Backward Simulation Methods for Monte Carlo Statistical Inference

Backward Simulation Methods for Monte Carlo Statistical Inference

Fredrik Lindsten

Linköping University
Sweden
lindsten@isy.liu.se

Thomas B. Schön

Linköping University
Sweden
schon@isy.liu.se

the essence of knowledge

Boston – Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is F. Lindsten and T. B. Schön, Backward Simulation Methods for Monte Carlo Statistical Inference, Foundations and Trends® in Machine Learning, vol 6, no 1, pp 1–143, 2013.

ISBN: 978-1-60198-698-6

Foundations and Trends® in Machine Learning
Volume 6 Issue 1, 2013

Editorial Board

Editor-in-Chief:
Michael Jordan
Department of Electrical Engineering and Computer Science
Department of Statistics
University of California, Berkeley
Berkeley, CA 94720-1776

Editors

Editorial Scope

Foundations and Trends® in Machine Learning publishes survey and tutorial articles in the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians
Foundations and Trends® in Machine Learning, 2013, Volume 6, 4 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Foundations and Trends® in
Machine Learning
Vol. 6, No. 1 (2013) 1–143

DOI: 10.1561/2200000045

now
the essence of knowledge

Backward Simulation Methods for Monte Carlo Statistical Inference

Fredrik Lindsten[1] and Thomas B. Schön[2]

[1] *Division of Automatic Control, Linköping University, Linköping, 581 83, Sweden, lindsten@isy.liu.se*
[2] *Division of Automatic Control, Linköping University, Linköping, 581 83, Sweden, schon@isy.liu.se*

Abstract

Monte Carlo methods, in particular those based on Markov chains and on interacting particle systems, are by now tools that are routinely used in machine learning. These methods have had a profound impact on statistical inference in a wide range of application areas where probabilistic models are used. Moreover, there are many algorithms in machine learning which are based on the idea of processing the data sequentially, first in the forward direction and then in the backward direction. In this tutorial, we will review a branch of Monte Carlo methods based on the forward–backward idea, referred to as backward simulators. These methods are useful for learning and inference in probabilistic models containing latent stochastic processes. The theory and practice of backward simulation algorithms have undergone a significant development in recent years and the algorithms keep finding new applications. The foundation for these methods is sequential Monte Carlo (SMC). SMC-based backward simulators are capable of addressing smoothing problems in sequential latent variable models, such as

general, nonlinear/non-Gaussian state-space models (SSMs). However, we will also clearly show that the underlying backward simulation idea is by no means restricted to SSMs. Furthermore, backward simulation plays an important role in recent developments of Markov chain Monte Carlo (MCMC) methods. Particle MCMC is a systematic way of using SMC within MCMC. In this framework, backward simulation gives us a way to significantly improve the performance of the samplers. We review and discuss several related backward-simulation-based methods for state inference as well as learning of static parameters, both using a frequentistic and a Bayesian approach.

Contents

1

Introduction

A basic strategy to address many inferential problems in machine learning is to process data sequentially, first in the forward direction and then in the backward direction. Examples of this approach are the well-known forward–backward algorithm for hidden Markov models (HMMs) and the Rauch–Tung–Striebel smoother [119] for linear Gaussian state-space models. Moreover, two decades of research on sequential Monte Carlo and Markov chain Monte Carlo have enabled inference in increasingly more challenging models. Many developments have been made in order to make use of the forward–backward idea together with these Monte Carlo methods, providing inferential techniques collectively referred to as backward simulation. This tutorial provides a unifying view of these methods. In this introductory section we review some relevant background materials and also derive a first backward simulator for the special case of linear Gaussian state-space models.

1.1 Background and Motivation

For over half a century, Monte Carlo methods have been recognized as potent tools for statistical inference in complex probabilistic

models; see [103] for an early discussion. A continuous development and refinement of these methods have enabled inference in increasingly more challenging models. A key milestone in this development was the introduction of Markov chain Monte Carlo (MCMC) methods through the inventions of the Metropolis–Hastings algorithm [71, 102] and the Gibbs sampler [58]. Parallel to this, sequential importance sampling [70] and sampling/importance resampling [122] laid the foundation of sequential Monte Carlo (SMC). In its modern form, SMC was first introduced in [64, 129]. During the 1990s, several independent developments were made by, among others, [77, 83]. Recently, SMC and MCMC have been combined in a systematic manner through the developments of pseudo-marginal methods [6, 11] and particle MCMC [3].

Backward simulation is a strategy which is useful as a Monte Carlo method for learning of probabilistic models containing latent stochastic processes. In particular, we will consider inference in dynamical systems, i.e., systems that evolve over time. Dynamical systems play a central role in a wide range of scientific fields, such as signal processing, automatic control, epidemiology and econometrics, to mention a few.

One of the most widely used models of a dynamical system is the state-space model (SSM), reviewed in more detail in Sections 1.4–1.6. The structure of an SSM can be seen as influenced by the notion of a physical system. At each time t, the system is assumed to be in a certain state x_t. The state contains all relevant information about the system, i.e., if we would know the state of the system we would have full insight into its internal condition. However, the state is typically not known. Instead, we measure some quantity y_t which depend on the state in some way. Given a sequence of observations $y_{1:T} \triangleq (y_1, \ldots, y_T)$, we seek to draw inference about the latent state process $x_{1:T}$ (state inference), as well as about unknown static parameters of the model (parameter inference).

The class of SSMs will play a central role in this tutorial. Indeed, many of the inferential methods that we will review have been developed explicitly for SSMs. However, as will become apparent in Sections 4 and 5, most of the methods are more general and can be used for learning interesting models outside the class of SSMs.

Backward simulation is based on the forward–backward idea. That is, the data is processed first in the forward direction and then in the backward direction. In the backward pass, the state process is simulated backward in time, i.e., by first simulating x_T, then x_{T-1} etc., until a complete state trajectory $x_{1:T}$ is generated. This procedure gives us a tool to address the state smoothing problem in models for which no closed form solution is available. This is done by simulating multiple backward trajectories from the smoothing distribution, i.e., conditionally on the observations $y_{1:T}$, which can then be used for Monte Carlo integration. State smoothing is of key relevance, e.g., to obtain refined state estimates in offline settings. Furthermore, it lies at the core of many parameter inference methods (see Section 1.5) and it can be used to address problems in optimal control (see Section 4.1).

Backward simulation is also useful in MCMC, as a way of grouping variables to improve the mixing of the sampler. A common way to construct an MCMC sampler for an SSM is to sample the state variables x_t, for different t, one at a time (referred to as single-state sampling). However, since the states are often strongly dependent across time, this can lead to poor performance. Backward simulation provides a mean of grouping the state variables and sampling the entire trajectory $x_{1:T}$ as one entity. As we will illustrate in Section 1.3, this can lead to a considerable improvement upon the single-state sampler.

