Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond

ZHE CHEN

Abstract— *In this self-contained survey/review paper, we systematically investigate the roots of Bayesian filtering as well as its rich leaves in the literature. Stochastic filtering theory is briefly reviewed with emphasis on nonlinear and non-Gaussian filtering. Following the Bayesian statistics, different Bayesian filtering techniques are developed given different scenarios. Under linear quadratic Gaussian circumstance, the celebrated Kalman filter can be derived within the Bayesian framework. Optimal/suboptimal nonlinear filtering techniques are extensively investigated. In particular, we focus our attention on the Bayesian filtering approach based on sequential Monte Carlo sampling, the so-called particle filters. Many variants of the particle filter as well as their features (strengths and weaknesses) are discussed. Related theoretical and practical issues are addressed in detail. In addition, some other (new) directions on Bayesian filtering are also explored.*

Index Terms— **Stochastic filtering, Bayesian filtering, Bayesian inference, particle filter, sequential Monte Carlo, sequential state estimation, Monte Carlo methods.**

> *"The probability of any event is the ratio between the value at which an expectation depending on the happening of the event ought to be computed, and the value of the thing expected upon its happening."*
> — Thomas Bayes (1702-1761), [29]

> *"Statistics is the art of never having to say you're wrong. Variance is what any two statisticians are at."*
> — C. J. Bradfield

Contents

The work is supported by the Natural Sciences and Engineering Research Council of Canada. Z. Chen was also partially supported by Clifton W. Sherman Scholarship.

The author is with the Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada L8S 4K1, e-mail: zhechen@soma.crl.mcmaster.ca, Tel: (905)525-9140 x27282, Fax:(905)521-2922.

I. Introduction

THE contents of this paper contain three major scientific areas: *stochastic filtering theory, Bayesian theory*, and *Monte Carlo methods*. All of them are closely discussed around the subject of our interest: Bayesian filtering. In the course of explaining this long story, some relevant theories are briefly reviewed for the purpose of providing the reader a complete picture. Mathematical preliminaries and background materials are also provided in detail for the self-containing purpose.

A. Stochastic Filtering Theory

Stochastic filtering theory was first established in the early 1940s due to the pioneering work by Norbert Wiener [487], [488] and Andrey N. Kolmogorov [264], [265], and it culminated in 1960 for the publication of classic Kalman filter (KF) [250] (and subsequent Kalman-Bucy filter in 1961 [249]), [1] though many credits should be also due to some earlier work by Bode and Shannon [46], Zadeh and Ragazzini [502], [503], Swerling [434], Levinson [297], and others. Without any exaggeration, it seems fair to say that the Kalman filter (and its numerous variants) have dominated the adaptive filter theory for decades in signal processing and control areas. Nowadays, Kalman filters have been applied in the various engineering and scientific areas, including communications, machine learning, neuroscience, economics, finance, political science, and many others. Bearing in mind that Kalman filter is limited by its assumptions, numerous nonlinear filtering methods along its line have been proposed and developed to overcome its limitation.

B. Bayesian Theory and Bayesian Filtering

Bayesian theory[2] was originally discovered by the British researcher Thomas Bayes in a posthumous publication in 1763 [29]. The well-known Bayes theorem describes the fundamental probability law governing the process of logical inference. However, Bayesian theory has not gained its deserved attention in the early days until its modern form was rediscovered by the French mathematician Pierre-Simon de Laplace in *Théorie analytique des probailités*.[3] Bayesian inference [38], [388], [375], devoted to applying Bayesian statistics to statistical inference, has become one of the important branches in statistics, and has been applied successfully in statistical decision, detection and estimation, pattern recognition, and machine learning. In particular, the November 19 issue of 1999 SCIENCE magazine has given the Bayesian research boom a four-page special attention [320]. In many scenarios, the solutions gained through Bayesian inference are viewed as "optimal". Not surprisingly, Bayesian theory was also studied in the filtering literature. One of the first exploration of iterative Bayesian estimation is found in Ho and Lee' paper [212], in which they specified the principle and procedure of Bayesian filtering. Sprangins [426] discussed the iterative application of Bayes rule to sequential parameter estimation and called it as "Bayesian learning". Lin and Yau [301] and Chien an Fu [92] discussed Bayesian approach to optimization of adaptive systems. Bucy [62] and Bucy and Senne [63] also explored the point-mass approximation method in the Bayesian filtering framework.

C. Monte Carlo Methods and Monte Carlo Filtering

The early idea of Monte Carlo[4] can be traced back to the problem of *Buffon's needle* when Buffon attempted in 1777 to estimate π (see e.g., [419]). But the modern formulation of Monte Carlo methods started from 1940s in physics [330], [329], [393] and later in 1950s to statistics [198]. During the World War II, John von Neumann, Stanislaw Ulam, Niick Metropolis, and others initialized the Monte Carlo method in Los Alamos Laboratory. von Neumann also used Monte Carlo method to calculate the elements of an inverse matrix, in which they redefined the "Russian roulette" and "splitting" methods [472]. In recent decades, Monte Carlo techniques have been rediscovered independently in statistics, physics, and engineering. Many new Monte Carlo methodologies (e.g. Bayesian bootstrap, hybrid Monte Carlo, quasi Monte Carlo) have been rejuvenated and developed. Roughly speaking, Monte Carlo

[1] Another important event in 1960 is the publication of the celebrated least-mean-squares (LMS) algorithm [485]. However, the LMS filter is not discussed in this paper, the reader can refer to [486], [205], [207], [247] for more information.

[2] A generalized Bayesian theory is the so-called Quasi-Bayesian theory (e.g. [100]) that is built on the convex set of probability distributions and a relaxed set of aximoms about *preferences*, which we don't discuss in this paper.

[3] An interesting history of Thomas Bayes and its famous essay is found in [110].

[4] The method is named after the city in the Monaco principality, because of a roulette, a simple random number generator. The name was first suggested by Stanislaw Ulam.

technique is a kind of stochastic sampling approach aiming to tackle the complex systems which are analytically intractable. The power of Monte Carlo methods is that they can attack the difficult numerical integration problems. In recent years, sequential Monte Carlo approaches have attracted more and more attention to the researchers from different areas, with many successful applications in statistics (see e.g. the March special issue of 2001 ANNALS OF THE INSTITUTE OF STATISTICAL MATHEMATICS), signal processing (see e.g., the February special issue of 2002 IEEE TRANSACTIONS ON SIGNAL PROCESSING), machine learning, econometrics, automatic control, tracking, communications, biology, and many others (e.g., see [141] and the references therein). One of the attractive merits of sequential Monte Carlo approaches lies in the fact that they allow on-line estimation by combining the powerful Monte Carlo sampling methods with Bayesian inference, at an expense of reasonable computational cost. In particular, the sequential Monte Carlo approach has been used in parameter estimation and state estimation, for the latter of which it is sometimes called particle filter.[5] The basic idea of particle filter is to use a number of *independent* random variables called particles,[6] sampled directly from the state space, to represent the posterior probability, and update the posterior by involving the new observations; the "particle system" is properly located, weighted, and propagated recursively according to the Bayesian rule. In retrospect, the earliest idea of Monte Carlo method used in statistical inference is found in [200], [201], and later in [5], [6], [506], [433], [258], but the formal establishment of particle filter seems fair to be due to Gordon, Salmond and Smith [193], who introduced certain novel resampling technique to the formulation. Almost in the meantime, a number of statisticians also independently rediscovered and developed the sampling-importance-resampling (SIR) idea [414], [266], [303], which was originally proposed by Rubin [395], [397] in a non-dynamic framework.[7] The rediscovery and renaissance of particle filters in the mid-1990s (e.g. [259], [222], [229], [304], [307], [143], [40]) after a long dominant period, partially thanks to the ever increasing computing power. Recently, a lot of work has been done to improve the performance of particle filters [69], [189], [428], [345], [456], [458], [357]. Also, many doctoral theses were devoted to Monte Carlo filtering and inference from different perspectives [191], [142], [162], [118], [221], [228], [35], [97], [365], [467], [86].

It is noted that particle filter is not the only leaf in the Bayesian filtering tree, in the sense that Bayesian filtering can be also tackled with other techniques, such as differential geometry approach, variational method, or conjugate method. Some potential future directions, will be considering combining these methods with Monte Carlo sampling techniques, as we will discuss in the paper. The attention of this paper, however, is still on the Monte Carlo methods and particularly sequential Monte Carlo estimation.

D. *Outline of Paper*

In this paper, we present a comprehensive review of stochastic filtering theory from Bayesian perspective. [It happens to be almost three decades after the 1974 publication of Prof. Thomas Kailath's illuminating review paper "A view of three decades of linear filtering theory" [244], we take this opportunity to dedicate this paper to him who has greatly contributed to the literature in stochastic filtering theory.] With the tool of Bayesian statistics, it turns out that the celebrated Kalman filter is a special case of Bayesian filtering under the LQG (linear, quadratic, Gaussian) circumstance, a fact that was first observed by Ho and Lee [212]; particle filters are also essentially rooted in Bayesian statistics, in the spirit of recursive Bayesian estimation. To our interest, the attention will be given to the *nonlinear, non-Gaussian* and *non-stationary* situations where we mostly encounter in the real world. Generally for nonlinear filtering, no exact solution can be obtained, or the solution is infinite-dimensional,[8] hence various numerical approximation methods come in to address the intractability. In particular, we focus our attention on sequential Monte Carlo method which allows on-line estimation in a Bayesian perspective. The historic root and remarks of Monte Carlo filtering are traced. Other Bayesian filtering approaches other than Monte Carlo framework are also reviewed. Besides, we extend our discussion from Bayesian filtering to Bayesian inference, in the latter of which the well-known hidden Markov model (HMM) (a.k.a. HMM filter), dynamic Bayesian networks (DBN) and Bayesian kernel machines are also briefly discussed.

Nowadays Bayesian filtering has become such a broad topic involving many scientific areas that a comprehensive survey and detailed treatment seems crucial to cater the ever growing demands of understanding this important field for many novices, though it is noticed by the author that in the literature there exist a number of excellent tutorial papers on particle filters and Monte Carlo filters [143], [144], [19], [438], [443], as well as relevant edited volumes [141] and books [185], [173], [306], [82]. Unfortunately, as observed in our comprehensive bibliographies, a lot of papers were written by statisticians or physicists with some special terminologies, which might be unfamiliar to many engineers. Besides, the papers were written with different nomenclatures for different purposes (e.g. the convergence and asymptotic results are rarely cared in engineering but are important for the statisticians). The author, thus, felt obligated to write a tutorial paper on this emerging and promising area for the readership of engineers, and to introduce the reader many techniques developed in statistics

[5] Many other terminologies also exist in the literature, e.g., SIS filter, SIR filter, bootstrap filter, sequential imputation, or CONDENSATION algorithm (see [224] for many others), though they are addressed differently in different areas. In this paper, we treat them as different variants within the generic Monte Carlo filter family. Monte Carlo filters are not all sequential Monte Carlo estimation.

[6] The particle filter is called *normal* if it produces i.i.d. samples; sometimes it is deliberately to introduce *negative* correlations among the particles for the sake of variance reduction.

[7] The earliest idea of multiple imputation due to Rubin was published in 1978 [394].

[8] Or the sufficient statistics is infinite-dimensional.

and physics. For this purpose again, for a variety of particle filter algorithms, the basic ideas instead of mathematical derivations are emphasized. The further details and experimental results are indicated in the references. Due to the dual tutorial/review nature of current paper, only few simple examples and simulation are presented to illustrate the essential ideas, no comparative results are available at this stage (see other paper [88]); however, it doesn't prevent us presenting the new thoughts. Moreover, many graphical and tabular illustrations are presented. Since it is also a survey paper, extensive bibliographies are included in the references. But there is no claim that the bibliographies are complete, which is due to the our knowledge limitation as well as the space allowance.

The rest of this paper is organized as follows: In Section II, some basic mathematical preliminaries of stochastic filtering theory are given; the stochastic filtering problem is also mathematically formulated. Section III presents the essential Bayesian theory, particularly Bayesian statistics and Bayesian inference. In Section IV, the Bayesian filtering theory is systematically investigated. Following the simplest LQG case, the celebrated Kalman filter is briefly derived, followed by the discussion of optimal nonlinear filtering. Section V discusses many popular numerical approximation techniques, with special emphasis on Monte Carlo sampling methods, which result in various forms of particle filters in Section VI. In Section VII, some other new Bayesian filtering approaches other than Monte Carlo sampling are also reviewed. Section VIII presents some selected applications and one illustrative example of particle filters. We give some discussions and critiques in Section IX and conclude the paper in Section X.

II. Mathematical Preliminaries and Problem Formulation

A. Preliminaries

Definition 1: Let S be a set and $\mathcal{F}$ be a family of subsets of S. $\mathcal{F}$ is a σ-algebra if (i) $\emptyset \in \mathcal{F}$; (ii) $A \in \mathcal{F}$ implies $A^c \in \mathcal{F}$; (iii) $A_1, A_2, \cdots \in \mathcal{F}$ implies $\cup_{i=1}^{\infty} A_i \in \mathcal{F}$.

A σ-algebra is closed under complement and union of countably infinitely many sets.

Definition 2: A probability space is defined by the elements $\{\Omega, \mathcal{F}, P\}$ where $\mathcal{F}$ is a σ-algebra of Ω and P is a complete, σ-additive probability measure on all $\mathcal{F}$. In other words, P is a set function whose arguments are random events (element of $\mathcal{F}$) such that axioms of probability hold.

Definition 3: Let $p(\mathbf{x}) = \frac{dP(\mathbf{x})}{d\mu}$ denote Radon-Nikodým density of probability distribution $P(\mathbf{x})$ w.r.t. a measure μ. When $\mathbf{x} \in X$ is discrete and μ is a counting measure, $p(\mathbf{x})$ is a probability mass function (pmf); when $\mathbf{x}$ is continuous and μ is a Lebesgue measure, $p(\mathbf{x})$ is a probability density function (pdf).

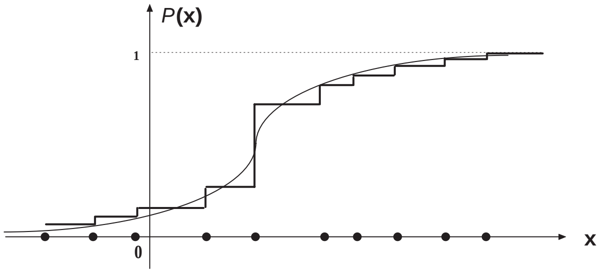

Fig. 1. Empirical probability distribution (density) function constructed from the discrete observations $\{\mathbf{x}^{(i)}\}$.

Intuitively, the true distribution $P(\mathbf{x})$ can be replaced by the *empirical distribution* given the simulated samples (see Fig. 1 for illustration)

$$\hat{P}(\mathbf{x}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

where $\delta(\cdot)$ is a Radon-Nikodým density w.r.t. μ of the point-mass distribution concentrated at the point $\mathbf{x}$. When $\mathbf{x} \in X$ is discrete, $\delta(\mathbf{x} - \mathbf{x}^{(i)})$ is 1 for $\mathbf{x} = \mathbf{x}^{(i)}$ and 0 elsewhere. When $\mathbf{x} \in X$ is continuous, $\delta(\mathbf{x} - \mathbf{x}^{(i)})$ is a Dirac-delta function, $\delta(\mathbf{x} - \mathbf{x}^{(i)}) = 0$ for all $\mathbf{x}^{(i)} \neq \mathbf{x}$, and $\int_X d\hat{P}(\mathbf{x}) = \int_X \hat{p}(\mathbf{x}) d\mathbf{x} = 1$.

B. Notations

Throughout this paper, the bold font is referred to vector or matrix; the subscript symbol t ($t \in \mathbb{R}^+$) is referred to the index in a continuous-time domain; and n ($n \in \mathbb{N}$) is referred to the index in a discrete-time domain. $p(\mathbf{x})$ is referred to the pdf in a Lebesque measure or the pmf in a counting measure. $\mathbb{E}[\cdot]$ and $\mathrm{Var}[\cdot]$ ($\mathrm{Cov}[\cdot]$) are expectation and variance (covariance) operators, respectively. Unless specified elsewhere, the expectations are taken w.r.t. the true pdf. Notations $\mathbf{x}_{0:n}$ and $\mathbf{y}_{0:n}$ [9] are referred to the state and observation sets with elements collected from time step 0 up to n. Gaussian (normal) distribution is denoted by $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. $\mathbf{x}_n$ represents the true state in time step n, whereas $\hat{\mathbf{x}}_n$ (or $\hat{\mathbf{x}}_{n|n}$) and $\hat{\mathbf{x}}_{n|n-1}$ represent the *filtered* state and *predicted* state of $\mathbf{x}_n$, respectively. $\mathbf{f}$ and $\mathbf{g}$ are used to represent vector-valued state function and measurement function, respectively. f is denoted as a generic (vector or scalar valued) nonlinear function. Additional nomenclatures will be given wherever confusion is necessary to clarify.

For the reader's convenience, a complete list of notations used in this paper is summarized in the Appendix G.

C. Stochastic Filtering Problem

Before we run into the mathematical formulation of stochastic filtering problem, it is necessary to clarify some basic concepts:

Filtering is an operation that involves the extraction of information about a quantity of interest at time t by using data measured up to and including t.

[9] Sometimes it is also denoted by $\mathbf{y}_{1:n}$, which differs in the assuming order of state and measurement equations.

Prediction is an *a priori* form of estimation. Its aim is to derive information about what the quantity of interest will be like at some time $t+\tau$ in the future ($\tau > 0$) by using data measured up to and including time t. Unless specified otherwise, prediction is referred to one-step ahead prediction in this paper.

Smoothing is an *a posteriori* form of estimation in that data measured after the time of interest are used for the estimation. Specifically, the smoothed estimate at time t' is obtained by using data measured over the interval $[0,t]$, where $t' < t$.

Now, let us consider the following generic stochastic filtering problem in a dynamic state-space form [238], [422]:

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{d}_t), && (1a) \\ \mathbf{y}_t &= \mathbf{g}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t), && (1b) \end{aligned}$$

where equations (1a) and (1b) are called state equation and measurement equation, respectively; $\mathbf{x}_t$ represents the state vector, $\mathbf{y}_t$ is the measurement vector, $\mathbf{u}_t$ represents the system input vector (as driving force) in a controlled environment; $\mathbf{f} : \mathbb{R}^{N_\mathbf{x}} \mapsto \mathbb{R}^{N_\mathbf{x}}$ and $\mathbf{g} : \mathbb{R}^{N_\mathbf{x}} \mapsto \mathbb{R}^{N_\mathbf{y}}$ are two vector-valued functions, which are potentially time-varying; $\mathbf{d}_t$ and $\mathbf{v}_t$ represent the process (dynamical) noise and measurement noise respectively, with appropriate dimensions. The above formulation is discussed in the continuous-time domain, in practice however, we are more concerned about the discrete-time filtering.[10] In this context, the following practical filtering problem is concerned:[11]

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{d}_n), && (2a) \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n), && (2b) \end{aligned}$$

where $\mathbf{d}_n$ and $\mathbf{v}_n$ can be viewed as white noise random sequences with unknown statistics in the discrete-time domain. The state equation (2a) characterizes the state transition probability $p(\mathbf{x}_{n+1}|\mathbf{x}_n)$, whereas the measurement equation (2b) describes the probability $p(\mathbf{y}_n|\mathbf{x}_n)$ which is further related to the measurement noise model.

The equations (2a)(2b) reduce to the following special case where a linear Gaussian dynamic system is considered:[12]

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{F}_{n+1,n}\mathbf{x}_n + \mathbf{d}_n, && (3a) \\ \mathbf{y}_n &= \mathbf{G}_n\mathbf{x}_n + \mathbf{v}_n, && (3b) \end{aligned}$$

for which the analytic filtering solution is given by the Kalman filter [250], [253], in which the sufficient statistics[13] of mean and state-error correlation matrix are calculated and propagated. In equations (3a) and (3b), $\mathbf{F}_{n+1,n}$, $\mathbf{G}_n$ are called transition matrix and measurement matrix, respectively.

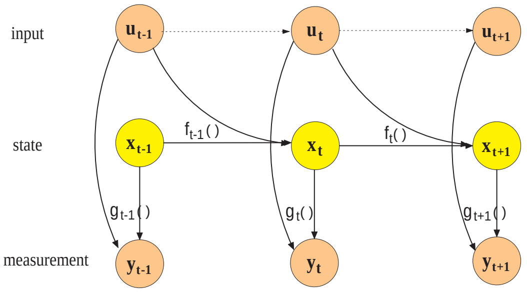

Fig. 2. A graphical model of generic state-space model.

Described as a generic state-space model, the stochastic filtering problem can be illustrated by a graphical model (Fig. 2). Given initial density $p(\mathbf{x}_0)$, transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, and likelihood $p(\mathbf{y}_n|\mathbf{x}_n)$, the objective of the filtering is to estimate the optimal current state at time n given the observations up to time n, which is in essence amount to estimating the posterior density $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ or $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$. Although the posterior density provides a complete solution of the stochastic filtering problem, the problem still remains intractable since the density is a function rather than a finite-dimensional point estimate. We should also keep in mind that most of physical systems are *not* finite dimensional, thus the infinite-dimensional system can only be modeled approximately by a finite-dimensional filter, in other words, the filter can only be suboptimal in this sense. Nevertheless, in the context of nonlinear filtering, it is still possible to formulate the exact finite-dimensional filtering solution, as we will discuss in Section IV.

In Table I, a brief and incomplete development history of stochastic filtering theory (from linear to nonlinear, Gaussian to non-Gaussian, stationary to non-stationary) is summarized. Some detailed reviews are referred to [244], [423], [247], [205].

D. Nonlinear Stochastic Filtering Is an Ill-posed Inverse Problem

D.1 Inverse Problem

Stochastic filtering is an inverse problem: Given collected $\mathbf{y}_n$ at discrete time steps (hence $\mathbf{y}_{0:n}$), provided $\mathbf{f}$ and $\mathbf{g}$ are known, one needs to find the optimal or suboptimal $\hat{\mathbf{x}}_n$. In another perspective, this problem can be interpreted as an inverse mapping learning problem: Find the inputs sequentially with a (composite) mapping function which yields the output data. In contrast to the forward learning (given inputs find outputs) which is a many-to-one mapping problem, the inversion learning problem is one-to-many, in a sense that the mapping from output to input space is generally non-unique.

A problem is said to be well-posed if it satisfies three con-

[10] The continuous-time dynamic system can be always converted into a discrete-time system by sampling the outputs and using "zero-order holds" on the inputs. Hence the derivative will be replaced by the difference, the operator will become a matrix.

[11] For discussion simplicity, no driving-force in the dynamic system (which is often referred to the stochastic control problem) is considered in this paper. However, the extension to the driven system is straightforward.

[12] An excellent and illuminating review of linear filtering theory is found in [244] (see also [385], [435], [61]); for a complete treatment of linear estimation theory, see the classic textbook [247].

[13] Sufficient statistics is referred to a collection of quantities which uniquely determine a probability density in its entirety.

TABLE I

A Development History of Stochastic Filtering Theory.

author(s) (year)	method	solution	comment
Kolmogorov (1941)	innovations	exact	linear, stationary
Wiener (1942)	spectral factorization	exact	linear, stationary, infinite memory
Levinson (1947)	lattice filter	approximate	linear, stationary, finite memory
Bode & Shannon (1950)	innovations, whitening	exact	linear, stationary,
Zadeh & Ragazzini (1950)	innovations, whitening	exact	linear, non-stationary
Kalman (1960)	orthogonal projection	exact	LQG, non-stationary, discrete
Kalman & Bucy (1961)	recursive Riccati equation	exact	LQG, non-stationary, continuous
Stratonovich (1960)	conditional Markov process	exact	nonlinear, non-stationary
Kushner (1967)	PDE	exact	nonlinear, non-stationary
Zakai (1969)	PDE	exact	nonlinear, non-stationary
Handschin & Mayne (1969)	Monte Carlo	approximate	nonlinear, non-Gaussian, non-stationary
Bucy & Senne (1971)	point-mass, Bayes	approximate	nonlinear, non-Gaussian, non-stationary
Kailath (1971)	innovations	exact	linear, non-Gaussian, non-stationary
Beneš (1981)	Beneš	exact solution of Zakai eqn.	nonlinear, finite-dimensional
Daum (1986)	Daum, virtual measurement	exact solution of FPK eqn.	nonlinear, finite-dimensional
Gordon, Salmond, & Smith (1993)	bootstrap, sequential Monte Carlo	approximate	nonlinear, non-Gaussian, non-stationary
Julier & Uhlmann (1997)	unscented transformation	approximate	nonlinear, (non)-Gaussian, derivative-free

ditions: *existence*, *uniqueness* and *stability*, otherwise it is said to be ill posed [87]. In this context, stochastic filtering problem is ill-posed in the following sense: (i) The ubiquitous presence of the unknown noise corrupts the state and measurement equations, given limited noisy observations, the solution is non-unique; (ii) Supposing the state equation is a *diffeomorphism* (i.e. differentiable and regular),[14] the measurement function is possibly a many-to-one mapping function (e.g. $g(\xi) = \xi^2$ or $g(\xi) = \sin(\xi)$, see also the illustrative example in Section VIII-G), which also violates the uniqueness condition; (iii) The filtering problem is per se a conditional posterior distribution (density) estimation problem, which is known to be stochastically ill posed especially in high-dimensional space [463], let alone on-line processing [412].

D.2 Differential Operator and Integral Equation

In what follows, we present a rigorous analysis of stochastic filtering problem in the continuous-time domain. To simplify the analysis, we first consider the simple *irregular* stochastic differential equation (SDE):

$$\frac{d\mathbf{x}_t}{dt} = f(t, \mathbf{x}_t) + \mathbf{d}_t, \quad t \in T \tag{4}$$

where $\mathbf{x}_t$ is a second-order stochastic process, $\boldsymbol{\omega}_t = \int_0^t \mathbf{d}_s ds$ is a Wiener process (Brownian motion) and $\mathbf{d}_t$ can be regarded as a white noise. $f : T \times L_2(\Omega, \mathcal{F}, P) \rightarrow L_2(\Omega, \mathcal{F}, P)$ is a mapping to a (Lebesque square-integrable) Hilbert space $L_2(\Omega, \mathcal{F}, P)$ with finite second-order moments. The solution of (4) is given by the stochastic integral

$$\mathbf{x}_t = \mathbf{x}_0 + \int_0^t f(s, \mathbf{x}_s) ds + \int_0^t d\boldsymbol{\omega}_s, \tag{5}$$

where the second integral is Itô stochastic integral (named after Japanese mathematician Kiyosi Ito [233]).[15]

Mathematically, the ill-posed nature of stochastic filtering problem can be understood from the operator theory.

Definition 4: [274], [87] Let $\boldsymbol{A} : Y \rightarrow X$ be an operator from a normed space Y to X. The equation $\boldsymbol{A}Y = X$ is said to be well posed if $\boldsymbol{A}$ is bijective and the inverse operator $\boldsymbol{A}^{-1} : X \rightarrow Y$ is continuous. Otherwise the equation is called ill posed.

Definition 5: [418] Suppose $\mathbb{H}$ is a Hilbert space and let $\boldsymbol{A} = \boldsymbol{A}(\gamma)$ be a stochastic operator mapping $\Omega \times \mathbb{H}$ in $\mathbb{H}$. Let $X = X(\gamma)$ be a generalized random variable (or function) in $\mathbb{H}$, then

$$\boldsymbol{A}(\gamma) Y = X(\gamma) \tag{6}$$

is a generalized stochastic operator equation for the element $Y \in \mathbb{H}$.

Since γ is an element of a measurable space $(\Omega, \mathcal{F})$ on which a complete probability measure P is defined, stochastic operator equation is a family of equations. The family of equations has a unique member when P is a Dirac measure. Suppose Y is a smooth functional with continuous first n derivatives, then (6) can be written as

$$\boldsymbol{A}(\gamma) Y(\gamma) = \sum_{k=0}^{N} a_k(t, \gamma) \frac{d^k Y}{dt^k} = X(\gamma), \tag{7}$$

which can be represented in the form of stochastic integral equations of Fredholm type or Voltera type [418], with an

[14] Diffeomorphism is referred to a smooth mapping with a smooth inverse, one-to-one mapping.

[15] The Itô stochastic integral is defined as $\int_{t_0}^{t} \sigma(t) d\omega(t) = \lim\limits_{n\to\infty} \left[\sum_{j=1}^{n} \sigma(t_{j-1}) \Delta\omega_j \right]$. The Itô calculus satisfies $d\omega^2(t) = dt$, $d\omega(t)dt = 0$, $dt^{N+1} = d\omega^{N+2}(t) = 0$ ($N > 1$). See [387], [360] for a detailed background about Itô calculus and Itô SDE.

appropriately defined kernel K:

$$Y(t,\gamma) = X(t,\gamma) + \int K(t,\tau,\gamma)Y(\tau,\gamma)d\tau, \qquad (8)$$

which takes a similar form as the continuous-time Wiener-Hopf equation (see e.g. [247]) when K is translation invariant.

Definition 6: [418] Any mapping $Y(\gamma) : \Omega \rightarrow \mathbb{H}$ which satisfies $\boldsymbol{A}(\gamma)Y(\gamma) = X(\gamma)$ for every $\gamma \in \Omega$, is said to be a wide-sense solution of (6).

The wide-sense solution is a stochastic solution if it is measurable w.r.t. P and $\Pr\{\gamma : \boldsymbol{A}(\gamma)Y(\gamma) = X(\gamma)\} = 1$. The *existence* and *uniqueness* conditions of the solution to the stochastic operator equation (6) is given by the probabilistic *Fixed-Point Theorem* [418]. The essential idea of Fixed-Point Theorem is to prove that $\boldsymbol{A}(\gamma)$ is a stochastic *contractive operator*, which unfortunately is not always true for the stochastic filtering problem.

Let's turn our attention to the measurement equation in an integral form

$$\mathbf{y}_t = \int_0^t g(s, \mathbf{x}_s)ds + \mathbf{v}_t, \qquad (9)$$

where $g : \mathbb{R}^{N_\mathbf{x}} \rightarrow \mathbb{R}^{N_\mathbf{y}}$. For any $\phi(\cdot) \in \mathbb{R}^{N_\mathbf{x}}$, the optimal (in mean-square sense) filter $\hat{\phi}(\mathbf{x}_t)$ is the one that seeks an minimum mean-square error, as given by

$$\hat{\phi}(\mathbf{x}_t) \equiv \arg\min\{\|\phi - \hat{\phi}\|^2\} = \frac{\int \pi(\mathbf{x}_t|\mathbf{y}_{0:t})\phi(\mathbf{x})d\mathbf{x}_t}{\int \pi(\mathbf{x}_t|\mathbf{y}_{0:t})d\mathbf{x}_t}, \qquad (10)$$

where $\pi(\cdot)$ is an unnormalized filtering density. A common way to study the unnormalized filtering density is to treat it as a solution of the Zakai equation, as will be detailed in Section II-E.

D.3 Relations to Other Problems

It is conducive to better understanding the stochastic filtering problem by comparing it with many other ill-posed problems that share some commons in different perspectives:

- **System identification**: System identification has many commons with stochastic filtering. Both of them belong to statistical inference problems. Sometimes, identification is also meant as filtering in stochastic control realm, especially with a driving-force as input. However, the measurement equation can admit the feedback of previous output, i.e. $\mathbf{y}_n = \mathbf{g}(\mathbf{x}_n, \mathbf{y}_{n-1}, \mathbf{v}_n)$. Besides, identification is often more concerned about the parameter estimation problem instead of state estimation. We will revisit this issue in the Section IX.
- **Regression**: In some perspective, filtering can be viewed as a sequential linear/nonlinear regression problem if state equation reduces to a random walk. But, regression differs from filtering in the following sense: Regression is aimed to find a deterministic mapping between the input and output given a finite number of observation pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{\ell}$, which is usually off-line; whereas filtering is aimed to sequentially infer the signal or state process given some observations by assuming the knowledge of the state and measurement models.
- **Missing data problem**: Missing data problem is well addressed in statistics, which is concerned about probabilistic inference or model fitting given limited data. Statistical approaches (e.g. EM algorithm, data augmentation) are used to help this goal by assuming auxiliary missing variables (unobserved data) with tractable (on-line or off-line) inference.
- **Density estimation**: Density estimation shares some commons with filtering in that both of them target at a dependency estimation problem. Generally, filtering is nothing but to learn the conditional probability distribution. However, density estimation is more difficult in the sense that it doesn't have any prior knowledge on the data (though sometimes people give some assumption, e.g. mixture distribution) and it usually works directly on the state (i.e. observation process is tantamount to the state process). Most of density estimation techniques are off-line.
- **Nonlinear dynamic reconstruction**: Nonlinear dynamic reconstruction arise from physical phenomena (e.g. sea clutter) in the real world. Given some limited observations (possibly not continuously or evenly recorded), it is concerned about inferring the physically meaningful state information. In this sense, it is very similar to the filtering problem. However, it is much more difficult than the filtering problem in that the nonlinear dynamics involving $\mathbf{f}$ is totally unknown (usually assuming a nonparametric model to estimate) and potentially complex (e.g. chaotic), and the prior knowledge of state equation is very limited, and thereby severely ill-posed [87]. Likewise, dynamic reconstruction allows off-line estimation.

E. Stochastic Differential Equations and Filtering

In the following, we will formulate the continuous-time stochastic filtering problem by SDE theory. Suppose $\{\mathbf{x}_t\}$ is a Markov process with an infinitesimal generator, rewriting state-space equations (1a)(1b) in the following form of Itô SDE [418], [360]:

$$\begin{aligned} d\mathbf{x}_t &= f(t, \mathbf{x}_t)dt + \sigma(t, \mathbf{x}_t)d\boldsymbol{\omega}_t, &(11a)\\ d\mathbf{y}_t &= g(t, \mathbf{x}_t)dt + d\mathbf{v}_t, &(11b)\end{aligned}$$

where $f(t, \mathbf{x}_t)$ is often called *nonlinear drift* and $\sigma(t, \mathbf{x}_t)$ called *volatility* or *diffusion coefficient*. Again, the noise processes $\{\boldsymbol{\omega}_t, \mathbf{v}_t, t \geq 0\}$ are two Wiener processes. $\mathbf{x}_t \in \mathbb{R}^{N_\mathbf{x}}, \mathbf{y}_t \in \mathbb{R}^{N_\mathbf{y}}$. First, let's look at the state equation (a.k.a. diffusion equation). For all $t \geq 0$, we define a backward diffusion operator $\boldsymbol{L}_t$ as[16]

$$\boldsymbol{L}_t = \sum_{i=1}^{N_\mathbf{x}} f_t^i \frac{\partial}{\partial \mathbf{x}_i} + \frac{1}{2}\sum_{i,j=1}^{N_\mathbf{x}} a_t^{ij} \frac{\partial^2}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, \qquad (12)$$

[16] $\boldsymbol{L}_t$ is a partial differential operator.

where $a_t^{ij} = \sigma^i(t, \mathbf{x}_t)\sigma^j(t, \mathbf{x}_t)$. Operator $\boldsymbol{L}$ corresponds to an infinitesimal generator of the diffusion process $\{\mathbf{x}_t, t \geq 0\}$. The goal now is to deduce conditions under which one can find a recursive and finite-dimensional (close form) scheme to compute the conditional probability distribution $p(\mathbf{x}_t|\mathcal{Y}_t)$, given the *filtration* $\mathcal{Y}_t$[17] produced by the observation process (1b).

