeck, and F. Ronquist, "Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference," *Bioinformatics*, vol. 20, no. 3, pp. 407–415, 2004.

[2] P. O. Amblard, J. M. Brossier, and E. Moisan, "Phase tracking: what do we gain from optimality? particle filtering versus phase-locked loops," *Signal Processing*, vol. 83, pp. 151–167, 2003.

[3] B. D. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, New Jersey, 1979.

[4] C. Andrieu, M. Davy, and A. Doucet, "Efficent particle filter for jump Markov systems. Application to time-varyimg autoregressions," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1762–1770, 2003.

[5] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, pp. 5–43, 2003.

[6] C. Andrieu, P. M. Djurić, and A. Doucet, "Model selection by Markov chain Monte Carlo computations," *Signal Processing*, vol. 81, pp. 19–38, 2001.

[7] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić, "Particle methods for change detection, system identification, and control," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 423–438, 2004.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[9] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *the Proceedings of the First International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, 2003, pp. 150–161.

[10] I. Beichl and F. Sullivan, "The Metropolis algorithm," *Computing in Science and Engineering*, vol. 2, no. 1, pp. 65–69, 2000.

[11] M. Bolić, A. Athalye, P. M. Djurić, and S. Hong, "A design study for practical physical implementation of Gaussian particle filters," *IEEE Transactions on Circuits and Systems I*, 2004, submitted.

[12] M. Bolić, P. M. Djurić, and S. Hong, "New resampling algorithms for particle filters," in *the Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, 2003.

[13] M. Bolić, P. M. Djurić, and S. Hong, "Resampling algorithms and architectures for distributed particle filters," *IEEE Transactions on Signal Processing*, 2003, submitted.

[14] M. Bolić, P. M. Djurić, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal of Applied Signal Processing*, 2003, submitted.

[15] S. Brooks and G. Roberts, "Assessing convergence of Markov chain Monte Carlo algorithms," *Statistics and Computing*, pp. 319–335, 1999.

[16] M. G. S. Bruno, "Sequential importance sampling filtering for target tracking in image sequences," *IEEE Signal Processing Letters*, vol. 10, no. 8, pp. 246–249, 2003.

[17] M. H. Chen, Q. M. Shao, and J. G. Ibrahim, Eds., *Monte Carlo Methods for Bayesian Computation*, Springer Verlag, New York, 2001.

[18] R. Chen and J. S. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society*, vol. 62, no. Part 3, pp. 493–508, 2000.

[19] R. Chen, J. S. Liu, and X. Wang, "Convergence analysis and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 255–270, 2002.

[20] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 485–494, 2004.

[21] M. Coates, "Distributed particle filters for sensor networks," in *the Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, San Francisco, CA, 2004.

[22] N. de Freitas, R. Dearden, F. Hutter, J. Mutch R. Morales-Menéndez, and D. Poole, "Diagnosis by a waiter and a Mars explorer," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 455–468, 2004.

[23] N. de Freitas, M. Niranjan, A.H. Gee, and A. Doucet, "Sequential Monte Carlo methods to train neural work models," *Neural Computation*, vol. 12, no. 4, pp. 955–993, 2000.

[24] X. Descombes, F. Kruggel, G. Wollny, and H. J. Gertz, "An object-based approach for detecting small brain lesions: application to Virchow-Robin spaces," *IEEE Transactions on Medical Imaging*, vol. 23, no. 2, pp. 246–255, 2004.

[25] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[26] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.

[27] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, pp. 197–208, 2000.

[28] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *Journal of the Royal Statistical Society, Series B*, vol. 65, Part 4, pp. 887–899, 2003.

[29] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall, 1995.

[30] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[31] T. Ghirmai, M. F. Bugallo, J. Míguez, and P. M. Djurić, "A sequential Monte Carlo method for adaptive blind timing estimation and data detection," *IEEE Transactions on Signal Processing*, 2004, submitted.

[32] T. Ghirmai, J. Kotecha, and P. M. Djurić, "Semi-blind equalization for time-varying channels using particle filtering," *Digital Signal Processing*, 2004, to be published.