In Section 1.7 we will derive a first backward simulator for the class of linear Gaussian state-space (LGSS) models. Apart from LGSS models, exact backward simulation is tractable, basically only for finite state-space HMMs (see also Section 4.1.1). The main focus in this tutorial will be on models outside these restricted classes, for which exact backward simulation is not possible. Instead, we will make use of SMC (and MCMC) to enable backward simulation in challenging probabilistic models, such as nonlinear/non-Gaussian SSMs, as well as more general non-Markovian latent variable models.

1.2 Notation and Definitions

For any sequence $\{x_k\}_{k\in\mathbb{N}}$ and integers $m \leq n$ we write $x_{m:n} \triangleq (x_m, \ldots, x_n)$. We let $\wedge$ be the minimum operator, i.e., $a \wedge b \triangleq \min(a, b)$.

For a matrix A, the matrix transpose is written as A^{T}. For two probability distributions μ_1 and μ_2, the total variation distance is given by $\|\mu_1 - \mu_2\|_{\mathrm{TV}} \triangleq \sup_A |\mu_1(A) - \mu_2(A)|$. A Dirac point-mass located at some point x' is denoted as $\delta_{x'}(dx)$. We write $X \sim \mu$ to mean that the random variable X is either distributed according to μ, or sampled from μ. The uniform probability distribution on the interval $[a, b]$ is written as $\mathcal{U}([a, b])$. $\mathrm{Cat}(\{p_i\}_{i=1}^n)$, with $\sum_{i=1}^n p_i = 1$, is the categorical (i.e., discrete) probability distribution on the set $\{1, \ldots, n\}$, with probabilities $\{p_i\}_{i=1}^n$. Finally, $\mathcal{N}(m, \Sigma)$ and $\mathcal{N}(x; m, \Sigma)$ are the Gaussian (i.e., normal) probability distribution and density function, respectively, with mean vector m, covariance matrix Σ and argument x.

1.3 A Preview Example

Before we continue with this section on background theory, we consider an example to illustrate the potential benefit of using backward simulation. A simple stochastic volatility SSM is given by,

$$x_{t+1} = ax_t + v_t, \qquad v_t \sim \mathcal{N}(0, q), \tag{1.1a}$$

$$y_t = e_t \exp\left(\tfrac{1}{2} x_t\right), \qquad e_t \sim \mathcal{N}(0, 1), \tag{1.1b}$$

where the state process $\{x_t\}_{t \geq 1}$ is latent and observations are made only via the measurement process $\{y_t\}_{t \geq 1}$. Similar models have been used to generalize the Black–Scholes option pricing equation to allow for the variance to change over time [27, 101]. The same model was used by [30] to illustrate the poor mixing of a single-state Gibbs sampler; an example which is replicated here.

For simplicity, we assume that the parameters $a = 0.99$ and $q = 0.01$ are known. We seek the density $p(x_{1:T} \mid y_{1:T})$, i.e., the conditional density of the state process $x_{1:T}$ given a sequence of observations $y_{1:T}$ for some fixed final time point T. This conditional density is referred to as the joint smoothing density (JSD). For the model under study, the JSD is not available in closed form due to the nonlinear measurement Equation (1.1b). To remedy this, we construct an MCMC method to approximately sample from it. MCMC will be reviewed in more detail in Section 2.2. However, the basic idea is to simulate a Markov chain which is constructed in such a way that it admits the target

distribution as limiting distribution. The sample path from the Markov chain can then be used to draw inference about the target density $p(x_{1:T} \mid y_{1:T})$.

As an initial attempt, we try a single-state Gibbs sampler. That is, we sample each state x_t conditionally on $\{x_{1:t-1}, x_{t+1:T}\}$ (and the observations $y_{1:T}$). At each iteration of the Gibbs sampler we thus simulate according to,

$$\begin{aligned} x'_1 &\sim p(x_1 \mid x_{2:T}, y_{1:T}); \\ &\vdots \\ x'_t &\sim p(x_t \mid x'_{1:t-1}, x_{t+1:T}, y_{1:T}); \\ &\vdots \\ x'_T &\sim p(x_T \mid x'_{1:T-1}, y_{1:T}). \end{aligned}$$

This procedure will leave $p(x_{1:T} \mid y_{1:T})$ invariant (see Section 2.2 for more on Gibbs sampling) and it results in a valid MCMC sampler. The conditional densities $p(x_t \mid x_{1:t-1}, x_{t+1:T}, y_{1:T})$ are not available in closed form. However, for this model (Equation (1.1)), they are log-concave and we can employ the efficient rejection sampling strategy by [145] to sample exactly from these distributions.

The single-state Gibbs sampler will indeed converge to samples from $p(x_{1:T} \mid y_{1:T})$. However, it is well recognized that single-state samplers can suffer from poor mixing, due to the often strong dependencies between consecutive state variables. That is, the convergence can be slow in the sense that we need to iterate the above sampling scheme a large number of times to get reliable samples.

To analyze this, we generate $T = 100$ samples from the model (Equation (1.1)) and run the Gibbs sampler for 100 000 iterations (in each iteration, we loop over all the state variables for $t = 1, \ldots, T$). The first 10 000 iterations are discarded, to avoid transient effects. We then compute the empirical autocorrelation function (ACF) of the state x_{50}, which is given in Figure 1.1. As can be seen, the ACF decreases very slowly, indicating a poorly mixing Gibbs kernel. This simply reflects the fact that, when the state variables are highly correlated, the single-state sampler will be inefficient at exploring the state-space. This is a

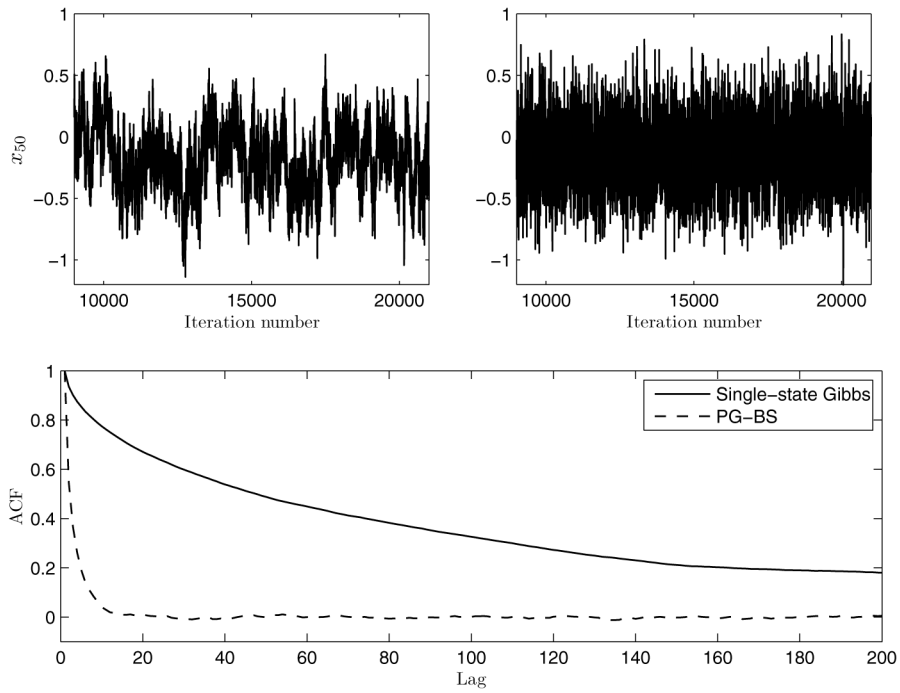

Fig. 1.1 (Top left) Part of sample path for the single-state Gibbs sampler; (Top right) Part of sample path for PGBS; (Bottom) Empirical ACF for x_{50} for the single-state Gibbs sampler and for PGBS using $N = 15$ particles.

common and well-recognized problem when addressing the state inference problem for SSMs.