Let's define an innovations process[18]

$$\mathbf{e}_t = \mathbf{y}_t - \int_0^t \mathbb{E}[g(s, \mathbf{x}_s)|\mathbf{y}_{0:s}]ds, \tag{13}$$

where $\mathbb{E}[g(s, \mathbf{x}_s)|\mathcal{Y}_s]$ is described as

$$\begin{aligned} \hat{g}(\mathbf{x}_t) &= \mathbb{E}[g(t, \mathbf{x}_t)|\mathcal{Y}_t] \\ &= \int_{-\infty}^{\infty} g(\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{Y}_s)d\mathbf{x}. \end{aligned} \tag{14}$$

For any test function $\phi \in \mathbb{R}^{N_\mathbf{x}}$, the forward diffusion operator $\tilde{\boldsymbol{L}}$ is defined as

$$\tilde{\boldsymbol{L}}_t\phi = -\sum_{i=1}^{N_\mathbf{x}} f_t^i \frac{\partial \phi}{\partial \mathbf{x}_i} + \frac{1}{2}\sum_{i,j=1}^{N_\mathbf{x}} a_t^{ij} \frac{\partial^2 \phi}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, \tag{15}$$

which essentially is the Fokker-Planck operator. Given initial condition $p(\mathbf{x}_0)$ at $t = 0$ as boundary condition, it turns out that the pdf of diffusion process satisfies the Fokker-Planck-Kolmogorov equation (FPK; a.k.a. Kolmogorov forward equation, [387]) [19]

$$\frac{\partial p(\mathbf{x}_t)}{\partial t} = \tilde{\boldsymbol{L}}_t p(\mathbf{x}_t). \tag{16}$$

By involving the innovation process (13) and assuming $\mathbb{E}[\mathbf{v}_t] = \Sigma_{\mathbf{v},t}$, we have the following Kushner's equation (e.g., [284]):

$$dp(\mathbf{x}_t|\mathcal{Y}_t) = \tilde{\boldsymbol{L}}_t p(\mathbf{x}_t|\mathcal{Y}_t)dt + p(\mathbf{x}_t|\mathcal{Y}_t)\mathbf{e}_t\Sigma_{\mathbf{v},t}^{-1}dt, \quad (t \geq 0) \tag{17}$$

which reduces to the FPK equation (16) when there are no observations or filtration $\mathcal{Y}_t$. Integrating (17), we have

$$\begin{aligned} p(\mathbf{x}_t|\mathcal{Y}_t) &= p(\mathbf{x}_0) + \int_0^t p(\mathbf{x}_s|\mathcal{Y}_s)ds \\ &\quad + \int_0^t \tilde{\boldsymbol{L}}_s p(\mathbf{x}_s|\mathcal{Y}_s)\mathbf{e}_s\Sigma_{\mathbf{v},s}^{-1}ds. \end{aligned} \tag{18}$$

Given conditional pdf (18), suppose we want to calculate $\hat{\phi}(\mathbf{x}_t) = \mathbb{E}[\phi(\mathbf{x}_t)|\mathcal{Y}_t]$ for any nonlinear function $\phi \in \mathbb{R}^{N_\mathbf{x}}$. By interchanging the order of integrations, we have

$$\begin{aligned} \hat{\phi}(\mathbf{x}_t) &= \int_{-\infty}^{\infty} \phi(\mathbf{x})p(\mathbf{x}_t|\mathcal{Y}_t)d\mathbf{x} \\ &= \int_{-\infty}^{\infty} \phi(\mathbf{x})p(\mathbf{x}_0)d\mathbf{x} \\ &\quad + \int_0^t \int_{-\infty}^{\infty} \phi(\mathbf{x})\tilde{\boldsymbol{L}}_s p(\mathbf{x}_s|\mathcal{Y}_s)d\mathbf{x}ds \\ &\quad + \int_0^t \int_{-\infty}^{\infty} \phi(\mathbf{x})p(\mathbf{x}_s|\mathcal{Y}_s)\mathbf{e}_s\Sigma_{\mathbf{v},s}^{-1}d\mathbf{x}ds \\ &= \mathbb{E}[\phi(\mathbf{x}_0)] + \int_0^t \int_{-\infty}^{\infty} p(\mathbf{x}_s|\mathcal{Y}_s)\boldsymbol{L}_s\phi(\mathbf{x})d\mathbf{x}ds \\ &\quad + \int_0^t \bigg[\int_{-\infty}^{\infty} \phi(\mathbf{x})g(s, \mathbf{x})p(\mathbf{x}_s|\mathcal{Y}_s)d\mathbf{x} \\ &\quad - \hat{g}(\mathbf{x}_s)\int_{-\infty}^{\infty} \phi(\mathbf{x})p(\mathbf{x}_s|\mathcal{Y}_s)d\mathbf{x}\bigg]\Sigma_{\mathbf{v},s}^{-1}ds. \end{aligned}$$

The Kushner equation lends itself a recursive form of filtering solution, but the conditional mean requests all of higher-order conditional moments and thus leads to an infinite-dimensional system.

On the other hand, under some mild conditions, the *unnormalized* conditional density of $\mathbf{x}_t$ given $\mathcal{Y}_s$, denoted as $\pi(\mathbf{x}_t|\mathcal{Y}_t)$, is the unique solution of the following stochastic partial differential equation (PDE), the so-called Zakai equation (see [505], [238], [285]):

$$d\pi(\mathbf{x}_t|\mathcal{Y}_t) = \tilde{\boldsymbol{L}}\pi(\mathbf{x}_t|\mathcal{Y}_t)dt + g(t, \mathbf{x}_t)\pi(\mathbf{x}_t|\mathcal{Y}_t)d\mathbf{y}_t \tag{19}$$

with the same $\tilde{\boldsymbol{L}}$ defined in (15). Zakai equation and Kushner equation have a *one-to-one* correspondence, but Zakai equation is much simpler,[20] hence we are usually turned to solve the Zakai equation instead of Kushner equation. In the early history of nonlinear filtering, the common way is to discretize the Zakai equation to seek the numerical solution. Numerous efforts were devoted along this line [285], [286], e.g. separation of variables [114], adaptive local grid [65], particle (quadrature) method [66]. However, these methods are neither recursive nor computationally efficient.

III. Bayesian Statistics and Bayesian Estimation

A. Bayesian Statistics

Bayesian theory (e.g., [38]) is a branch of mathematical probability theory that allows people to model the uncertainty about the world and the outcomes of interest by incorporating prior knowledge and observational evidence.[21] Bayesian analysis, interpreting the probability as

[17] One can imagine *filtration* as sort of information coding the previous history of the state and measurement.

[18] Innovations process is defined as a white Gaussian noise process. See [245], [247] for detailed treatment.

[19] The stochastic process is determined equivalently by the FPK equation (16) or the SDE (11a). The FPK equation can be interpreted as follows: The first term is the equation of motion for a cloud of particles whose distribution is $p(\mathbf{x}_t)$, each point of which obeys the equation of motion $\frac{d\mathbf{x}}{dt} = f(\mathbf{x}_t, t)$. The second term describes the disturbance due to Brownian motion. The solution of (16) can be solved exactly by Fourier transform. By inverting the Fourier transform, we can obtain

$$p(\mathbf{x}, t + \Delta t|\mathbf{x}_0, t) = \frac{1}{\sqrt{2\pi\sigma_0\Delta t}} \exp\Big\{-\frac{(\mathbf{x} - \mathbf{x}_0 - f(\mathbf{x}_0)\Delta t)^2}{2\sigma_0\Delta t}\Big\},$$

which is a Guaussian distribution of a deterministic path.

[20] This is true because (19) is linear w.r.t. $\pi(\mathbf{x}_t|\mathcal{Y}_t)$ whereas (17) involves certain nonlinearity. We don't extend discussion here due to space constraint.

[21] In the circle of statistics, there are slightly different treatments to probability. The frequentists condition on a hypothesis of choice and put the probability distribution on the data, either observed or not;

a *conditional measure of uncertainty*, is one of the popular methods to solve the inverse problems. Before running into Bayesian inference and Bayesian estimation, we first introduce some fundamental Bayesian statistics.

Definition 7: (Bayesian Sufficient Statistics) Let $p(\mathbf{x}|\mathcal{Y})$ denote the probability density of $\mathbf{x}$ conditioned on measurements $\mathcal{Y}$. A statistics, $\Psi(\mathbf{x})$, is said to be "sufficient" if the distribution of $\mathbf{x}$ conditionally on Ψ does not depend on $\mathcal{Y}$. In other words, $p(\mathbf{x}|\mathcal{Y}) = p(\mathbf{x}|\mathcal{Y}')$ for any two sets $\mathcal{Y}$ and $\mathcal{Y}'$ s.t. $\Psi(\mathcal{Y}) = \Psi(\mathcal{Y}')$.

The sufficient statistics $\Psi(\mathbf{x})$ contains all of information brought by $\mathbf{x}$ about $\mathcal{Y}$. The *Rao-Blackwell Theorem* says that when an estimator is evaluated under a convex loss, the optimal procedure only depends on the sufficient statistics. *Sufficiency Principle* and *Likelihood Principle* are two axiomatic principles in the Bayesian inference [388].

There are three types of intractable problems inherently related to the Bayesian statistics:

- **Normalization:** Given the prior $p(\mathbf{x})$ and likelihood $p(\mathbf{y}|\mathbf{x})$, the posterior $p(\mathbf{x}|\mathbf{y})$ is obtained by the product of prior and likelihood divided by a normalizing factor as
$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int_X p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}}. \tag{20}$$
- **Marginalization:** Given the joint posterior $(\mathbf{x}, \mathbf{z})$, the marginal posterior is
$$p(\mathbf{x}|\mathbf{y}) = \int_Z p(\mathbf{x}, \mathbf{z}|\mathbf{y})d\mathbf{z}, \tag{21}$$
as shown later, marginalization and factorization plays an important role in Bayesian inference.
- **Expectation:** Given the conditional pdf, some averaged statistics of interest can be calculated
$$\mathbb{E}_{p(\mathbf{x}|\mathbf{y})}[f(\mathbf{x})] = \int_X f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}. \tag{22}$$

In Bayesian inference, all of uncertainties (including states, parameters which are either time-varying or fixed but unknown, priors) are treated as random variables.[22] The inference is performed within the Bayesian framework given all of available information. And the objective of Bayesian inference is to use priors and causal knowledge, quantitatively and qualitatively, to infer the conditional probability, given finite observations. There are usually three levels of probabilistic reasoning in Bayesian analysis (so-called hierarchical Bayesian analysis): (i) starting with model selection given the data and assumed priors; (ii) estimating the parameters to fit the data given the model and priors; (iii) updating the hyperparameters of the prior. *Optimization* and *integration* are two fundamental numerical problems arising in statistical inference. Bayesian inference can be illustrated by a directed graph, a Bayesian network (or belief network) is a probabilistic graphical model with a set of vertices and edges (or arcs), the probability dependency is described by a directed arrow between two nodes that represent two random variables. Graphical models also allow the possibility of constructing more complex hierarchical statistical models [239], [240].

B. Recursive Bayesian Estimation

In the following, we present a detailed derivation of recursive Bayesian estimation, which underlies the principle of sequential Bayesian filtering. Two assumptions are used to derive the recursive Bayesian filter: (i) The states follow a first-order Markov process $p(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$; (ii) the observations are independent of the given states. For notation simplicity, we denote $\mathcal{Y}_n$ as a set of observations $\mathbf{y}_{0:n} := \{\mathbf{y}_0, \cdots, \mathbf{y}_n\}$; let $p(\mathbf{x}_n|\mathcal{Y}_n)$ denote the conditional pdf of $\mathbf{x}_n$. From Bayes rule we have

$$\begin{aligned}
p(\mathbf{x}_n|\mathcal{Y}_n) &= \frac{p(\mathcal{Y}_n|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathcal{Y}_n)} \\
&= \frac{p(\mathbf{y}_n, \mathcal{Y}_{n-1}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n, \mathcal{Y}_{n-1})} \\
&= \frac{p(\mathbf{y}_n|\mathcal{Y}_{n-1}, \mathbf{x}_n)p(\mathcal{Y}_{n-1}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})} \\
&= \frac{p(\mathbf{y}_n|\mathcal{Y}_{n-1}, \mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})p(\mathbf{x}_n)} \\
&= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})}{p(\mathbf{y}_n|\mathcal{Y}_{n-1})}. \qquad (23)
\end{aligned}$$

As shown in (23), the posterior density $p(\mathbf{x}_n|\mathcal{Y}_n)$ is described by three terms:

- **Prior:** The prior $p(\mathbf{x}_n|\mathcal{Y}_{n-1})$ defines the knowledge of the model
$$p(\mathbf{x}_n|\mathcal{Y}_{n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathcal{Y}_{n-1})d\mathbf{x}_{n-1}, \tag{24}$$
where $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ is the transition density of the state.
- **Likelihood:** the likelihood $p(\mathbf{y}_n|\mathbf{x}_n)$ essentially determines the measurement noise model in the equation (2b).
- **Evidence:** The denominator involves an integral
$$p(\mathbf{y}_n|\mathcal{Y}_{n-1}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})d\mathbf{x}_n. \tag{25}$$

Calculation or approximation of these three terms are the essences of the Bayesian filtering and inference.

IV. Bayesian Optimal Filtering

Bayesian filtering is aimed to apply the Bayesian statistics and Bayes rule to probabilistic inference problems, and specifically the stochastic filtering problem. To our knowledge, Ho and Lee [212] were among the first authors to

only one hypothesis is regarded as true; they regard the probability as frequency. The Bayesians only condition on the observed data and consider the probability distributions on the hypotheses; they put probability distributions on the several hypotheses given some priors; probability is not viewed equivalent to the frequency. See [388], [38], [320] for more information.

[22] This is the true spirit of Bayesian estimation which is different from other estimation schemes (e.g. least-squares) where the unknown parameters are usually regarded as deterministic.

discuss iterative Bayesian filtering, in which they discussed in principle the sequential state estimation problem and included the Kalman filter as a special case. In the past few decades, numerous authors have investigated the Bayesian filtering in a dynamic state space framework [270], [271], [421], [424], [372], [480]-[484].

A. Optimal Filtering

An optimal filter is said "optimal" only in some specific sense [12]; in other other words, one should define a criterion which measures the optimality. For example, some potential criteria for measuring the optimality can be:

1. Minimum mean-squared error (MMSE): It can be defined in terms of prediction or filtering error (or equivalently the trace of state-error covariance)

$$\mathbb{E}[\|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2|\mathbf{y}_{0:n}] = \int \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2 p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n,$$

which is aimed to find the *conditional mean* $\hat{\mathbf{x}}_n = \mathbb{E}[\mathbf{x}_n|\mathbf{y}_{0:n}] = \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n$.

2. Maximum a posteriori (MAP): It is aimed to find the *mode* of posterior probability $p(\mathbf{x}_n|\mathbf{y}_{0:n})$,[23] which is equal to minimize a loss function

$$\mathcal{E} = \mathbb{E}[1 - \mathbb{I}_{\mathbf{x}_n:\|\mathbf{x}_n - \hat{\mathbf{x}}_n\| \le \zeta}(\mathbf{x}_n)],$$

where $\mathbb{I}(\cdot)$ is an indicator function and ζ is a small scalar.

3. Maximum likelihood (ML): which reduces to a special case of MAP where the prior is neglected.[24]
4. Minimax: which is to find the *median* of posterior $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. See Fig. 3 for an illustration of the difference between mode, mean and median.
5. Minimum conditional inaccuracy[25]: Namely,

$$\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[-\log \hat{p}(\mathbf{x}|\mathbf{y})] = \int p(\mathbf{x},\mathbf{y}) \log \frac{1}{\hat{p}(\mathbf{x}|\mathbf{y})} d\mathbf{x}d\mathbf{y}.$$

6. Minimum conditional KL divergence [276]: The conditional KL divergence is given by

$$\mathrm{KL} = \int p(\mathbf{x},\mathbf{y}) \log \frac{p(\mathbf{x},\mathbf{y})}{\hat{p}(\mathbf{x}|\mathbf{y})p(\mathbf{x})} d\mathbf{x}d\mathbf{y}.$$

7. Minimum free energy[26]: It is a lower bound of maximum log-likelihood, which is aimed to minimize

$$\begin{aligned}\mathcal{F}(Q;P) &\equiv \mathbb{E}_{Q(\mathbf{x})}[-\log P(\mathbf{x}|\mathbf{y})] \\ &= \mathbb{E}_{Q(\mathbf{x})}\Big[\log \frac{Q(\mathbf{x})}{P(\mathbf{x}|\mathbf{y})}\Big] - \mathbb{E}_{Q(\mathbf{x})}[\log Q(\mathbf{x})],\end{aligned}$$

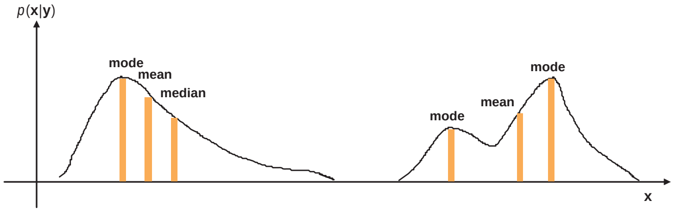

Fig. 3. **Left:** An illustration of three optimal criteria that seek different solutions for a skewed unimodal distribution, in which the *mean*, *mode* and *median* do not coincide. **Right:** MAP is misleading for the multimodal distribution where multiple modes (maxima) exist.

where $Q(\mathbf{x})$ is an arbitrary distribution of $\mathbf{x}$. The first term is called Kullback-Leibler (KL) divergence between distributions $Q(\mathbf{x})$ and $P(\mathbf{x}|\mathbf{y})$, the second term is the entropy w.r.t. $Q(\mathbf{x})$. The minimization of free energy can be implemented iteratively by the expectation-maximization (EM) algorithm [130]:

$$\begin{aligned} Q(\mathbf{x}_{n+1}) &\longleftarrow \arg\max_{Q}\{Q, \mathbf{x}_n\}, \\ \mathbf{x}_{n+1} &\longleftarrow \arg\max_{\mathbf{x}}\{Q(\mathbf{x}_n), \mathbf{x}\}. \end{aligned}$$

Remarks:

- The above criteria are valid not only for state estimation but also for parameter estimation (by viewing $\mathbf{x}$ as unknown parameters).
- Both MMSE and MAP methods require the estimation of the posterior distribution (density), but MAP doesn't require the calculation of the denominator (integration) and thereby more computational inexpensive; whereas the former requires full knowledge of the *prior*, *likelihood* and *evidence*. Note that however, MAP estimate has a drawback especially in a high-dimensional space. High probability density does not imply high probability mass. A narrow spike with very small width (support) can have a very high density, but the actual probability of estimated state (or parameter) belonging to it is small. Hence, the width of the mode is more important than its height in the high-dimensional case.
- The last three criteria are all ML oriented. By minimizing the negative log-likelihood $-\log \hat{p}(\mathbf{x}|\mathbf{y})$ and taking the expectation w.r.t. a fixed or variational pdf. Criterion 5 chooses the expectation w.r.t. joint pdf $p(\mathbf{x},\mathbf{y})$; when $Q(\mathbf{x}) = p(\mathbf{x},\mathbf{y})$, it is equivalent to Criterion 7; Criterion 6 is a modified version of the upper bound of Criterion 5.

The criterion of optimality used for Bayesian filtering is the Bayes risk of MMSE.[27] Bayesian filtering is optimal in a sense that it seeks the posterior distribution which integrates and uses all of available information expressed by probabilities (assuming they are quantitatively correct). However, as time proceeds, one needs infinite computing power and unlimited memory to calculate the "optimal"

[23] When the *mode* and the *mean* of distribution coincide, the MAP estimation is correct; however, for multimodal distributions, the MAP estimate can be arbitrarily bad. See Fig. 3.

[24] This can be viewed as a least-informative prior with uniform distribution.

[25] It is a generalization of Kerridge's inaccuracy for the case of i.i.d. data.

[26] Free energy is a variational approximation of ML in order to minimize its upper bound. This criterion is usually used in off-line Bayesian estimation.

[27] For a discussion of difference between Bayesian risk and frequentist risk, see [388].

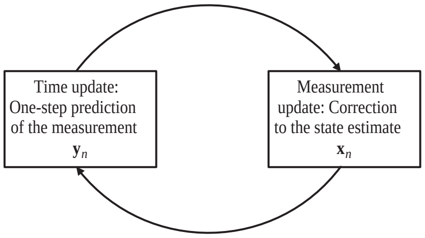

Fig. 4. Schematic illustration of Kalman filter's update as a predictor-corrector.

solution, except in some special cases (e.g. linear Gaussian or conjugate family case). Hence in general, we can only seek a suboptimal or locally optimal solution.

B. Kalman Filtering

Kalman filtering, in the spirit of Kalman filter [250], [253] or Kalman-Bucy filter [249], consists of an iterative *prediction-correction* process (see Fig. 4). In the prediction step, the time update is taken where the one-step ahead prediction of observation is calculated; in the correction step, the measurement update is taken where the correction to the estimate of current state is calculated. In a stationary situation, the matrices $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n$ in (3a) and (3b) are constant, Kalman filter is precisely the Wiener filter for stationary least-squares smoothing. In other words, Kalman filter is a time-variant Wiener filter [11], [12]. Under the LQG circumstance, Kalman filter was originally derived with the orthogonal projection method. In the late 1960s, Kailath [245] used the innovation approach developed by Wold and Kolmogorov to reformulate the Kalman filter, with the tool of martingales theory.[28] From innovations point of view, Kalman filter is a whitening filter.[29] Kalman filter is also optimal in the sense that it is unbiased $\mathbb{E}[\hat{\mathbf{x}}_n] = \mathbb{E}[\mathbf{x}_n]$ and is a minimum variance estimate. A detailed history of Kalman filter and its many variants can be found in [385], [244], [246], [247], [238], [12], [423], [96], [195].

Kalman filter has a very nice Bayesian interpretation [212], [497], [248], [366]. In the following, we will show that the celebrated Kalman filter can be derived within a Bayesian framework, or more specifically, it reduces to a MAP solution. The derivation is somehow similar to the ML solution given by [384]. For presentation simplicity, we assume the dynamic and measurement noises are both Gaussian distributed with zero mean and constant covariance. The derivation of Kalman filter in the linear Gaussian scenario is based on the following assumptions:

- $\mathbb{E}[\mathbf{d}_n\mathbf{d}_m^T] = \Sigma_{\mathbf{d}}\delta_{mn}$; $\mathbb{E}[\mathbf{v}_n\mathbf{v}_m^T] = \Sigma_{\mathbf{v}}\delta_{mn}$.
- The state and process noise are mutually independent: $\mathbb{E}[\mathbf{x}_n\mathbf{d}_m^T] = 0$ for $n \le m$; $\mathbb{E}[\mathbf{x}_n\mathbf{v}_m^T] = 0$ for all n, m.
- The process noise and measurement noise are mutually independent: $\mathbb{E}[\mathbf{d}_n\mathbf{v}_m^T] = 0$ for all n, m.

Let $\hat{\mathbf{x}}_n^{\text{MAP}}$ denote the MAP estimate of $\mathbf{x}_n$ that maximizes $p(\mathbf{x}_n|\mathcal{Y}_n)$, or equivalently $\log p(\mathbf{x}_n|\mathcal{Y}_n)$. By using the Bayes rule, we may express $p(\mathbf{x}_n|\mathcal{Y}_n)$ by

$$\begin{aligned} p(\mathbf{x}_n|\mathcal{Y}_n) &= \frac{p(\mathbf{x}_n, \mathcal{Y}_n)}{p(\mathcal{Y}_n)} \\ &= \frac{p(\mathbf{x}_n, \mathbf{y}_n, \mathcal{Y}_{n-1})}{p(\mathbf{y}_n, \mathcal{Y}_{n-1})}, \end{aligned} \tag{26}$$

where the expression of joint pdf in the numerator is further expressed by

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{y}_n, \mathcal{Y}_{n-1}) &= p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{Y}_{n-1})p(\mathbf{x}_n, \mathcal{Y}_{n-1}) \\ &= p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{Y}_{n-1})p(\mathbf{x}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1}) \\ &= p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1}). \end{aligned} \tag{27}$$

The third step is based on the fact that $\mathbf{v}_n$ does not depend on $\mathcal{Y}_{n-1}$. Substituting (27) into (26), we obtain

$$\begin{aligned} p(\mathbf{x}_n|\mathcal{Y}_n) &= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})}{p(\mathbf{y}_n, \mathcal{Y}_{n-1})} \\ &= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})}{p(\mathbf{y}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})} \\ &= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathcal{Y}_{n-1})}{p(\mathbf{y}_n|\mathcal{Y}_{n-1})}, \end{aligned} \tag{28}$$

which shares the same form as (23). Under the Gaussian assumption of process noise and measurement noise, the mean and covariance of $p(\mathbf{y}_n|\mathbf{x}_n)$ are calculated by

$$\mathbb{E}[\mathbf{y}_n|\mathbf{x}_n] = \mathbb{E}[\mathbf{G}_n\mathbf{x}_n + \mathbf{v}_n] = \mathbf{G}_n\mathbf{x}_n \tag{29}$$

and

$$\text{Cov}[\mathbf{y}_n|\mathbf{x}_n] = \text{Cov}[\mathbf{v}_n|\mathbf{x}_n] = \Sigma_{\mathbf{v}}, \tag{30}$$

respectively. And the conditional pdf $p(\mathbf{y}_n|\mathbf{x}_n)$ can be further written as

$$p(\mathbf{y}_n|\mathbf{x}_n) = A_1 \exp\Big(-\frac{1}{2}(\mathbf{y}_n - \mathbf{G}_n\mathbf{x}_n)^T\Sigma_{\mathbf{v}}^{-1}(\mathbf{y}_n - \mathbf{G}_n\mathbf{x}_n)\Big), \tag{31}$$

where $A_1 = (2\pi)^{-N_{\mathbf{y}}/2}|\Sigma_{\mathbf{v}}|^{-1/2}$.

Consider the conditional pdf $p(\mathbf{x}_n|\mathcal{Y}_{n-1})$, its mean and covariance are calculated by

$$\begin{aligned} \mathbb{E}[\mathbf{x}_n|\mathcal{Y}_{n-1}] &= \mathbb{E}[\mathbf{F}_{n,n-1}\hat{\mathbf{x}}_n + \mathbf{d}_{n-1}|\mathcal{Y}_{n-1}] \\ &= \mathbf{F}_{n-1,n}\hat{\mathbf{x}}_{n-1} = \hat{\mathbf{x}}_{n|n-1}, \end{aligned} \tag{32}$$

and

$$\begin{aligned} \text{Cov}[\mathbf{x}_n|\mathcal{Y}_{n-1}] &= \text{Cov}[\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1}] \\ &= \text{Cov}[\mathbf{e}_{n,n-1}], \end{aligned} \tag{33}$$

respectively, where $\hat{\mathbf{x}}_{n|n-1} \equiv \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ represents the state estimate at time n given the observations up to $n-1$,

[28] The martingale process was first introduced by Doob and discussed in detail in [139].

[29] Innovations concept can be used straightforward in nonlinear filtering [7]. From innovations point of view, one of criteria to justify the optimality of the solution to a nonlinear filtering problem is to check how *white* the pseudo-innovations are, the whiter the more optimal.

$\mathbf{e}_{n,n-1}$ is the state-error vector. Denoting the covariance of $\mathbf{e}_{n,n-1}$ by $\mathbf{P}_{n,n-1}$, by Gaussian assumption, we may obtain

$$\begin{aligned} p(\mathbf{x}_n|\mathcal{Y}_{n-1}) &= A_2 \exp\Big(-\frac{1}{2}(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})^T \\ &\quad \times \mathbf{P}^{-1}_{n,n-1}(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})\Big), \end{aligned} \tag{34}$$

where $A_2 = (2\pi)^{-N_\mathbf{x}/2}|\mathbf{P}_{n,n-1}|^{-1/2}$. By substituting equations (31) and (34) to (26), it further follows

$$\begin{aligned} p(\mathbf{x}_n|\mathcal{Y}_n) &\propto A \exp\Big(-\frac{1}{2}(\mathbf{y}_n - \mathbf{G}_n\mathbf{x}_n)^T\Sigma_\mathbf{v}^{-1}(\mathbf{y}_n - \mathbf{G}_n\mathbf{x}_n) \\ &\quad -\frac{1}{2}(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})^T\mathbf{P}^{-1}_{n,n-1}(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})\Big), \end{aligned} \tag{35}$$

where $A = A_1A_2$ is a constant. Since the denominator is a normalizing constant, (35) can be regarded as an *unnormalized* density, the fact doesn't affect the following derivation.

Since the MAP estimate of the state is defined by the condition

$$\frac{\partial \log p(\mathbf{x}_n|\mathcal{Y}_n)}{\partial \mathbf{x}_n}\Big|_{\mathbf{x}_n = \hat{\mathbf{x}}^{\mathrm{MAP}}} = 0, \tag{36}$$

substituting equation (35) into (36) yields

$$\begin{aligned} \hat{\mathbf{x}}_n^{\mathrm{MAP}} &= \Big(\mathbf{G}_n^T\Sigma_\mathbf{v}^{-1}\mathbf{G}_n + \mathbf{P}^{-1}_{n,n-1}\Big)^{-1} \\ &\quad \times \Big(\mathbf{P}^{-1}_{n,n-1}\hat{\mathbf{x}}_{n|n-1} + \mathbf{G}_n^T\Sigma_\mathbf{v}^{-1}\mathbf{y}_n\Big). \end{aligned}$$

By using the lemma of inverse matrix,[30] it is simplified as

$$\hat{\mathbf{x}}_n^{\mathrm{MAP}} = \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{G}_n\hat{\mathbf{x}}_{n|n-1}), \tag{37}$$

where $\mathbf{K}_n$ is the Kalman gain as defined by

$$\mathbf{K}_n = \mathbf{F}_{n+1,n}\mathbf{P}_{n,n-1}\mathbf{G}_n^T(\mathbf{G}_n\mathbf{P}_{n,n-1}\mathbf{G}_n^T + \Sigma_\mathbf{v})^{-1}. \tag{38}$$

Observing

$$\begin{aligned} \mathbf{e}_{n,n-1} &= \mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} \\ &= \mathbf{F}_{n,n-1}\mathbf{x}_{n-1} + \mathbf{d}_n - \mathbf{F}_{n,n-1}\hat{\mathbf{x}}_{n-1}^{\mathrm{MAP}} \\ &= \mathbf{F}_{n,n-1}\mathbf{e}_{n-1}^{\mathrm{MAP}} + \mathbf{d}_{n-1}, \end{aligned} \tag{39}$$

and by virtue of $\mathbf{P}_{n-1} = \mathrm{Cov}[\mathbf{e}_{n-1}^{\mathrm{MAP}}]$, we have

$$\begin{aligned} \mathbf{P}_{n,n-1} &= \mathrm{Cov}[\mathbf{e}_{n,n-1}] \\ &= \mathbf{F}_{n,n-1}\mathbf{P}_{n-1}\mathbf{F}_{n,n-1}^T + \Sigma_\mathbf{d}. \end{aligned} \tag{40}$$

Since

$$\begin{aligned} \mathbf{e}_n &= \mathbf{x}_n - \hat{\mathbf{x}}_n^{\mathrm{MAP}} \\ &= \mathbf{x}_n - \mathbf{x}_{n|n-1} - \mathbf{K}_n(\mathbf{y}_n - \mathbf{G}_n\hat{\mathbf{x}}_{n|n-1}), \end{aligned} \tag{41}$$

noting that $\mathbf{e}_{n,n-1} = \mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1}$ and $\mathbf{y}_n = \mathbf{G}_n\mathbf{x}_n + \mathbf{v}_n$, we further have

$$\begin{aligned} \mathbf{e}_n &= \mathbf{e}_{n,n-1} - \mathbf{K}_n(\mathbf{G}_n\mathbf{e}_{n,n-1} + \mathbf{v}_n) \\ &= (\mathbf{I} - \mathbf{K}_n\mathbf{G}_n)\mathbf{e}_{n,n-1} - \mathbf{K}_n\mathbf{v}_n, \end{aligned} \tag{42}$$

and it further follows

$$\begin{aligned} \mathbf{P}_n &= \mathrm{Cov}[\mathbf{e}_n^{\mathrm{MAP}}] \\ &= (\mathbf{I} - \mathbf{K}_n\mathbf{G}_n)\mathbf{P}_{n,n-1}(\mathbf{I} - \mathbf{K}_n\mathbf{G}_n)^T + \mathbf{K}_n\Sigma_\mathbf{v}\mathbf{K}_n^T. \end{aligned}$$

Rearranging the above equation, it reduces to

$$\mathbf{P}_n = \mathbf{P}_{n,n-1} - \mathbf{F}_{n,n+1}\mathbf{K}_n\mathbf{G}_n\mathbf{P}_{n,n-1}. \tag{43}$$

Thus far, the Kalman filter is completely derived from MAP principle, the expression of $\mathbf{x}_n^{\mathrm{MAP}}$ is exactly the same solution derived from the innovations framework (or others).

The above procedure can be easily extended to ML case without much effort [384]. Suppose we want to maximize the *marginal maximum likelihood* of $p(\mathbf{x}_n|\mathcal{Y}_n)$, which is equivalent to maximizing the log-likelihood

$$\log p(\mathbf{x}_n|\mathcal{Y}_n) = \log p(\mathbf{x}_n, \mathcal{Y}_n) - \log p(\mathcal{Y}_n), \tag{44}$$

and the optimal estimate near the solution should satisfy

$$\frac{\partial \log p(\mathbf{x}_n|\mathcal{Y}_n)}{\partial \mathbf{x}_n}\Big|_{\mathbf{x}_n = \hat{\mathbf{x}}^{\mathrm{ML}}} = 0. \tag{45}$$

Substituting (35) to (45), we actually want to minimize the the cost function of two combined Mahalanobis norms [31]

$$\mathcal{E} = \|\mathbf{y}_n - \mathbf{G}_n\mathbf{x}_n\|^2_{\Sigma_\mathbf{v}^{-1}} + \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2_{\mathbf{P}^{-1}_{n,n-1}}. \tag{46}$$

Taking the derivative of $\mathcal{E}$ with respect to $\mathbf{x}_n$ and setting as zero, we also obtain the same solution as (37).

Remarks:

- The derivation of the Kalman-Bucy filter [249] was rooted in the SDE theory [387], [360], it can be also derived within the Bayesian framework [497], [248].
- The optimal filtering solution described by Wiener-Hopf equation is achieved by spectral factorization technique [487]. By admitting state-space formulation, Kalman filter elegantly overcomes the stationarity assumption and provides a fresh look at the filtering problem. The signal process (i.e."state") is regarded as a linear stochastic dynamical system driven by white noise, the optimal filter thus has a stochastic differential structure which makes the recursive estimation possible. Spectral factorization is replaced by the solution of an ordinary differential equation (ODE) with known initial conditions. Wiener filter doesn't treat the difference between the white and colored noises, it also permits the infinite-dimensional systems; whereas Kalman filter works for

[30] For $\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T$, it follows from the matrix inverse lemma that $\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{B}$.

[31] The Mahalanobis norm is defined as a weighted norm: $\|\mathbf{A}\|^2_\mathbf{B} = \mathbf{A}^T\mathbf{B}\mathbf{A}$.

finite-dimensional systems with white noise assumption.

- Kalman filter is an unbiased minimum variance estimator under LOG circumstance. When the Gaussian assumption of noise is violated, Kalman filter is still optimal in a mean square sense, but the estimate doesn't produce the condition mean (i.e. it is biased), and neither the minimum variance. Kalman filter is not robust because of the underlying assumption of noise density model.
- Kalman filter provides an exact solution for linear Gaussian prediction and filtering problem. Concerning the smoothing problem, the off-line estimation version of Kalman filter is given by the Rauch-Tung-Striebel (RTS) smoother [384], which consists of a forward filter in a form of Kalman filter and a backward recursive smoother. The RTS smoother is computationally efficient than the optimal smoother [206].
- The conventional Kalman filter is a point-valued filter, it can be also extended to set-valued filtering [39], [339], [80].
- In the literature, there exists many variants of Kalman filter, e.g., covariance filter, information filter, square-root Kalman filters. See [205], [247] for more details and [403] for a unifying review.