[33] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall, New York, 1996.

[34] N. Gordon and B. Ristić, "Tracking airborne targets occasionally hidden in the blind Doppler," *Digital Signal Processing*, vol. 12, pp. 383–393, 2002.

[35] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.

[36] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 4, pp. 711–732, 1995.

[37] D. Guo and X. Wang, "Bayesian inference of network loss and delay characteristics with applications to TCP performance prediction," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2205–2218, 2003.

[38] J. M. Hammersley and K. W. Morton, "Poor man's Monte Carlo," *Journal of the Royal Statistical Society, Series B*, vol. 16, no. 1, pp. 23–38, 1954.

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

[40] S. Haykin, K. Huber, and Z. Chen, "Bayesian sequential state estimation for MIMO wireless communications," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 439–454, 2004.

[41] S. Hong, M. Bolić, and P. M. Djurić, "Design complexity comparison method for loop-based signal processing algorithms: Particle filters," in *the Proceedings of the IEEE International Symposium on Circuits and Systems*, 2004.

[42] S. Hong, M. Bolić, and P. M. Djurić, "An efficient fixed-point implementation of residual systematic resampling scheme for high-speed particle filters," *IEEE Signal Processing Letters*, 2004, accepted for publication.

[43] S. Hong, S. S. Chin, M. Bolić, and P. M. Djurić, "Design and implementation of flexible resampling mechanisms for high-speed parallel particle filters," *Journal of VLSI Signal Processing*, 2003, submitted.

[44] Y. Huang and P. M. Djurić, "A blind particle filtering detector for joint channel estimation and symbol detection over flat fading channels," *IEEE Transactions on Signal Processing*, accepted for publication.

[45] R. A. Iltis, "A sequential Monte Carlo filter for joint linear/nonlinear state estimation with application to DS-CDMA," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 417–426, 2003.

[46] M. S. Johnson and B. W. Junker, “Using data augmentation and Markov chain Monte Carlo for the estimation of unfolding response models,” *Journal of Educational and Behavioral Statistics*, vol. 28, no. 3, pp. 195–230, 2003.

[47] J. M. Keith, D. P. Kroese, and D. Bryant, “A generalised Markov sampler,” *Methodology and Computing in Applied Probability*, vol. 6, no. 1, pp. 29–53, 2004.

[48] S. Kirkpatrick, C. D. Gelat, and M. P. Vechi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–680, 1983.

[49] A. Kokaram, “Practical, unified, motion and missing data treatment in degraded video,” *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 163–177, 2004.

[50] J. Kotecha and P. M. Djurić, “Blind sequential detection for Rayleigh fading channels using hybrid Monte Carlo - recursive identification algorithms,” *Signal Processing*, accepted for publication.

[51] J. Kotecha and P. M. Djurić, “Gaussian particle filtering,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.

[52] J. Kotecha and P. M. Djurić, “Gaussian sum particle filtering,” *IEEE Transactions on Signal Processing*, vol. 10, no. 10, pp. 2602–2612, 2003.

[53] C. Kwok, D. Fox, and M. Meila, “Real-time particle filters,” vol. 92, no. 3, pp. 469–484, 2004.

[54] J. Liu and M. West, “Combined parameter and state estimation in simulation-based filtering,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., pp. 197–223. Springer, 2001.

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

[56] J. S. Liu and R. Chen, “Blind deconvolution via sequential imputations,” *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, 1995.

[57] J. S. Liu, F. Liang, and W. H. Wong, “The use of multiple-try method and local optimization in Metropolis sampling,” *Journal of the American Statistical Association*, vol. 95, pp. 121–134, 2000.

[58] X. W. Liu, A. Srivastava, and D. H. Sun, “Image and video retrieval,” *Lecture Notes in Computer Science*, vol. 2728, pp. 50–60, 2003.

[59] N. Metropolis, A. W. Rosenblath, M. N. Rosenblath, A. H. Teller, and E. Teller, “Equations of state calculations by fast computing machines,” *Journal of Chemical Physics*, vol. 21, pp. 1087–1091, 1954.