One way to remedy this is to group the variables and sample a full state trajectory $x_{1:T}$ jointly. This is what a backward simulator aims to accomplish. Grouping variables in a Gibbs sampler will in general improve upon the mixing of the single-state sampler [97, Section 6.7], and in practice the improvement can be quite considerable.

To illustrate this, we have included the ACF for a backward-simulation-based method in Figure 1.1. Since the model (Equation (1.1)) is nonlinear, exact backward simulation is not possible. Instead, the results reported here are from a backward simulator based on SMC, using (only) $N = 15$ particles. The specific method that we have used is denoted as particle Gibbs with backward simulation (PGBS), and it will be discussed in detail in Section 5.4. For the PGBS,

the ACF drops off much more rapidly, indicating a more efficient sampler. Furthermore, a key property of PGBS is that, despite the fact that it relies on a crude SMC approximation, it does not alter the stationary distribution of the Gibbs sampler, nor does it introduce any additional bias. That is, PGBS will, just as the single-state Gibbs sampler, target the exact JSD $p(x_{1:T} \mid y_{1:T})$. This property is known as *exact approximation*, a concept that we will return to in Section 5.

1.4 State-Space Models

State-space models (SSMs) are commonly used to model time series and dynamical systems. Additionally, many models that are not sequential "by nature" can also be written on state-space form. It is a comprehensive and important class of models, and it serves as a good starting point for introducing the concepts that will be discussed throughout this tutorial.

We consider here discrete-time SSMs on a general state-space X. The system state is a Markov process $\{x_t\}_{t\geq 1}$ on X, evolving according to a Markov transition kernel $F(dx_{t+1} \mid x_t)$ and with initial distribution $\nu(dx_1)$. The state x_t is assumed to summarize all relevant information about the system at time t. However, the state process is latent and it is observed only implicitly through the observations $\{y_t\}_{t\geq 1}$, taking values in some set Y. Given x_t, the measurement y_t is conditionally independent of past and future states and observations, and it is distributed according to a kernel $G(dy_t \mid x_t)$. A graphical model, illustrating the conditional dependencies in an SSM, is given in Figure 1.2.

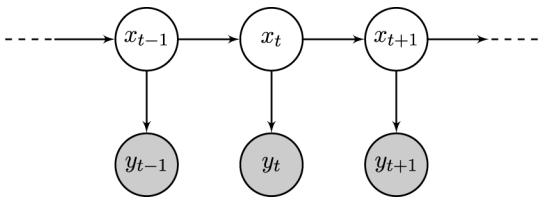

Fig. 1.2 Graphical model of an SSM. The white nodes represent latent variables and the gray nodes represent observed variables.

We shall assume that the observation kernel G admits a probability density g w.r.t. some dominating measure, which we simply denote dy. Such models are referred to as partially dominated. If, in addition, the transition kernel F admits a density f and the initial distribution ν admits a density μ, both w.r.t. some dominating measure dx, the model is called fully dominated. In summary, a fully dominated SSM can be expressed as,

$$x_{t+1} \sim f(x_{t+1} \mid x_t), \tag{1.2a}$$
$$y_t \sim g(y_t \mid x_t), \tag{1.2b}$$

and $x_1 \sim \mu(x_1)$. Two examples of SSMs follow below.

Example 1.1 (Finite state-space hidden Markov model). A finite state-space HMM, or simply HMM, is an SSM with $\mathsf{X} = \{1, \ldots, K\}$ for some finite K. The transition density (w.r.t. counting measure) can be summarized in a $K \times K$ transition matrix Π, where the (i,j)th entry is given by,

$$\Pi_{i,j} = P(x_{t+1} = j \mid x_t = i) = f(j \mid i).$$

Hence, $f(j \mid i)$ denotes the probability of moving from state i at time t, to state j at time $t+1$.

Example 1.2 (Additive noise model). In engineering applications, SSMs are often expressed on functional form with additive noise,

$$x_{t+1} = a(x_t) + v_t,$$
$$y_t = c(x_t) + e_t,$$

for some functions a and c. Here, the noises v_t and e_t are commonly referred to as process noise and measurement noise, respectively. If the noise distributions admit densities w.r.t. dominating measures, then the model is fully dominated. The transition density is then given by $f(x_{t+1} \mid x_t) = p_{v_t}(x_{t+1} - a(x_t))$ and similarly for the observation density.

Throughout this tutorial, we will mostly be concerned with fully dominated SSMs and therefore do most of our derivations in terms of probability densities. There are, however, several examples of interesting models that are *degenerate*, i.e., that are not fully dominated. We will return to this in the sequel and discuss how it affects the methods presented in here.

1.5 Parameter Learning in SSMs

The basic inference problem for SSMs is typically that of state inference, i.e., to infer the latent states given measurements from the system. In fact, even when the actual task is to learn a model of the system dynamics, state inference tends to play a crucial role as an intermediate step of the learning algorithm. To illustrate this, assume that the SSM (Equation (1.2)) is parameterized by some unknown parameter $\theta \in \Theta$,

$$x_{t+1} \sim f_\theta(x_{t+1} \mid x_t), \tag{1.3a}$$

$$y_t \sim g_\theta(y_t \mid x_t), \tag{1.3b}$$

and $x_1 \sim \mu_\theta(x_1)$. Given a batch of measurements $y_{1:T}$, we wish to draw inference about θ. In the Bayesian setting, a prior distribution $\pi(\theta)$ is assigned to the parameter and the learning problem amounts to computing the posterior distribution $p(\theta \mid y_{1:T})$.

A complicating factor is that the likelihood $p(y_{1:T} \mid \theta)$ in general cannot be computed in closed form. To address this difficulty, it is common to make use of *data augmentation* [136, 132]. That is, we target the joint state and parameter posterior $p(\theta, x_{1:T} \mid y_{1:T})$, rather than the marginal posterior $p(\theta \mid y_{1:T})$. The latent states are thus viewed as auxiliary variables. This opens up for using Gibbs sampling (see Section 2.2), for instance by initializing $\theta[0] \in \Theta$ and iterating;

(i) Draw $x_{1:T}[r] \sim p(x_{1:T} \mid \theta[r-1], y_{1:T})$;
(ii) Draw $\theta[r] \sim p(\theta \mid x_{1:T}[r], y_{1:T})$.