C. Optimum Nonlinear Filtering

In practice, the use of Kalman filter is limited by the ubiquitous nonlinearity and non-Gaussianity of physical world. Hence since the publication of Kalman filter, numerous efforts have been devoted to the generic filtering problem, mostly in the Kalman filtering framework. A number of pioneers, including Zadeh [503], Bucy [61], [60], Wonham [496], Zakai [505], Kushner [282]-[285], Stratonovich [430], [431], investigated the nonlinear filtering problem. See also the papers seeking optimal nonlinear filters [420], [289], [209]. In general, the nonlinear filtering problem per sue consists in finding the conditional probability distribution (or density) of the state given the observations up to current time [420]. In particular, the solution of nonlinear filtering problem using the theory of conditional Markov processes [430], [431] is very attractive from Bayesian perspective and has a number of advantages over the other methods. The recursive transformations of the posterior measures are characteristics of this theory. Strictly speaking, the number of variables replacing the density function is infinite, but not all of them are of equal importance. Thus it is advisable to select the important ones and reject the remainder.

The solutions of nonlinear filtering problem have two categories: *global* method and *local* method. In the global approach, one attempts to solve a PDE instead of an ODE in linear case, e.g. Zakai equation, Kushner-Stratonovich equation, which are mostly analytically intractable. Hence the numerical approximation techniques are needed to solve the equation. In special scenarios (e.g. exponential family) with some assumptions, the nonlinear filtering can admit the tractable solutions. In the local approach, finite sum approximation (e.g. Gaussian sum filter) or linearization techniques (i.e. EKF) are usually used. In the EKF, by defining

$$\hat{\mathbf{F}}_{n+1,n} = \frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_n}, \quad \hat{\mathbf{G}}_n = \frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{n|n-1}},$$

the equations (2a)(2b) can be linearized into (3a)(3b), and the conventional Kalman filtering technique is further employed. The details of EKF can be found in many books, e.g. [238], [12], [96], [80], [195], [205], [206]. Because EKF always approximates the posterior $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ as a Gaussian, it works well for some types of nonlinear problems, but it may provide a poor performance in some cases when the true posterior is non-Gaussian (e.g. heavily skewed or multimodal). Gelb [174] provided an early overview of the uses of EKF. It is noted that the estimate given by EKF is usually biased since in general $\mathbb{E}[f(\mathbf{x})] \neq f(\mathbb{E}[\mathbf{x}])$.

In summary, a number of methods have been developed for nonlinear filtering problems:

- Linearization methods: first-order Taylor series expansion (i.e. EKF), and higher-order filter [20], [437].
- Approximation by finite-dimensional nonlinear filters: Beneš filter [33], [34], Daum filter [111]-[113], and projection filter [202], [55].
- Classic PDE methods, e.g. [282], [284], [285], [505], [496], [497], [235].
- Spectral methods [312].
- Neural filter methods, e.g. [209].
- Numerical approximation methods, as to be discussed in Section V.

C.1 Finite-dimensional Filters

The on-line solution of the FPK equation can be avoided if the unnormalized filtered density admits a finite-dimensional sufficient statistics. Beneš [33], [34] first explored the exact finite-dimensional filter[32] in the nonlinear filtering scenario. Daum [111] extended the framework to a more general case and included Kalman filter and Beneš filter as special cases [113]. Some new development of Daum filter with virtual measurement was summarized in [113]. The recently proposed projection filters [202], [53]-[57], also belong to the finite-dimensional filter family.

In [111], starting with SDE filtering theory, Daum introduced a gradient function

$$r(t,\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \ln \psi(t,\mathbf{x})$$

where $\psi(t,\mathbf{x})$ is the solution of the FPK equation of (11a) with a form

$$\frac{\partial \psi(t,\mathbf{x})}{\partial t} = -\frac{\partial \psi(t,\mathbf{x})}{\partial \mathbf{x}} f - \psi \mathrm{tr}\Big(\frac{\partial f}{\partial \mathbf{x}}\Big) + \frac{1}{2}\mathrm{tr}\Big(A\frac{\partial^2 \psi}{\partial \mathbf{x}\mathbf{x}^T}\Big),$$

with an appropriate initial condition (see [111]), and $A = \sigma(t,\mathbf{x}_t)\sigma(t,\mathbf{x}_t)^T$. When the measurement equation (11b) is

[32] Roughly speaking, a finite-dimensional filter is the one that can be implemented by integrating a finite number of ODE, or the one has the sufficient statistics with finite variables.

linear with Gaussian noise (recalling the discrete-time version (3b)), Daum filter admits a finite-dimensional solution

$$p(\mathbf{x}_t|\mathcal{Y}_t) = \psi^s(\mathbf{x}_t) \exp\left[\frac{1}{2}(\mathbf{x}_t - \boldsymbol{m}_t)^T \boldsymbol{P}_t^{-1}(\mathbf{x}_t - \boldsymbol{m}_t)\right],$$

where s is real number in the interval $0 < s < 1$ defined in the initial condition, $\boldsymbol{m}_t$ and $\boldsymbol{P}_t$ are two sufficient statistics that can be computed recursively.[33] The calculation of $\psi(\mathbf{x}_t)$ can be done off line which does not rely on the measurement, whereas $\boldsymbol{m}_t$ and $\boldsymbol{P}_t$ will be computed on line using numerical methods. See [111]-[113] for more details.

The problem of the existence of a finite-dimensional filter is concerned with the necessary and sufficient conditions. In [167], a necessary condition is that the observations and the filtering densities belong to the exponential class. In particular, we have the *Generalized Fisher-Darmois-Koopman-Pitman Theorem*:

Theorem 1: e.g. [388], [112] For smooth nowhere vanishing densities, a fixed finite-dimensional filter exists if and only if the *unnormalized* conditional density is from an exponential family

$$\pi(\mathbf{x}_n|\mathbf{y}_{0:n}) = \pi(\mathbf{x}_n) \exp[\lambda^T(\mathbf{x}_n)\Psi(\mathbf{y}_{0:n})], \tag{47}$$

where $\Psi(\cdot)$ is a sufficient statistics, $\lambda(\cdot)$ is a function in X (which turns out to be the solution of specific PDE's).

The nonlinear finite-dimensional filtering is usually performed with the conjugate approach, where the prior and posterior are assumed to come from some parametric probability function family in order to admit the exact and analytically tractable solution. We will come back to this topic in Section VII. On the other hand, for general nonlinear filtering problem, no exact solution can be obtained, various numerical approximation are hence need. In the next section, we briefly review some popular numerical approximation approaches in the literature and focus our attention on the sequential Monte Carlo technique.

V. Numerical Approximation Methods

A. Gaussian/Laplace Approximation

Gaussian approximation is the simplest method to approximate the numerical integration problem because of its analytic tractability. By assuming the posterior as Gaussian, the nonlinear filtering can be taken with the EKF method.

Laplace approximation method is to approximate the integral of a function $\int f(\mathbf{x})d\mathbf{x}$ by fitting a Gaussian at the maximum $\hat{\mathbf{x}}$ of $f(\mathbf{x})$, and further compute the volume under the Gaussian [319]:

$$\int f(\mathbf{x})d\mathbf{x} \approx (2\pi)^{N_x/2} f(\hat{\mathbf{x}}) \Big| -\nabla\nabla \log f(\mathbf{x})\Big|^{-1/2} \tag{48}$$

The covariance of the fitted Gaussian is determined by the Hessian matrix of $\log f(\mathbf{x})$ at $\hat{\mathbf{x}}$. It is also used to approximate the posterior distribution with a Gaussian centered at the MAP estimate, which is partially justified by the fact that under certain regularity conditions the posterior distribution asymptotically approaches Gaussian distribution as the number of samples increases to infinity. Laplace approximation is useful in the MAP or ML framework, this method usually works for the unimodal distribution but produces a poor approximation result for the multimodal distribution, especially in a high-dimensional space. Some new development of Laplace approximation can be found in MacKay's paper [319].

B. Iterative Quadrature

Iterative quadrature is an important numerical approximation method, which was widely used in computer graphics and physics in the early days. One of the popular quadrature methods is Gaussian quadrature [117], [377]. In particular, a finite integral is approximated by a weighted sum of samples of the integrand based on some quadrature formula

$$\int_a^b f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \sum_{k=1}^m c_k f(\mathbf{x}_k), \tag{49}$$

where $p(\mathbf{x})$ is treated as a weighting function, and $\mathbf{x}_k$ is the quadrature point. For example, it can be the k-th zero the m-th order orthogonal Hermite polynomial $H_m(\mathbf{x})$,[34] for which the weights are given by

$$c_k = \frac{2^{m-1}m!\sqrt{m}}{m^2(H_{m-1}(\mathbf{x}_k))^2}.$$

The approximation is good if $f(\mathbf{x})$ is a polynomial of degree not bigger than $2m-1$. The values $\mathbf{x}_k$ are determined by the weighting function $p(\mathbf{x})$ in the interval $[a, b]$.[35] This method can produce a good approximation if the nonlinear function is smooth. Quadrature methods, alone or combined with other methods, were used in nonlinear filtering (e.g. [475], [287]). The quadrature formulae will be used after a centering about the current estimate of the conditional mean and rescaling according to the current estimate of the covariance.

C. Mulitgrid Method and Point-Mass Approximation

If the state is discrete and finite (or it can be discretized and approximated as finite), grid-based methods can provide a good solution and optimal way to update the filtered density $p(\boldsymbol{z}_n|\mathbf{y}_{0:n})$ (To discriminate from the continuous-valued state $\mathbf{x}$, we denote the discrete-valued state as $\boldsymbol{z}$ from now on). Suppose the discrete state $\boldsymbol{z} \in \mathbb{N}$ consists of a finite number of distinct discrete states $\{1, 2, \cdots, N_z\}$. For the state space $\boldsymbol{z}_{n-1}$, let $w^i_{n-1|n-1}$ denote the conditional probability of each $\boldsymbol{z}^i_{n-1}$ given measurement up to

[33] They degenerate into the mean and error covariance when (11a) is linear Gaussian, and the filter reduces to the Kalman-Bucy filter.

[34] Other orthogonal approximation techniques can be also considered.

[35] The *Fundamental Theorem of Gaussian Quadrature* states that: the abscissas of the m-point Gaussian quadrature formula are precisely the roots of the orthogonal polynomial for the same interval and weighting function.

$n-1$, i.e. $p(\boldsymbol{z}_{n-1}=\boldsymbol{z}^i|\mathbf{y}_{0:n-1})=w^i_{n-1|n-1}$. Then the posterior pdf at $n-1$ can be represented as

$$p(\boldsymbol{z}_{n-1}|\mathbf{y}_{0:n-1})=\sum_{i=1}^{N_z} w^i_{n-1|n-1}\delta(\boldsymbol{z}_{n-1}-\boldsymbol{z}^i_{n-1}), \tag{50}$$

and the prediction and filtering equations are further derived as

$$p(\boldsymbol{z}_n|\mathbf{y}_{0:n-1}) = \sum_{i=1}^{N_z} w^i_{n|n-1}\delta(\boldsymbol{z}_n-\boldsymbol{z}^i_n), \tag{51}$$

$$p(\boldsymbol{z}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^{N_z} w^i_{n|n}\delta(\boldsymbol{z}_n-\boldsymbol{z}^i_n), \tag{52}$$

where

$$w^i_{n|n-1} = \sum_{j=1}^{N_z} w^j_{n-1|n-1}p(\boldsymbol{z}^i_n|\boldsymbol{z}^j_n), \tag{53}$$

$$w^i_{n|n} = \frac{w^i_{n|n-1}p(\mathbf{y}_n|\boldsymbol{z}^i_n)}{\sum_{j=1}^{N_z} w^j_{n|n-1}p(\mathbf{y}_n|\boldsymbol{z}^j_n)}. \tag{54}$$

If the state space is continuous, the approximate-grid based method can be similarly derived (e.g. [19]). Namely, we can always discretize the state space into N_z discrete cell states, then a grid-based method can be further used to approximate the posterior density. The grid must be sufficiently dense to obtain a good approximation, especially when the dimensionality of $N_\mathbf{x}$ is high, however the increase of N_z will increase the computational burden dramatically. If the state space is not finite, then the accuracy of grid-based methods is not guaranteed. As we will discuss in Section VII, HMM filter is quite fitted to the grid-based methods. The disadvantage of grid-based method is that it requires the state space cannot be partitioned unevenly to give a great resolution to the state with high density [19]. Some adaptive grid-based methods were proposed to overcome this drawback [65]. Given the predefined grid, different methods were used to approximate the functions and carry out the dynamic Bayesian estimation and forecasting [62], [258], [271], [424], [373], [372].

In studying the nonlinear filtering, Bucy [62] and Bucy and Senne [63] introduced the point-mass method, which is a global function approximation method. Such method uses a simple rectangular grid, spline basis, step function, the quadrature methods are used to determine the grid points [64], [475], [271], the number of grid points is prescribed to provide an adequate approximation. The density is assumed to be represented by a set of point masses which carry the information about the data; mesh grid and directions are given in terms of eignevalues and eigenvectors of conditional error covariance; the floating grid is centered at the current mean estimate and rotated from the state coordinate frame into the principal axes of error ellipsoid (covariance); the grid along the axes is chosen to extend over a sufficient distance to cover the true state. For the multimodal density, it is suggested to define a grid for each mode rather than for the entire density. Even so, the computation of multigrid-based point-mass approximation method is nontrivial and the complexity is high (see [271]).

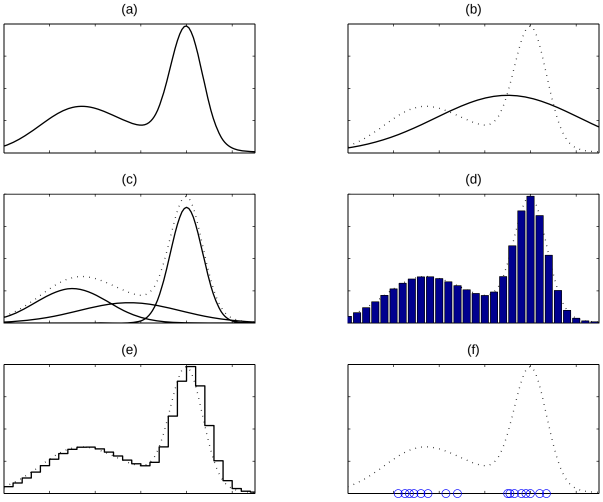

Fig. 5. Illustration of non-Gaussian distribution approximation: (a) true distribution; (b) Gaussian approximation; (c) Gaussian sum approximation; (d) histogram approximation; (e) Riemannian sum (step function) approximation; (f) Monte Carlo sampling approximation.

Another sophisticated approximation method, based on the piecewise constant approximation of density, was proposed in [271], [258]. The method is similar but not identical to the point-mass approximation. It defines a simple grid based on tiling the state space with a number of identical parallelepipeds, over each of them the density approximation is constant, and the integration is replaced by a discrete linear convolution problem. The method also allows error propagation analysis along the calculation [271].

D. Moment Approximation

Moment approximation is targeted at approximating the moments of density, including mean, covariance, and higher order moments. The approximation of the first two moments is widely used in filtering [367]. Generally, we can empirically use the sample moment to approximate the true moment, namely

$$m_k=\mathbb{E}[\mathbf{x}^k]=\int_X \mathbf{x}^k p(\mathbf{x})d\mathbf{x}=\frac{1}{N}\sum_{i=1}^{N}|\mathbf{x}^{(i)}|^k$$

where m_k denotes the m-th order moment and $\mathbf{x}^{(i)}$ are the samples from true distribution. Among many, Gram-Charlier and Edgeworth expansion are two popular higher-order moment approximation approaches. Due to space constraint, we cannot run into the details here, and refer the reader to [ ] for more information. The applications of higher-order moment approximation to nonlinear filters are found in [427]. However, the computation cost of these approaches are rather prohibitive, especially in high-dimensional space.

E. Gaussian Sum Approximation

Different from the linearized EKF or second-order approximation filter that only concentrate on the vicinity of the mean estimate, Gaussian sum approximation uses a weighted sum of Gaussian densities to approximate the posterior density(the so-called Gaussian mixture model):

$$p(\mathbf{x}) = \sum_{j=1}^{m} c_j \mathcal{N}(\hat{\mathbf{x}}_j, \Sigma_j) \tag{55}$$

where the weighting coefficients $c_j > 0$ and $\sum_{j=1}^{m} c_j = 1$. The approximation is motivated by the observation that any non-Gaussian density can be approximated to some accurate degree by a sufficiently large number of Gaussian mixture densities, which admits tractable solution by calculating individual first and second order moments. The Gaussian sum filter [421], [8], essentially uses this idea and runs a bank of EKFs in parallel to obtain the suboptimal estimate. The following theorem reads the underlying principle:

Theorem 2: [12] Suppose in equations (2a)(2b) the noise vectors $\mathbf{d}_n$ and $\mathbf{v}_n$ are white Gaussian noises with zero mean and covariances $\Sigma_\mathbf{d}$ and $\Sigma_\mathbf{v}$, respectively. If $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n|n-1}, \Sigma_{n|n-1})$, then for fixed $\mathbf{g}(\cdot)$, $\boldsymbol{\mu}_{n|n-1}$ and $\Sigma_\mathbf{v}$, the filtered density $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = c_n p(\mathbf{x}_n|\mathbf{y}_{0:n-1})p(\mathbf{y}_n|\mathbf{x}_n)$ (where c_n is a normalizing constant) converges uniformly to $\mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n|n}, \Sigma_{n|n})$ as $\Sigma_{n|n-1} \rightarrow \mathbf{0}$. If $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n|n}, \Sigma_{n|n})$, then for fixed $\mathbf{f}(\cdot)$, $\boldsymbol{\mu}_{n|n}$ and $\Sigma_\mathbf{d}$, the predicted density $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n$ converges uniformly to $\mathcal{N}(\mathbf{x}_{n+1}; \boldsymbol{\mu}_{n+1|n}, \Sigma_{n+1|n})$ as $\Sigma_{n|n} \rightarrow \mathbf{0}$.

Some new development of Gaussian sum filter (as well as Gaussian-quadrature filter) is referred to [235], [234], where the recursive Bayesian estimation is performed, and no Jacobian matrix evaluation is needed (similar to the unscented transformation technique discussed below).

F. Deterministic Sampling Approximation

The deterministic sampling approximation we discussed below is a kind of method called unscented transformation (UT). [36] It can be viewed as a special numerical method to approximate the sufficient statistics of mean and covariance. The intuition of UT is somewhat similar to the point-mass approximation discussed above: it uses the so-called sigma-points with additional skewed parameters to cover and propagate the information of the data. Based on UT, the so-called unscented Kalman filter (UKF) was derived. The most mentionable advantage of UKF over EKF is its derivative-nonlinear estimation (no need of calculation of Jacobians and Hessians), though its computational complexity is little higher than the EKF's. There are also other derivative-free estimation techniques available. In [355], a polynomial approximation using interpolation formula was developed and subsequently applied to nonlinear Kalman filtering, with a name of `nprKF`. The nprKF filtering technique was also used to train the neural networks [166].

The idea of derivative-free state estimation is following: In order to estimate the state information (mean, covariance, and higher-order moments) after a nonlinear transformation, it is favorable to approximate the probability distribution directly instead of approximating the nonlinear function (by linear localization) and apply the Kalman filter in the transformed domain. The derivative-free UKF can overcome the drawback by using a deterministic sampling approach to calculate the mean and covariance. In particular, the $(2N_\mathbf{x} + 1)$ sigma-points are generated and propagated through the true nonlinearity, and the weighted mean and covariance are further calculated [242], [474]. Compared with the EKF's first-order accuracy, the estimation accuracy of UKF is improved to the third-order for Gaussian data and at least second-order for non-Gaussian data [242], [474].

However, UT and UKF often encounter the ill-conditioned [37] problem of covariance matrix in practice (though it is theoretically positive semi-definite), although the regularization trick and square-root UKF [460] can alleviate this. For enhancing the numerical robustness, we propose another derivative-free KF based on singular-value decomposition (SVD).

The SVD-based KF is close in spirit to UKF, it only differs in that the UT is replaced by SVD and the sigma-point covariance becomes an eigen-covariance matrix, in which the pairwise (±) eigenvectors are stored into the column vector of the new covariance matrix. The number of eigen-points to store is the same as the sigma points in UT. The idea behind SVD is simple: We assume the covariance matrix be characterized by a set of eigenvectors which correspond to a set of eigenvalues.[38] For the symmetric covariance matrix $\mathbf{C}$, ED and SVD are equivalent, and the eigenvalues are identical to the singular values. We prefer to calculate SVD instead of eigen-decomposition because the former is more numerically robust. The geometrical interpretation of SVD compared with UT is illustrated in Fig. 6. By SVD of square-root of the covariance matrix $\mathbf{C}$

$$\mathbf{C}^{1/2} = \mathbf{U} \left[\begin{array}{cc} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array} \right] \mathbf{V}^T \tag{56}$$

where $\mathbf{C}^{1/2} = \mathtt{chol}(\mathbf{C})$ and `chol` represents Cholesky factorization; $\mathbf{S}$ is a diagonal matrix $\mathbf{S} = \mathrm{diag}\{s_1, \cdots, s_k\}$, when $\mathbf{C}^{1/2}$ is symmetric, $\mathbf{U} = \mathbf{V}$. Thus the eigenvalues are $\lambda_k = s_k^2$, and the eigenvectors of $\mathbf{C}$ is represented by the column vectors of matrix $\mathbf{U}\mathbf{U}^T$. A Monte Carlo sampling of a two-dimensional Gaussian distribution passing a Gaussian nonlinearity is shown in Fig. 6. As shown, the sigma points and eigen-points can both approximately characterize the structure of the transformed covariance

[36] The name is somehow *ad hoc* and the word "unscented" does not imply its original meaning (private communication with S. Julier).

[37] Namely, the conditional number of the covariance matrix is very large.

[38] By assuming that, we actually assume that the sufficient statistics of underlying data is second-order, which is quite not true.

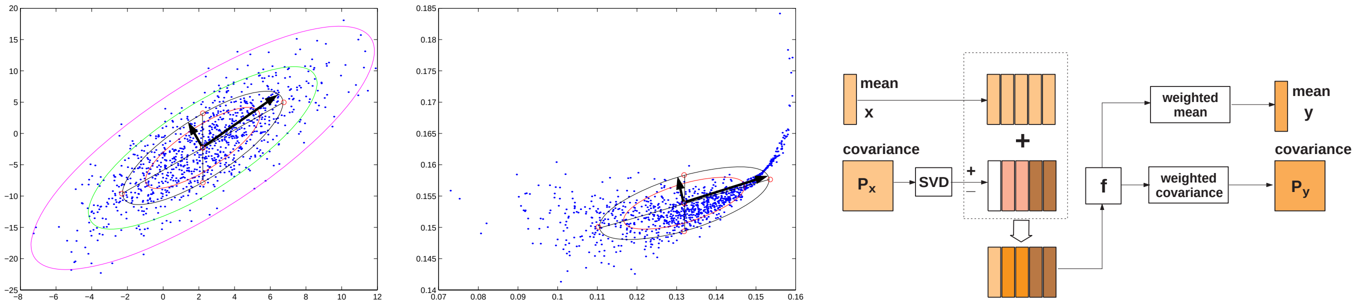

Fig. 6. SVD against Choleksy factorization in UT. **Left:** 1,000 data points are generated from a two-dimensional Gaussian distribution. The small red circles linked by two thin lines are sigma points using UT (parameters $\alpha = 1, \beta = 2, \kappa = 0$; see the paper [ ] for notations); the two black arrows are the eigenvector multiplied by $\rho = 1.4$; the ellipses from inside to outside correspond to the scaling factors $\sigma = 1, 1.4, 2, 3$; **Middle:** After the samples pass a Gaussian nonlinearity, the sigma points and eigen-points are calculated again for the transformed covariance; **Right:** SVD-based derivative-free estimation block diagram.

matrix. For state space equations (2a)(2b) with additive noise, the SVD-based derivative-free KF algorithm for the state estimation is summarized in Table X in Appendix E.

G. Monte Carlo Sampling Approximation

Monte Carlo methods use statistical sampling and estimation techniques to evaluate the solutions to mathematical problems. Monte Carlo methods have three categories: (i) *Monte Carlo sampling*, which is devoted to developing efficient (variance-reduction oriented) sampling technique for estimation; (ii) *Monte Carlo calculation*, which is aimed to design various random or pseudo-random number generators; and (iii) *Monte Carlo optimization*, which is devoted to applying the Monte Carlo idea to optimize some (nonconvex or non-differentiable) functions, to name a few, simulated annealing [257], dynamic weighting [494], [309], [298], and genetic algorithm. In recent decades, modern Monte Carlo techniques have attracted more and more attention and have been developed in different areas, as we will briefly overview in this subsection. Only Monte Carlo sampling methods are discussed. A detailed background of Monte Carlo methods can refer to the books [168], [389], [306], [386] and survey papers [197], [318].

The underlying mathematical concept of Monte Carlo approximation is simple. Consider a statistical problem estimating a Lebesque-Stieltjes integral:

$$\int_X f(\mathbf{x})dP(\mathbf{x}),$$

where $f(\mathbf{x})$ is an integrable function in a measurable space. As a *brute force* technique, Monte Carlo sampling uses a number of (independent) random variables in a probability space $(\Omega, \mathcal{F}, P)$ to approximate the true integral. Provided one draws a sequence of N_p i.i.d. random samples $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_p)}\}$ from probability distribution $P(\mathbf{x})$, then the Monte Carlo estimate of $f(\mathbf{x})$ is given by

$$\hat{f}_{N_p} = \frac{1}{N_p}\sum_{i=1}^{N_p} f(\mathbf{x}^{(i)}), \tag{57}$$

for which $\mathbb{E}[\hat{f}_{N_p}] = \mathbb{E}[f]$ and $\mathrm{Var}[\hat{f}_{N_p}] = \frac{1}{N_p}\mathrm{Var}[f] = \frac{\sigma^2}{N_p}$ (see Appendix A for a general proof). By the *Kolmogorov Strong Law of Large Numbers* (under some mild regularity conditions), $\hat{f}_{N_p}(\mathbf{x})$ converges to $\mathbb{E}[f(\mathbf{x})]$ almost surely (a.s.) and its convergence rate is assessed by the *Central Limit Theorem*

$$\sqrt{N_p}(\hat{f}_{N_p} - \mathbb{E}[f]) \sim \mathcal{N}(0, \sigma^2),$$

where σ^2 is the variance of $f(\mathbf{x})$. Namely, the error rate is of order $\mathcal{O}(N_p^{-1/2})$, which is slower than the order $\mathcal{O}(N_p^{-1})$ for deterministic quadrature in one-dimensional case. One crucial property of Monte Carlo approximation is the estimation accuracy is independent of the dimensionality of the state space, as opposed to most deterministic numerical methods.[39] The variance of estimate is inversely proportional to the number of samples.

There are two fundamental problems arising in Monte Carlo sampling methods: (i) *How to draw random samples* $\{\mathbf{x}^{(i)}\}$ *from a probability distribution* $P(\mathbf{x})$*?*; and (ii) *How to estimate the expectation of a function w.r.t. the distribution or density, i.e.* $\mathbb{E}[f(\mathbf{x})] = \int f(\mathbf{x})dP(\mathbf{x})$*?* The first problem is a design problem, and the second one is an inference problem invoking integration. Besides, several central issues are concerned in the Monte Carlo sampling:

- **Consistency**: An estimator is *consistent* if the estimator converges to the true value almost surely as the number of observations approaches infinity.
- **Unbiasedness**: An estimator is *unbiased* if its expected value is equal to the true value.
- **Efficiency**: An estimator is *efficient* if it produces the smallest error covariance matrix among all unbiased estimators, it is also regarded optimally using the information in the measurements. A well-known efficiency criterion is the Cramér-Rao bound.
- **Robustness**: An estimator is *robust* if it is insensitive to the gross measurement errors and the uncertainties of the model.
- **Minimal variance**: Variance reduction is the central issue of various Monte Carlo approximation methods, most improvement techniques are variance-reduction oriented.

[39] Note that, however, it doesn't mean Monte Carlo methods can beat the curse of dimensionality, an issue that will be discussed in Section VI-P.

In the rest of this subsection, we will provide a brief introduction of many popular Monte Carlo method relevant to our paper. No attempt is made here to present a complete and rigorous theory. For more theoretical details or applications, reader is referred to the books [199], [389], [168], [306].

G.1 Importance Sampling

Importance sampling (IS) was first introduced by Marshall [324] and received a well-founded treatment and discussion in the seminal book by Hammersley and Hanscomb [199]. The objective of importance sampling is aimed to sample the distribution in the region of "importance" in order to achieve computational efficiency. This is important especially for the high-dimensional space where the data are usually sparse, and the region of interest where the target lies in is relatively small in the whole data space. The idea of importance sampling is to choose a proposal distribution $q(\mathbf{x})$ in place of the true probability distribution $p(\mathbf{x})$, which is hard-to-sample. The support of $q(\mathbf{x})$ is assumed to cover that of $p(\mathbf{x})$. Rewriting the integration problem as

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}, \tag{58}$$

Monte Carlo importance sampling is to use a number of (say N_p) independent samples drawn from $q(\mathbf{x})$ to obtain a weighted sum to approximate (58):

$$\hat{f} = \frac{1}{N_p}\sum_{i=1}^{N_p} W(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)}), \tag{59}$$

where $W(\mathbf{x}^{(i)}) = p(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$ are called the *importance weights* (or importance ratios). If the normalizing factor of $p(\mathbf{x})$ is not known, the importance weights can be only evaluated up to a normalizing constant, hence $W(\mathbf{x}^{(i)}) \propto p(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$. To ensure that $\sum_{i=1}^{N_p} W(\mathbf{x}^{(i)}) = 1$, we normalize the importance weights to obtain

$$\hat{f} = \frac{\frac{1}{N_p}\sum_{i=1}^{N_p} W(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)})}{\frac{1}{N_p}\sum_{j=1}^{N_p} W(\mathbf{x}^{(j)})} \equiv \sum_{i=1}^{N_p} \tilde{W}(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)}), \tag{60}$$

where $\tilde{W}(\mathbf{x}^{(i)}) = \frac{W(\mathbf{x}^{(i)})}{\sum_{j=1}^{N_p} W(\mathbf{x}^{(j)})}$ are called the *normalized importance weights*. The variance of importance sampler estimate (59) is given by [59]

$$\begin{aligned}
\mathrm{Var}_q[\hat{f}] &= \frac{1}{N_p}\mathrm{Var}_q[f(\mathbf{x})W(\mathbf{x})] \\
&= \frac{1}{N_p}\mathrm{Var}_q[f(\mathbf{x})p(\mathbf{x})/q(\mathbf{x})] \\
&= \frac{1}{N_p}\int \Big[\frac{f(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})} - \mathbb{E}_p[f(\mathbf{x})]\Big]^2 q(\mathbf{x})d\mathbf{x} \\
&= \frac{1}{N_p}\int \Big[\Big(\frac{(f(\mathbf{x})p(\mathbf{x}))^2}{q(\mathbf{x})}\Big) - 2p(\mathbf{x})f(\mathbf{x})\mathbb{E}_p[f(\mathbf{x})]\Big]d\mathbf{x} \\
&\quad + \frac{(\mathbb{E}_p[f(\mathbf{x})])^2}{N_p} \\
&= \frac{1}{N_p}\int \Big[\Big(\frac{(f(\mathbf{x})p(\mathbf{x}))^2}{q(\mathbf{x})}\Big)\Big]d\mathbf{x} - \frac{(\mathbb{E}_p[f(\mathbf{x})])^2}{N_p}.
\end{aligned} \tag{61}$$

The variance can be reduced when an appropriate $q(\mathbf{x})$ is chosen to (i) match the shape of $p(\mathbf{x})$ so as to approximate the true variance; or (ii) match the shape of $|f(\mathbf{x})|p(\mathbf{x})$ so as to further reduce the true variance.[40] Importance sampling estimate given by (60) is *biased* (thus a.k.a. biased sampling)[41] but *consistent*, namely the bias vanishes rapidly at a rate $\mathcal{O}(N_p)$. Provided q is appropriately chosen, as $N_p \to \infty$, from the *Weak Law of Large Numbers*, we know

$$\hat{f} \to \frac{\mathbb{E}_q[W(\mathbf{x})f(\mathbf{x})]}{\mathbb{E}_q[W(\mathbf{x})]}.$$

It was also shown [180] that if $\mathbb{E}[\tilde{W}(\mathbf{x})] < \infty$ and $\mathbb{E}[\tilde{W}(\mathbf{x})f^2(\mathbf{x})] < \infty$, (60) converges to $\mathbb{E}_p[f]$ a.s. and the *Lindeberg-Lévy Central Limit Theorem* still holds:

$$\sqrt{N_p}(\hat{f} - \mathbb{E}_p[f]) \sim \mathcal{N}(0, \Sigma_f),$$

where

$$\Sigma_f = \mathrm{Var}_q\Big[\tilde{W}(\mathbf{x})(f(\mathbf{x}) - \mathbb{E}_p[f(\mathbf{x})])\Big]. \tag{62}$$

A measure of efficiency of importance sampler is given by the normalized version of (62): $\frac{\Sigma_f}{\mathrm{Var}_p[f]}$, which is related to the effective sample size, as we will discuss later.

Remarks:

- Importance sampling is useful in two ways [86]: (i) it provides an elegant way to reduce the variance of the estimator (possibly *even* less than the true variance); and (ii) it can be used when encountering the difficulty to sample from the true probability distribution directly.
- As shown in many empirical experiments [318], importance sampler (proposal distribution) should have a heavy tail so as to be insensitive to the outliers. The super-Gaussian distributions usually have long tails, with kurtosis bigger than 3. Alternatively, we can roughly verify the "robust" behavior from the *activation function* defined as $\varphi(\mathbf{x}) = \frac{-d\log q(\mathbf{x})}{d\mathbf{x}}$: if $\varphi(\mathbf{x})$ is bounded, $q(\mathbf{x})$ has a long tail, otherwise not.

[40] In an ideal case, $q(\mathbf{x}) \propto |f(\mathbf{x})|p(\mathbf{x})$, the variance becomes zero.

[41] It is unbiased only when all of importance weights $\tilde{W}^{(i)} = 1$ (namely $p(\cdot) = q(\cdot)$, and it reduces to the estimate $\hat{f}_{N_p}$ in (57)).

- Although theoretically the bias of importance sampler vanishes at a rate $\mathcal{O}(N_p)$, the accuracy of estimate is not guaranteed even with a large N_p. If $q(\cdot)$ is not close to $p(\cdot)$, it can be imagined that the weights are very uneven, thus many samples are almost useless because of their negligible contributions. In a high-dimensional space, the importance sampling estimate is likely dominated by a few samples with large importance weights.
- Importance sampler can be mixed with Gibbs sampling or Metropolis-Hastings algorithm to produce more efficient techniques [40], [315].
- Some advanced (off-line) importance sampling schemes, such as adaptive importance sampling [358], annealed importance sampling [348], [350], smoothed importance sampling [49], [322], dynamic importance sampling [494], (regularized) greedy importance sampling, Bayesian importance sampling [382] etc. are also available.