[60] J. Míguez, M. Bugallo, and P. M. Djurić, “A new class of particle filters for random dynamic systems with unknown statistics,” *EURASIP Journal of Applied Signal Processing*, submitted.

[61] J. Míguez and P. M. Djurić, “Blind equalization of frequency selective channels by sequential importance sampling,” *IEEE Transactions on Signal Processing*, 2004, to be published.

[62] R. M. Neal, *Bayesian learning for neural networks*, Springer-Verlag, New York, 1996.

[63] J. A. A. Nylander, F. Ronquist, J. P. Huelsenbeck, and J. L. Nieves-Aldrey, “Bayesian phylogenetic analysis of combined data,” *Systematic Bilogy*, vol. 53, no. 1, pp. 47–67, 2004.

[64] P. Pérez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particle filters,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, 2004.

[65] E. Punskaya, C. Andrieu, A. Doucet, and W. Fitzgerald, “Particle filtering for demodulation in fading channels with non-Gaussian additive noise,” *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 579–582, 2001.

[66] S. Ray and B. K. Malick, “A Bayesian transformation mode for wavelet shrinkage,” *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1512–1521, 2003.

[67] B. D. Ripley, *Stochastic Simulation*, John Wiley & Sons, New York, 1987.

[68] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, New York, 1999.

[69] O. Rosec, J.-M. Boucher, B. Nsiri, and T. Chonavel, “Blind marine seismic deconvolution using statistical MCMC methods,” *IEEE Journal of Oceanic Engineering*, vol. 28, no. 3, pp. 502–512, 2003.

[70] M. N. Rosenbluth and A. W. Rosenbluth, “Monte Carlo calculation of the average extension of molecular chains,” *Journal of Chemical Physics*, vol. 23, no. 2, pp. 356–359, 1956.

[71] M. Rosenkrantz, G. Gordon, and S. Thrun, “Decentralized sensor fusion with distributed particle filters,” in *the Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, 2003.

[72] P. S. Torr and C. Davidson, “IMPSAC: synthesis of importance sampling and random sample consensus,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 354–364, 2003.

[73] J.-Y. Tourneret, M. Doisy, and M. Lavielle, “Bayesian off-line detection of multiple change-points corrupted by multiplicative noise: application to SAR image edge detection,” *Signal Processing*, vol. 83, pp. 1871–1887, 2003.

[74] J. Venna, S. Kaski, and J. Peltonen, “Visualizations for assesing convergence and mixing of MCMC,” *Machine Learning: EMCL, Lecture Notes in Artificial Intelligence*, vol. 1837, pp. 432–443, 2003.

[75] X. Wang and R. Chan, “Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and non-Gaussian noise,” *IEEE Transactions on Signal Processing*, vol. 48, no. 7, pp. 2013–2029, 2000.

[76] D. B. Ward, E. A. Lehmann, and R. C. Williamson, “Particle filtering algorithms for tracking an acoustic source in a reverberant environment,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, 2003.

[77] M. West, “Mixture models, Monte Carlo, Bayesian updating and dynamic models,” *Computer Science and Statistics*, vol. 24, pp. 325–333, 1993.

[78] M. W. Woolrich, M. Jenkinson, J. M. Brady, and S. M. Smith, “Fully Bayesian spatio-temporal modeling of FMRI data,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 2, pp. 213–231, 2004.

[79] Z. Yang and X. Wang, “Blind turbo multiuser detection for long-code multipath CDMA,” *IEEE Transactions on Communications*, vol. 50, no. 1, pp. 112–125, 2002.

[80] Z. Yang and X. Wang, “A sequential Monte Carlo blind receiver for OFDM systems in frequency-selective fading channels,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 271–280, 2002.

[81] J. Zhang and P. M. Djurić, “Joint estimation and decoding of space-time trellis codes,” *EURASIP Journal on Applied Signal Processing*, vol. 2002, pp. 305–315, 2002.

[82] S. Zhou, V. Krueger, and R. Chellapa, “Probabilistic recognition of human faces from video,” *Computer Vision and Image Understanding*, vol. 91, pp. 214–245, 2003.