Under weak assumptions, this procedure will generate a Markov chain $\{\theta[r], x_{1:T}[r]\}_{r \geq 1}$ with stationary distribution $p(\theta, x_{1:T} \mid y_{1:T})$. Consequently, the stationary distribution of the subchain $\{\theta[r]\}_{r \geq 1}$ will be the

marginal parameter posterior distribution $p(\theta \mid y_{1:T})$. Note that Step (i) of the above sampling scheme requires the computation of the JSD, for a fixed value of the parameter θ. That is, we need to address an intermediate smoothing problem in order to implement this Gibbs sampler.

Data augmentation is commonly used also in the frequentistic setting. Assume that we, instead of the posterior distribution, seek the maximum likelihood estimator (MLE),

$$\widehat{\theta}_{\mathrm{ML}} = \arg\max_{\theta \in \Theta} \log p_\theta(y_{1:T}), \tag{1.4}$$

where $p_\theta(y_{1:T})$ is the likelihood of the observed data for a given value of the system parameter θ. Again, since the log-likelihood $\log p_\theta(y_{1:T})$ is not available in closed form, direct maximization in Equation (1.4) is problematic. Instead, we can make use of the expectation maximization (EM) algorithm [33] (see also [100]). The EM algorithm is an iterative method, which maximizes $p_\theta(y_{1:T})$ by iteratively maximizing an auxiliary quantity,

$$Q(\theta, \theta') = \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta'}(x_{1:T} \mid y_{1:T})\, dx_{1:T}. \tag{1.5}$$

The EM algorithm is useful when maximization of $\theta \mapsto Q(\theta, \theta')$, for fixed θ', is simpler than direct maximization of the log-likelihood, $\theta \mapsto \log p_\theta(y_{1:T})$. The procedure is initialized at some $\theta[0] \in \Theta$ and then iterates between two steps, expectation (E) and maximization (M);

(E) Compute $Q(\theta, \theta[r-1])$;
(M) Compute $\theta[r] = \arg\max_{\theta \in \Theta} Q(\theta, \theta[r-1])$.

The resulting sequence $\{\theta[r]\}_{r \geq 0}$ will, under weak assumptions, converge to a stationary point of the likelihood $p_\theta(y_{1:T})$ [148].

Using the conditional independence properties of an SSM, we can write the complete data log-likelihood as

$$\begin{aligned} &\log p_\theta(x_{1:T}, y_{1:T}) \\ &\quad = \log \mu_\theta(x_1) + \sum_{t=1}^{T} \log g_\theta(y_t \mid x_t) + \sum_{t=1}^{T-1} \log f_\theta(x_{t+1} \mid x_t). \end{aligned} \tag{1.6}$$

From Equation (1.5), we note that the auxiliary quantity is defined as the expectation of expression (1.6) under the JSD. Hence, to carry out the E-step of the EM algorithm, we again need to address an intermediate smoothing problem for fixed values of the system parameters.

1.6 Smoothing Recursions

As noted above, the JSD is a quantity of central interest for learning and inference problems in SSMs. It summarizes all the information about the latent states which is available in the observations. Many densities that arise in various state inference problems are given as marginals of the JSD. There are a few that are of particular interest, which we summarize in Table 1.1. To avoid a cluttered notation, we now drop the (possible) dependence on an unknown parameter θ from the notation and write the JSD as $p(x_{1:T} \mid y_{1:T})$.

As in Equation (1.6), the conditional independence properties of an SSM implies that the complete data likelihood can be written as,

$$p(x_{1:T}, y_{1:T}) = \mu(x_1) \prod_{t=1}^{T} g(y_t \mid x_t) \prod_{t=1}^{T-1} f(x_{t+1} \mid x_t). \tag{1.7}$$

The JSD is related to the above expression by Bayes' rule,

$$p(x_{1:T} \mid y_{1:T}) = \frac{p(x_{1:T}, y_{1:T})}{\int p(x_{1:T}, y_{1:T})\, dx_{1:T}}. \tag{1.8}$$

Despite the simplicity of this expression, it is of limited use in practice due to the high-dimensional integration needed to compute the normalization factor in the denominator. Instead, most practical methods

Table 1.1 Filtering and smoothing densities of particular interest.

	Density
Filtering[a]	$p(x_t \mid y_{1:t})$
Joint smoothing	$p(x_{1:T} \mid y_{1:T})$
Marginal smoothing ($t \leq T$)	$p(x_t \mid y_{1:T})$
Fixed-interval smoothing ($s < t \leq T$)	$p(x_{s:t} \mid y_{1:T})$
Fixed-lag smoothing (ℓ fixed)[a]	$p(x_{t-\ell+1:t} \mid y_{1:t})$

[a] The filtering and fixed-lag smoothing densities are marginals of the JSD at time t, $p(x_{1:t} \mid y_{1:t})$.

(and in particular the ones discussed in this tutorial) are based on a recursive evaluation of the JSD.

Again by using Bayes' rule, we get the following two-step procedure,

$$p(x_{1:t} \mid y_{1:t}) = \frac{g(y_t \mid x_t)p(x_{1:t} \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}, \tag{1.9a}$$

$$p(x_{1:t+1} \mid y_{1:t}) = f(x_{t+1} \mid x_t)p(x_{1:t} \mid y_{1:t}). \tag{1.9b}$$

The above equations will be denoted as the forward recursion for the JSD, since they evolve forward in time. Step (1.9a) is often referred to as the measurement update, since the current measurement y_t is taken into account. Step (1.9b) is known as the time update, moving the density forward in time, from t to $t+1$.

An interesting fact about SSMs is that, conditioned on $y_{1:T}$, the state process $\{x_t\}_{t=1}^T$ is an inhomogeneous Markov process. Under weak assumptions (see [23, Section 3.3.2] for details), the same holds true for the time-reversed chain, starting at time T and evolving backward in time according to the so-called backward kernel,

$$B_t(A \mid x_{t+1}) \triangleq P(x_t \in A \mid x_{t+1}, y_{1:T}). \tag{1.10}$$

Note that the backward kernel is time inhomogeneous. In the general case, it is not possible to give an explicit expression for the backward kernel. However, for a fully dominated model, this can always be done, and its density is given by

$$p(x_t \mid x_{t+1}, y_{1:T}) = \frac{f(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})}{\int f(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})\,dx_t}. \tag{1.11}$$

From the conditional independence properties of an SSM, it also holds that $p(x_t \mid x_{t+1}, y_{1:T}) = p(x_t \mid x_{t+1}, y_{1:t})$.

Using the backward kernel, we get an alternative recursion for the JSD, evolving backward in time,

$$p(x_{t:T} \mid y_{1:T}) = p(x_t \mid x_{t+1}, y_{1:t})p(x_{t+1:T} \mid y_{1:T}), \tag{1.12}$$

starting with the filtering density at time T, $p(x_T \mid y_{1:T})$. This is known as the backward recursion. At time $t = 1$, the JSD for the time interval $1, \ldots, T$ is obtained.