G.2 Rejection Sampling

Rejection sampling (e.g. [199]) is useful when we know (pointwise) the upper bound of underlying distribution or density. The basic assumption of rejection sampling is similar to that of importance sampling. Assume there exists a known constant $C < \infty$ such that $p(\mathbf{x}) < Cq(\mathbf{x})$ for every $\mathbf{x} \in X$, the sampling procedure reads as follows:

- Generate a uniform random variable $u \sim \mathcal{U}(0, 1)$;
- Draw a sample $\mathbf{x} \sim q(\mathbf{x})$;
- If $u < \frac{p(\mathbf{x})}{Cq(\mathbf{x})}$, return $\mathbf{x}$, otherwise go to step 1.

The samples from rejection sampling are *exact*, and the acceptance probability for a random variable is inversely proportional to the constant C. In practice, the choice of constant C is critical (which relies on the knowledge of $p(\mathbf{x})$): if C is too small, the samples are not reliable because of low rejection rate; if C is too large, the algorithm will be inefficient since the acceptance rate will be low. In Bayesian perspective, rejection sampling naturally incorporates the normalizing denominator into the constant C. If the prior $p(\mathbf{x})$ is used as proposal distribution $q(\mathbf{x})$, and the likelihood $p(\mathbf{y}|\mathbf{x}) \le C$ where C is assumed to be known, the bound on the posterior is given by

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \le \frac{Cq(\mathbf{x})}{p(\mathbf{y})} \equiv C'q(\mathbf{x}),$$

and the acceptance rate for drawing a sample $\mathbf{x} \in X$ is

$$\frac{p(\mathbf{x}|\mathbf{y})}{C'q(\mathbf{x})} = \frac{p(\mathbf{y}|\mathbf{x})}{C}, \tag{63}$$

which can be computed even the normalizing constant $p(\mathbf{y})$ is not known.

Remarks:

- The draws obtained from rejection sampling are exact [414].
- The prerequisite of rejection sampling is the prior knowledge of constant C, which is sometimes unavailable.
- It usually takes a long time to get the samples when the ratio $p(\mathbf{x})/Cq(\mathbf{x})$ is close to zero [441].

G.3 Sequential Importance Sampling

A good proposal distribution is essential to the efficiency of importance sampling, hence how to choose an appropriate proposal distribution $q(\mathbf{x})$ is the key to apply a successful importance sampling [200], [506], [266]. However, it is usually difficult to find a good proposal distribution especially in a high-dimensional space. A natural way to alleviate this problem is to construct the proposal distribution sequentially, which is the basic idea of sequential importance sampling (SIS) [198], [393].

In particular, if the proposal distribution is chosen in a factorized form [144]

$$q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = q(\mathbf{x}_0)\prod_{t=1}^{n} q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}), \tag{64}$$

then the importance sampling can be performed recursively. We will give the derivation detail when discussing the SIS particle filter in Section VI. At this moment, we consider a simplified (unconditional pdf) case for the ease of understanding. According to the “telescope” law of probability, we have the following:

$$\begin{aligned} p(\mathbf{x}_{0:n}) &= p(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0)\cdots p(\mathbf{x}_n|\mathbf{x}_0, \cdots, \mathbf{x}_{n-1}), \\ q(\mathbf{x}_{0:n}) &= q_0(\mathbf{x}_0)q_1(\mathbf{x}_1|\mathbf{x}_0)\cdots q_n(\mathbf{x}_n|\mathbf{x}_0, \cdots, \mathbf{x}_{n-1}). \end{aligned}$$

Hence the importance weights $W(\mathbf{x}_{0:n})$ can be written as

$$W(\mathbf{x}_{0:n}) = \frac{p(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0)\cdots p(\mathbf{x}_n|\mathbf{x}_0, \cdots, \mathbf{x}_{n-1})}{q_0(\mathbf{x}_0)q_1(\mathbf{x}_1|\mathbf{x}_0)\cdots q_n(\mathbf{x}_n|\mathbf{x}_0, \cdots, \mathbf{x}_{n-1})},$$

which be recursively calculated as

$$W_n(\mathbf{x}_{0:n}) = W_{n-1}(\mathbf{x}_{0:n-1})\frac{p(\mathbf{x}_n|\mathbf{x}_{0:n-1})}{q_n(\mathbf{x}_n|\mathbf{x}_{0:n-1})}.$$

Remarks:

- The advantage of SIS is that it doesn’t rely on the underlying Markov chain. Instead, many i.i.d. replicates are run to create an importance sampler, which consequently improves the efficiency. The disadvantage of SIS is that the importance weights may have large variances, resulting in inaccurate estimate [315].
- SIS method can be also used in a non-Bayesian computation, such as evaluation of the likelihood function in the missing-data problem [266].
- It was shown in [266] that the unconditional variance of the importance weights increases over time, which is the so-called weight degeneracy problem: Namely, after a few iterations of algorithm, only few or one of $W(\mathbf{x}^{(i)})$ will be nonzero. This is disadvantageous since a lot of computing effort is wasted to update those trivial weight coefficients. In order to cope with this situation, resampling step is suggested to be used after weight normalization.

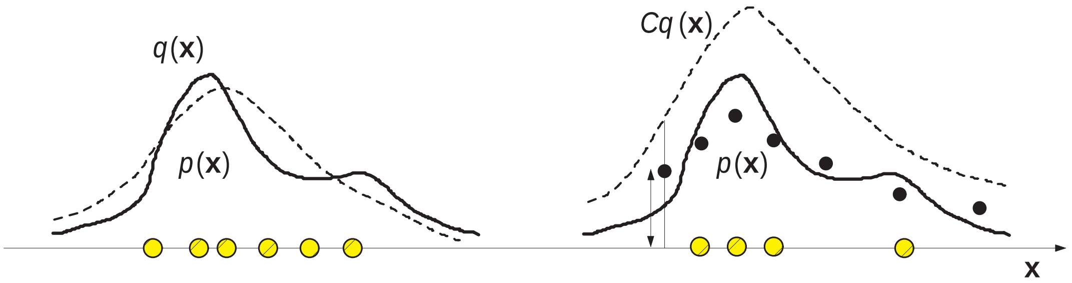

Fig. 7. Illustration of importance sampling (left) and acceptance-rejection sampling (right). $p(\mathbf{x})$ is the true pdf (solid line), $q(\mathbf{x})$ is the proposal distribution (dashed line). For rejection sampling, some random samples $\mathbf{x}^{(i)}$ are generated below $Cq(\mathbf{x})$, which are rejected if they lie in the region between $p(\mathbf{x})$ and $Cq(\mathbf{x})$; if they also lie below $p(\mathbf{x})$, they are accepted.

G.4 Sampling-Importance Resampling

The sampling-importance resampling (SIR) is motivated from the Bootstrap and jackknife techniques. Bootstrap technique is referred to a collection of computationally intensive methods that are based on resampling from the observed data [157], [408], [321]. The seminal idea originated from [155] and was detailed in [156], [157]. The intuition of bootstrapping is to evaluate the properties of an estimator through the empirical cumulative distribution function (cdf) of the samples instead of the true cdf.

In the statistics literature, Rubin [395], [396] first applied SIR technique to Monte Carlo inference, in which the resampling is inserted between two importance sampling steps. The resampling step[42] is aimed to eliminate the samples with small importance weights and duplicate the samples with big weights. The generic principle of SIR proceeds as follows:

- Draw N_p random samples $\{\mathbf{x}^{(i)}\}_{i=1}^{N_p}$ from proposal distribution $q(\mathbf{x})$;
- Calculate importance weights $W^{(i)} \propto p(\mathbf{x})/q(\mathbf{x})$ for each sample $\mathbf{x}^{(i)}$;
- Normalize the importance weights to obtain $\tilde{W}^{(i)}$;
- Resample with replacement N times from the discrete set $\{\mathbf{x}^{(i)}\}_{i=1}^{N_p}$, where the probability of resampling from each $\mathbf{x}^{(i)}$ is proportional to $\tilde{W}^{(i)}$.

Remarks (on features):

- Resampling usually (but not necessarily) occurs between two importance sampling steps. In resampling step, the particles and associated importance weights $\{\mathbf{x}^{(i)}, \tilde{W}^{(i)}\}$ are replaced by the new samples with equal importance weights (i.e. $\tilde{W}^{(i)} = 1/N_p$). Resampling can be taken at every step or only taken if regarded necessary.
- As justified in [303], resampling step plays an critical role in importance sampling since (i) if importance weights are uneven distributed, propagating the "trivial" weights through the dynamic system is a waste of computing power; (ii) when the importance weights are skewed, resampling can provide chances for selecting "important" samples and rejuvenate the sampler for the future use, though resampling doesn't necessarily improve the current state estimate because it also introduces extra Monte Carlo variation.
- Resampling schedule can be deterministic or dynamic [304], [308]. In deterministic framework, resampling is taken at every k time step (usually $k = 1$). In a dynamic schedule, a sequence of thresholds (that can be constant or time-varying) are set up and the variance of the importance weights are monitored; resampling is taken only when the variance is over the threshold.

The validity of inserting a resampling step in SIS algorithm has been justified by [395], [303], since resampling step also brings extra variation, some special schemes are needed. There are many types of resampling methods available in the literature:

1. **Multinomial resampling** [395], [414], [193]: the procedure reads as follows (see also [19])

   - Produce a uniform distribution $u \sim \mathcal{U}(0,1)$, construct a cdf for importance weights (see Fig. 1), calculate $s_i = \sum_{j=1}^{i} \tilde{W}^{(j)}$;
   - Find s_i s.t. $s_{i-1} \leq u < s_i$, the particle with index i is chosen;
   - Given $\{\mathbf{x}^{(i)}, \tilde{W}^{(i)}\}$, for $j = 1, \cdots, N_p$, generate new samples $\mathbf{x}^{(j)}$ by duplicating $\mathbf{x}^{(i)}$ according to the associated $\tilde{W}^{(i)}$;
   - Reset $W^{(i)} = 1/N_p$.

   Multinomial resampling uniformly generates N_p new independent particles from the old particle set. Each particle is replicated N_i times (N_i can be zero), namely each $\mathbf{x}^{(i)}$ produces N_i children. Note that here $\sum_{i=1}^{N_p} N_i = N_p$, $\mathbb{E}[N_i] = N_p\tilde{W}^{(i)}$, $\mathrm{Var}[N_i] = N_p\tilde{W}^{(i)}(1-\tilde{W}^{(i)})$.

2. **Residual resampling** [211], [304]: Liu and Chen [304] suggested a partially deterministic resampling method. The two-step selection procedure is as follows [304]:

   - For each $i = 1, \cdots, N_p$, retain $k_i = \lfloor N_p\tilde{W}^{(i)} \rfloor$ copies of $\mathbf{x}_n^{(i)}$;
   - Let $N_r = N_p - k_1 - \cdots - k_{N_p}$, obtain N_r i.i.d. draws from $\{\mathbf{x}_n^{(i)}\}$ with probabilities proportional to $N_p\tilde{W}^{(i)} - k_i$ $(i = 1, \cdots, N_p)$;

[42] It is also called selection step.

- Reset $W^{(i)} = 1/N_p$.

Residual resampling procedure is computationally cheaper than the conventional SIR and achieves a lower sampler variance, and it doesn't introduce additional bias. Every particle in residual resampling is replicated.

3. **Systematic resampling** (or Minimum variance sampling) [259], [69], [70]: the procedure proceeds as follows:

```
• $u \sim \mathcal{U}(0,1)/N_p$; $j = 1$;$\ell = 0$;$i = 0$;
• do while $u < 1$
•     if $\ell > u$ then
•         $u = u + 1/N_p$; output $\mathbf{x}^{(i)}$
•     else
•         pick $k$ in $\{j, \cdots, N_p\}$
•         $i = \mathbf{x}^{(k)}$, $\ell = \ell + W^{(i)}$
•         switch $(\mathbf{x}^{(k)}, W^{(k)})$ with $(\mathbf{x}^{(j)}, W^{(j)})$
•         $j = j + 1$
•     end if
• end do
```

The systematic resampling treats the weights as continuous random variables in the interval $(0,1)$, which are randomly ordered. The number of grid points $\{u+k/N_p\}$ in each interval is counted [70]. Every particle is replicated and the new particle set is chosen to minimize $\text{Var}[N_i] = \mathbb{E}[(N_i - \mathbb{E}[N_i])^2]$. The complexity of systematic resampling is $\mathcal{O}(N_p)$.

4. **Local Monte Carlo resampling** [304]: The samples are redrawn using rejection method or Metropolis-Hastings method. We will briefly describe this scheme later in Section VI.

Remarks (on weakness):

- Different from the rejection sampling that achieves exact draws from the posterior, SIR only achieves approximate draws from the posterior as $N_p \to \infty$. Some variations of combining rejection sampling and importance sampling are discussed in [307].
- Although resampling can alleviate the weight degeneracy problem, it unfortunately introduces other problems [144]: after one resampling step, the simulated trajectories are not statistically independent any more, thus the convergence result due to the original central limit theorem is invalid; resampling causes the samples that have high importance weights to be statistically selected many times, thus the algorithm suffers from the *loss of diversity*.
- Resampling step also limits the opportunity to parallelize since all of the particles need to be combined for selection.

G.5 Stratified Sampling

The idea of stratified sampling is to distribute the samples evenly (or unevenly according to their respective variance) to the subregions dividing the whole space. Let $\hat{f}$ (statistics of interest) denote the Monte Carlo sample average of a generic function $f(\mathbf{x}) \in \mathbb{R}^{\mathbf{N_x}}$, which is attained from importance sampling. Suppose the state space is decomposed into two *equal, disjoint* strata (subvolumes), denoted as a and b, for stratified sampling, the total number of N_p samples are drawn from two strata separately and we have the stratified mean $\hat{f}' = \frac{1}{2}(\hat{f}_a + \hat{f}_b)$, and the stratified variance

$$\begin{aligned}\text{Var}[\hat{f}'] &= \frac{\text{Var}_a[\hat{f}] + \text{Var}_b[\hat{f}]}{4} \\ &= \frac{\text{Var}_a[f] + \text{Var}_b[f]}{2N_p}, \end{aligned} \tag{65}$$

where the second equality uses the facts that $\text{Var}_a[\hat{f}] = \frac{2}{N_p}\text{Var}_a[f]$ and $\text{Var}_b[\hat{f}] = \frac{2}{N_p}\text{Var}_b[f]$. In addition, it can be proved that[43]

$$\begin{aligned}N_p\text{Var}[\hat{f}] &= \text{Var}[f] \\ &= \frac{\text{Var}_a[f] + \text{Var}_b[f]}{2} + \frac{(\mathbb{E}_a[f] - \mathbb{E}_b[f])^2}{4} \\ &= N_p\text{Var}[\hat{f}'] + \frac{(\mathbb{E}_a[f] - \mathbb{E}_b[f])^2}{4} \\ &\geq N_p\text{Var}[\hat{f}'], \end{aligned} \tag{66}$$

where the third line follows from (65). Hence, the variance of stratified sampling $\text{Var}[\hat{f}']$ is never bigger than that of conventional Monte Carlo sampling $\text{Var}[\hat{f}]$, whenever $\mathbb{E}_a[f] \neq \mathbb{E}_b[f]$.

In general, provided the numbers of simulated samples from strata a and b are N_a and $N_b \equiv N_p - N_a$, respectively, (65) becomes

$$\text{Var}[\hat{f}'] = \frac{1}{4}\Big[\frac{\text{Var}_a[f]}{N_a} + \frac{\text{Var}_b[f]}{N_p - N_a}\Big], \tag{67}$$

the variance is minimized when

$$\frac{N_a}{N_p} = \frac{\sigma_a}{\sigma_a + \sigma_b}, \tag{68}$$

and the achieved minimum variance is

$$\text{Var}[\hat{f}']_{\min} = \frac{(\sigma_a + \sigma_b)^2}{4N_a}. \tag{69}$$

Remarks:

- In practice, it is suggested [376] that (67) be changed to the generic form

$$\text{Var}[\hat{f}'] = \frac{1}{4}\Big[\frac{\text{Var}_a[f]}{(N_a)^\alpha} + \frac{\text{Var}_b[f]}{(N_p - N_a)^\alpha}\Big],$$

  with an empirical value $\alpha = 2$.
- Stratified sampling works very well and is efficient in a not-too-high dimension space (say $N_{\mathbf{x}} \leq 4$), when $N_{\mathbf{x}}$ grows higher, the use of this technique is limited because one needs to estimate the variance of each stratum. In [376], an adaptive recursive stratified sampling procedure was developed to overcome this weakness (see [377] for implementation details).

[43] The inequality (66) is called the "parallel axis theorem" in physics.

G.6 Markov Chain Monte Carlo

Consider a state vector $\mathbf{x} \in \mathbb{R}^{N_x}$ in a probability space $(\Omega, \mathcal{F}, P)$, $K(\cdot,\cdot)$ is assumed to be a transition kernel in the state space, which represents the probability of moving from $\mathbf{x}$ to a point in a set $S \in \mathcal{B}$ (where $\mathcal{B}$ s a Borel σ-field on $\mathbb{R}^{N_x}$), a Markov chain is a sequence of random variable $\{\mathbf{x}_n\}_{n\geq 0}$ such that

$$\Pr(\mathbf{x}_n \in \mathcal{B}|\mathbf{x}_0, \cdots, \mathbf{x}_{n-1}) = \Pr(\mathbf{x}_n \in \mathcal{B}|\mathbf{x}_{n-1}),$$

and $K(\mathbf{x}_{n-1}, \mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$. A Markov chain is characterized by the properties of its states, e.g. transiency, periodicity, irreducibility,[44] and ergodicity. The foundation of Markov chain theory is the *Ergodicity Theorem*, which establishes under which a Markov chain can be analyzed to determine its steady state behavior.

Theorem 3: If a Markov chain is ergodic, then there exists a unique steady state distribution π independent of the initial state.

Markov chain theory is mainly concerned about finding the conditions under which there exists an invariant distribution Q and conditions under which iterations of transition kernel converge to the invariant distribution [185], [91]. The invariant distribution satisfies

$$\begin{aligned} Q(d\mathbf{x}') &= \int_X K(\mathbf{x}, d\mathbf{x}')\pi(\mathbf{x})d\mathbf{x}, \\ \pi(\mathbf{x}') &= \int_X K(\mathbf{x}, \mathbf{x}')\pi(\mathbf{x})d\mathbf{x} \end{aligned}$$

where $\mathbf{x}' \in S \subset \mathbb{R}^{N_x}$, and π is the density w.r.t. Lebesgue measure of Q such that $Q(d\mathbf{x}') = \pi(\mathbf{x}')d\mathbf{x}'$. The n-th iteration is thus given by $\int_X K^{(n-1)}(\mathbf{x}, d\mathbf{x}')K(\mathbf{x}', S)$. When $n \to \infty$, the initial state $\mathbf{x}$ will converge to the invariant distribution Q.

Markov chain Monte Carlo (MCMC) algorithms turn around the Markov chain theory. The invariant distribution or density is assumed to be known which correspond to the target density $\pi(\mathbf{x})$, but the transition kernel is unknown. In order to generate samples from $\pi(\cdot)$, the MCMC methods attempt to find a $K(\mathbf{x}, d\mathbf{x}')$ whose n-th iteration (for large n) converges to $\pi(\cdot)$ given an arbitrary starting point.

One of important properties of Markov chain is the *reversible condition* (a.k.a. "detailed balance")[45]

$$\pi(\mathbf{x})K(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}')K(\mathbf{x}', \mathbf{x}), \tag{70}$$

which states that the unconditional probability of moving $\mathbf{x}$ to $\mathbf{x}'$ is equal to the unconditional probability of moving $\mathbf{x}'$ to $\mathbf{x}$, where $\mathbf{x}, \mathbf{x}'$ are both generated from $\pi(\cdot)$. The distribution Q is thus the invariant distribution for $K(\cdot,\cdot)$.

In the MCMC sampling framework, unlike the importance or rejection sampling where the samples are drawn independently, the samples are generated by a *homogeneous, reversible, ergodic* Markov chain with invariant distribution Q.[46] Generally, we don't know how fast the Markov chain will converge to an equilibrium,[47] neither the rate of convergence or error bounds. Markov chain can be also used for importance sampling, in particular, we have the following theorem:

Theorem 4: [315] Let $K(\mathbf{x}, \mathbf{x}')$ denote a transitional kernel for a Markov chain on $\mathbb{R}^{N_x}$ with $p(\mathbf{x})$ as the density of its invariant distribution, let $q(\mathbf{x})$ denote the proposal distribution with $W(\mathbf{x})$ as importance weights, then $\int W(\mathbf{x})q(\mathbf{x})K(\mathbf{x}, \mathbf{x}')d\mathbf{x} = p(\mathbf{x}')$ for all $\mathbf{x}' \in \mathbb{R}^{N_x}$.

Metropolis-Hastings Algorithm. Metropolis-Hastings algorithm,[48] initially studied by Metropolis [329], and later redeveloped by Hastings [204], is a kind of MCMC algorithm whose transition is associated with the acceptance probability. Assume $q(\mathbf{x}, \mathbf{x}')$ as the proposal distribution (candidate target) that doesn't satisfy the reversibility condition, without loss of generality, suppose $\pi(\mathbf{x})q(\mathbf{x}, \mathbf{x}') > \pi(\mathbf{x}')q(\mathbf{x}', \mathbf{x})$, which means the probability moving from $\mathbf{x}$ to $\mathbf{x}'$ is bigger (more frequent) than the probability moving from $\mathbf{x}'$ to $\mathbf{x}$. Intuitively, we want to change this situation to reduce the number of moves from $\mathbf{x}$ to $\mathbf{x}'$. By doing this, we introduce a *probability of move*, $0 < \alpha(\mathbf{x}, \mathbf{x}') < 1$, if the move is not performed, the process returns $\mathbf{x}$ as a value from the target distribution. Hence the the transition from $\mathbf{x}$ to $\mathbf{x}'$ now becomes:

$$p_{\text{MH}}(\mathbf{x}, \mathbf{x}') = q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}'), \tag{71}$$

where $\mathbf{x} \neq \mathbf{x}'$. In order to make (71) satisfy reversibility condition, $\alpha(\mathbf{x}, \mathbf{x}')$ need to be set to [204]:

$$\alpha(\mathbf{x}, \mathbf{x}') = \begin{cases} \min\left[\frac{\pi(\mathbf{x}')q(\mathbf{x}', \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x}, \mathbf{x}')}, 1\right], & \text{if } \pi(\mathbf{x})q(\mathbf{x}, \mathbf{x}') > 0, \\ 1 & \text{otherwise} \end{cases} \tag{72}$$

Hence the probability that Markov process stays at $\mathbf{x}$ is given by

$$1 - \int_X q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}')d\mathbf{x}', \tag{73}$$

and the transition kernel is given by

$$\begin{aligned} K_{\text{MH}}(\mathbf{x}, d\mathbf{x}') &= q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}')d\mathbf{x}' \\ &\quad + \Big[1 - \int_X q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}')d\mathbf{x}'\Big]\delta_{\mathbf{x}}(d\mathbf{x}') \end{aligned} \tag{74}$$

In summary, a generic Metropolis-Hastings algorithm proceeds as follows [91]:

- For $i = 1, \cdots, N_p$, at iteration $n = 0$, draw a starting point $\mathbf{x}_0$ from a prior density;

[44] A Markov chain is called irreducible if any state can be reached from any other state in a finite number of iterations.

[45] Markov chains that satisfy the detailed balance are called reversible Markov chains.

[46] Note that the samples are independent only when the Markov chain is reversible and uniformly ergodic, otherwise they are dependent for which the Central Limit Theorem doesn't hold for the convergence.

[47] Only the samples that are drawn after the Markov chain approaches the equilibrium are regarded as the representative draws from the posterior. The time for Markov chain converging to equilibrium is called the *burn-in* time.

[48] This algorithm appears as the first entry of a recent list of great algorithms of 20th-century scientific computing.

- generate a uniform random variable $u \sim \mathcal{U}(0,1)$, and $\mathbf{x}' \sim q(\mathbf{x}_n, \cdot)$;
- If $u < \alpha(\mathbf{x}_n, \mathbf{x}')$, set $\mathbf{x}_{n+1} = \mathbf{x}'$, else $\mathbf{x}_{n+1} = \mathbf{x}_n$;
- $n = n+1$, repeat steps 2 and 3, until certain (say k) steps (i.e. burn-in time), store $x^{(i)} = x_k$.
- $i = i+1$, repeat the procedure until N_p samples are drawn, return the samples $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_p)}\}$.

Remarks:

- If the candidate-generating density is symmetric (e.g. random walk), i.e. $q(\mathbf{x}, \mathbf{x}') = q(\mathbf{x}', \mathbf{x})$, the probability of move reduces to $\pi(\mathbf{x}')/\pi(\mathbf{x})$, hence (72) will reduce to: if $\pi(\mathbf{x}') \geq \pi(\mathbf{x})$ the chain moves to $\mathbf{x}'$; and remains the same otherwise. This is the original algorithm in [329], it was also used in simulated annealing [257].
- The probability of move doesn't need the knowledge of normalizing constant of $\pi(\cdot)$.
- The draws are regarded as the samples from the target density only after the chain has passed the transient phase, the convergence to the invariant distribution occurs under mild regularity conditions (irreducibility and aperiodicity) [416].
- The efficiency of Metropolis algorithm is determined by the ratio of the accepted samples to the total number of samples. Too large or too small variance of the driving-force noise may result in inefficient sampling.
- It was suggested in [95] to use a Gaussian proposal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ for Metropolis-Hastings algorithm (or in MCMC step of particle filter), where the mean and covariance are determined by

$$\begin{aligned}
\boldsymbol{\mu} &= \frac{\sum_{i=1}^{N_p} W^{(i)} \mathbf{x}^{(i)}}{\sum_{i=1}^{N_p} W^{(i)}}, \\
\Sigma &= \frac{\sum_{i=1}^{N_p} W^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^T}{\sum_{i=1}^{N_p} W^{(i)}}.
\end{aligned}$$

Gibbs Sampling. Gibbs sampling, initially developed by Geman and Geman in image restoration [178], is a special form of MCMC [185], [173], or a special form of Metropolis-Hastings algorithm [329], [204], [175], [176]. The Gibbs sampler uses the concept of alternating (marginal) conditional sampling. Given an $N_\mathbf{x}$-dimensional state vector $\mathbf{x} = [x_1, x_2, \cdots, x_{N_\mathbf{x}}]^T$, we are interested in drawing the samples from the marginal density in the case where joint density is inaccessible or hard to sample. The generic procedure is as follows (e.g., [73]):

- At iteration $n = 0$, draw $\mathbf{x}_0$ from the prior density $p(\mathbf{x}_0)$;
- At iterations $n = 1, 2, \cdots$, draw a sample $x_{1,n}$ from $p(x_1 | x_{2,n-1}, x_{3,n-1}, \cdots, x_{N_\mathbf{x},n-1})$;
- draw a sample $x_{2,n}$ from $p(x_2 | x_{1,n}, x_{3,n-1}, \cdots, x_{N_\mathbf{x},n-1})$; $\cdots$
- draw a sample $x_{N_\mathbf{x},n}$ from $p(x_{N_\mathbf{x}} | x_{1,n}, x_{2,n}, \cdots, x_{N_\mathbf{x}-1,n})$;

To illustrate the idea of Gibbs sampling, an example with four-step iterations in a two-dimensional probability space $p(x_1, x_2)$ is presented in Fig. 8.

Remarks:

- Gibbs sampling is an alternating sampling scheme, since the conditional density to be sampled is low-dimensional, the Gibbs sampler is a nice solution to estimation of hierarchical or structured probabilistic model.
- Gibbs sampling can be viewed as a Metropolis method in which the proposal distribution is defined in terms of the conditional distributions of the joint distribution and every proposal is always accepted [318].
- Gibbs sampling has been extensively used for dynamic state space model [71] within the Bayesian framework.
- Adaptive rejection Gibbs sampling algorithm was also developed in [187].

In addition to Metropolis-Hastings algorithm and Gibbs sampling, MCMC methods are powerful and have a huge literature. We cannot extend the discussions due to the space constraint and refer the reader to [176], [182], [185] for more discussions on MCMC methods, and the review paper [416] for Bayesian estimation using MCMC methods.

In the context of sequential state estimation, Metropolis-Hastings algorithm and Gibbs sampling are less attractive because of their computational inefficiency in a non-iterative fashion. On the other hand, both of them use random walk to explore the state space, the efficiency is low when $N_\mathbf{x}$ is big. Another important issue about MCMC methods is their convergence: How long it takes a MCMC to converge to an equilibrium? How fast is the convergence rate? Many papers were devoted to investigating these questions [99], [140].[49] One way to reduce the reducing the "blind" random-walk behavior in Gibbs sampling is the methods of over-relaxation [2], [349], [318]; another way is the so-called hybrid Monte Carlo method as we discuss next.

G.7 Hybrid Monte Carlo

Hybrid Monte Carlo (HMC) algorithm [152] is a kind of asymptotically unbiased MCMC algorithm for sampling from complex distributions. In particular, it can be viewed as a Metropolis method which uses gradient information to reduce random walk behavior. Assume the probability distribution is written as [346], [318]

$$P(\mathbf{x}) = \frac{\exp(-\mathcal{E}(\mathbf{x}))}{\mathcal{Z}}, \tag{75}$$

where $\mathcal{Z}$ is a normalizing constant. The key idea of HMC is not only use the energy $\mathcal{E}(\mathbf{x})$ but also its gradient (w.r.t. to $\mathbf{x}$), since the gradient direction might indicate the way to find the state with a higher probability [318].

In the HMC,[50] the state space $\mathbf{x}$ is augmented by a *momentum variable* $\boldsymbol{\eta}$; and two proposals are alternately used. The first proposal randomizes the momentum variable with the state $\mathbf{x}$ unchanged; the second proposal changes both $\mathbf{x}$ and $\boldsymbol{\eta}$ using the simulated Hamilton dynamics as follows [318]

$$H(\mathbf{x}, \boldsymbol{\eta}) = \mathcal{E}(\mathbf{x}) + \mathcal{K}(\boldsymbol{\eta}), \tag{76}$$

[49] See also the recent special MCMC issue in *Statistical Science*, vol. 16, no. 4, 2001.

[50] A pseudocode of HMC algorithm was given in [318].

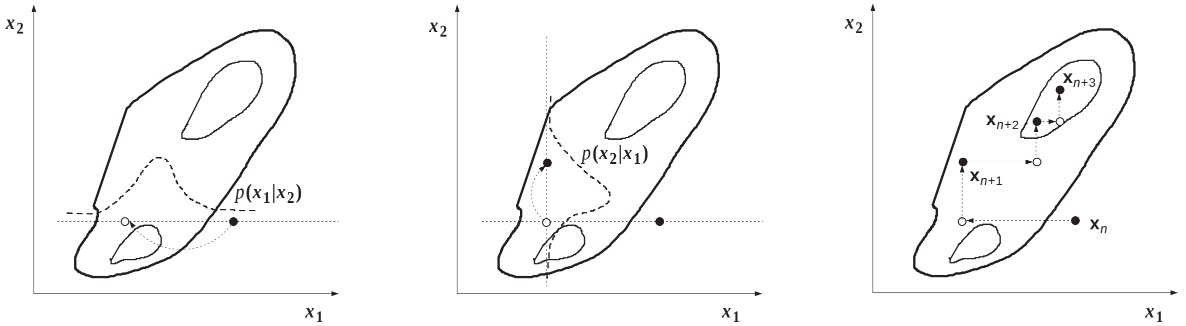

Fig. 8. An illustration of Gibbs sampling in a two-dimensional space (borrowed and changed from MacKay (1998) with permission). **Left:** Starting from state $\mathbf{x}_n$, x_1 is sampled from the conditional pdf $p(x_1|x_{2,n-1})$. **Middle:** A sample is drawn from the conditional pdf $p(x_2|x_{1,n})$. **Right:** Four-step iterations in the probability space (contour).

where $\mathcal{K}(\boldsymbol{\eta})$ is a *kinetic energy* with the form $\mathcal{K}(\boldsymbol{\eta}) = \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta}$. These two proposals are used to produce samples from the joint distribution:

$$\begin{aligned} P_H(\mathbf{x},\boldsymbol{\eta}) &= \frac{1}{\mathcal{Z}_H}\exp[-H(\mathbf{x},\boldsymbol{\eta})] \\ &= \frac{1}{\mathcal{Z}_H}\exp[-\mathcal{E}(\mathbf{x})]\exp[-\mathcal{K}(\boldsymbol{\eta})], \qquad (77) \end{aligned}$$

where $\mathcal{Z}_H = \mathcal{Z}\mathcal{Z}_K$ is a normalizing constant. The distribution $P_H(\mathbf{x},\boldsymbol{\eta})$ is separable and the marginal distribution of $\mathbf{x}$ is the desired distribution $\exp[-\mathcal{E}(\mathbf{x})]/\mathcal{Z}$. By discarding the momentum variables, a sequence of random samples $\mathbf{x}^{(i)}$ can be generated that can be viewed as asymptotically being drawn from $P(\mathbf{x})$. The first proposal draws a new momentum from the Gaussian density $\exp[-\mathcal{K}(\boldsymbol{\eta})]/\mathcal{Z}_K$. In the second proposal, the momentum determines where the state should go, and the gradient of $\mathcal{E}(\mathbf{x})$ determines how the momentum $\boldsymbol{\eta}$ changes according to the following differential equations

$$\dot{\mathbf{x}} = \boldsymbol{\eta} \qquad (78a)$$

$$\dot{\boldsymbol{\eta}} = -\frac{\partial\mathcal{E}(\mathbf{x})}{\partial\mathbf{x}}. \qquad (78b)$$

Since the motion of $\mathbf{x}$ is driven by the direction of momentum $\boldsymbol{\eta}$, intuitively the state converges faster than the conventional MC methods. With perfect simulatio of Hamilton dynamics, the total energy $H(\mathbf{x},\boldsymbol{\eta})$ is a constant, thus (72) is always 1 and the proposal is always accepted; with imperfect simulation, we can obtain, asymptotically, the samples from $P_H(\mathbf{x},\boldsymbol{\eta})$ [318].

Remarks:

- HMC method can be used for particle filter [94]: Instead of being weighted by the likelihood, each particle produces a Markov chain that follows the gradient of the posterior over large distances, which allows it to rapidly explore the state space and produce samples from the target distribution.
- Some improved HMC methods were developed in [347], [346].
- The idea of using gradient information in HMC can be extended to sequential framework, e.g. the HySIR algorithm [120].

G.8 Quasi-Monte Carlo

Another important Monte Carlo method attempting to accelerate the convergence is quasi-Monte Carlo (QMC) (e.g., see [353], [425], [363]), which was extensively used in computer graphics. The mathematical foundation of QMC is the number theory instead of probability theory, hence it is a *deterministic* method. The idea of QMC methods is to substitute the pseudo-randomly generated sequence used in the regular MC methods with a deterministic sequence in order to minimize the divergence, and also to replace the probabilistic error bounds of regular MC with deterministic bounds. In the QMC, a popular class of deterministic sequence called low-discrepancy sequence (LDS) is often used to generate the samples points [353]. The LDS has a minimum discrepancy[51] $\mathcal{O}((\log N_p)^{N_\mathbf{x}-1}/N_p)$ (for a large N_p), which is faster than the regular MC methods' error bound $\mathcal{O}(1/\sqrt{N_p})$ (from Central Limit Theorem). There are many methods for constructing LDS, among them the lattice rule (LR) is a popular one due to its simplicity and potential variance redundancy advantage [295], [296]. By using some lattice rule to generate a point set

$$S = \Big\{\frac{(i-1)}{N_p}(1, a, \cdots, a^{N_\mathbf{x}-1}) \bmod 1,\ i = 1, \cdots, N_p\Big\},$$

where N_p is the number of lattice points in S and a is an integer between 1 and $N_p - 1$. For a square-integrable function f over $[0,1)^{N_\mathbf{x}}$, the estimator of QMC via a lattice rule is given by

$$\hat{f}_{\mathrm{LR}} = \frac{1}{N_p}\sum_{i=1}^{N_p} f((\mathbf{x}_i + \Delta) \bmod 1). \qquad (79)$$

It was shown in [295] that the estimate (79) is unbiased and $\mathrm{Var}[\hat{f}_{\mathrm{LR}}] \ll \mathrm{Var}[\hat{f}_{\mathrm{MC}}]$; in particular when f is linear, $\mathrm{Var}[\hat{f}_{\mathrm{LR}}] = \frac{1}{N_p}\mathrm{Var}[\hat{f}_{\mathrm{MC}}]$; in some cases where f is nonlinear, the convergence rate $\mathcal{O}(1/N_p^2)$ might be achieved.