The backward kernel density at time t depends only on the transition density $f(x_{t+1} \mid x_t)$ and on the filtering density $p(x_t \mid y_{1:t})$, a property which is of key relevance. Hence, to utilise the backward recursion (Equation (1.12)) for computing the JSD, the filtering densities must first be computed for $t = 1, \ldots, T$. Consequently, this procedure is generally called forward filtering/backward smoothing.

1.7 Backward Simulation in Linear Gaussian SSMs

An important special case of Equation (1.2) is the class of linear Gaussian state-space models. A functional representation of an LGSS model is given by,

$$x_{t+1} = Ax_t + v_t, \qquad v_t \sim \mathcal{N}(0, Q), \tag{1.13a}$$

$$y_t = Cx_t + e_t, \qquad e_t \sim \mathcal{N}(0, R). \tag{1.13b}$$

Here, y_t is an n_y-dimensional vector of observations, x_t is an n_x-dimensional state vector and the system matrices A and C are of appropriate dimensions. The process and measurement noises are multivariate Gaussian with zero means and covariances Q and R, respectively.

Example 1.3 (Partially or fully dominated SSM). Assume that the measurement noise covariance R in Equation (1.13b) is full rank. Then, the observation kernel is Gaussian and dominated by Lebesgue measure. Hence, the model is partially dominated. If, in addition, the process noise covariance Q in Equation (1.13a) is full rank, then the transition kernel is also Gaussian and dominated by Lebesgue measure. In this case, the model is fully dominated.

However, for singular Q the model is degenerate (i.e., not fully dominated). Rank deficient process noise covariances arise in many applications, for instance if there is a physical connection between some of the states (such as between position and velocity).

A fully dominated LGSS model can equivalently be expressed as in Equation (1.2) with,

$$f(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1}; Ax_t, Q), \tag{1.14a}$$

$$g(y_t \mid x_t) = \mathcal{N}(y_t; Cx_t, R). \tag{1.14b}$$

LGSS models are without doubt one of the most important and well-studied classes of SSMs. There are basically two reasons for this. First, LGSS models provide sufficiently accurate descriptions of many interesting dynamical systems. Second, LGSS models are one of the few model classes, simple enough to allow for a fully analytical treatment.

When addressing inferential problems for SSMs, we are often asked to generate samples from the JSD, typically as part of an MCMC sampler used to learn a model of the system dynamics, as discussed above. For an LGSS model, the JSD is Gaussian and it can be computed using Kalman filtering and smoothing techniques (see e.g., [80]). Hence, we can make use of standard results for Gaussian distributions to generate a sample from $p(x_{1:T} \mid y_{1:T})$. This is possible for small T, but for increasing T it soon becomes infeasible due to the large matrix inversions involved.

To address this issue, it was recognized by [24, 56] that we can instead use the backward recursion (Equation (1.12)). It follows that the JSD can be factorized as,

$$p(x_{1:T} \mid y_{1:T}) = \left(\prod_{t=1}^{T-1} p(x_t \mid x_{t+1}, y_{1:t}) \right) p(x_T \mid y_{1:T}). \tag{1.15}$$

Initially, we generate a sample from the filtering density at time T,

$$\widetilde{x}_T \sim p(x_T \mid y_{1:T}). \tag{1.16a}$$

We then, successively, augment this *backward trajectory* by generating samples from the backward kernel,

$$\widetilde{x}_t \sim p(x_t \mid \widetilde{x}_{t+1}, y_{1:t}), \tag{1.16b}$$

for $t = T-1, \ldots, 1$. After a complete backward sweep, the backward trajectory $\widetilde{x}_{1:T}$ is (by construction) a realization from the JSD (Equation (1.15)).

To compute the backward kernel, we first run a forward filter to find the filtering densities $p(x_t \mid y_{1:t})$ for $t = 1, \ldots, T$. For an LGSS model, this is done by a standard Kalman filter [81]. It follows that the filtering densities are Gaussian according to,

$$p(x_t \mid y_{1:t}) = \mathcal{N}(x_t; \hat{x}_{t|t}, P_{t|t}), \tag{1.17}$$

for some tractable sequences of mean vectors $\{\hat{x}_{t|t}\}_{t\geq 1}$ and covariance matrices $\{P_{t|t}\}_{t\geq 1}$, respectively. From Equation (1.14a), we note that the transition density function is Gaussian and affine in x_t. Using Equations (1.11) and (1.17) and standard results on affine transformations of Gaussian variables, it then follows that

$$p(x_t \mid x_{t+1}, y_{1:t}) = \mathcal{N}(x_t; \mu_t, M_t), \tag{1.18a}$$

with

$$\mu_t = \hat{x}_{t|t} + P_{t|t}A^{\mathsf{T}}(Q + AP_{t|t}A^{\mathsf{T}})^{-1}(x_{t+1} - A\hat{x}_{t|t}), \tag{1.18b}$$
$$M_t = P_{t|t} - P_{t|t}A^{\mathsf{T}}(Q + AP_{t|t}A^{\mathsf{T}})^{-1}AP_{t|t}. \tag{1.18c}$$

Note that, if more than one sample is desired, multiple backward trajectories can be generated independently, without having to rerun the forward Kalman filter. We illustrate the backward simulator in the example below.

Example 1.4. To illustrate the possibility of generating samples from the JSD using backward simulation, we consider a first-order LGSS model,

$$x_{t+1} = 0.9x_t + v_t, \qquad v_t \sim \mathcal{N}(0, 0.1),$$
$$y_t = x_t + e_t, \qquad e_t \sim \mathcal{N}(0, 1),$$

and $x_1 \sim \mathcal{N}(x_1; 0, 10)$. We simulate $T = 50$ samples $y_{1:T}$ from the model. Since the model is linear Gaussian, the marginal smoothing densities $p(x_t \mid y_{1:T})$ can be computed by running a Kalman filter followed by a Rauch–Tung–Striebel smoother [119]. However, we can also generate samples from the JSD $p(x_{1:T} \mid y_{1:T})$ by running a backward simulator. We simulate $M = 5\,000$ independent trajectories $\{\widetilde{x}^j_{1:T}\}_{j=1}^M$, by first running a Kalman filter and then repeating the backward simulation procedure given by Equations (1.16) and (1.18) M times. Histograms over the simulated states at three specific time points, $t = 1$, $t = 25$ and $t = 50$, are given in Figure 1.3. As expected, the histograms are in close agreement with the true marginal smoothing distributions.