Remarks:

- QMC can be viewed as a special quadrature technique with a different scheme choosing the quadrature

[51] It is a measure of the uniformity of distribution of finite point sets.

TABLE II
A List of Popular Monte Carlo Methods.

author(s)	method	inference	references
Metropolis	MCMC	off line	[330], [329]
Marshall	importance sampling	on/off line	[324], [199], [180]
N/A	rejection sampling	off line	[199], [197]
N/A	stratified sampling	on/off line	[376], [377], [69]
Hastings	MCMC	off line	[204]
Geman & Geman	Gibbs sampling	off line	[178], [175]
Handschin & Mayne	SIS	off line	[200], [506], [266]
Rubin	multiple imputation	off line	[394], [395]
Rubin	SIR	on/off line	[397], [176]
Gordon *et al.*	bootstrap	on line	[191], [193]
Duane *et al.*	HMC	on/off line	[152], [347], [346]
N/A	QMC	on/off line	[353], [425], [354]
Chen & Schmeiser	hit-and-run MC	off line	[81], [417]
N/A	slice sampling	off line	[336], [351]
N/A	perfect sampling	off line	[133], [490]

points, it can be used for marginal density estimation [363].

- QMC method can be also applied to particle filters [361].

To the end of this subsection, we summarize some popular Monte Carlo methods available in the literature in Table II for the reader's convenience.

VI. Sequential Monte Carlo Estimation: Particle Filters

With the background knowledge of stochastic filtering, Bayesian statistics, and Monte Carlo techniques, we are now in a good position to discuss the theory and paradigms of particle filters. In this section, we focus the attention on the sequential Monte Carlo approach for sequential state estimation. Sequential Monte Carlo technique is a kind of recursive Bayesian filter based on Monte Carlo simulation, it is also called bootstrap filter [193] and shares many common features with the so-called interacting particle system approximation [104], [105], [122], [123], [125], CONDENSATION [229], [230], Monte Carlo filter [259]-[261], [49], sequential imputation [266], [303], survival of fittest [254], and likelihood weighting algorithm [254].

The working mechanism of particle filters is following: The state space is partitioned as many parts, in which the particles are filled according to some probability measure. The higher probability, the denser the particles are concentrated. The particle system evolves along the time according to the state equation, with evolving pdf determined by the FPK equation. Since the pdf can be approximated by the point-mass histogram, by random sampling of the state space, we get a number of particles representing the evolving pdf. However, since the posterior density model is unknown or hard to sample, we would rather choose another distribution for the sake of efficient sampling.

To avoid intractable integration in the Bayesian statistics, the posterior distribution or density is empirically represented by a weighted sum of N_p samples drawn from the posterior distribution

$$p(\mathbf{x}_n|\mathcal{Y}_n) \approx \frac{1}{N_p}\sum_{n=1}^{N_p}\delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}) \equiv \hat{p}(\mathbf{x}_n|\mathcal{Y}_n), \tag{80}$$

where $\mathbf{x}_n^{(i)}$ are assumed to be i.i.d. drawn from $p(\mathbf{x}_n|\mathcal{Y}_n)$. When N_p is sufficiently large, $\hat{p}(\mathbf{x}_n|\mathcal{Y}_n)$ approximates the true posterior $p(\mathbf{x}_n|\mathcal{Y}_n)$. By this approximation, we can estimate the mean of a nonlinear function

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_n)] &\approx \int f(\mathbf{x}_n)\hat{p}(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n \\ &= \frac{1}{N_p}\sum_{i=1}^{N_p}\int f(\mathbf{x}_n)\delta(\mathbf{x}_n - \mathbf{x}_n^{(i)})d\mathbf{x}_n \\ &= \frac{1}{N_p}\sum_{i=1}^{N_p} f(\mathbf{x}_n^{(i)}) \equiv \hat{f}_{N_p}(\mathbf{x}). \qquad (81)\end{aligned}$$

Since it is usually impossible to sample from the true posterior, it is common to sample from an easy-to-implement distribution, the so-called *proposal distribution* [52] denoted by $q(\mathbf{x}_n|\mathcal{Y}_n)$, hence

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathcal{Y}_n)}{q(\mathbf{x}_n|\mathcal{Y}_n)}q(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n \\ &= \int f(\mathbf{x}_n)\frac{W_n(\mathbf{x}_n)}{p(\mathcal{Y}_n)}q(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n \\ &= \frac{1}{p(\mathcal{Y}_n)}\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n, (82)\end{aligned}$$

where

$$W_n(\mathbf{x}_n) = \frac{p(\mathcal{Y}_n|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathcal{Y}_n)}. \tag{83}$$

Equation (82) can be rewritten as

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_n)] &= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n}{\int p(\mathcal{Y}_n|\mathbf{x}_n)p(\mathbf{x}_n)d\mathbf{x}_n} \\ &= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n}{\int W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathcal{Y}_n)d\mathbf{x}_n} \\ &= \frac{\mathbb{E}_{q(\mathbf{x}_n|\mathcal{Y}_n)}[W_n(\mathbf{x}_n)f(\mathbf{x}_n)]}{\mathbb{E}_{q(\mathbf{x}_n|\mathcal{Y}_n)}[W_n(\mathbf{x}_n)]}. \qquad (84)\end{aligned}$$

By drawing the i.i.d. samples $\{\mathbf{x}_n^{(i)}\}$ from $q(\mathbf{x}_n|\mathcal{Y}_n)$, we can approximate (84) by

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_n)] &\approx \frac{\frac{1}{N_p}\sum_{i=1}^{N_p} W_n(\mathbf{x}_n^{(i)})f(\mathbf{x}_n^{(i)})}{\frac{1}{N_p}\sum_{i=1}^{N_p} W_n(\mathbf{x}_n^{(i)})} \\ &= \sum_{i=1}^{N_p}\tilde{W}_n(\mathbf{x}_n^{(i)})f(\mathbf{x}_n^{(i)}) \equiv \hat{f}(\mathbf{x}), \qquad (85)\end{aligned}$$

[52] It is also called *importance density* or *important function*. The optimal proposal distribution is the one that minimizes the conditional variance given the observations up to n.

where

$$\tilde{W}_n(\mathbf{x}_n^{(i)}) = \frac{W_n(\mathbf{x}_n^{(i)})}{\sum_{j=1}^{N_p} W_n(\mathbf{x}_n^{(j)})}. \tag{86}$$

Suppose the proposal distribution has the following factorized form

$$\begin{aligned} q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) &= q(\mathbf{x}_n|\mathbf{x}_{0:n-1},\mathbf{y}_{0:n})q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \\ &= q(\mathbf{x}_0)\prod_{t=1}^{n} q(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{0:t}). \end{aligned}$$

Similar to the derivation steps in (23), the posterior $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ can be factorized as

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})}$$

Thus the importance weights $W_n^{(i)}$ can be updated recursively

$$\begin{aligned} W_n^{(i)} &= \frac{p(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n})}{q(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n})} \\ &\propto \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})p(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n})q(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n-1})} \\ &= W_{n-1}^{(i)}\frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n})}. \end{aligned} \tag{87}$$

A. Sequential Importance Sampling (SIS) Filter

In practice, we are more interested in the current filtered estimate $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ instead of $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$. Provided $q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n})$ is assumed to be equivalent to $q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_n)$, (87) can be simplified as

$$W_n^{(i)} = W_{n-1}^{(i)}\frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_n)}. \tag{88}$$

As discussed earlier, the problem of the SIS filter is that the distribution of the importance weights becomes more and more skewed as time increases. Hence, after some iterations, only very few particles have non-zero importance weights. This phenomenon is often called *weight degeneracy* or *sample impoverishment* [396], [193], [40], [304]. An intuitive solution is to multiply the particles with high normalized importance weights, and discard the particles with low normalized importance weights, which can be be done in the resampling step. To monitor how bad is the weight degeneration, we need a measure. A suggested measure for degeneracy, the so-called effective sample size, N_{eff}, was introduced in [266] (see also [303], [305], [315], [144], [350])[53]

$$\begin{aligned} N_{eff} &= \frac{N_p}{1+\mathrm{Var}_{q(\cdot|\mathbf{y}_{0:n})}[\tilde{W}(\mathbf{x}_{0:n})]} \\ &= \frac{N_p}{\mathbb{E}_{q(\cdot|\mathbf{y}_{0:n})}[(\tilde{W}(\mathbf{x}_{0:n}))^2]} \leq N_p. \end{aligned} \tag{89}$$

TABLE III

SIS Particle Filter with Resampling.

```
For time steps $n = 0, 1, 2, \cdots$
1: For $i = 1, \cdots, N_p$, draw the samples $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})$
   and set $\mathbf{x}_{0:n}^{(i)} = \{\mathbf{x}_{0:n-1}^{(i)}, \mathbf{x}_n^{(i)}\}$.
2: For $i = 1, \cdots, N_p$, calculate the importance weights $W_n^{(i)}$ according to (88).
3: For $i = 1, \cdots, N_p$, normalize the importance weights $\tilde{W}_n^{(i)}$ according to (86).
4: Calculate $\hat{N}_{eff}$ according to (90), return if $\hat{N}_{eff} > N_T$,
   otherwise generate a new particle set $\{\mathbf{x}_n^{(j)}\}$ by resampling
   with replacement $N_p$ times from the previous set $\{\mathbf{x}_{0:n}^{(i)}\}$ with
   probabilities $\Pr(\mathbf{x}_{0:n}^{(j)} = \mathbf{x}_{0:n}^{(i)}) = \tilde{W}_{0:n}^{(i)}$, reset the weights
   $\tilde{W}_n^{(i)} = 1/N_p$.
```

The second equality above follows from the facts that $\mathrm{Var}[\xi] = \mathbb{E}[\xi^2] - (\mathbb{E}[\xi])^2$ and $\mathbb{E}_q[\tilde{W}] = 1$. In practice, the true N_{eff} is not available, thus its estimate, $\hat{N}_{eff}$, is alternatively given [305], [303]:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p}(\tilde{W}_n^{(i)})^2}. \tag{90}$$

When $\hat{N}_{eff}$ is below a predefined threshold N_T (say $N_p/2$ or $N_p/3$), the resampling procedure is performed. The above procedure was also used in the rejection control [304] that combines the rejection method [472] and importance sampling. The idea is following: when the $\hat{N}_{eff} < N_T$ (where N_T can be either a predefined value or the median of the weights), then each sample is accepted with probability $\min\{1, W_n^{(i)}/N_T\}$; all the accepted samples are given a new weight $W_n^{(j)} = \max\{N_T, W_n^{(i)}\}$, and the rejected samples are restarted and rechecked at the *all* previously violated thresholds. It is obvious that this procedure is computational expensive as n increases. Some advanced scheme like partial rejection control [308] was thus proposed to reduce the computational burden, while preserving the dynamic control of the resampling schedule. A generic algorithm of SIS particle filter with resampling is summarized in Table III.

B. Bootstrap/SIR filter

The Bayesian bootstrap filter due to Gordon, Salmond and Smith [193], is very close in spirit to the sampling importance resampling (SIR) filter developed independently in statistics by different researchers [304], [307], [369], [370], [69], with a slight difference on the resampling scheme. Here we treat them as the same class for discussion. The key idea of SIR filter is to introduce the resampling step as we have discussed in Section V-G.4. The resampling step is flexible and varies from problems as well as the selection scheme and schedule. It should be noted that resampling does *not* really prevent the weight degeneracy problem, it just saves further calculation time by discarding the particles associated with insignificant weights. What it really does is artificially concealing the impoverishment by re-

[53] It was claimed that [70], [162] the estimate N_{eff} is not robust, see discussion in Section VI-P.3.

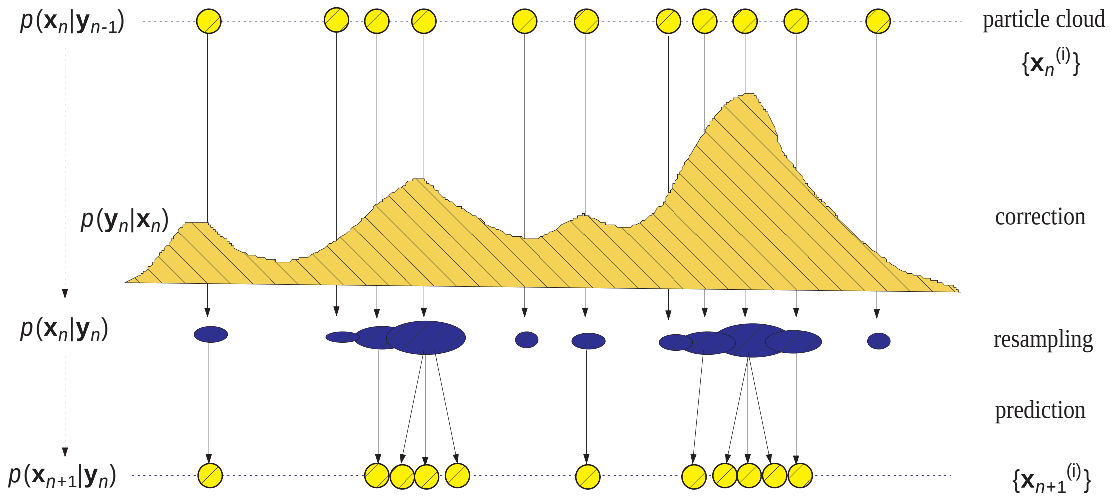
$p(\mathbf{x}_n|\mathbf{y}_{n-1})$
particle cloud
$\{\mathbf{x}_n^{(i)}\}$
$p(\mathbf{y}_n|\mathbf{x}_n)$
correction
$p(\mathbf{x}_n|\mathbf{y}_n)$
resampling
prediction
$p(\mathbf{x}_{n+1}|\mathbf{y}_n)$
$\{\mathbf{x}_{n+1}^{(i)}\}$

Fig. 9. An illustration of generic particle filter with importance sampling and resampling.

TABLE IV

SIR Particle Filter Using Transition Prior as Proposal Distribution.

```
For time steps $n = 0, 1, 2, \cdots$
1: Initialization: for $i = 1, \cdots, N_p$, sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, $W_0^{(i)} = \frac{1}{N_p}$.
2: Importance Sampling: for $i = 1, \cdots, N_p$, draw samples $\hat{\mathbf{x}}_n^{(i)} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$, set $\hat{\mathbf{x}}_{0:n}^{(i)} = \{\mathbf{x}_{0:n-1}^{(i)}, \hat{\mathbf{x}}_n^{(i)}\}$.
3: Weight update: Calculate the importance weights $W_n^{(i)} = p(\mathbf{y}_n|\hat{\mathbf{x}}_n^{(i)})$.
4: Normalize the importance weights: $\tilde{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^{N_p} W_n^{(j)}}$.
5: Resampling: Generate $N_p$ new particles $\mathbf{x}_n^{(i)}$ from the set $\{\hat{\mathbf{x}}_n^{(i)}\}$ according to the importance weights $\tilde{W}_n^{(i)}$.
6: Repeat Steps 2 to 5.
```

placing the high important weights with many replicates of particles, thereby introducing high correlation between particles.

A generic algorithm of Bayesian bootstrap/SIR filter using transition prior density as proposal distribution is summarized in Table IV, where the resampling step is performed at each iteration using any available resampling method discussed earlier.

Remarks:

- Both SIS and SIR filters use importance sampling scheme. The difference between them is that in SIR filter, the resampling is always performed (usually between two importance sampling steps); whereas in SIS filter, importance weights are calculated sequentially, resampling is only taken whenever needed, thus SIS filter is less computationally expensive.
- The choice of proposal distributions in SIS and SIR filters plays an crucial role in their final performance.
- Resampling step is suggested to be done after the filtering [75], [304], because resampling brings extra random variation to the current samples. Normally (eps. in off-line processing), the posterior estimate (and its relevant statistics) should be calculated before resampling.
- As suggested by some authors [259], [308], in the resampling stage, the new importance weights of the surviving particles are not necessarily reset to $1/N_p$, but rather abide certain procedures.
- To alleviate the sample degeneracy in SIS filter, we can change (88) as

$$W_n = W_{n-1}^{\alpha} \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})}{q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_n)},$$

where the scalar $0 < \alpha < 1$ plays a role as annealing factor that controls the impact of previous importance weights.

C. Improved SIS/SIR Filters

In the past few years, many efforts have been devoted to improving the particle filters' performance [69], [189], [428], [345], [456], [458], [357]. Here, due to space limitation, we only focus on the improved schemes on (efficient) sampling/resampling and variance reduction.

In order to alleviate the sample impoverishment problem, a simple improvement strategy is *prior boosting* [193]. Namely, in the sampling step, one can increase the number of simulated samples drawn from the proposal, $N_p' > N_p$; but in the resampling step, only N_p particles are preserved.

Carpenter, Clifford, and Fearnhead [69] proposed using a sophisticated stratified sampling (also found in [259]) for particle filtering. In particular, the posterior density is assumed to comprise of N_p distinct mixture strata[54]

$$p(\mathbf{x}) = \sum_{i=1}^{N_p} c_i p_i(\mathbf{x}), \quad \sum_{i=1}^{N_p} c_i = 1, \tag{91}$$

According to [69], a population quantity can be estimated efficiently by sampling a fixed number M_i from each stra-

[54] This is the so-called survey sampling technique [199], [162].

tum, with $\sum_{i=1}^{N_p} M_i = N_p$ ($N_p \gg M_i$). The efficiency is attained with *Neyman allocation* $M_i \propto c_i\sigma_i$ (where σ_i is the variance of generic function $f(\mathbf{x})$ in the i-th stratum), or with proportional allocation $M_i = c_i N_p$ for simplicity. It was argued that in most of cases the proportional allocation is more efficient than simple random sampling from $p(\mathbf{x})$. In the particle filtering context, the coefficients c_i and $p_i(\mathbf{x})$ are determined recursively [69]:

$$c_i = \frac{\int p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})p(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n}{\sum_{i=1}^{N_p}\int p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})p(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n}, \quad (92)$$

$$p_i(\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})p(\mathbf{y}_n|\mathbf{x}_n)}{\int p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})p(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n}. \quad (93)$$

For the i-th stratum, the importance weights associated with the M_i particles are updated recursively by

$$W_n^{(j)} = W_{n-1}^{(i)}\frac{p(\mathbf{x}_n^{(j)}|\mathbf{x}_{n-1}^{(i)})p(\mathbf{y}_n|\mathbf{x}_n^{(j)})}{c_i p_i(\mathbf{x}_n^{(j)})} \quad (94)$$

for $\sum_{\ell=1}^{i-1} M_\ell < j \le \sum_{\ell=1}^{i} M_\ell$. By stratified sampling in the update stage, the variance reduction is achieved.[55] In the resampling stage, a sample set of size N_p is selected from the $10 \times N_p$ predicted values to keep the size of particle set unchanged.[56] By taking advantage of the method of simulating order statistics [386], an improved SIR algorithm with $\mathcal{O}(N_p)$ complexity via stratified sampling was developed [69], to which reader is referred for more details.

Many improved particle filters are devoted to the resampling step. For instance, given the discrete particle set $\{\mathbf{x}_n^{(i)}, \tilde{W}_n^{(i)}\}_{i=1}^{N_p}$, it was suggested [308] that in the resampling stage, a new independent particle set $\{\mathbf{x}_n^{(j)}, \tilde{W}_n^{(j)}\}_{j=1}^{N_p}$ is generated as follows:

- For $j = 1, \cdots, N_p$, $\mathbf{x}_n^{(j)}$ replaces $\mathbf{x}_n^{(i)}$ with probability proportional to $a^{(i)}$;
- The associated new weights $\tilde{W}_n^{(j)}$ is updated as $\tilde{W}_n^{(j)} = \tilde{W}_n^{(i)}/a^{(i)}$.

In the conventional multinomial resampling scheme (Section V-G.4), $a^{(i)} = N_p W_n^{(i)}$; however in general, the choices of $a^{(i)}$ are flexible, e.g. $a^{(i)} = \sqrt{W_n^{(i)}}$, or $a^{(i)} = |W_n^{(i)}|^\alpha$. Liu, Chen and Logvinenko [308] also proposed to use a partially deterministic *reallocation* scheme instead of resampling to overcome the extra variation in resampling step. The reallocation procedure proceeds as follows [308]:

- For $i = 1, \cdots, N_p$, if $a^{(i)} \ge 1$, retain $k_i = \lfloor a^{(i)} \rfloor$ (or $k_i = \lfloor a^{(i)} \rfloor + 1$) copies of the $\mathbf{x}_n^{(i)}$; assign the weight $W_n^{(j)} = W_n^{(i)}/k_i$ for each copy;
- if $a^{(i)} < 1$, remove the sample with probability $1 - a^{(i)}$; assign the weight $W_n^{(j)} = W_n^{(i)}/a^{(i)}$ to the survived sample.
- Return the new particle set $\{\mathbf{x}^{(j)}, W_n^{(j)}\}$.

D. Auxiliary Particle Filter

A potential weakness of generic particle filters discussed above is that the particle-based approximation of filtered density is not sufficient to characterize the tail behavior of true density, due to the use of finite mixture approximation; this is more severe when the outliers are existent. To alleviate this problem, Pitt and Shephard [370], [371] introduced the so-called auxiliary particle filter (APF). The idea behind it is to augment the existing "good" particles $\{\mathbf{x}^{(i)}\}$ in a sense that the *predictive* likelihoods $p(\mathbf{y}_n|\mathbf{x}_{0:n-1}^{(i)})$ are large for the "good" particles. When $p(\mathbf{y}_n|\mathbf{x}_{0:n-1}^{(i)})$ cannot be computed analytically, it uses an analytic approximation; when $p(\mathbf{y}_n|\mathbf{x}_{0:n-1}^{(i)})$ can be computed exactly, it uses the optimal proposal distribution (which is thus called "perfect adaptation" [370]). The APF differs from SIR in that it reverses the order of sampling and resampling, which is possible when the importance weights are dependent on $\mathbf{x}_n$. By inserting the likelihood inside the empirical density mixture, we may rewrite the filtered density as

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|\mathbf{x}_n)\int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1} \\ &\propto \sum_{i=1}^{N_p} W_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}), \end{aligned} \quad (95)$$

where $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^{N_p} W_{n-1}^{(i)}\delta(\mathbf{x}_{n-1} - \mathbf{x}_{n-1}^{(i)})$. Now the product $W_{n-1}^{(i)}p(\mathbf{y}_n|\mathbf{x}_n)$ is treated as a combined probability contributing to the filtered density. By introducing an auxiliary variable ξ ($\xi \in \{1, \cdots, N_p\}$) that plays a role of index of the mixture component, the augmented joint density $p(\mathbf{x}_n, \xi|\mathbf{y}_{0:n})$ is updated as

$$\begin{aligned} p(\mathbf{x}_n, \xi = i|\mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n, \xi = i|\mathbf{y}_{0:n-1}) \\ &= p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\xi = i, \mathbf{y}_{0:n-1})p(i|\mathbf{y}_{0:n-1}) \\ &= p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})W_{n-1}^{(i)}. \end{aligned} \quad (96)$$

Henceforth a sample can be drawn from joint density (96) via simply neglecting the index ξ, by which a set of particles $\{\mathbf{x}_n^{(i)}\}_{i=1}^{N_p}$ are drawn from the marginalized density $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ and the indices ξ are simulated with probabilities proportional to $p(\xi|\mathbf{y}_{0:n})$. Thus, (95) can be approximated by

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) \propto \sum_{i=1}^{N_p} W_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_n^{(i)}, \xi^i)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}), \quad (97)$$

where ξ^i denotes the index of the particle $\mathbf{x}_n^{(i)}$ at time step $n-1$, namely $\xi^i \equiv \{\xi = i\}$. The proposal distribution used to draw $\{\mathbf{x}_n^{(i)}, \xi^i\}_{i=1}^{N_p}$ is chosen as a factorized form

$$q(\mathbf{x}_n, \xi|\mathbf{y}_{0:n}) \propto q(\xi|\mathbf{y}_{0:n})q(\mathbf{x}_n|\xi, \mathbf{y}_{0:n}), \quad (98)$$

[55] Intuitively, they use the weighted measure before resampling rather than resampling and then using the unweighted measure, because the weighted samples are expected to contain more information than an equal number of unweighted points.

[56] The number 10 was suggested by Rubin [395] where $N_p \gg 10$. The number of particle set is assumed to be unchanged.

where

$$\begin{aligned} q(\xi^i|\mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|\mu_n^{(i)})W_{n-1}^{(i)} \qquad (99)\\ q(\mathbf{x}_n|\xi^i,\mathbf{y}_{0:n}) &= p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}). \qquad (100)\end{aligned}$$

where $\mu^{(i)}$ is a value (e.g. mean, mode, or sample value) associated with $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$ from which the i-th particle is drawn. Thus the true posterior is further approximated by

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) \propto \sum_{i=1}^{N_p} W_{n-1}^{(i)} p(\mathbf{y}_n|\mu_n^{(\xi=i)})p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(\xi=i)}). \qquad (101)$$

From (99) and (100), the important weights are recursively updated as

$$\begin{aligned} W_n^{(i)} &= W_{n-1}^{(\xi=i)} \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(\xi=i)})}{q(\mathbf{x}_n^{(i)},\xi^i|\mathbf{y}_{0:n})}\\ &\propto \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})}{p(\mathbf{y}_n|\mu_n^{(\xi=i)})}. \qquad (102)\end{aligned}$$

The APF is essentially a two-stage procedure: At the first stage, simulate the particles with large predictive likelihoods; at the second stage, reweigh the particles and draw the augmented states. This is equivalent to making a proposal that has a high conditional likelihood a priori, thereby avoiding inefficient sampling [370]. The auxiliary variable idea can be used for SIS or SIR filters. An auxiliary SIR filter algorithm is summarized in Table V.

It is worthwhile to take a comparison between APF and SIR filter on the statistical efficiency in the context of the random measure $\mathbb{E}[\tilde{W}^2(\mathbf{x}^{(i)})]$. Pitt and Shephard [370] showed that when the likelihood does not vary over different ξ, then the variance of APF is smaller than that of SIR filter. APF can be understood as a one-step ahead filtering [369]-[371]: the particle $\mathbf{x}_{n-1}^{(i)}$ is propagated to $\xi_n^{(i)}$ in the next time step in order to assist the sampling from the posterior. On the other hand, APF resamples $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ instead of $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ used in SIR, hence it usually achieves lower variance because the past estimate is more reliable. Thus APF actually takes advantage of beforehand the information from likelihood model to avoid inefficient sampling because the particles with low likelihood are deemed less informative; in other words, the particles to be sampled are intuitively pushed to the high likelihood region. But when the conditional likelihood is not insensitive to the state, the difference between APF and SIR filter is insignificant. APF calculates twice the likelihood and importance weights, in general it achieves better performance than SIS and SIR filters.

Remarks:

- In conventional particle filters, estimation is usually performed *after* the resampling step, which is less efficient because resampling introduces extra random variation in the current state [75], [304]. APF basically overcomes this problem by doing one-step ahead estimation based on the point estimate $\mu_n^{(i)}$ that characterizes $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$.

TABLE V

Auxiliary Particle Filter.

```
For time steps $n = 1, 2, \cdots$
1: For $i = 1, \cdots, N_p$, calculate $\mu_n^{(i)}$ (e.g. $\mu_n^{(i)} = \mathbb{E}[p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})]$.
2: For $i = 1, \cdots, N_p$, calculate the first-stage weights $W_n^{(i)} = W_{n-1}^{(i)} p(\mathbf{y}_n|\mu_n^{(i)})$ and normalize weights $\tilde{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^{N_p} W_n^{(j)}}$.
3: Use the resampling procedure in SIR filter algorithm to obtain new $\{\mathbf{x}_n^{(i)}, \xi^i\}_{i=1}^{N_p}$.
4: For $i = 1, \cdots, N_p$, sample $\mathbf{x}_n^{(i)} \sim p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}, \xi^i)$, update the second-stage weights $W_n^{(i)}$ according to (102).
```

- When the process noise is small, the performance of APF is usually better than that of SIR filter, however, when the process noise is large, the point estimate $\mu_n^{(i)}$ doesn't provide sufficient information about $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$, then the superiority of APF is not guaranteed [19].
- In the APF, the proposal distribution is proposed as a mixture density that depends upon the past state and the most recent observations.
- The idea of APF is also identical to that of local Monte Carlo method proposed in [304], where the authors proposed two methods for draw samples $\{\mathbf{x}, \xi\}$, based on either joint distribution or marginal distribution.
- The disadvantage of APF is that the sampling is drawn in an augmented (thus higher) space, if the auxiliary index varies a lot for a fixed prior, the gain is negligible and the variance of importance weights will be higher.
- The APF is computationally slower since the proposal is used twice. It was argued that [162] (chap. 5) the resampling step of APF is unnecessary, which introduces nothing but inaccuracy. This claim, however, is not justified sufficiently.
- The idea of auxiliary variable can be also used for MCMC methods [210], [328].

E. Rejection Particle Filter

It was suggested in [222], [441], [444], [49] that the rejection sampling method is more favorable than the importance sampling method for particle filters, because rejection sampling achieves exact draws from the posterior. Usually rejection sampling doesn't admit a recursive update, hence how to design a sequential procedure is the key issue for the rejection particle filter.

Tanizaki [441]-[444] has developed a rejection sampling framework for particle filtering. The samples are drawn from the filtering density $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ without evaluating any integration. Recalling (20) and inserting equations (24) and (25) to (23), the filtering density can be approximated

TABLE VI
Rejection Particle Filter.

```
For time steps $n = 1, 2, \cdots$
1:  For $i = 1$, draw $\mathbf{x}_{n-1}^{(i)}$ with probability $\lambda_n^{(i)}$;
2:  Generate a random draw $\mathbf{z} \sim q(\mathbf{x}_n)$;
3:  Draw a uniform random variable $u \sim \mathcal{U}(0, 1)$;
4:  If $u \le \alpha(\mathbf{z})$, accept $\mathbf{z}$ as $\mathbf{x}_n^{(i)}$; otherwise go back to step 2;
5:  $i = i + 1$, repeat the procedure until $i = N_p$;
6:  Calculate the sample average $\hat{f}_{N_p}$, and calculate the posterior
    according to (103).
```

as

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{0:n}) &= \frac{1}{C_n}\int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1} \\ &\approx \frac{1}{N_p}\sum_{i=1}^{N_p}\frac{C_n^{(i)}}{C_n}\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})}{C_n^{(i)}} \\ &= \sum_{i=1}^{N_p}\lambda_n^{(i)}\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})}{C_n^{(i)}}, \end{aligned} \tag{103}$$

where $\lambda_n^{(i)} = C_n^{(i)}/N_pC_n$. The normalizing constant C_n is given as

$$\begin{aligned} C_n &= \int\int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1}d\mathbf{x}_n \\ &\approx \frac{1}{N_p^2}\sum_{i=1}^{N_p}\sum_{j=1}^{N_p}p(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(ji)}) \equiv \hat{C}_n, \end{aligned} \tag{104}$$

where $\mathbf{x}_{n|n-1}^{(ji)}$ is obtained from $\mathbf{f}(\mathbf{x}_{n-1}^{(i)}, \mathbf{d}_n^{(j)})$. In addition, $C_n^{(i)}$ is given as

$$\begin{aligned} C_n^{(i)} &= \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})d\mathbf{x}_n \\ &\approx \frac{1}{N_p}\sum_{j=1}^{N_p}p(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(ji)}) \equiv \hat{C}_n^{(i)}. \end{aligned} \tag{105}$$

Hence the filtering density is approximated as a mixture distribution associated with the weights $\lambda_n^{(i)}$, which are approximated by $\hat{C}_n^{(i)}/N_p\hat{C}_n$. The acceptance probability, denoted by $\alpha(\cdot)$, is defined as

$$\alpha(\mathbf{z}) = \frac{p(\mathbf{y}_n|\mathbf{z})p(\mathbf{z}|\mathbf{x}_{n-1}^{(i)})/q(\mathbf{z})}{\sup_{\mathbf{z}}\{p(\mathbf{y}_n|\mathbf{z})p(\mathbf{z}|\mathbf{x}_{n-1}^{(i)})/q(\mathbf{z})\}}, \tag{106}$$

where $q(\cdot)$ is a proposal distribution. The estimation procedure of rejection particle filter is summarized in Table VI.

The proposal distribution $q(\mathbf{x}_n)$ can be chosen as transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ or a mixture distribution (e.g. Gaussian mixture, see Section VI-M.4). But the variance of proposal distribution should be bigger than the posterior density's, since it is supposed to have a broad support.

Remarks:

- Rejection particle filter usually produces better results than SIR filter if the proposal distribution is appropriate and the supremum of the ratio $p(\cdot)/q(\cdot)$ exists. However, if the acceptance probability $\alpha(\mathbf{z})$ is small, it takes a long time to produce a sufficient sample set.
- Another drawback of rejection particle filter is that the computing time for every time step is fluctuating because of the uncertainty of acceptance probability, if the acceptance rate is too low, real-time processing requirement is not satisfied.
- It was suggested by Liu [305] to use $\mathrm{Var}[\hat{f}]/N_p$ as a measure to verify the efficiency for rejection sampling and importance sampling. It was claimed based on many experiments that, for a large N_p, importance sampling is more efficient in practice.
- Rejection sampling can be also used for APF. In fact, the proposal of APF accounts for the most recent observations and thus is more close to true posterior, thereby may increase the average acceptance rate.

F. *Rao-Blackwellization*

Rao-Blackwellization, motivated by the *Rao-Blackwell theorem*, is a kind of marginalization technique. It was first used in [175] to calculate the marginal density with Monte Carlo sampling method. Casella and Robert [74] also developed Rao-Blackwellization methods for rejection sampling and Metropolis algorithm with importance sampling procedure. Because of its intrinsic property of variance reduction, it has been used in particle filters to improve the performance [304], [14], [315], [145], [119]. There are couple ways to use Rao-Blackwellization: (i) state decomposition; (ii) model simplification; and (iii) data augmentation, all of which are based on the underlying Rao-Blackwell theorem:

Theorem 5: [388] Let $\hat{f}(Y)$ be an unbiased estimate of $f(\mathbf{x})$ and Ψ is a sufficient statistics for $\mathbf{x}$. Define $\hat{f}(\Psi(\mathbf{y})) = \mathbb{E}_{p(\mathbf{x})}[\hat{f}(Y)|\Psi(Y) = \Psi(\mathbf{y})]$, then $\hat{f}[\Psi(Y)]$ is also an unbiased estimate of $f(\mathbf{x})$. Furthermore,

$$\mathrm{Var}_{p(\mathbf{x})}[\hat{f}(\Psi(Y))] \le \mathrm{Var}_{p(\mathbf{x})}[\hat{f}(Y)],$$

and equality if and and only if $\Pr(\hat{f}(Y) = \hat{f}(\Psi(Y))) = 1$.

The proof of this theorem is based on *Jensen's Inequality* (see e.g., [462]). The importance of Rao-Blackwellization theorem is that, with a sufficient statistics Ψ, we can improve any unbiased estimator that is not a function of Ψ by conditioning on Ψ; in addition, if Ψ is sufficient for $\mathbf{x}$ and if there is a unique function of Ψ that is an unbiased estimate of $f(\mathbf{x})$, then such function is a minimum variance unbiased estimate for $f(\mathbf{x})$.