The strategy given by Equation (1.16), i.e., to sequentially sample (either exactly or approximately) from the backward kernel to generate a realization from the JSD, is what we collectively refer to as

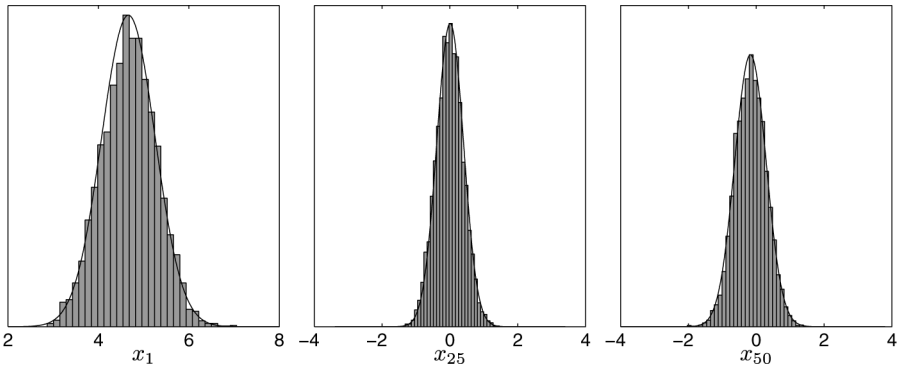

Fig. 1.3 Histograms of $\{\widetilde{x}_t^j\}_{j=1}^M$ for $t = 1$, $t = 25$ and $t = 50$ (from left to right). The true marginal smoothing densities $p(x_t \mid y_{1:T})$ are shown as black lines.

backward simulation. We will now leave the world of LGSS models. In the remainder of this tutorial we address backward simulation for general nonlinear/non-Gaussian models. In these cases, the backward kernels will in general not be available in closed form. Instead, we will rely on SMC approximations of the kernels to carry out the backward simulation.

Before we leave this section, it should be noted that the backward simulator for LGSS models derived here is provided primarily to illustrate the concept. For LGSS models, more efficient samplers exist, e.g., based on disturbance simulation. See [30, 47, 146] for further details and extensions.

1.8 Outline

The rest of this tutorial is organized as follows. Section 2 reviews the two main Monte Carlo methods that are used throughout SMC and MCMC. The section is self-contained, but for obvious reasons it does not provide an in-depth coverage of these methods. Several references which may be useful for readers with no background in this area are given in Section 2.

Section 3 addresses SMC-based backward simulation for SSMs. The focus in this section is on smoothing in general nonlinear/non-Gaussian SSMs. More precisely, we discuss algorithms for generating

state trajectories, approximately distributed according to the joint smoothing distribution. These algorithms can be categorized as *particle smoothers*. Hence, readers with particular interest in smoothing problems may want to focus their attention on this section. However, smoothing is also addressed in Section 5 (see in particular Section 5.7), and the methods presented there can be useful alternatives to the particle smoothers discussed in Section 3.

Section 4 generalizes the backward simulation idea to latent variable models outside the class of SSMs. A general backward simulator is introduced and we discuss its properties and the type of models for which it is applicable. As a special case of the general backward simulator, we derive a Rao–Blackwellized particle smoother for conditionally linear Gaussian SSMs.

In Section 5, we discuss backward simulation in the context of so-called particle MCMC (PMCMC) methods. The focus in this section is on parameter inference, primarily in the Bayesian setting, but we also discuss PMCMC for maximum-likelihood-based inference. As mentioned above, the smoothing problem is also addressed. Finally, in Section 6 we conclude with a discussion about the various methods reviewed throughout this tutorial and outline possible directions for future work.

References

[1] D. Aldous, "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII–1983*, (P. L. Hennequin, ed.), Springer, 1985.

[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, no. 1, pp. 5–43, 2003.

[3] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B*, vol. 72, no. 3, pp. 269–342, 2010.

[4] C. Andrieu and S. J. Godsill, "A particle filter for model based audio source separation," in *Proceedings of the 2000 International Workshop on Independent Component Analysis and Blind Signal Separation (ICA)*, Helsinki, Finland, June 2000.

[5] C. Andrieu, E. Moulines, and P. Priouret, "Stability of stochastic approximation under verifiable conditions," *SIAM Journal on Control and Optimization*, vol. 44, no. 1, pp. 283–312, 2005.

[6] C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient Monte Carlo computations," *The Annals of Statistics*, vol. 37, no. 2, pp. 697–725, 2009.

[7] C. Andrieu and M. Vihola, "Markovian stochastic approximation with expanding projections," arXiv.org, arXiv:1111.5421, November 2011.

[8] C. Andrieu and M. Vihola, "Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms," arXiv.org, arXiv:1210.1484, October 2012.

[9] C. E. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The Annals of Statistics*, vol. 2, no. 6, pp. 1152–1174, 1974.

[10] 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.

[11] M. A. Beaumont, "Estimation of population growth or decline in genetically monitored populations," *Genetics*, vol. 164, no. 3, pp. 1139–1160, 2003.

[12] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York, USA: Springer-Verlag, 1990.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics. New York, USA: Springer, 2006.

[14] D. Blackwell and J. B. MacQueen, "Ferguson distributions via Polya urn schemes," *The Annals of Statistics*, vol. 1, no. 2, pp. 353–355, 1973.

[15] A. Blake, P. Kohli, and C. Rother, eds., *Markov Random Fields For Vision And Image Processing*. MIT Press, 2011.

[16] A. Bouchard-Côté, S. Sankararaman, and M. I. Jordan, "Phylogenetic inference via sequential Monte Carlo," *Systematic Biology*, vol. 61, no. 4, pp. 579–593, 2012.

[17] Y. Bresler, "Two-filter formulae for discrete-time non-linear bayesian smoothing," *International Journal of Control*, vol. 43, no. 2, pp. 629–641, 1986.

[18] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, February 2010.

[19] M. Briers, A. Doucet, and S. S. Singh, "Sequential auxiliary particle belief propagation," in *Proceedings of the International Conference on Information Fusion (FUSION)*, July 2005.

[20] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, eds., *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC, 2011.

[21] P. Bunch and S. Godsill, "Improved particle approximations to the joint smoothing distribution using Markov chain Monte Carlo," *IEEE Transactions on Signal Processing (submitted)*, 2012.

[22] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[23] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer, 2005.

[24] C. K. Carter and R. Kohn, "On Gibbs sampling for state space models," *Biometrika*, vol. 81, no. 3, pp. 541–553, 1994.

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

[26] R. Chen, X. Wang, and J. S. Liu, "Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2079–2094, 2000.

[27] M. Chesney and L. Scott, "Pricing European currency options: A comparison of the modified Black-Scholes model and a random variance model," *The Journal of Financial and Quantitative Analysis*, vol. 24, no. 3, pp. 267–284, 1989.

[28] N. Chopin, "Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference," *The Annals of Statistics*, vol. 32, no. 6, pp. 2385–2411, 2004.

[29] N. Chopin and S. S. Singh, "On the particle Gibbs sampler," arXiv.org, arXiv:1304.1887, April 2013.

[30] P. de Jong and N. Shephard, "The simulation smoother for time series models," *Biometrika*, vol. 82, no. 2, pp. 339–350, 1995.

[31] P. Del Moral, *Feynman-Kac Formulae — Genealogical and Interacting Particle Systems with Applications*, Probability and its Applications. Springer, 2004.

[32] B. Delyon, M. Lavielle, and E. Moulines, "Convergence of a stochastic approximation version of the EM algorithm," *The Annals of Statistics*, vol. 27, no. 1, pp. 94–128, 1999.

[33] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[34] S. Donnet and A. Samson, "EM algorithm coupled with particle filter for maximum likelihood parameter estimation of stochastic differential mixed-effects models," Technical Report hal-00519576, v2, Université Paris Descartes, MAP5, 2011.

[35] R. Douc, A. Garivier, E. Moulines, and J. Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models," *Annals of Applied Probability*, vol. 21, no. 6, pp. 2109–2145, 2011.

[36] R. Douc and E. Moulines, "Limit theorems for weighted samples with applications to sequential Monte Carlo," *The Annals of Statistics*, vol. 36, no. 5, pp. 2344–2376, 2008.

[37] R. Douc, E. Moulines, and J. Olsson, "Optimality of the auxiliary particle filter," *Probability and Mathematical Statistics*, vol. 29, pp. 1–28, 2009.

[38] A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. New York, USA: Springer Verlag, 2001.

[39] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[40] A. Doucet, S. J. Godsill, and M. West, "Monte Carlo filtering and smoothing with application to time-varying spectral estimation," in *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.

[41] A. Doucet and A. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, (D. Crisan and B. Rozovsky, eds.), Oxford University Press, 2011.

[42] A. Doucet, A. M. Johansen, and V. B. Tadić, "On solving integral equations using Markov chain Monte Carlo methods," *Applied Mathematics and Computation*, vol. 216, pp. 2869–2880, 2010.

[43] A. Doucet, M. K. Pitt, and R. Kohn, "Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator," arXiv.org, arXiv:1210.1871, October 2012.

[44] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216–222, 1987.

[45] C. Dubarry and S. L. Corff, "Non-asymptotic deviation inequalities for smoothed additive functionals in non-linear state-space models," arXiv.org, arXiv:1012.4183v2, April 2012.

[46] C. Dubarry and R. Douc, "Particle approximation improvement of the joint smoothing distribution with on-the-fly variance estimation," arXiv.org, arXiv:1107.5524, July 2011.

[47] J. Durbin and S. J. Koopman, "A simple and efficient simulation smoother for state space time series analysis," *Biometrika*, vol. 89, no. 3, pp. 603–616, 2002.

[48] P. Fearnhead, "Particle filters for mixture models with an unknown number of components," *Statistics and Computing*, vol. 14, pp. 11–21, 2004.

[49] P. Fearnhead, "Using random quasi-Monte-Carlo within particle filters, with application to financial time series," *Journal of Computational and Graphical Statistics*, vol. 14, no. 4, pp. 751–769, 2005.

[50] P. Fearnhead, O. Papaspiliopoulos, and G. O. Roberts, "Particle filters for partially observed diffusions," *Journal of the Royal Statistical Society: Series B*, vol. 70, no. 4, pp. 755–777, 2008.

[51] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[52] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[53] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 438–449, February 2002.

[54] G. Fort and E. Moulines, "Convergence of the Monte Carlo expectation maximization for curved exponential families," *The Annals of Statistics*, vol. 31, no. 4, pp. 1220–1259, 2003.

[55] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle MCMC," arXiv.org, arXiv:1306.2861, June 2013.

[56] S. Frühwirth-Schnatter, "Data augmentation and dynamic linear models," *Journal of Time Series Analysis*, vol. 15, no. 2, pp. 183–202, 1994.

[57] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990.

[58] 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, no. 6, pp. 721–741, 1984.

[59] R. Gerlach, C. Carter, and R. Kohn, "Efficient Bayesian inference for dynamic mixture models," *Journal of the American Statistical Association*, vol. 95, no. 451, pp. 819–828, 2000.

[60] W. R. Gilks and C. Berzuini, "Following a moving target — Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[61] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 2, pp. 1–37, 2011.

[62] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, March 2004.

[63] A. Golightly and D. J. Wilkinson, "Bayesian inference for nonlinear multivariate diffusion models observed with error," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1674–1693, 2008.

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

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

[66] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[67] J. Hall, M. K. Pitt, and R. Kohn, "Bayesian inference for nonlinear structural time series models," arXiv.org, arXiv:1209.0253v2, September 2012.

[68] F. Hamze and N. de Freitas, "From fields to trees," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.

[69] F. Hamze, J.-N. Rivasseau, and N. de Freitas, "Information theory tools to rank MCMC algorithms on probabilistic graphical models," in *Proceedings of the UCSD Information Theory Workshop*, 2006.

[70] J. Handschin and D. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *International Journal of Control*, vol. 9, no. 5, pp. 547–559, May 1969.

[71] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[72] S. E. Hills and A. F. M. Smith, "Parameterization issues in Bayesian inference (with discussion)," in *Bayesian Statistics 4*, (J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, eds.), pp. 641–649, Oxford University Press, 1992.

[73] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, eds., *Bayesian Nonparametrics*. Cambridge University Press, 2010.

[74] M. Hoffman, N. de Freitas, A. Doucet, and J. Peters, "An expectation maximization algorithm for continuous Markov decision processes with arbitrary rewards," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, Clearwater Beach, FL, USA, 2009.

[75] M. Hoffman, H. Kueck, N. de Freitas, and A. Doucet, "New inference strategies for solving Markov decision processes using reversible jump MCMC," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 223–231, Corvallis, OR, USA, 2009.

[76] A. Ihler and D. McAllester, "Particle belief propagation," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, Clearwater Beach, FL, USA, 2009.

[77] M. Isard and A. Blake, "Condensation — conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[78] H. Ishwaran, "Applications of hybrid Monte Carlo to Bayesian generalized linear models: Quasicomplete separation and neural networks," *Journal of Computational and Graphical Statistics*, vol. 8, no. 4, pp. 779–799, 1999.

[79] A. M. Johansen and A. Doucet, "A note on auxiliary particle filters," *Statistics & Probability Letters*, vol. 78, no. 12, pp. 1498–1504, 2008.

[80] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ, USA: Prentice Hall, 2000.

[81] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[82] R. Karlsson and F. Gustafsson, "Particle filter for underwater navigation," in *Proceedings of the 2003 IEEE Workshop on Statistical Signal Processing (SSP)*, pp. 509–512, St. Louis, USA, September 2003.

[83] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[84] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: if I had a million particles," in *Proceedings of the International Conference on Machine Learning*, Pittsburgh, USA, June 2006.

[85] M. Klaas, D. Lang, and N. de Freitas, "Fast maximum a posteriori inference in Monte Carlo state spaces," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[86] E. Kuhn and M. Lavielle, "Coupling a stochastic approximation version of EM with an MCMC procedure," *ESAIM: Probability and Statistics*, vol. 8, pp. 115–131, 2004.

[87] H. R. Künsch, "Recursive Monte Carlo filters: Algorithms and theoretical analysis," *The Annals of Statistics*, vol. 33, no. 5, pp. 1983–2021, 2005.

[88] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

[89] D. Lang and N. de Freitas, "Beat tracking the graphical model way," in *Proceedings of the 2004 Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2004.

[90] E. L. Lehmann and G. Casella, *Theory of Point Estimation*. Springer Texts in Statistics. New York, USA: Springer, 2nd ed., 1998.