For dynamic state space model, the basic principle of Rao-Blackwellization is to exploit the model structure in order to improve the inference efficiency and consequently to reduce the variance. For example, we can attempt to decompose the dynamic state space into two parts, one part being calculated exactly using Kalman filter, the other part being inferred approximately using particle filter. Since the first part is inferred exactly and quickly, the computing power is saved and the variance is reduced. The following observations were given in [143], [144]. Let the states vector be partitioned into two parts $\mathbf{x}_n = [\mathbf{x}_n^1 \ \ \mathbf{x}_n^2]$, where marginal density $p(\mathbf{x}_n^2|\mathbf{x}_n^1)$ is assumed to be tractable analytically. The expectation of $f(\mathbf{x}_n)$ w.r.t. the posterior can be rewritten by:

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n^1, \mathbf{x}_n^2) p(\mathbf{x}_n^1, \mathbf{x}_n^2|\mathbf{y}_{0:n}) d\mathbf{x}_n \\ &= \frac{\int \lambda(\mathbf{x}_{0:n}^1) p(\mathbf{x}_{0:n}^1) d\mathbf{x}_{0:n}^1}{\int\int p(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}^1, \mathbf{x}_{0:n}^2) p(\mathbf{x}_{0:n}^2|\mathbf{x}_{0:n}^1) d\mathbf{x}_{0:n}^2 p(\mathbf{x}_{0:n}^1) d\mathbf{x}_{0:n}^1} \\ &= \frac{\int \lambda(\mathbf{x}_{0:n}^1) p(\mathbf{x}_{0:n}^1) d\mathbf{x}_{0:n}^1}{\int p(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}^1) p(\mathbf{x}_{0:n}^1) d\mathbf{x}_{0:n}^1}\end{aligned}$$

where

$$\lambda(\mathbf{x}_{0:n}^1) = \int f(\mathbf{x}_n^1, \mathbf{x}_n^2) p(\mathbf{y}_{0:n}|\mathbf{x}_n^1, \mathbf{x}_n^2) p(\mathbf{x}_{0:n}^2|\mathbf{x}_{0:n}^1) d\mathbf{x}_{0:n}^2.$$

And the weighted Monte Carlo estimate is given by

$$\hat{f}_{\mathrm{RB}} = \frac{\sum_{i=1}^{N_p} \lambda(\mathbf{x}_{0:n}^{1,(i)}) W(\mathbf{x}_{0:n}^{1,(i)})}{\sum_{i=1}^{N_p} W(\mathbf{x}_{0:n}^{1,(i)})}. \tag{107}$$

The lower variance of marginalized estimate is achieved because of the Rao-Blackwellization theorem

$$\mathrm{Var}[f(\mathbf{x})] = \mathrm{Var}\Big[\mathbb{E}[f(\mathbf{x}^1, \mathbf{x}^2)|\mathbf{x}^1]\Big] + \mathbb{E}\Big[\mathrm{Var}[f(\mathbf{x}^1, \mathbf{x}^2)|\mathbf{x}^1]\Big].$$

It has been proved that [143], [315], the variance of ratio of two joint densities is not less than that of two marginal densities

$$\begin{aligned}\mathrm{Var}_q\Big[\frac{p(\mathbf{x}^1, \mathbf{x}^2)}{q(\mathbf{x}^1, \mathbf{x}^2)}\Big] &= \mathrm{Var}_q\Big[\frac{\int p(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}{\int q(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}\Big] \\ & \quad + \mathbb{E}_q\Big[\mathrm{Var}_q\Big[\frac{p(\mathbf{x}^1, \mathbf{x}^2)}{q(\mathbf{x}^1, \mathbf{x}^2)}\Big|\mathbf{x}^1\Big]\Big] \\ &\geq \mathrm{Var}_q\Big[\frac{\int p(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}{\int q(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}\Big], \end{aligned} \tag{108}$$

where

$$\frac{\int p(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}{\int q(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2} = \mathbb{E}_q\Big[\frac{p(\mathbf{x}^1, \mathbf{x}^2)}{q(\mathbf{x}^1, \mathbf{x}^2)}\Big|\mathbf{x}^1\Big].$$

Hence by decomposing the variance, it is easy to see that the variance of the importance weights via Rao-Blackwellization is smaller than that obtained using direct Monte Carlo method.

Rao-Blackwellization technique is somewhat similar to the data augmentation method based on marginalization [445] in that it introduces a latent variable with assumed knowledge to ease the probabilistic inference. For instance, consider the following state-space model

$$\begin{aligned}\mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{d}_n), &\text{(109a)}\\ \mathbf{z}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n), &\text{(109b)}\\ \mathbf{y}_n &\sim p(\mathbf{y}_n|\mathbf{z}_n), &\text{(109c)}\end{aligned}$$

where the latent variable $\mathbf{z}_n$ is related to the measurement $\mathbf{y}_n$ with an analytic (e.g. exponential family) conditional pdf $p(\mathbf{y}_n|\mathbf{z}_n)$. Hence, the state estimation problem can be written by

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) p(\mathbf{z}_{0:n}|\mathbf{y}_{0:n}) d\mathbf{z}_{0:n}. \tag{110}$$

The probability distribution $p(\mathbf{z}_{0:n}|\mathbf{y}_{0:n})$ is approximated by the Monte Carlo simulation:

$$p(\mathbf{z}_{0:n}|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} W_n^{(i)} \delta(\mathbf{z}_{0:n} - \mathbf{z}_{0:n}^{(i)}), \tag{111}$$

thus the filtered density $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ is obtained by

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} W_n^{(i)} p(\mathbf{x}_n|\mathbf{z}_{0:n}^{(i)}), \tag{112}$$

which is a form of mixture model. When $p(\mathbf{x}_n|\mathbf{z}_{0:n}^{(i)})$ is Gaussian, this can be done by conventional Kalman filter technique, as exemplified in [83], [14], [325]; if $\mathbf{f}$ and $\mathbf{g}$ are either/both nonlinear, $p(\mathbf{x}_n|\mathbf{z}_{0:n}^{(i)})$ can be inferred by running a bank of EKFs. For any nonlinear function $f(\mathbf{x})$, Rao-Blackwellization achieves a lower variance estimate

$$\mathrm{Var}[f(\mathbf{x}_n)|\mathbf{y}_{0:n}] \geq \mathrm{Var}\Big[\mathbb{E}[f(\mathbf{x}_n)|\mathbf{z}_{0:n}, \mathbf{y}_{0:n}]\Big|\mathbf{y}_{0:n}\Big].$$

Remarks:

- In practice, appropriate model transformation (e.g. from Cartesian coordinate to polar coordinate) may simplify the model structure and admit Rao-Blackwellization.[57]
- Two examples of marginalized Rao-Blackwellization in particle filtering are *Conditionally Gaussian State-Space Model*, *Partially Observed Gaussian State-Space Model* and *Finite State HMM Model*. Rao-Blackwellization can be also used for MCMC [74].
- Similar to the idea of APF, Rao-Blackwellization can be also done one-step ahead [338], in which the sampling and resampling steps are switched when the important weights are independent on the measurements and the important proposal distribution can be analytically computed.

G. Kernel Smoothing and Regularization

In their seminal paper [193], Gordon, Salmond and Smith used an ad hoc approach called *jittering* to alleviate the sample impoverishment problem. In each time step, a small amount of Gaussian noise is added to each resampled particle, which is equivalent to using a Gaussian kernel to smooth the posterior. Another byproduct of jittering is to prevent the filter from divergence, as similarly done in the EKF literature.

Motivated by the kernel smoothing techniques in statistics, we can use a kernel to smooth the posterior estimate

[57] The same idea was often used in the EKF for improving the linearization accuracy.

by replacing the Dirac-delta function with a kernel function[58]

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} W_n^{(i)} K_h(\mathbf{x}_n, \mathbf{x}_n^{(i)}), \tag{113}$$

where $K_h(\mathbf{x}) = h^{-N_\mathbf{x}} K(\frac{\mathbf{x}}{h})$ with K being a symmetric, unimodal and smooth kernel and $h > 0$ being the bandwidth of the kernel. Some candidate kernels can be Gaussian or Epanechnikov kernel [345]

$$K(\mathbf{x}) = \begin{cases} \frac{N_\mathbf{x}+2}{2V_{N_\mathbf{x}}}(1 - \|\mathbf{x}\|^2), & \text{if } \|\mathbf{x}\| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{114}$$

where $V_{N_\mathbf{x}}$ denotes the volume of the unit hypersphere in $\mathbb{R}^{N_\mathbf{x}}$. The advantage of variance reduction of kernel smoothing is at a cost of increase of bias, but this problem can be alleviated by gradually decreasing the kernel width h as time progresses, an approach being employed in [481].

Kernel smoothing is de facto a regularization technique [87]. Some regularized particle filters were also developed in the past few years [222], [364], [365], [345]. Within particle filtering update, regularization can be taken before or after the correction step, resulting in the so-called pre-regularized particle filter (pre-RPF) and post-regularized particle filter (post-RPF) [345]. The pre-PRF is also close to the kernel particle filter [222] where the kernel smoothing is performed in the resampling step. The implementation of RPF is similar to the regular particle filter, except in the resampling stage. For the post-RPF, the resampling procedure reads as follows [345]:

- Generate $\xi \in \{1, \cdots, N_p\}$, with $\Pr(\xi = i) = W_n^{(i)}$;
- Draw a sample from a selected kernel $\mathbf{s} \sim K(\mathbf{x})$;
- Generate the particles $\mathbf{x}_n^{(i)} = \mathbf{x}_n^{(\xi)} + hA_n\mathbf{s}$, where h is the the optimal bandwidth of the kernel, A_n is chosen to be the square root of the empirical covariance matrix if whitening is used, otherwise $A_n = \xi$.

The resampling of pre-PRF is similar to the that of post-RPF except an additional rejection step is performed, reader is referred to [222], [345] for details. It was proved that the RPF converge to the optimal filter in the weak sense, with a rate $\mathcal{O}(h^2 + 1/\sqrt{N_p})$, when $h = 0$, it reduces to the rate of regular particle filter $\mathcal{O}(1/\sqrt{N_p})$.

In [364], [345], an algorithm called "progressive correction" was proposed for particle filters, in which the correction step is split into several subcorrection steps associated with a decreasing sequence of (fictitious) variance matrices for the observation noise (similar to the idea of *annealing*). The intuition of progressive correction is to decompose the likelihood function into multiple stages since the error induced in the correction step is usually unbounded (e.g. the measurement noise is small) and thus more attention is deserved. Though theoretically attractive, the implementation of partitioned sampling is quite complicated, the details are left for the interested reader and not discussed here.

H. Data Augmentation

The data augmentation idea arises from the missing data problem, it is referred to a scheme of augmenting the observed data, thereby making the probabilistic inference easier. Data augmentation was first proposed by Dempster *et al.* [130] in a deterministic framework for the EM algorithm, and later generalized by Tanner and Wong [445] for posterior distribution estimation in a stochastic framework, which can be viewed as a Rao-Blackwell approximation of the marginal density.

H.1 Data Augmentation is an Iterative Kernel Smoothing Process

Data augmentation is an iterative procedure for solving a fixed operator equation (the following content follows closely [445], [446]). Simply suppose

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \int_Z p(\mathbf{x}|\mathbf{y},\mathbf{z})p(\mathbf{z}|\mathbf{y})d\mathbf{z}, & (115) \\ p(\mathbf{z}|\mathbf{y}) &= \int_X p(\mathbf{z}|\mathbf{x}',\mathbf{y})p(\mathbf{x}'|\mathbf{y})d\mathbf{x}'. & (116) \end{aligned}$$

Substituting (116) to (115), it follows that the posterior satisfies

$$\begin{aligned} \pi(\mathbf{x}) &= \int K(\mathbf{x},\mathbf{x}')\pi(\mathbf{x}')d\mathbf{x}', & (117) \\ K(\mathbf{x},\mathbf{x}') &= \int p(\mathbf{x}|\mathbf{y},\mathbf{z})p(\mathbf{z}|\mathbf{x}',\mathbf{y})d\mathbf{z}, & (118) \end{aligned}$$

where (118) is a *Fredholm integral equation of the first kind*, which can be written in the following operator form

$$\boldsymbol{T}f(\mathbf{x}) = \int K(\mathbf{x},\mathbf{x}')f(\mathbf{x}')d\mathbf{x}', \tag{119}$$

where f is an arbitrary integrable function, $\boldsymbol{T}$ is an integral operator, and (119) is an operator fixed point equation. Noticing the mutual dependence of $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{z}|\mathbf{y})$, by applying successive substitution we can obtain an iterative method

$$\pi_{n+1}(\mathbf{x}) = (\boldsymbol{T}\pi_n)(\mathbf{x}) = (\boldsymbol{T}^{n+1}\pi_0)(\mathbf{x}). \tag{120}$$

It was shown in [445] that under some regularity condition, $\|\pi_{n+1}(\mathbf{x}) - p\| \le \|\pi_n(\mathbf{x}) - p\|$, thus $\pi_n(\mathbf{x}) \to p(\mathbf{x}|\mathbf{y})$ when $n \to \infty$.

If $(\boldsymbol{T}\pi_n)(\mathbf{x})$ cannot be calculated analytically, then $\pi_{n+1}(\mathbf{x})$ can be approximated by the Monte Carlo sampling

$$\pi_{n+1}(\mathbf{x}) = \frac{1}{N_p}\sum_{i=1}^{N_p} p(\mathbf{x}|\mathbf{y},\mathbf{z}^{(i)}). \tag{121}$$

The quantities $\mathbf{z}^{(i)}$ are called *multiple imputations* by Rubin [395], [397]. The data augmentation algorithm consists

[58] It was also called the localization sampling or local multiple imputation [3].

of iterating the *Imputation* (I) step and the *Posterior* (P) step.

1. **I-Step**: Draw the samples $\{\mathbf{z}^{(i)}\}_{i=1}^{N_p}$ from current approximation $\pi_n(\mathbf{x})$ to the predictive distribution $p(\mathbf{z}|\mathbf{y})$, which comprises of two substeps
   - Generate $\mathbf{x}^{(i)}$ from $\pi_n(\mathbf{x})$;
   - Generate $\mathbf{z}^{(i)}$ from $p(\mathbf{z}|\mathbf{y}, \mathbf{x}^{(i)})$.
2. **P-Step**: Update the current approximation to $p(\mathbf{x}|\mathbf{y})$ to be the mixture of conditional densities via (121), where $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$ is supposed to be analytically calculated or sampled easily.

H.2 Data Augmentation as a Bayesian Sampling Method

Data augmentation can be used as a Bayesian sampling technique in MCMC [388]. In order to generate a sample from a distribution $\pi(\mathbf{x}|\mathbf{y})$, the procedure proceeds as follows:

- Start with an arbitrary $\mathbf{z}_{(0)}$.
- For $1 \leq k \leq N$, generate
- $\mathbf{x}_{(k)}$ according to marginal distribution $\pi(\mathbf{x}|\mathbf{y}, \mathbf{z}_{(k-1)})$;
- $\mathbf{z}_{(k)}$ according to marginal distribution $\pi(\mathbf{z}|\mathbf{y}, \mathbf{x}_{(k)})$.

When N is large and the chain $\mathbf{x}_{(k)}$ is ergodic with invariant distribution $\pi(\mathbf{x}|\mathbf{y})$, the final sample $\mathbf{x}_{(N)}$ can be regarded a sample $\mathbf{x}^{(i)} \sim \pi(\mathbf{x}|\mathbf{y})$.

The sample set $\{\mathbf{x}^{(i)}\}_{i=1}^{N_p}$ obtained in this way has a conditional structure [175], [388]. It is interestingly found that one can take advantage of the dual samples $\{\mathbf{z}^{(i)}\}_{i=1}^{N_p}$. Indeed, if the quantity of interest is $\mathbb{E}_\pi[f(\mathbf{x})|\mathbf{y}]$, one can calculate the average of conditional expectation whenever it is analytically computable

$$\hat{\rho}_2 = \frac{1}{N_p}\sum_{i=1}^{N_p} \mathbb{E}_\pi[f(\mathbf{x})|\mathbf{y}, \mathbf{z}^{(i)}] \qquad (122)$$

instead of the unconditional Monte Carlo average

$$\hat{\rho}_1 = \frac{1}{N_p}\sum_{i=1}^{N_p} f(\mathbf{x}^{(i)}). \qquad (123)$$

The justification of substituting (123) with (122) is the Rao-Blackwell Theorem, since

$$\begin{aligned}\mathbb{E}_\pi\Big[(\hat{\rho}_1 - \mathbb{E}_\pi[f(\mathbf{x})|\mathbf{y}])\Big|\mathbf{y}\Big] &= \frac{1}{N_p}\mathrm{Var}_\pi[f(\mathbf{x})|\mathbf{y}] \\ &\geq \frac{1}{N_p}\mathrm{Var}_\pi\Big[\mathbb{E}_\pi[f(\mathbf{x})|\mathbf{y}, \mathbf{z}]\Big|\mathbf{y}\Big] \\ &= \mathbb{E}_\pi\Big[(\hat{\rho}_2 - \mathbb{E}_\pi[f(\mathbf{x})|\mathbf{y}, \mathbf{z}])\Big|\mathbf{y}\Big].\end{aligned}$$

Generally, under a quadratic loss (or any other strictly convex loss), it is favorable to work with conditional expectations. Hence, data augmentation provides a way to approximate the posterior $p(\mathbf{x}|\mathbf{y})$ by the average of the conditional densities [388]

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{N_p}\sum_{i=1}^{N_p} p(\mathbf{x}|\mathbf{y}, \mathbf{z}^{(i)}), \qquad (124)$$

which is identical to (121).

Remarks:

- Data augmentation can be viewed as a two-step Gibbs sampling, where the augmented data $\mathbf{z}$ and true state $\mathbf{x}$ are alternatingly marginalized.
- In the APF, the auxiliary variable can be viewed as a sort of data augmentation technique.
- Similar to the EM algorithm [130], data augmentation algorithm exploits the simplicity of the posterior distribution of the parameter given the augmented data. A detailed discussion on state-of-the-art data augmentation techniques was found in [461], [328].
- A comparative discussion between data augmentation and SIR methods is referred to [445].

I. MCMC Particle Filter

When the state space is very high (say $N_{\mathbf{x}} > 10$), the performance of particle filters depends to a large extent on the choices of proposal distribution. In order to tackle more general and more complex probability distribution, MCMC methods are needed. In particle filtering framework, MCMC is used for drawing the samples from an invariance distribution, either in sampling step or resampling step.

Many authors have tried to integrate the MCMC technique to particle filtering, e.g., [40], [304], [162], [315], [370], [164]. Berzuini *et al.* [40] used the Metropolis-Hastings importance sampling for filtering problem. Recalling the Metropolis-Hastings algorithm in Section V-G.6, within the Bayesian estimation framework, $\pi(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$, the proposal $q(\mathbf{x}', \mathbf{x})$ is rewritten as $q(\mathbf{x}|\mathbf{x}')$, the acceptance probability (moving from $\mathbf{x}$ to $\mathbf{x}'$) (72) can be rewritten by

$$\alpha(\mathbf{x}, \mathbf{x}') = \min\Big[\frac{p(\mathbf{y}|\mathbf{x}')p(\mathbf{x}')q(\mathbf{x}|\mathbf{x}')}{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})q(\mathbf{x}'|\mathbf{x})}, 1\Big]. \qquad (125)$$

Provided we use the prior as proposal (i.e. $q(\mathbf{x}|\mathbf{x}') = p(\mathbf{x})$), (125) will reduce to

$$\alpha(\mathbf{x}, \mathbf{x}') = \min\Big[\frac{p(\mathbf{y}|\mathbf{x}')}{p(\mathbf{y}|\mathbf{x})}, 1\Big], \qquad (126)$$

which says that the acceptance rate only depends on the likelihood. Equivalently, we can define the transition function $K(\mathbf{x}, \mathbf{x}') = p(\mathbf{x}'|\mathbf{x})$ as

$$K(\mathbf{x}, \mathbf{x}') = \begin{cases} q(\mathbf{x}')\min\Big[1, \frac{W(\mathbf{x}')}{W(\mathbf{x})}\Big], & \text{if } \mathbf{x}' \neq \mathbf{x} \\ 1 - \int_{\mathbf{z}\neq\mathbf{x}} q(\mathbf{z})\min[1, \frac{W(\mathbf{z})}{W(\mathbf{x})}]d\mathbf{z}, & \text{if } \mathbf{x}' = \mathbf{x} \end{cases}$$

where $W(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ represents the importance weight. The samples are drawn from Metropolis-Hastings algorithm only after the “burn-in” time of Markov chain, namely the samples during the burn-in time are discarded, and the next N_p samples are stored.[59] However, there are some disadvantages of this algorithm. When the dynamic

[59] It was also suggested by some authors to discard the burn-in period for particle filters for the purpose of on-line processing.

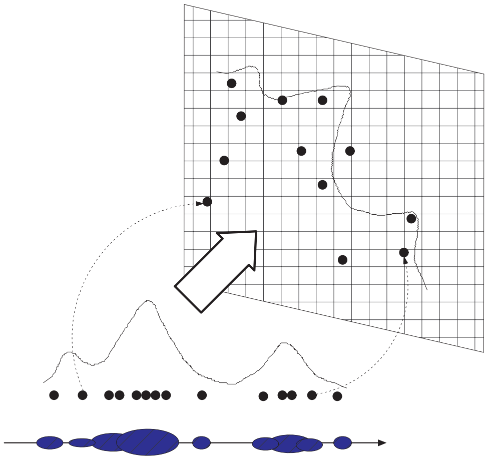

Fig. 10. Sampling-importance-resampling (SIR) followed by a reversible jump MCMC step. The particles are moved w.r.t. an invariant transitional kernel without changing the distribution.

noise ($\Sigma_{\mathbf{d}}$) is small,[60] the Markov chain usually takes a long time to converge, and the burn-in time is varied.

It was also suggested to perform a reversible jump MCMC step, after the resampling, to each particle in order to increase the diversity of simulated samples without affecting the estimated posterior distribution (see Fig. 10). The advantages are twofold [41]: (i) If particles are already distributed according to the posterior, then applying a Markov-chain transition kernel with the same invariant distribution to particles will not change the new particles' distribution, in addition, it also reduces the correlations between the particles; (ii) on the other hand, if particles are not in the region of interest, the MCMC step may have possibility to move them to the interesting state space. Nevertheless, adding MCMC move step also increase the computation burden of the particle filter, thus the merit of such step should be only justified by specific application.

One special MCMC particle filter is the resample-move algorithm [186], [41], which combines SIR and MCMC sampling; it was shown experimentally that this methodology can somehow alleviate the progressive degeneration problem. The basic idea is as follows [186]: The particles are grouped into a set $S_n = \{\mathbf{x}_n^{(i)}\}_{i=1}^{N_p}$ at time step n, and they are propagated through the state-space equations by using SIR and MCMC sampling, at time $n+1$, the resampled particles are moved according to a Markov chain transition kernel to form a new set S_{n+1}; in the *rejuvenation* stage, two steps are performed: (i) in the resample step, draw the samples $\{\mathbf{x}_n^{(i)}\}$ from S_n such that they are selected with probability proportional to $\{W(\mathbf{x}_n^{(i)})\}$; (ii) in the move step, the selected particles are moved to a new position by sampling from a Markov chain transitional kernel. The resample-move algorithm essentially includes SIS [200], [506], [266] as special case, where the rejuvenation step is neglected, as well as the previous work by West [481] and Liu and Chen [304], in the latter of which a Gibbs sampling form of the move step was performed.

Lately, Fearnhead [164] has proposed an efficient method to implement the MCMC step for particle filter based on the sufficient statistics. Usually, the whole trajectories of particles need to be stored [186], Fearnhead instead used the summary of trajectories as sufficient statistics on which the MCMC move is applied. Let $\Psi = \Psi(\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n})$ denote the sufficient statistics for $\mathbf{x}_n$, according to the Factorization theorem (e.g. [388]), the unnormalized joint distribution can be factorized by two functions' product

$$\pi(\mathbf{x}_n, \mathbf{x}_{0:n-1}, \mathbf{z}_{0:n}) = \lambda_1(\mathbf{x}_n, \Psi)\lambda_2(\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n}).$$

The implementation idea is to assume the invariant distribution is $p(\mathbf{x}_n|\Psi)$ conditioning on the sufficient statistics instead of the whole state and measurement trajectories. The sufficient statistics are also allowed to be updated recursively, see [164] for some examples.

J. Mixture Kalman Filters

Mixture Kalman filters (MKF) is essentially a stochastic bank of (extended) Kalman filters, each Kalman filter is run with Monte Carlo sampling approach. The idea was first explored in [6], and further explored by Chen and Liu [83] (also implicitly in [144]) with resampling and rejection control schemes. This also follows West's idea that the posterior can be approximated by a mixture model [481]. In fact, MKF can viewed as a special case of particle filter with marginalization and Rao-Blackwellization on conditionally Gaussian linear dynamic model. The advantage of MKF is its obvious computational efficiency, it also found many successful applications in tracking and communications [83], [84], [476].

K. Mixture Particle Filters

It is necessary to discriminate two kinds of mixture particle filters in the literature: (i) mixture posterior (arising from mixture transitional density or mixture measurement density), and (ii) mixture proposal distribution. The example of the first kind is the Gaussian sum particle filter [268], where the posterior is approximated by a Gaussian sum, which can be further used a sampling-based particle filter for inference. The examples of the second kind were proposed by many authors from different perspectives [162], [69], [370], [144], [459]. The mixture proposal is especially useful and efficient for the situations where the posterior is multimodal. We give more general discussion as follows.

The idea is to assume the underlying posterior is a mixture distribution such that we can decompose the proposal distribution in a similar way. For instance, to calculate a

[60] $\Sigma_{\mathbf{d}}$ is directly related to the variation of samples drawn from transition prior, and consequently related to the sample impoverishment problem.

expected function of interest, we have

$$
\begin{aligned}
\mathbb{E}[f(\mathbf{x})] &= \int f(\mathbf{x}) \sum_{j=1}^{m} c_j p_j(\mathbf{x}) d\mathbf{x}, \\
&= \sum_{j=1}^{m} c_j \int f(\mathbf{x}) p_j(\mathbf{x}) d\mathbf{x}, \\
&= \sum_{j=1}^{m} c_j \int f(\mathbf{x}) \frac{p_j(\mathbf{x})}{q_j(\mathbf{x})} q_j(\mathbf{x}) d\mathbf{x} \\
&= \sum_{j=1}^{m} W'_j \int f(\mathbf{x}) q_j(\mathbf{x}) d\mathbf{x} \qquad (127)
\end{aligned}
$$

where $W'_j = c_j \frac{p_j(\mathbf{x})}{q_j(\mathbf{x})}$. Namely, for m mixtures of $q_i(\mathbf{x})$ with total number of N_p particles, each mixture has N_p/m particles if allocated evenly (but not necessarily). However, the form of $q_i(\mathbf{x})$ can differ and the number of particles associated to $q_i(\mathbf{x})$ can be also different according to the prior knowledge (e.g. their variances). In this context, we have the mixture particle filters (MPF). Each particle filter has individual proposal. The idea of MPF is similar to the stratified sampling and partitioned sampling idea, and includes the idea using EKF/UKF as Gaussian proposal approximation as special cases, as to be discussed sooner. Also note that MPF allow the parallel implementation, and each proposal distribution allows different form and sampling scheme.

The estimate given by MPF is represented as

$$
\begin{aligned}
\mathbb{E}[f(\mathbf{x}_n)] &= \sum_{j=1}^{m} W'_{n,j} \int f(\mathbf{x}_n) q_j(\mathbf{x}_n|\mathcal{Y}_n) d\mathbf{x}_n \\
&= \sum_{j=1}^{m} \frac{\mathbb{E}_{q_j(\mathbf{x}_n|\mathcal{Y}_n)}[W'_{n,j}(\mathbf{x}_n) f(\mathbf{x}_n)]}{\mathbb{E}_{q_j(\mathbf{x}_n|\mathcal{Y}_n)}[W'_{n,j}(\mathbf{x}_n)]} \\
&\approx \sum_{j=1}^{m} \sum_{i=1}^{N_p/m} \tilde{W}'_{j,n}(\mathbf{x}_{j,n}^{(i)}) f(\mathbf{x}_{j,n}^{(i)}), \qquad (128)
\end{aligned}
$$

where $\tilde{W}'_{j,n}(\mathbf{x}_{j,n}^{(i)})$ is the normalized importance weights from the j-th mixture associated with the i-th particle.

L. Other Monte Carlo Filters

There are also some other Monte Carlo filters that has not been covered in our paper, which are either not updated sequentially (but still with recursive nature), or based on HMC or QMC methods. Due to space constraint, we do not extend the discussion and only refer the reader to the specific references.

- Gibbs sampling for dynamic state space model [71], [72]. Those Monte Carlo filters are useful when the real-time processing is not too demanding.
- Quasi Monte Carlo filters or smoothers, which use Metropolis-Hastings algorithm [440], [443].
- Non-recursive Monte Carlo filters [439], [438], [443].
- Particle filters based on HMC technique [94].
- Particle filters based on QMC and lattice technique [361].
- Annealed particle filter [131].
- The branching and interacting particle filters discussed in continuous-time domain [122], [123], [125], [104], [105].
- Genetic particle filter via evolutionary computation [455].

M. Choices of Proposal Distribution

The potential criteria of choosing a good proposal distribution should include:

- The support of proposal distribution should cover that of posterior distribution, in other words, the proposal should have a broader distribution.
- The proposal distribution has a long-tailed behavior to account for outliers.
- Ease of sampling implementation, preferably with linear complexity.
- Taking into account of transition prior and likelihood, as well as most recent observation data.
- Achieving minimum variance.
- Being close (in shape) to the true posterior.

However, achieving either of these goals is not easy and we don't know what the posterior suppose to look like. Theoretically, it was shown [506], [6], [266] that the choice of proposal distribution $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n)$ minimizes the variance of importance weights $W_n^{(i)}$ conditional upon $\mathbf{x}_{0:n-1}^{(i)}$ and $\mathbf{y}_{0:n}$ (see [144] for a simple proof). By this, the importance weights can be recursively calculated as $W_n^{(i)} = W_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_{n-1}^{(i)})$. However, this optimal proposal distribution suffers from certain drawbacks [144]: It requires sampling from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n)$ and evaluating the integral $p(\mathbf{y}_n|\mathbf{x}_{n-1}^{(i)}) = \int p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}) d\mathbf{x}_n$.[61] On the other hand, it should be also pointed out that there is no universal choice for proposal distribution, which is usually problem dependent. Choosing an appropriate proposal distribution requires a good understanding of the underlying problem. In the following, we present some rules of thumb available in the literature and discuss their features.

M.1 Prior Distribution

Prior distribution was first used for proposal distribution [200], [201] because of its intuitive simplicity. If $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$, the importance weights are updated by

$$
W_n^{(i)} = W_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_n^{(i)}), \qquad (129)
$$

which essentially neglects the effect of the most recent observation $\mathbf{y}_n$. In the CONDENSATION (CONditional DENSity propagATION) algorithm [229], [230], a transition prior was used as the proposal distribution for visual

[61] Generally the integral has no analytic form and thus requires approximation; however, it is possible to obtain the analytic evaluation in some cases, e.g. the Gaussian state-space model with nonlinear state equation.

tracking. This kind of proposal distribution is easy to implement, but usually results in a high variance because the most recent observation $\mathbf{y}_n$ is neglected in $p(\mathbf{x}_n|\mathbf{x}_{n-1})$. The problem becomes more serious when the likelihood is peaked and the predicted state is near the likelihood's tail (see Fig. 11 for illustration), in other words, the measurement noise model is sensitive to the outliers.

From (129), we know that importance weights are proportional to the likelihood model. It is obvious that $W(\mathbf{x})$ will be very uneven if the likelihood model is not flat. In the Gaussian measurement noise situation, the flatness will be determined by the variance. If $\Sigma_{\mathbf{v}}$ is small, the distribution of the measurement noise is peaked, hence $W(\mathbf{x})$ will be peaked as well, which makes the the sample impoverishment problem more severe. Hence we can see that, choosing transition prior as proposal is really a brute force approach whose result can be arbitrarily bad, though it was widely used in the literature and sometimes produced reasonably good results (really depending on the noise statistics!). Our caution is: Do *not* run into this proposal model unless you know something about your problem; do *not* use something just because of its simplicity!

For some applications, state equations are modeled as an autoregressive (AR) model $\mathbf{x}_{n+1} = \mathbf{A}_n\mathbf{x}_n + \mathbf{d}_n$, where time-varying $\mathbf{A}_n$ can be determined sequentially or block-by-block way (by solving Yule-Walker equation). In the on-line estimation, it can be augmented into a pseudo-state vector. However, it should be cautioned that for time-varying AR model, the use of transitional prior proposal is not recommended. Many experimental results have confirmed this [189], [467]. This is due to the special stability condition of AR process.[62] When the Monte Carlo samples of AR coefficients are generated violating the stability condition, the AR-driven signal will oscillate and the filtered states will deviate from the true ones. The solution to this problem is Rao-Blackwellization [466] or careful choice of proposal distribution [189].

M.2 Annealed Prior Distribution

The motivation of using transition prior as proposal is its simplicity. However, it doesn't take account of the noise statistics $\Sigma_{\mathbf{d}}$ and $\Sigma_{\mathbf{v}}$. Without too much difficulty, one can imagine that if the samples drawn from prior doesn't cover the likelihood region, the performance of the particle filter will be very poor since the contributions of most particles are insignificant. This fact further motivates us to use annealed prior as proposal to alleviate this situation.

Recall the update equation of importance weights (88), if we let $q(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1})^\beta$, [63] then

$$\begin{aligned} W_n &= W_{n-1}\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})}{q(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_n)} \\ &= W_{n-1}\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})}{p(\mathbf{x}_n|\mathbf{x}_{n-1})^\beta} \\ &= W_{n-1}p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})^\alpha \end{aligned}$$

where $\alpha = 1-\beta$, and $0 \le \alpha \le 1$. When $\alpha = 1$, it reduces to the normal SIR filter (129); when $\alpha = 0$, it is equivalent to taking a uniform distribution (infinitely flat) as proposal. The choice of annealing parameter α depends on the knowledge of the noise statistics:

- When $\Sigma_{\mathbf{d}} < \Sigma_{\mathbf{v}}$, the support of prior distribution is largely outside the flat likelihood (see the first illustration of Fig. 11). In this case, we let $0 < \alpha < 1$, which thus makes the shape of the prior more flat. This is also tantamount to the effect of "jitter": adding some artificial noise makes the drawn samples broadly located.[64]
- When $\Sigma_{\mathbf{d}} \approx \Sigma_{\mathbf{v}}$, the most support of prior overlap that of the likelihood (see the second illustration of Fig. 11). In this case, prior proposal is fine and we let $\alpha = 1$.
- When $\Sigma_{\mathbf{d}} > \Sigma_{\mathbf{v}}$, the prior is flat compared to the peaked likelihood (see the third illustration of Fig. 11). In this case, we cannot do much about it by changing α.[65] And we will discuss this problem in detail in subsections M.3 and M.5.

Another perspective to understand the parameter β is following: by taking the logarithm of the posterior, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, we have

$$\log p(\mathbf{x}_n|\mathbf{y}_{0:n}) \propto \log p(\mathbf{y}_n|\mathbf{x}_n) + \beta \log p(\mathbf{x}_n|\mathbf{x}_{n-1}),$$

which essentially states that the log-posterior can be interpreted as a penalized log-likelihood, with $\log p(\mathbf{x}_n|\mathbf{x}_{n-1})$ as a smoothing prior, β is a tuning parameter controlling the trade-off between likelihood and prior.

M.3 Likelihood

When the transition prior is used as proposal, the current observation $\mathbf{y}_n$ is neglected. However, the particles that have larger importance weights at previous time step $n-1$ don't necessarily have large weights at current step n. In some cases, the likelihood is far tighter than the prior and is comparably closer (in shape) to the posterior. Hence we can employ the likelihood as proposal distribution,[66] which results in the likelihood particle filter. The idea behind that is instead of drawing samples from the state transition density and then weighting them according to their likelihood, samples are drawn from the likelihood

[62] A sufficient condition for stability of AR model is that the poles are strictly within the unit circle.

[63] β can be viewed as a variational parameter.

[64] The pdf of the sum of two random variables is the convolution of the two pdf's of respective random variables.

[65] Note that letting $\alpha > 1$ doesn't improve the situation.

[66] Here likelihood can be viewed as an "observation density" in terms of the states.

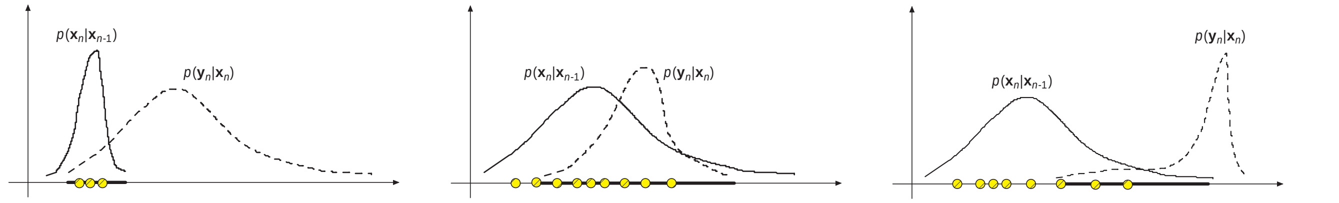

Fig. 11. **Left:** $\Sigma_{\mathbf{d}} < \Sigma_{\mathbf{v}}$, transition prior $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ is peaked compared to the flat likelihood $p(\mathbf{y}_n|\mathbf{x}_n)$, and their overlapping region is indicated by the thick line; **Middle:** $\Sigma_{\mathbf{d}} \approx \Sigma_{\mathbf{v}}$, the support of prior and likelihood largely overlap, where the prior proposal works well; **Right:** an illustration of poor approximation of transition prior as proposal distribution when the likelihood is peaked $\Sigma_{\mathbf{d}} > \Sigma_{\mathbf{v}}$. Sampling from the prior doesn't generate sufficient particles in the overlapping region.

and then assigned weights proportional to the state transition density.[67] In some special cases where the likelihood model can be inverted easily $\mathbf{x}_n = \mathbf{g}^{-1}(\mathbf{y}_n, \mathbf{v}_n)$, one can alteratively use likelihood as proposal distribution. To give an example [19], assume the likelihood model is quadratic, say $\mathbf{y}_n = \mathbf{G}_n\mathbf{x}_n^2 + \mathbf{v}_n$, without loss of generality. Denote $\mathbf{s}_n = |\mathbf{x}_n|^2$, then we can sample $\mathbf{s}_n$ from the equation $\mathbf{s}_n = \mathbf{G}_n^{-1}(\mathbf{y}_n - \mathbf{v}_n)$. From the Bayes rule, the proposal can be chosen to be [19]

$$p(\mathbf{s}_n|\mathbf{y}_n) \propto \begin{cases} p(\mathbf{y}_n|\mathbf{s}_n), & \text{if } \mathbf{s}_n \geq 0 \\ 0, & \text{otherwise} \end{cases}, \tag{130}$$

then $p(\mathbf{x}_n|\mathbf{s}_n^{(i)})$ is chosen to be a pair of Dirac delta functions

$$p(\mathbf{x}_n|\mathbf{s}_n^{(i)}) = \frac{\delta\Big(\mathbf{x}_n - \sqrt{\mathbf{s}_n^{(i)}}\Big) + \delta\Big(\mathbf{x}_n + \sqrt{\mathbf{s}_n^{(i)}}\Big)}{2}. \tag{131}$$

By letting the proposal $q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{y}_{0:n}) \propto p(\mathbf{x}_n|\mathbf{s}_n)p(\mathbf{s}_n|\mathbf{y}_n)$, The importance weights $W_n^{(i)}$ are updated as [19]

$$W_n^{(i)} \propto W_{n-1}^{(i)} p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}) \frac{p(\mathbf{x}_n^{(i)}|\mathbf{y}_n)}{p(\mathbf{s}_n^{(i)}|\mathbf{y}_n)}, \tag{132}$$

where the ratio $\frac{p(\mathbf{x}_n^{(i)}|\mathbf{y}_n)}{p(\mathbf{s}_n^{(i)}|\mathbf{y}_n)}$ is the determinant of the Jacobian of the transformation from $\mathbf{s}_n$ to $\mathbf{x}_n$ [19]

$$\frac{p(\mathbf{x}_n^{(i)}|\mathbf{y}_n)}{p(\mathbf{s}_n^{(i)}|\mathbf{y}_n)} \propto \Big|\frac{d\mathbf{s}_n}{d\mathbf{x}_n}\Big| = 2|\mathbf{x}_n|. \tag{133}$$

Hence (132) is rewritten as

$$W_n^{(i)} \propto W_{n-1}^{(i)} p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})|\mathbf{x}_n^{(i)}|. \tag{134}$$

Taking the likelihood as proposal amounts to pushing the particles to the high likelihood region, this is efficient when the transition prior is broad ($\Sigma_{\mathbf{d}}$ is large) compared to the peaked likelihood ($\Sigma_{\mathbf{v}}$ is small). In above quadratic likelihood example, the procedure of likelihood particle filter is given in Table VII.

Remarks:

TABLE VII

Likelihood Particle Filter (an example in the text).

```
For time steps $n = 0, 1, 2, \cdots$
1: Draw i.i.d. samples $\mathbf{s}_n^{(i)} \sim \hat{p}(\mathbf{s}_n|\mathbf{y}_n) \propto p(\mathbf{y}_n|\mathbf{s}_n)$;
2: $u = \mathcal{U}(0,1)$, $\mathbf{x}_n^{(i)} = \text{sgn}(u - \frac{1}{2})\sqrt{\mathbf{s}_n^{(i)}}$;
3: Importance weight update: $W_n^{(i)} = W_{n-1}^{(i)} p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})|\mathbf{x}_n^{(i)}|$;
4: Weight normalization to get $\tilde{W}_n^{(i)}$;
5: Resampling to get new $\{\mathbf{x}_n^{(i)}, W_n^{(i)}\}_{i=1}^{N_p}$ using SIS procedure.
```

- Note that it is *not* always possible to sample from likelihood because the mapping $\mathbf{y}_n = \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n)$ is usually many-to-one. Above example is only a two-to-one mapping whose distribution $p(\mathbf{x}_n|\mathbf{y}_n)$ is bimodal.
- It is cautioned that using likelihood as proposal distribution will increase the variance of the simulated samples. For instance, from the measurement equation $\mathbf{y}_n = \mathbf{x}_n + \mathbf{v}_n$ ($\mathbf{v}_n \sim \mathcal{N}(0, \Sigma_{\mathbf{v}})$), we can draw samples from $\mathbf{x}_n^{(i)} = \mathbf{y}_n - \mathbf{v}_n^{(i)}$, thus $\mathbb{E}[\mathbf{x}_n] = \mathbb{E}[\mathbf{y}_n]$, $\text{Var}[\mathbf{x}_n] = \text{Var}[\mathbf{y}_n] + \Sigma_{\mathbf{v}}$. This is a disadvantage for the Monte Carlo estimate. Hence it is often not recommended especially when $\Sigma_{\mathbf{v}}$ is large.

M.4 Bridging Density and Partitioned Sampling

Bridging density [189], was proposed for proposal distribution as an intermediate distribution between the prior and likelihood. The particles are reweighed according to the intermediate distribution and resampled.

Partitioned sampling [313], was also proposed for a proposal distribution candidate, especially when the distributions are the functions of part of the states and the peaked likelihood can be factorized into several broader distributions. The basic procedure is as follows [313], [314]:

- Partition the state space into two or more parts;
- Draw the samples in the partitioned space, and pass the samples into the factorized dynamics respectively;
- Generate new particle sets via resampling.

Since the particles are drawn independently from different partitioned spaces, which are little or not correlated, partitioned sampling leads to a considerable improvement in sampling efficiency and reduction of the need of the samples. This scheme is very useful especially when the measurement components are independent and have different individual likelihood models, e.g. [313], [464].

[67] The likelihood particle filter is similar but not identical to the APF in that neither the auxiliary variable is introduced, nor is the mixture density proposal involved.

M.5 Gradient-Based Transition Density

Bearing in mind the second and third proposal criteria in the beginning of this subsection, we also proposed another proposal distribution by using the gradient information [88]. Before sampling from the transition density $\mathbf{x}_n^{(i)} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1})$, we attempt to use the information ignored in the current observation $\mathbf{y}_n$. To do that, we plug in an intermediate step (MOVE-step) to move the particles in previous step towards the gradient descent direction, [68] by using first-order information. The idea behind that is to push the particles into the high likelihood region, where the likelihood is evaluated by current observation $\mathbf{y}_n$ and previous state $\mathbf{x}_{n-1}$. For instance, the MOVE-step can be implemented through

- *Gradient descent*

$$\hat{\mathbf{x}}_{n|n-1} = \hat{\mathbf{x}}_{n-1|n-1} - \eta \frac{\partial(\mathbf{y}_n - \mathbf{g}(\mathbf{x}))^2}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{n-1|n-1}},$$

  where the scalar $0 < \eta < 1$ is the learning rate parameter.

- *Natural gradient*

$$\hat{\mathbf{x}}_{n|n-1} = \hat{\mathbf{x}}_{n-1|n-1} - \eta \Sigma_{\mathbf{d}}^{-1} \frac{\partial(\mathbf{y}_n - \mathbf{g}(\mathbf{x}))^2}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{n-1|n-1}},$$

- *EKF updates* [120]

$$\begin{aligned}
\mathbf{P}_{n|n-1} &= \mathbf{P}_{n-1|n-1} + \Sigma_{\mathbf{d}} \\
\mathbf{K}_n &= \mathbf{P}_{n|n-1}\hat{\mathbf{G}}_n^T(\hat{\mathbf{G}}_n\mathbf{P}_{n|n-1}\hat{\mathbf{G}}_n^T + \Sigma_{\mathbf{v}})^{-1} \\
\hat{\mathbf{x}}_{n|n-1} &= \hat{\mathbf{x}}_{n-1|n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{g}(\hat{\mathbf{x}}_{n-1|n-1})) \\
\mathbf{P}_{n|n} &= \mathbf{P}_{n|n-1} - \mathbf{K}_n\hat{\mathbf{G}}_n\mathbf{P}_{n|n-1},
\end{aligned}$$

  where $\hat{\mathbf{G}}_n = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}_{n|n-1}}$.

The MOVE-step is followed by the normal sampling from transition density, this new proposal distribution can be understood as a one-step-ahead transition density in a sense that it uses the likelihood model (gradient information) a priori to help choose samples. In this sense, it is similar to the APF and likelihood particle filter. For more discussions and experimental results of this gradient-based SIR filter, see [88].

M.6 EKF as Proposal Distribution

The proposal distribution $q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ can be assumed to be a parameterized mixture distribution (e.g. Gaussian mixture), with finite-dimensional parameters determined by $\mathbf{x}_{n-1}$ and $\mathbf{y}_n$. If the optimal proposal distribution is nonlinear, it can be approximated by an EKF, as shown in [144], [83]. In this case, the state-space model reduces to a nonlinear additive Gaussian model:

$$\begin{aligned}
\mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n) + \mathbf{d}_n, && (135a) \\
\mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n) + \mathbf{v}_n, && (135b)
\end{aligned}$$

where $\mathbf{d}_n$ and $\mathbf{v}_n$ are assumed to be Gaussian distributed. Following [143], [144], we denote the log-likelihood of $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ as $l(\mathbf{x}) = \log p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$, and

$$l'(\mathbf{x}) = \frac{\partial l(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_n}, \quad l''(\mathbf{x}) = \frac{\partial l^2(\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^T}\Big|_{\mathbf{x}=\mathbf{x}_n},$$

thus $l(\mathbf{x}_n)$ can be approximated by the second-order Taylor series:

$$l(\mathbf{x}_n) \approx l(\mathbf{x}) + l'(\mathbf{x})(\mathbf{x}_n - \mathbf{x}) + \frac{1}{2}(\mathbf{x}_n - \mathbf{x})^T l''(\mathbf{x})(\mathbf{x}_n - \mathbf{x}).$$

Under the assumption that $l(\mathbf{x}_n)$ being concave, the proposal distribution can be shown to have a Gaussian distribution

$$q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}) + \mathbf{x}, \Sigma(\mathbf{x})), \tag{136}$$

where the covariance and mean are given by $\Sigma(\mathbf{x}) = -l''(\mathbf{x})^{-1}$ and $\boldsymbol{\mu}(\mathbf{x}) = \Sigma(\mathbf{x})l'(\mathbf{x})$, respectively; when $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ is unimodal, it reduces to the zero mean $\boldsymbol{\mu}(\mathbf{x}) = \mathbf{0}$.

M.7 Unscented Particle Filter

In [459], [474], the unscented Kalman filter (UKF) was used to approximate the proposal distribution of the particle filter, which results in the so-called unscented particle filter (UPF). The advantage of UKF over EKF to approximate the proposal distribution lies in the fact that UKF can better handle the heavy-tailed distributions thus more tailored for non-Gaussian scenarios. In fact, UPF has been successfully applied in object tracking [398], financial time series modeling, robot navigation. Detailed implementation of UPF is referred to [459], [474]. EKF proposal and UPF both use Gaussian approximation of proposal, but UKF produces more accurate estimate than EKF and it is derivative-free.

N. Bayesian Smoothing

As discussed in the beginning, filtering technique can be extended to the smoothing problem,[69] where the future observations are allowed to estimate current state. In the Bayesian/particle filtering framework, the task is to estimate the posterior density $p(\mathbf{x}_n|\mathbf{y}_{0:n+\tau})$. In particular, three kinds of smoothing are discussed in the below.

N.1 Fixed-point smoothing

Fixed-point smoothing is concerned with achieving smoothed estimate of state $\mathbf{x}_n$ at a fixed point n, i.e. with obtaining $\hat{\mathbf{x}}_{n|n+\tau}$ for fixed n and all $\tau \geq 1$. In linear case, the fixed-point smoothing problem is a Kalman filtering problem in disguise and therefore able to be solved by direct use of Kalman filter techniques [12]. Suppose the index of the fixed point is m at time step n ($m \leq n$), we want to estimate the posterior $p(\mathbf{x}_m|\mathbf{y}_{0:n})$. By *forward filtering*

[68] Similar idea was also used in [120] for training neural networks.

[69] The multiple-step ahead prediction was discussed in [144], [443].

forward sampling, at time n we know the posterior distribution $P(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$, by marginalization, we can obtain

$$P(\mathbf{x}_m|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} \tilde{W}_n^{(i)} \delta(\mathbf{x}_m - \mathbf{x}_m^{(i)}),$$

namely, we use current important weights to replace the previous values.

In the simplest case where only one-step backward smoothing (i.e. $\tau = 1$) is considered, it reduces to

$$P(\mathbf{x}_{n-1}|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} \tilde{W}_n^{(i)} \delta(\mathbf{x}_{n-1} - \mathbf{x}_{n-1}^{(i)}),$$

the justification for this approximation is to assume the important weights $\tilde{W}_n^{(i)}$ are more accurate than $\tilde{W}_m^{(i)}$ (and $\tilde{W}_{n-1}^{(i)}$), since they are calculated based on more information.

If the fixed point is the current time step (i.e. $\tau = 0$), we can also smooth the estimate by sampling the state trajectory history [162]: $\mathbf{x}_n^{(i)} \sim p(\mathbf{x}_n|\mathcal{X}_{n-1}^{(i)})$ where $\mathcal{X}_{n-1}^{(i)} = \{\mathbf{x}_n^{(n-\tau)}, \cdots, \mathbf{x}_{n-1}^{(i)}\}$ $(1 \leq \tau \leq n)$. Namely, the current particles are sampled from a τ-length state history, and consequently the memory requirement is τN_p. The new state history $\mathcal{X}_n^{(i)}$ is generated by simply augmenting the $\mathbf{f}(\mathbf{x}_{n-1}^{(i)}, \mathbf{d}_{n-1})$ to $\mathcal{X}_{n-1}^{(i)}$ and discard the least recent one. This procedure certainly is more computationally demanding.

N.2 Fixed-lag smoothing

Fixed-lag smoothing is concerned with on-line smoothing of data where there is a fixed delay τ between state reception and the availability of its estimate, i.e. with obtaining $\hat{\mathbf{x}}_{n|n+\tau}$ for all n and fixed τ.

Similar to the fixed-point smoothing, at the step $n + \tau$, the particle filter yields the approximated distribution $\hat{P}(\mathbf{x}_{0:n+\tau}|\mathbf{y}_{0:n+\tau})$

$$\hat{P}(\mathbf{x}_{0:n+\tau}|\mathbf{y}_{0:n+\tau}) = \sum_{i=1}^{N_p} \tilde{W}_{n+\tau}^{(i)} \delta(\mathbf{x}_{0:n+\tau} - \mathbf{x}_{0:n+\tau}^{(i)}). \quad (137)$$

By marginalization, we can obtain the approximated fixed-lag smoothing distribution

$$\hat{P}(\mathbf{x}_n|\mathbf{y}_{0:n+\tau}) \approx \sum_{i=1}^{N_p} \tilde{W}_{n+\tau}^{(i)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}). \quad (138)$$

Hence in order to get the smoothing density, we need to restore the trajectories of states and draw the samples from respective distribution. Ideally this will give a better result, in practice however, this is not true. First, when τ is big, the approximations (137) and (138) are poor [144]; second, resampling brings inaccuracy to the approximation especially in SIR where resampling is performed in every iteration. To overcome these problems, Clapp and Godsill [98] proposed an alternative way. Using Bayes rule, the fixed-lag smoothing density is factorized by

$$\begin{aligned} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n+\tau}) &= \frac{p(\mathbf{y}_{n+\tau}|\mathbf{y}_{0:n+\tau-1}, \mathbf{x}_{0:n}) p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n+\tau-1})}{p(\mathbf{y}_{n+\tau}|\mathbf{y}_{0:n+\tau-1})} \\ &= \frac{p(\mathbf{y}_{n+\tau}|\mathbf{y}_{0:n+\tau-1}, \mathbf{x}_n)}{p(\mathbf{y}_{n+\tau}|\mathbf{y}_{0:n+\tau-1})} \times \\ &\quad p(\mathbf{x}_n|\mathbf{y}_{n:n+\tau-1}, \mathbf{x}_{0:n-1}) p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n+\tau-1}). \end{aligned}$$

Using a factorized proposal distribution

$$\begin{aligned} q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n+\tau}) &= q(\mathbf{x}_0|\mathbf{y}_{0:\tau}) \prod_{t=1}^{n} q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t+\tau}) \\ &= q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n+\tau}) q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n+\tau-1}), \end{aligned}$$

the unnormalized importance weights can be updated by

$$W(\mathbf{x}_{0:n+\tau}) = W(\mathbf{x}_{0:n+\tau-1}) \times \\ \frac{p(\mathbf{y}_{n+\tau}|\mathbf{y}_{n-1:n+\tau-1}, \mathbf{x}_{0:n}) p(\mathbf{x}_n|\mathbf{y}_{n:n+\tau-1}, \mathbf{x}_{0:n-1})}{q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n+\tau}) p(\mathbf{y}_{n+\tau}|\mathbf{y}_{0:n+\tau-1})}.$$

Generally, $p(\mathbf{y}_{n+\tau}|\mathbf{y}_{n-1:n+\tau-1}, \mathbf{x}_{0:n})$ is not evaluated, but for sufficiently large τ, it can be approximately viewed as a constant for all $\mathbf{x}_{0:n}$ [98]. The fixed-lag smoothing is a *forward sampling backward chaining* procedure. However, the smoothing density $p(\mathbf{x}_n|\mathbf{y}_{n+\tau})$ can be also obtained using the filtered density instead of fixed-lag smoothing technique by using the *forward filtering backward sampling* technique [71], [143], [98], [466]. Besides, the joint estimation problem (with state and uncertain parameter) can be also tackled using fixed-lag smoothing technique, reader is referred to [98] for details.

N.3 Fixed-interval smoothing

Fixed-interval smoothing is concerned with the smoothing of a finite set of data, i.e. with obtaining $\hat{\mathbf{x}}_{n|M}$ for fixed M and all n in the interval $0 \leq n \leq M$. Fixed-interval smoothing is usually discussed in an off-line estimation framework. But for short interval, the sequential estimation is still possible with the increasing computer power nowadays.

Firstly in the forward step, we run a particle filter to obtain $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ for all $0 < n < M$. Secondly in the backward step, the smoothing process is recursively updated by

$$\begin{aligned} p(\mathbf{x}_{n:M}|\mathbf{y}_{0:M}) &= p(\mathbf{x}_{n+1:M}|\mathbf{y}_{0:M}) p(\mathbf{x}_n|\mathbf{x}_{n+1:M}, \mathbf{y}_{0:M}) \\ &= p(\mathbf{x}_{n+1:M}|\mathbf{y}_{1:M}) p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n}) \\ &= p(\mathbf{x}_{n+1:M}|\mathbf{y}_{1:M}) \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_{0:n}) p(\mathbf{x}_n|\mathbf{y}_{0:n})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})} \end{aligned} \quad (139)$$

where the second step uses the assumption of first-order Markov dynamics. In (139), $p(\mathbf{x}_{n:M}|\mathbf{y}_{0:M})$ denotes current smoothed estimate, $p(\mathbf{x}_{n+1:M}|\mathbf{y}_{0:M})$ denotes future smoothed estimate, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ is the current filtered estimate, $\frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_{0:n})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})}$ is the incremental ratio of modified dynamics.

Similar to the fixed-lag smoothing, at time step n, we can have the following distribution

$$\hat{P}(\mathbf{x}_{0:M}|\mathbf{y}_{0:M}) = \sum_{i=1}^{N_p} \tilde{W}_M^{(i)} \delta(\mathbf{x}_{0:M} - \mathbf{x}_{0:M}^{(i)}).$$

By marginalizing the above distribution, we can further obtain $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:M})$ for any $0 \le n \le M$. In practice, this is infeasible because of the weight degeneracy problem [144]: At time M, the state trajectories $\{\mathbf{x}_{0:M}^{(i)}\}_{i=1}^{N_p}$ have been possibly resampled many times ($M-1$ times in the worst case), hence there are only a few distinct trajectories at times n for $n \ll M$. Doucet, Godsill and Andrieu proposed [144] a new fixed-interval smoothing algorithm as follows. Rewriting $p(\mathbf{x}_n|\mathbf{y}_{0:M})$ via [258]

$$p(\mathbf{x}_n|\mathbf{y}_{0:M}) = p(\mathbf{x}_n|\mathbf{y}_{0:n}) \int \frac{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:M})p(\mathbf{x}_{n+1}|\mathbf{x}_n)}{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})} d\mathbf{x}_{n+1},$$

the smoothing density $p(\mathbf{x}_n|\mathbf{y}_{0:M})$ is approximated by

$$\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:M}) = \sum_{i=1}^{N_p} \tilde{W}_{n|M}^{(i)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}), \tag{140}$$

where $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:M})$ is assumed to have the same support (described by the particles) as the filtering density $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n})$ but with different important weights. The normalized importance weights $\tilde{W}_{n|M}^{(i)}$ are calculated as follows:

- Initialization: At time $n = M$, set $\tilde{W}_{n|M}^{(i)} = \tilde{W}_M^{(i)}$.
- Evaluation: For $n = M-1, \cdots, 0$,

$$\tilde{W}_{n|M}^{(i)} = \sum_{j=1}^{N_p} \tilde{W}_{n+1|M}^{(i)} \frac{\tilde{W}_n^{(i)} p(\mathbf{x}_{n+1}^{(j)}|\mathbf{x}_n^{(i)})}{\sum_{i=1}^{N_p} \tilde{W}_n^{(i)} p(\mathbf{x}_{n+1}^{(j)}|\mathbf{x}_n^{(i)})} \tag{141}$$

The derivation of (141) is referred to [144]. The algorithmic complexity is $\mathcal{O}(MN_p^2)$ with memory requirement $\mathcal{O}(MN_p)$. Some other work on fixed-interval smoothing using rejection particle filters are found in [259], [438], [222].

O. Likelihood Estimate

Particle filters can be also used to estimate the likelihood [259], [144], [223], wherever the maximum-likelihood estimation principle can be applied.[70]

Suppose we want to estimate the likelihood of the data

$$p(\mathbf{y}_{0:n}) = \int W(\mathbf{x}_{0:n}) q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) d\mathbf{x}_{0:n}, \tag{142}$$

as discussed earlier, if the proposal distribution is transition prior, the *conditional* likelihood (observation density) will be given by

$$\hat{p}(\mathbf{y}_n|\mathbf{x}_n) = \frac{1}{N_p} \sum_{i=1}^{N_p} W_n^{(i)}(\mathbf{x}_n),$$

which can be used to approximate (142) to get $\hat{P}(\mathbf{y}_n) = \frac{1}{N_p}\sum_{i=1}^{N_p} W_n^{(i)}$. However, this is an a priori likelihood $\hat{P}_{n|n-1}(\mathbf{y}_n)$ which uses the predicted estimate $\hat{\mathbf{x}}_{n|n-1}$ instead of the filtered estimate $\hat{\mathbf{x}}_{n|n}$; on the other hand, the resampling step makes the a posteriori likelihood estimate impossible. Alternatively, we can use another method for estimating likelihood [144]. By factorization of (142), we obtain

$$p(\mathbf{y}_{0:n}) = p(\mathbf{y}_0) \prod_{t=1}^{n} p(\mathbf{y}_t|\mathbf{y}_{0:t-1}), \tag{143}$$

where

$$\begin{aligned} p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) &= \int p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) d\mathbf{x}_n \\ &= \int p(\mathbf{y}_n|\mathbf{x}_{n-1}) p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}. \end{aligned}$$

where the first equality uses the predicted estimate (at time step n) based on $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$, and second equality uses the filtered estimate at time step $n-1$. The likelihood based these estimates are given respectively by

$$\hat{P}(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \sum_{i=1}^{N_p} \tilde{W}_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_n^{(i)}), \tag{144a}$$

$$\hat{P}(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \sum_{i=1}^{N_p} \tilde{W}_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_{n-1}^{(i)}). \tag{144b}$$

A detailed discussion on the likelihood estimate using different particle filters and different sampling schemes is referred to [443].

P. Theoretical and Practical Issues

P.1 Convergence and Asymptotic Results

As discussed earlier, although the convergence[71] of Monte Carlo approximation is quite clear (e.g. [180]), the convergence behavior of sequential Monte Carlo method or particle filter is different and deserves special attention. Many authors have explored this issue from different perspectives, but most results are available in the probability literature. In particular, it has been theoretically shown that under some mild conditions the particle methods converge to the solution of the Zakai equation [103], [107] and Kushner-Stratonovich equation [104]. Crisan [106] presented a rigorous mathematical treatment of convergence of particle filters and gave the sufficient and necessary conditions for the a.s. convergence of particle filter to the true posterior. A review of convergence results on particle filtering methods has been recently given by Crisan and Doucet from practical point of view [106], [102]. We summarize the main results from their survey paper.

Almost Sure Convergence: If the the transition kernel $K(\mathbf{x}_t|\mathbf{x}_{t-1})$ is *Feller*,[72], importance weights are upper bounded, and the likelihood function is continuous,

[70] In fact, the Monte Carlo EM (MCEM), or quasi Monte Carlo EM algorithms can be developed within this framework [389], however, further discussion is beyond the scope of current paper.

[71] A brief introduction of different concepts of convergence is given in Appendix B.

[72] A kernel is *Feller* means that for any continuous bounded function ϕ, $K\phi$ is also a continuous bounded function.

bounded, and strictly positive, then with $N_p \to \infty$ the filtered density given by particle filter converges asymptotically to the true posterior.

Mean Square Convergence: If likelihood function is bounded, for any bounded function $\phi \in \mathbb{R}^{N_\mathbf{x}}$, then for $t \geq 0$, there exists a $C_{t|t}$ independent of N_p s.t.

$$\mathbb{E}\Big[\Big((\hat{P}_{t|t}, \phi) - (P_{t|t}, \phi)\Big)^2\Big] \leq C_{t|t}\frac{\|\phi\|^2}{N_p}, \tag{145}$$

where $(\hat{P}_{t|t}, \phi) = \int \phi(\mathbf{x}_{0:t})P(d\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$, $\|\phi\| = \sup_{\mathbf{x}_{0:t}} |\phi(\mathbf{x}_{0:t})|$. It should be cautioned that, it seems at the first sight that particle filtering method beats the curse of dimensionality,[73] as the rate of convergence, $1/N_p$, is independent on the state dimension $N_\mathbf{x}$. This is nevertheless *not* true because in order to assure (145) holds, the number of particles N_p needs to increase over the time since it depends on $C_{t|t}$, a term that further relies on $N_\mathbf{x}$. As discussed in [102], in order to assure the uniform convergence, both $C_{t|t}$ and the approximation error accumulates over the time.[74] This phenomenon was actually observed in practice and exemplified in [359], [116], [361]. Daum and Huang particularly gave a critical comment on this problem and presented some empirical formula for complexity estimate. Besides, the uniform convergence and stability issues were also discussed in [294].

In a high-dimensional space (order of tens or higher), particle filters still suffer the problem of curse of dimensionality. Empirically, we can estimate the requirement of the number of particles, although this bound in practice is loose and usually data/problem dependent. Suppose the minimum number is determined by the effective volume (variance) of the search space (proposal) against the target space (posterior). If the proposal and posterior are uniform in two $N_\mathbf{x}$-dimensional hyperspheres with radii r and R ($R > r$) respectively,[75] the effective particle number N_{eff} is approximately measured by the the volume ratio in the proposal space against posterior space, namely

$$N_{eff} \approx N_p \times (r/R)^{N_\mathbf{x}}$$

when the ratio is low ($r \ll R$), the effective number decreases exponentially as $N_\mathbf{x}$ increases; on the other hand, if we want to keep the effective number as a constant, we need to increase N_p exponentially as $N_\mathbf{x}$ increases.

An important asymptotic result is the error bound of the filter. According to the Cramér-Rao theorem, the expected square error of an estimate is generally given by

$$\begin{aligned}\mathcal{E}(\mathbf{x}) &= \mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}})^2] \\ &\geq \frac{\Big[1 + \frac{d\mathbb{E}[\hat{\mathbf{x}} - \mathbf{x}]}{d\mathbf{x}}\Big]^2}{\mathbf{J}(\mathbf{x})} + (\mathbb{E}[\hat{\mathbf{x}} - \mathbf{x}])^2, \end{aligned} \tag{146}$$

where $\mathbf{J}(\mathbf{x})$ is the Fisher information matrix defined by

$$\mathbf{J}(\mathbf{x}) = \mathbb{E}\Big[\Big(\frac{\partial}{\partial \mathbf{x}}\log p(\mathbf{x}, \mathbf{y})\Big)\Big(\frac{\partial}{\partial \mathbf{x}}\log p(\mathbf{x}, \mathbf{y})\Big)^T\Big].$$

If the estimate is unbiased (namely $\mathbb{E}[\hat{\mathbf{x}} - \mathbf{x}] = 0$), then $\mathcal{E}(\mathbf{x})$ is equal to the variance, and (146) reduces to

$$\mathcal{E}(\mathbf{x}) \geq \mathbf{J}^{-1}(\mathbf{x}) \tag{147}$$

and the estimate satisfying (147) is called Fisher efficient. Kalman filter is Fisher-efficient under LQG circumstance in which the state-error covariance matrix plays a similar role as the inverse Fisher information matrix.[76] Many efforts were also devoted to studying the error bounds of nonlinear filtering [504], [45], [138], [188], [407], [451] (see also [410] for a review and unified treatment, and the references therein). Naturally, the issue is also interesting within the particle filtering framework. Recently, it has been established in [36] that under some regularity conditions, the particle filters also satisfy the Cramér-Rao bound[77]

$$\mathbb{E}[\tilde{\mathbf{x}}_n\tilde{\mathbf{x}}_n^T] \geq \mathbf{P}_n \tag{148}$$

$$\mathbb{E}[\|\tilde{\mathbf{x}}_n\|^2] \geq \mathrm{tr}(\mathbf{P}_n) \tag{149}$$

where $\tilde{\mathbf{x}}_n = \mathbf{x}_n - \hat{\mathbf{x}}_{n|n}$ is the one-step ahead prediction error, and

$$\begin{aligned}
\mathbf{P}_{n+1} &= \mathbf{F}_n(\mathbf{P}_n^{-1} + \mathbf{R}_n^{-1})^{-1}\mathbf{F}_n^T + \mathbf{G}_n\mathbf{Q}_n\mathbf{G}_n^{-1}, \\
\mathbf{P}_0^{-1} &= \mathbb{E}\Big[-\frac{\partial}{\partial \mathbf{x}_0\mathbf{x}_0}\log p(\mathbf{x}_0)\Big], \\
\mathbf{F}_n &= \mathbb{E}\Big[\frac{\partial}{\partial \mathbf{x}_n}\mathbf{f}(\mathbf{x}_n, \mathbf{d}_n)\Big], \\
\mathbf{R}_n^{-1} &= \mathbb{E}\Big[-\frac{\partial}{\partial \mathbf{x}_n\mathbf{x}_n}\log p(\mathbf{y}_n|\mathbf{x}_n)\Big], \\
\mathbf{G}_n^T &= \mathbb{E}\Big[\frac{\partial}{\partial \mathbf{d}_n}\mathbf{f}(\mathbf{x}_n, \mathbf{d}_n)\Big], \\
\mathbf{Q}_n^{-1} &= \mathbb{E}\Big[-\frac{\partial}{\partial \mathbf{d}_n\mathbf{d}_n}\log p(\mathbf{d}_n)\Big].
\end{aligned}$$

The upper bound is time-varying and can be recursively updated by replacing the expectation with Monte Carlo average. For derivation details and discussions, see [35], [36]; for more general unified treatment (filtering, prediction, smoothing) and extended situations, see [410]. A specific Cramér-Rao bound in multi-target tracking scenario was also given in [218].

P.2 Bias-Variance

Let's first consider the exact Monte Carlo sampling. The true and Monte Carlo state-error covariance matrices are defined by

$$\begin{aligned}
\Sigma &= \mathbb{E}_p[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T], \\
\Sigma_{\hat{\boldsymbol{\mu}}} &= \mathbb{E}_p[(\mathbf{x} - \hat{\boldsymbol{\mu}})(\mathbf{x} - \hat{\boldsymbol{\mu}})^T],
\end{aligned}$$

[73] This term was first used by Bellman in 1961, which refers to the exponential growth of hypervolume as a function of dimensionality.

[74] Unfortunately, most convergence results did not specify very clearly and might produce confusion for the reader. We must caution that any claim of an established theoretical result should not violate the underlying assumption, e.g. smoothness, regularity, exponential forgetting; any unsatisfied condition will invalidate the claim.

[75] More generalized discussion for hyperellipses is given in [94].

[76] For the information filter, the information matrix is equivalent to the $\mathbf{J}(\mathbf{x})$.

[77] In contrast to the conventional Cramér-Rao bound for deterministic parameters, it is not required that the estimated state $\hat{\mathbf{x}}$ be unbiased, as many authors have suggested [462], [410].