[91] F. Lindsten, "An efficient stochastic approximation EM algorithm using conditional particle filters," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[92] F. Lindsten, P. Bunch, S. J. Godsill, and T. B. Schön, "Rao-Blackwellized particle smoothers for mixed linear/nonlinear state-space models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[93] F. Lindsten, M. I. Jordan, and T. B. Schön, "Ancestor sampling for particle Gibbs," in *Advances in Neural Information Processing Systems 25*, (P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 2600–2608, 2012.

[94] F. Lindsten and T. B. Schön, "On the use of backward simulation in the particle Gibbs sampler," in *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.

[95] F. Lindsten, T. B. Schön, and M. I. Jordan, "Bayesian semiparametric Wiener system identification," *Automatica*, vol. 49, no. 7, pp. 2053–2063, 2013.

[96] F. Lindsten, T. B. Schön, and J. Olsson, "An explicit variance reduction expression for the Rao-Blackwellised particle filter," in *Proceedings of the 18th IFAC World Congress*, Milan, Italy, August 2011.

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

[98] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[99] S. N. MacEachern, M. Clyde, and J. S. Liu, "Sequential importance sampling for nonparametric Bayes models: The next generation," *The Canadian Journal of Statistics*, vol. 27, no. 2, pp. 251–267, 1999.

[100] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Whiley Series in Probability and Statistics. New York, USA: John Wiley & Sons, second ed., 2008.

[101] A. Melino and S. M. Turnbull, "Pricing foreign currency options with stochastic volatility," *Journal of Econometrics*, vol. 45, no. 1–2, pp. 239–265, 1990.

[102] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[103] N. Metropolis and S. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.

[104] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. Cambridge University Press, 2nd ed., 2009.

[105] M. Montemerlo and S. Thrun, *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*. Berlin, Germany: Springer, 2007.

[106] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002.

[107] P. D. Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 3, pp. 411–436, 2006.

[108] L. M. Murray, E. M. Jones, and J. Parslow, "On collapsed state-space models and the particle marginal Metropolis-Hastings sampler," arXiv.org, arXiv:1202.6159v1, February 2012.

[109] R. M. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, (S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, eds.), pp. 113–162, Chapman & Hall/CRC, 2011.

[110] R. M. Neal, M. J. Beal, and S. T. Roweis, "Inferring state sequences for non-linear systems with embedded hidden Markov models," in *Proceedings of the 2003 Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2003.

[111] M. L. A. Netto, L. Gimeno, and M. J. Mendes, "A new spline algorithm for non-linear filtering of discrete time systems," in *Proceedings of the 7th Triennial World Congress*, pp. 2123–2130, Helsinki, Finland, 1979.

[112] J. Olsson, R. Douc, O. Cappé, and E. Moulines, "Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models," *Bernoulli*, vol. 14, no. 1, pp. 155–179, 2008.

[113] J. Olsson and T. Rydén, "Rao-Blackwellization of particle Markov chain Monte Carlo methods using forward filtering backward sampling," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4606–4619, 2011.

[114] O. Papaspiliopoulos, G. O. Roberts, and M. Sköld, "Non-centered parameterisations for hierarchical models and data augmentation," in *Bayesian Statistics 7*, (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, eds.), pp. 307–326, Oxford University Press, 2003.

[115] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference*. San Francisco, CA, USA: Morgan Kaufmann, 2nd ed., 1988.

[116] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[117] M. K. Pitt, R. S. Silva, P. Giordani, and R. Kohn, "On some properties of Markov chain Monte Carlo simulation methods based on the particle filter," *Journal of Econometrics*, vol. 171, pp. 134–151, 2012.

[118] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[119] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, August 1965.

[120] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.

[121] G. O. Roberts and S. K. Sahu, "Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler," *Journal of the Royal Statistical Society: Series B*, vol. 59, no. 2, pp. 291–317, 1997.

[122] D. B. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, June 1987. Comment to Tanner and Wong: The Calculation of Posterior Distributions by Data Augmentation.

[123] S. Särkkä, P. Bunch, and S. Godsill, "A backward-simulation based Rao-Blackwellized particle smoother for conditionally linear Gaussian models," in *Proceedings of the 16th IFAC Symposium on System Identification*, Brussels, Belgium, July 2012.

[124] M. N. Schmidt, "Function factorization using warped Gaussian processes," in *Proceedings of the International Conference on Machine Learning*, pp. 921–928, 2009.

[125] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, July 2005.

[126] T. B. Schön and F. Lindsten, *Computational Learning in Dynamical Systems*. 2013. (forthcoming, draft manuscript is available from the authors).

[127] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.

[128] J. C. Spall, "Estimation via Markov chain Monte Carlo," *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 34–45, 2003.

[129] L. Stewart and P. McCarty, "The use of Bayesian belief networks to fuse continuous and discrete information for target recognition, tracking, and situation assessment," in *Proceedings of the SPIE 1699, Signal Processing, Sensor Fusion, and Target Recognition*, 1992.

[130] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *Communications of the ACM*, vol. 53, no. 10, pp. 95–103, 2010.

[131] E. Taghavi, F. Lindsten, L. Svensson, and T. B. Schön, "Adaptive stopping for fast particle smoothing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[132] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 528–540, June 1987.

[133] Y. W. Teh, H. Daumé III, and D. Roy, "Bayesian agglomerative clustering with coalescents," *Advances in Neural Information Processing*, pp. 1473–1480, 2008.

[134] L. Tierney, "Markov chains for exploring posterior distributions," *The Annals of Statistics*, vol. 22, no. 4, pp. 1701–1728, 1994.

[135] M. Toussaint and A. Storkey, "Probabilistic inference for solving discrete and continuous state Markov decision processes," in *Proceedings of the International Conference on Machine Learning*, Pittsburgh, PA, USA, 2006.

[136] D. A. van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, March 2001.

[137] D. A. Van Dyk and T. Park, "Partially collapsed Gibbs samplers: Theory and methods," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 790–796, 2008.

[138] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[139] X. Wang, R. Chen, and D. Guo, "Delayed-pilot sampling for mixture Kalman filter with application in fading channels," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 241–254, 2002.

[140] G. C. G. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[141] N. Whiteley, "Discussion on Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B*, vol. 72, no. 3, pp. 306–307, 2010.

[142] N. Whiteley, C. Andrieu, and A. Doucet, "Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods," Technical report, Bristol Statistics Research Report 10:04, 2010.

[143] N. Whiteley, C. Andrieu, and A. Doucet, "Bayesian computational methods for inference in multiple change-points models," *Submitted*, 2011.

[144] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, pp. 65–85, 1994.

[145] P. Wild and W. R. Gilks, "Algorithm AS 287: Adaptive rejection sampling from log-concave density functions," *Journal of the Royal Statistical Society: Series C*, vol. 42, no. 4, pp. 701–709, 1993.

[146] D. J. Wilkinson and S. K. H. Yeung, "Conditional simulation from highly structured Gaussian systems, with application to blocking-MCMC for the Bayesian analysis of very large linear models," *Statistics and Computing*, vol. 12, no. 3, pp. 287–300, July 2002.

[147] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, "Identification of Hammerstein–Wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[148] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.