TABLE VIII

A List of Statistics Notations.

notation	definition	comment
$f(\mathbf{x})$	N/A	nonlinear function in $\mathbb{R}^{N_\mathbf{x}}$
$\hat{f}_{N_p}(\mathbf{x})$	(58)	exact MC estimate
$\hat{f}(\mathbf{x})$	(60)	weighted estimate of IS
$\mathbb{E}_p[f]$	$\int p(\mathbf{x})f(\mathbf{x})d\mathbf{x}$	true mean
$\Sigma_f \equiv \mathrm{Var}_p[f]$	$\mathbb{E}_p[(f - \mathbb{E}_p[f])^2]$	true variance
$\hat{\Sigma}_{\hat{f}}$	(151)	sample variance
$\mathbb{E}_q[f]$	$\int q(\mathbf{x})f(\mathbf{x})d\mathbf{x}$	mean w.r.t. proposal distribution q
$\mathbb{E}_p[\hat{f}_{N_p}]$	$\int p(\mathbf{x})\hat{f}_{N_p}(\mathbf{x})d\mathbf{x}$	mean of $\hat{f}_{N_p}$, equal to $\mathbb{E}_p[f]$
$\mathrm{Var}_p[\hat{f}_{N_p}]$	$\mathbb{E}_p[(\hat{f}_{N_p} - \mathbb{E}_p[\hat{f}_{N_p}])^2]$	variance of exact MC estimate
$\mathbb{E}_q[\hat{f}]$	$\int q(\mathbf{x})\hat{f}(\mathbf{x})d\mathbf{x}$	mean of $\hat{f}$ w.r.t. q, equal to $\mathbb{E}_q[f]$
$\mathrm{Var}_q[\hat{f}]$	$\mathbb{E}_q[(\hat{f} - \mathbb{E}_q[\hat{f}])^2]$	variance of weighted sampler w.r.t q
$\mathrm{Var}_{\mathrm{MC}}[\hat{f}_{N_p}]$	$\mathbb{E}_{\mathrm{MC}}[(f - \mathbb{E}_p[\hat{f}_{N_p}])^2]$	w.r.t. Monte Carlo runs
$\mathrm{Var}_{\mathrm{MC}}[\hat{f}]$	$\mathbb{E}_{\mathrm{MC}}[(\hat{f} - \mathbb{E}_q[\hat{f}])^2]$	w.r.t. Monte Carlo runs

where $\boldsymbol{\mu} = \mathbb{E}_p[\mathbf{x}]$, $\hat{\boldsymbol{\mu}} = \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}^{(i)}$ where $\{\mathbf{x}^{(i)}\}$ are i.i.d. samples drawn from true pdf $p(\mathbf{x})$. It can be proved that [49]

$$\begin{aligned}\Sigma_{\hat{\boldsymbol{\mu}}} &= (1+\frac{1}{N_p})\Sigma \\ &= \Sigma + \mathrm{Var}_p[\hat{\boldsymbol{\mu}}], \end{aligned} \tag{150}$$

where the second line follows the fact that $\mathbb{E}_p[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}})(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}})^T] = \frac{1}{N_p}\Sigma$ (see Appendix A). Hence, the uncertainty from the exact Monte Carlo sampling part is the order of N_p^{-1}, for example, $N_p = 20$ adds an extra 5% to the true variance. In practice, we usually calculate the sample variance in place of true variance, for Monte Carlo simulation, we have

$$\hat{\Sigma}_{\hat{\boldsymbol{\mu}}} = \frac{1}{N_p - 1}\sum_{i=1}^{N_p}(\hat{\boldsymbol{\mu}} - \mathbf{x}^{(i)})(\hat{\boldsymbol{\mu}} - \mathbf{x}^{(i)})^T. \tag{151}$$

It should be cautioned that $\hat{\Sigma}_{\hat{\boldsymbol{\mu}}}$ is an unbiased estimate of Σ instead of $\Sigma_{\hat{\boldsymbol{\mu}}}$, the unbiased estimate of $\Sigma_{\hat{\boldsymbol{\mu}}}$ is given by $(1+N_p^{-1})\hat{\Sigma}_{\hat{\boldsymbol{\mu}}}$.

Second, we particularly consider the importance sampling where the i.i.d. samples are drawn from the proposal distribution. Recalling some notations defined earlier (for the reader's convenience, they are summarized in Table VIII, a geometrical interpretation of Monte Carlo estimates is shown in Fig. 12), it must be cautioned again that although $\hat{f}_{N_p}$ is unbiased (i.e. $\mathbb{E}_p[f(\mathbf{x})] = \mathbb{E}_p[\hat{f}_{N_p}(\mathbf{x})]$), however, $\hat{f}$ is biased (i.e. $\mathbb{E}_p[f(\mathbf{x})] \neq \mathbb{E}_q[\hat{f}(\mathbf{x})]$). In practice, with moderate sample size, it was shown in [256] that the bias is *not* negligible.[78] The bias accounts for the following sources: limited simulated samples, limited computing power and limited memory (calculation of posterior $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ needs storing the data up to n), not to mention the sampling inaccuracy as well as the existence of noise.

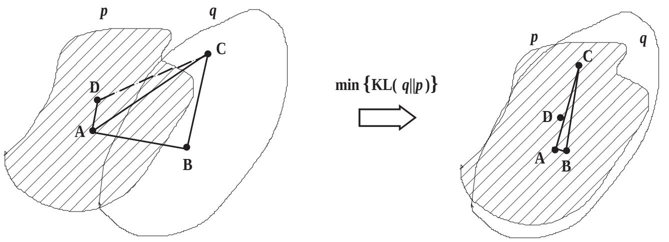

Fig. 12. A geometrical interpretation of Monte Carlo estimate statistics. The points A, B, C, D represent $\mathbb{E}_p[f]$, $\mathbb{E}_q[\hat{f}]$, $\hat{f}$, $\hat{f}_{N_p}$, respectively. $|AB| = |\mathbb{E}_p[f] - \mathbb{E}_q[\hat{f}]|$ represents the bias, $|AC| = |\mathbb{E}_p[f] - \hat{f}|$, p, q represent two probability densities in the convex set, p is target density, q is the proposal distribution. **Left:** when $q \neq p$, the estimate is biased, the variance $\mathbb{E}_q[\|AC\|^2]$ varies. **Right:** when q is close to p, or $\mathrm{KL}(q\|p)$ is small, bias vanishes (A approaches B) and C approaches D, the variance decrease with increasing N_p; when A overlaps B, $\|AC\|^2$ represents the total error.

In the Monte Carlo filtering context, suppose $\hat{\mathbf{x}}_n$ is an estimate given by the particle filter, by writing

$$\mathbf{x}_n - \hat{\mathbf{x}}_n = (\mathbf{x}_n - \mathbb{E}_q[\hat{\mathbf{x}}_n|\mathbf{y}_{0:n}]) + (\mathbb{E}_q[\hat{\mathbf{x}}_n|\mathbf{y}_{0:n}] - \hat{\mathbf{x}}_n),$$

we may calculate the expected gross error

$$\begin{aligned}\mathcal{E} &= \mathbb{E}_q\Big[\mathtt{tr}\Big((\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^T\Big)\Big|\mathbf{y}_{0:n}\Big] \\ &= \mathtt{tr}\Big(\mathbb{E}_q\Big[(\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^T\Big|\mathbf{y}_{0:n}\Big]\Big) \\ &= \mathtt{tr}\Big(\underbrace{\mathbb{E}_q\Big[(\hat{\mathbf{x}}_n - \mathbb{E}_q[\hat{\mathbf{x}}_n|\mathbf{y}_{0:n}])(\hat{\mathbf{x}}_n - \mathbb{E}_q[\hat{\mathbf{x}}_n|\mathbf{y}_{0:n}])^T\Big|\mathbf{y}_{0:n}\Big]}_{\text{Covariance}} \\ &\quad + \underbrace{(\mathbb{E}_q[\hat{\mathbf{x}}_n|\mathbf{y}_{0:n}] - \mathbf{x}_n)(\mathbb{E}_q[\hat{\mathbf{x}}_n|\mathbf{y}_{0:n}] - \mathbf{x}_n)^T}_{\text{Bias}^2}\Big) \end{aligned} \tag{152}$$

where

$$\mathbb{E}_q[\mathbf{x}_n|\mathbf{y}_{0:n}] = \int \mathbf{x}_n W(\mathbf{x}_n) q(\mathbf{x}_n|\mathbf{y}_{0:n}) d\mathbf{x}_n,$$

and $W(\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{y}_{0:n})/q(\mathbf{x}_n|\mathbf{y}_{0:n})$. If $p = q$, the bias vanishes to zero at a rate $\mathcal{O}(N_p)$, then $\mathcal{E}$ only accounts for variance, and the state-error covariance is the true covariance. If $p \neq q$, $\mathcal{E}$ generally consists of both bias and variance where the bias is a nonzero constant. Hence, equation (152) represents the bias-(co)variance dilemma.[79] When the loss $\mathcal{E}$ is fixed, the bias and variance is a trade-off.[80] As suggested in [322], generally, we can define the bias and variance of importance sampling or MCMC estimate as:

$$\begin{aligned}\text{Bias} &= \mathbb{E}_q[\hat{f}(\mathbf{x})] - \mathbb{E}_p[f(\mathbf{x})], \\ \text{Var} &= \mathbb{E}_q\Big[\Big(\hat{f}(\mathbf{x}) - \mathbb{E}_q[\hat{f}(\mathbf{x})]\Big)^2\Big], \end{aligned}$$

[78] An improved Bayesian bootstrap method was proposed for reducing the bias of the variance estimator, which is asymptotically equivalent to the Bayesian bootstrap method but has better finite sample properties [256].

[79] It is also called the trade-off between approximation error and estimation error.

[80] In a very loose sense, Kalman filter can be imagined as a special particle filter with only one "perfect" particle propagation, in which the unique sample characterizes the sufficient information of the prototype data from the distribution. The variance estimate of Kalman filter or EKF is small, whereas its bias (innovation error) is relatively larger than that of particle filter.

where $\hat{f}(\mathbf{x})$ is given by the weighted importance sampling. The quality of approximation is measured by a loss function $\mathcal{E}$, as decomposed by

$$\begin{aligned}\mathcal{E} &= \mathbb{E}_q\Big[\Big(\hat{f}(\mathbf{x}) - \mathbb{E}_p[f(\mathbf{x})]\Big)^2\Big] \\ &= \text{Bias}^2 + \text{Var}.\end{aligned}$$

Example 1: Consider two bounded functions

$$\begin{aligned}f_1(x) &= \begin{cases} Cx, & \text{if } 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}, \\ f_2(x) &= \begin{cases} Cx^3, & \text{if } 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases},\end{aligned}$$

where the constant $C = 1$. The true pdf $p(x)$ is a Cauchy density and the proposal distribution $q(x)$ is a Gaussian pdf (see the illustration in Fig. 14), as follows

$$\begin{aligned}p(x) &= \frac{1}{\pi\sigma(1 + x^2/\sigma^2)}, \\ q(x) &= \frac{1}{\sqrt{2\pi}\sigma}\exp(-x^2/2\sigma^2),\end{aligned}$$

both with variance $\sigma^2 = 1$. Hence the means of $f_1(x)$ and $f_2(x)$ w.r.t. two distributions are calculated as

$$\begin{aligned}\mathbb{E}_p[f_1(x)] &= \int_0^1 \frac{x}{\pi(1+x^2)}dx = \frac{\ln 2}{2\pi}, \\ \mathbb{E}_p[f_2(x)] &= \int_0^1 \frac{x^3}{\pi(1+x^2)}dx = \frac{(1-\ln 2)}{2\pi}, \\ \mathbb{E}_q[f_1(x)] &= \int_0^1 \frac{1}{\sqrt{2\pi}}x\exp(-x^2/2)dx = \frac{1}{\sqrt{8\pi}} - \frac{1}{\sqrt{8\pi e}}, \\ \mathbb{E}_q[f_2(x)] &= \int_0^1 \frac{1}{\sqrt{2\pi}}x^3\exp(-x^2/2)dx = \sqrt{\frac{2}{\pi}} - \sqrt{\frac{9}{2\pi e}}.\end{aligned}$$

We draw Monte Carlo samples from two distributions (see Appendix C for implementation) with N_p varying from 100 to 10,000. The analytic calculation results are compared with the ensemble average over 100 independent runs of Monte Carlo simulation with different initial random seeds. The experimental results are partially summarized in Table IX and shown in Fig. 13.

Remarks (on experimental results):

- As observed in Fig. 13, as N_p increases, the estimates of both $\hat{f}_{N_p}$ and $\hat{f}$ become more accurate; and the variances decrease at a rate $\mathcal{O}(N_p^{-1})$.
- As seen from Table IX, $\hat{f}$ is equal to $\hat{f}_{N_p}$ (mean value based on 100 Monte Carlo runs), but their variances are different (see right plot of Fig. 13).
- Noting in experiments we use $C = 1$, but it can be expected that when $C > 1$ ($C < 1$), the variance increases (decreases) by a ratio C^2.

TABLE IX

MONTE CARLO EXPERIMENTAL RESULTS OF EXAMPLE 1. (THE RESULTS ARE AVERAGED ON 100 INDEPENDENT RUNS USING 10,000 SAMPLES WITH DIFFERENT RANDOM SEEDS. THE BOLD FONT INDICATES THE STATISTICS ARE EXPERIMENTALLY MEASURED, WHEREAS THE OTHERS ARE ANALYTICALLY CALCULATED.)

statistics	$f_1(x)$	$f_2(x)$
$\mathbb{E}_p[f]$	0.1103	0.0488
$\mathbb{E}_p[\hat{f}_{N_p}]$	0.1103	0.0488
$\mathbb{E}_q[f]$	0.1570	0.0720
$\hat{f}_{N_p}(x)$	**0.1103**	**0.0489**
$\hat{f}(x)$	**0.1103**	**0.0489**
$\Sigma_f \equiv \text{Var}_p[f]$	0.0561	0.0235
$\hat{\Sigma}_{\hat{f}_{N_p}}$	**0.0562**	**0.0236**
$\hat{\Sigma}_{\hat{f}}$	**0.0646**	**0.0329**
$\text{Var}_p[\hat{f}_{N_p}]$	0.0561×10^{-4}	0.0235×10^{-4}
$\text{Var}_q[f]$	0.0748	0.0336
$\text{Var}_{\text{MC}}[\hat{f}_{N_p}]$	$\mathbf{(0.0012)^2}$	$\mathbf{(0.0009)^2}$
$\text{Var}_{\text{MC}}[\hat{f}]$	$\mathbf{(0.0014)^2}$	$\mathbf{(0.0012)^2}$
$\hat{N}'_{eff}$	**3755**	**6124**
N_{eff}/N_p	**2208/10000 (22.8%)**	
$\hat{N}_{eff}/N_p$	**6742/10000 (67.4%)**	
N_{KL}	**4.0431**	
$\text{Var}[N_{\text{KL}}]$	**0.0161**	

To the end of the discussion of bias-variance, we summarize the popular variance reduction techniques as follows:

- Data augmentation [445], [446].
- Rao-Blackwellization [74], [304], [315], [144], [145], [119], [458], [338], [23].
- Stratified sampling [376], [69].
- Importance sampling [199], slicing sampling [351].
- Survey sampling [199], [162].
- Partition sampling [313].
- Antithetic variate [200], [201], [442] and control variate [5], [201] (see Appendix D).
- Group averaging [267].
- Moment matching method [52].
- jitter and prior boosting [193].
- Kernel smoothing [222], [345].
- QMC and lattice method [413], [299], [368], [361], [295], [296].

P.3 Robustness

Robustness (both algorithmic robustness and numerical robustness) issue is important for the discrete-time filtering. In many practical scenarios, the filter might encounter the possibility of divergence where the algorithmic assumption is violated or the numerical problem is encountered (e.g., ill-conditioned matrix factorization). In retrospect, many authors have explored this issue from different perspectives, e.g. robust KF [80], robust EKF [80], [158], minimax filter [273], or hybrid Kalman/minimax filter. Many useful rules of thumb for improving robustness were discussed in [80]. Here we focus our attention on the particle filters.

There are two fundamental problems concerning the robustness in particle filters. First, when there is an outlier,

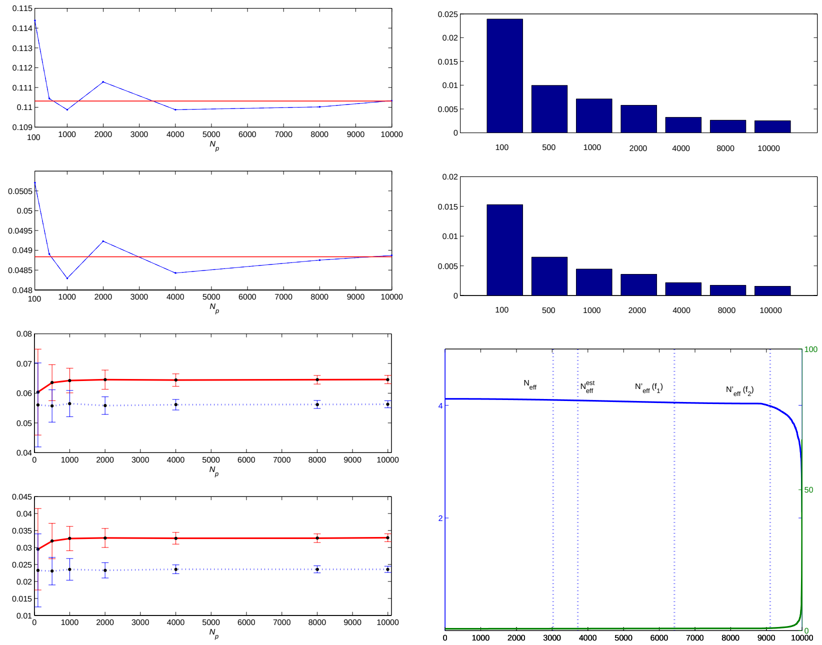

Fig. 13. Monte Carlo experimental results of Example 1. The first row shows the results of $f_1(x)$ and the second row for $f_2(x)$. **Top Left:** Monte Carlo Mean of $\hat{f}$ compared to the true mean $\mathbb{E}_p[f]$ (solid line). **Top Right:** Monte Carlo varianceof $\hat{f}$ within 100 independent runs. **Bottom Left:** Error bar of the sample variance of $\hat{\Sigma}_{\hat{f}}$ (solid line) compared to the sample variance $\hat{\Sigma}_{\hat{f}_{N_p}}$ (dotted line). The dots are given by the means of 100 trial results of sample variance, the bars denote their standard deviations. **Bottom Right:** Ordered $-\log_{10}\tilde{W}(\mathbf{x}^{(i)})$ (left ordinate) and $W(\mathbf{x}^{(i)})$ (right ordinate; both in ascending order of abscissa) and effective sample size estimates (in one trial).

the importance weights will be very unevenly distributed and it usually requires a large number of N_p to assure the accuracy of empirical density approximation. Hence the measurement density $p(\mathbf{y}_n|\mathbf{x}_n)$ is supposed to insensitive to the $\mathbf{x}_n$. Second, the empirical distribution from the samples often approximates poorly for the long-tailed distribution, either for proposal distribution or for posterior. This is imaginable because the probability sampling from the tail part of distribution is very low, and resampling somehow makes this problem more severe. Many results have shown that even the mixture distribution can not well describe the tail behavior of the target distribution. Hence, outliers will possibly cause the divergence of filter or produce a very bad performance.

Recently, it has been shown in [162], [70] that the sample size estimate given by (89) is not robust, the approximated expression might be infinitely wrong for certain $f(\mathbf{x})$, $p(\mathbf{x})$ and $q(\mathbf{x})$. It can be derived that

$$\begin{aligned}\mathrm{Var}_q[\hat{f}] &= \frac{1}{N_p}\mathrm{Var}_q[f(\mathbf{x})W(\mathbf{x})] \\ &= \frac{1}{N_p}\mathbb{E}_q\Big[\Big(f(\mathbf{x})-\mathbb{E}_p[f(\mathbf{x})]\Big)^2 W^2(\mathbf{x})\Big]+\mathcal{O}(N_p^{-2}),\end{aligned}$$

where $W(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$. For a large N_p, the true effective sample size is given as [162], [70]

$$\begin{aligned}N'_{eff} &= \frac{\mathrm{Var}_p[f]}{\mathrm{Var}_q[\hat{f}]} \\ &\approx \frac{N_p\mathbb{E}_p[(f(\mathbf{x})-\mathbb{E}_p[f(\mathbf{x})])^2]}{\mathbb{E}_q\Big[(f(\mathbf{x})-\mathbb{E}_p[f(\mathbf{x})])^2W^2(\mathbf{x})\Big]}. \qquad (153)\end{aligned}$$

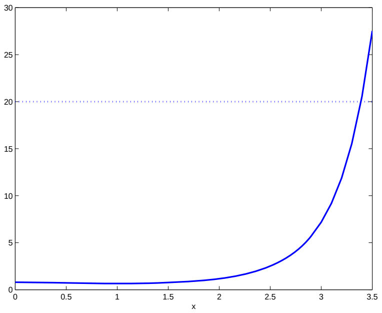

Fig. 14. The ratio curve of important ratio function $W(x)$ of Example 1. Solid line: true $W(x) = \sqrt{2/\pi}\frac{\exp(x^2/2)}{1+x^2}$; dotted line: bounded curve specified by C.

Fearnhead gave a simple example and illustrated that, the estimate expression (89) of N_{eff} (derived by using first two moments of $W(\mathbf{x})$ and $f(\mathbf{x})$) can be very poor (for two simple cases, one leads to $N'_{eff}/N_{eff} \to 0$ and the other $N'_{eff}/N_{eff} \to \infty$). In [70], a more robust effective sample size estimate was proposed

$$\hat{N}'_{eff} = \frac{N_p\sum_{i=1}^{N_p}(f(\mathbf{x}^{(i)})-\hat{f}(\mathbf{x}))^2 W(\mathbf{x}^{(i)})}{\sum_{i=1}^{N_p}(f(\mathbf{x}^{(i)})-\hat{f}(\mathbf{x}))^2 W^2(\mathbf{x}^{(i)})}. \qquad (154)$$

Another critical issue is the estimate of the important weights within the IS, SIS, SIR framework. Note that $W(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ is a *function*[81] instead of a point esti-

[81] More precisely, $W(\mathbf{x})$ is a ratio function between two pdf's. Es-

mate. Being a function usually implies certain prior knowledge, e.g. smoothness, non-negativeness, finite support. However, when we use a finite number of random (uneven) samples to represent this function, the inaccuracy (both bias and variance) is significant. This problem becomes more severe if the outliers come in. Experimentally, we found that in a simple problem (Example 1), the distribution of important weights are very peaked, even with a very large N_p (e.g. 10,000 to 100,000). Most importantly, as we can see in Fig. 14, the ratio curve (for Example 1) $W(x) = \sqrt{2/\pi}\frac{\exp(x^2/2)}{1+x^2}$ is unbounded.[82] When x is bigger than 3 (namely $3\sigma^2$ where $\sigma^2 = 1$; for Gaussian it accounts for 99% support of the distribution), the ratio becomes very large.[83] Imaginably, this phenomenon is the intrinsic reason of weight unevenness when outliers come in, no matter in sequential or non-sequential framework. To alleviate this problem, a natural solution is simply to bound the important ratio function:

$$W(\xi) = \left\{ \begin{array}{ll} p(\xi)/q(\xi) & 0 \le \xi < C \\ p(C)/q(C) & \xi \ge C \end{array} \right. ,$$

or

$$W(\xi) = \varphi(p(\xi)/q(\xi)),$$

where $\varphi(\cdot)$ is a bounded function, e.g. piecewise linear function or scaled sigmoid function. The constant C here plays a similar role of C in the rejection sampling discussed in Section V-G.2, both of which determine the acceptance rate of the samples. The choices of the bound C or scaling parameters also requires strong prior knowledge of the problem (e.g. the support of target density). The use of bounded important weights essentially implies that we only use the reliable samples, ignoring the samples with very big weights. The reason is intuitively justified by the following: Since $W(\mathbf{x})$ is an ratio function between two pdf's, in practice, the support of these pdf's are often limited or compact, which means the distributions are sparse (esp. when $N_\mathbf{x}$ is high). In order to handle the outliers and improve the robustness, we only use the samples from the reliable support based on prior knowledge and discard the others as outliers, though we also encounter the risk of neglecting the tail behavior of target density. This is tantamount to specifying a upper bound for the important ratio function $W(\mathbf{x})$.

Another improved strategy is use kernel smoothing technique (Section VI-G) to smooth the importance ratio function, namely $K(W(\xi))$, where $K(\cdot)$ can be a Gaussian kernel. The disadvantage of this strategy is the increase of computational cost, which brings inefficiency in on-line processing.

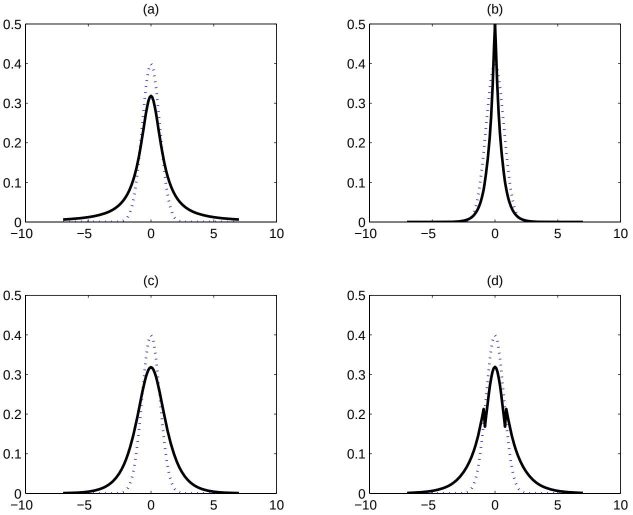

Fig. 15. An illustration of some heavy tailed densities and robust density model. (a) Cauchy density $p(\xi) = \frac{1}{\pi\sigma(1+\xi^2/\sigma^2)}$; (b) Laplace density $p(\xi) = \frac{1}{2\sigma}\exp(-|\xi|/\sigma)$; (c) Hyperbolic cosine $p(\xi) = \frac{1}{\pi\cosh(\xi)}$; (d) Huber's robust density with $\epsilon = 0.2$ and $c = 0.8616$. The dashed line is zero-mean Gaussian density for comparison, all of densities have unity variances $\sigma^2 = 1$.

Robust issues can be addressed in the robust statistics framework [214], [255]. Here we are particularly interested in the robust proposal or likelihood model. As discussed earlier, proposal distribution used in importance sampler is preferred to be a heavy-tailed density. In the Bayesian perspective, we know that the proposal distribution $q(\mathbf{x}|\mathbf{y})$ is assumed to approximate the posterior $p(\mathbf{x}|\mathbf{y})$ and $q(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$. If the likelihood $p(\mathbf{y}|\mathbf{x})$ is upper-bounded, say $p(\mathbf{y}|\mathbf{x}) \le C$, then the prior can be a good candidate for proposal distribution since $q(\mathbf{x}|\mathbf{y}) \propto Cp(\mathbf{x})$ and it is also easy to implement. This fact motivates us to come up with a robust loss function or robust likelihood density $p(\mathbf{y}|\mathbf{x})$,[84] which assumes an ϵ-contaminated mixture density. In spirit of robustness, the following likelihood model is used

$$p(\xi) = \left\{ \begin{array}{ll} \frac{1-\epsilon}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{\xi^2}{2\sigma^2}\right\} & |\xi| < c\sigma \\ \frac{1-\epsilon}{\sqrt{2\pi}\sigma}\exp\left\{\frac{c^2}{2\sigma^2} - \frac{c|\xi|}{\sigma}\right\} & |\xi| > c\sigma \end{array} \right. \tag{155}$$

where $0 < \epsilon < 1$, and c is determined from the normalization condition [463]

$$1 = \frac{1-\epsilon}{\sqrt{2\pi}\sigma}\Big(\int_{-c\sigma}^{c\sigma}\exp(-\xi^2/2)d\xi + \frac{2}{c}\exp(-c^2/2)\Big).$$

The idea here is to bound the error and discard the influence of outliers;[85] it was also suggested by West [480], in which he developed a robust sequential approximate Bayesian estimation for some special non-Gaussian distribution families. In Fig. 15, some heavy-tailed densities

timating the ratio of two pdf's given limited observations is stochastically ill-posed [463] (chap. 7). This amounts to solve the integral equation $\int_{-\infty}^{\mathbf{x}} W(\mathbf{x})dQ(\mathbf{x}) = P(\mathbf{x})$. Given N_p simulated samples $\{\mathbf{x}^{(i)}\}$, it turns out to solve an approximated operator equation: $\boldsymbol{A}_{N_p}W = \int_0^{\mathbf{x}} W(\mathbf{x})dQ_{N_p}(\mathbf{x})$.

[82] That is the reason we are recommended to choose a proposal with heavy tail.

[83] This can be arbitrary bad if $W(\mathbf{x})$ is not upper bounded.

[84] The relationship between loss function and likelihood is established by $\mathcal{E} = -\log p(\mathbf{y}|\mathbf{x})$.

[85] The idea of "local search" in prediction [456] is close in spirit to this.

and Huber's robust density are illustrated. Those density models are more insensitive to the outliers because of their bounded activation function. In addition, there is a large amount of literature on robust Bayesian analysis (e.g. [226]) in terms of robust priors, robust likelihoods, and robust (minimax) risks, however, extended discussion is beyond the scope of current paper.

P.4 Adaptive Procedure

Another way to enhance robustness is the adaptive particle methods [262], [447], which allow to adjust the number of particles through the filtering process. The common criterion is based on the likelihoods (which are equal to importance weights if the proposal is transition prior) [262]. The intuition behind that is if the samples are well suited to the real posterior, each individual importance weight is large, and the variance of the importance weights is large, which means the mismatch between proposal distribution and true posterior is large, and we keep N_p small. Another method proposed in [171] is based on the stochastic bounds on the sample-based approximation quality. The idea is to bound the error induced by the samples and sequentially approximate the upper bound with additional computational overhead.

To monitor the efficiency of sampling in each step, we propose another adaptive procedure as follows. Besides effective sample number N_{eff} or N'_{eff}, another useful efficiency measure will be $W(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ itself. Since proposal $q(\cdot)$ is supposed to be close to posterior $p(\cdot)$, the closeness of two probability distribution (density) is naturally the Kullback-Leibler (KL) divergence $\mathrm{KL}(q\|p)$,[86] which is approximated by

$$\begin{aligned}\mathrm{KL}(q\|p) &= \mathbb{E}_q\Big[\log\frac{q(\mathbf{x})}{p(\mathbf{x})}\Big] \approx \frac{1}{N_p}\sum_{i=1}^{N_p}\log\frac{q(\mathbf{x}^{(i)})}{p(\mathbf{x}^{(i)})}\\ &= -\frac{1}{N_p}\sum_{i=1}^{N_p}\log(W(\mathbf{x}^{(i)})) \qquad (156)\end{aligned}$$

when $q(\cdot) = p(\cdot)$ and $W(\mathbf{x}^{(i)}) = 1$ for all i, $\mathrm{KL}(q\|p) = 0$. From (156), we can also see that if the proposal is chosen as transition prior, $\mathrm{KL}(q\|p)$ will only depend on the likelihood $\sum_{i=1}^{N_p}\log p(\mathbf{y}|\mathbf{x}^{(i)})$, thus the KL divergence reduces to a log-likelihood measure; in a sequential framework, (88) can be rewritten as

$$-\sum_{i=1}^{N_p}\log W(\mathbf{x}_n^{(i)}) = -\sum_{i=1}^{N_p}\log W(\mathbf{x}_{n-1}^{(i)}) - \sum_{i=1}^{N_p}\log p(\mathbf{y}_n|\mathbf{x}_n^{(i)}).$$

Generally, $\mathrm{KL}(q\|p) \neq 0$, thus (156) can be used as a measure to monitor the efficiency of proposal. Intuitively, if $\mathrm{KL}(q\|p)$ is small or decreases, we can remain or decrease the particle number N_p; if $\mathrm{KL}(q\|p)$ is big or increases, we can increase the N_p. In order to let $-\log(W(\mathbf{x}^{(i)}))$ be non-negative (since $\mathrm{KL}(q\|p) \geq 0$), we calculate the normalized weights and obtain

$$\mathrm{KL}(q\|p) \approx -\frac{1}{N_p}\sum_{i=1}^{N_p}\log(\tilde{W}(\mathbf{x}^{(i)})) \equiv N_{\mathrm{KL}}, \qquad (157)$$

which achieves the minimum value $N_{\mathrm{KL}}^{\min} = \log(N_p)$ when all $\tilde{W}(\mathbf{x}^{(i)}) = 1/N_p$. Equation (157) can be also used as a measure of effective samples (for reampling), which leads the following adaptive procedure:

```
• If $N_{\mathrm{KL}}(n) > \kappa\log(N_p)$
•       resample and increase $N_p$ (i.e. prior boosting) via
•       $N_p(n+1) = \kappa N_p$
• Else
•       $N_p(n+1) = N_p$, and resample if $\hat{N}_{eff} < N_T$
• End
```

where $\kappa > 1$ is a threshold defined a priori. We can also calculate the variance approximately by

$$\mathrm{Var}[-\log(\tilde{W})] \approx \frac{1}{N_p}\sum_{i=1}^{N_p}(\log(\tilde{W}(\mathbf{x}^{(i)})))^2 - (N_{\mathrm{KL}})^2.$$

Although above adaptive procedure is sort of hindsight in a sense that it can only boost the samples in next step based on current N_{KL}, while $N_{\mathrm{KL}}(n+1)$ may not be less than $\kappa\log(N_p)$. Our empirical results show that it is still a useful measure for monitoring the sample efficiency. This procedure is particularly useful for APF when the importance weights are evaluated after the first stage.

P.5 Evaluation and Implementation

We should keep in mind that designing particular particle filter is problem dependent. In other words, there is no general rule or universal good particle filter. For instance, in certain case like robot global localization [332], we prefer to keep the spread of particles wide (to prevent missing hypothesis), but in another case like target tracking [357], we instead prefer to keep the support of particles bounded (to improve the accuracy). To give another example, in many cases we want the particle filter robust to the outliers, thereby an insensitive likelihood model is preferred, however in some case where the cost is unaffordable even the likelihood is low, a risk-sensitive model is needed [448]. On the other hand, one particle filter `Algorithm A` works well (better than another particle filter `Algorithm B`) doesn't necessarily mean that it has the gain over `Algorithm B` on the other problems - this is the spirit of no-free-lunch (NFL) theorem! (see Appendix F) Hence it is not fair to conclude that `Algorithm A` is superior to `Algorithm B` for only one particular problem being tested. Justification of the superiority of certain algorithm over the others even on a specific problem is also unfair without Monte Carlo simulations.

One of the merits about particle filter is the implementation complexity is $\mathcal{O}(N_p)$, independent of the state dimension $N_{\mathbf{x}}$. As to the evaluation criteria of Monte Carlo or particle filters, a straightforward indicator of performance of different algorithms can be seen from the MSE between

[86] KL divergence can be viewed as the expected log-likelihood, where the likelihood is defined by $q(\cdot)/p(\cdot)$.

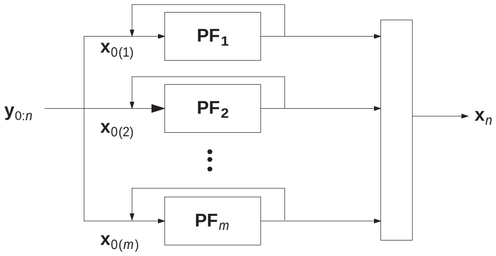

Fig. 16. A Parallel particle filters structure.

the estimate and true value. Due to the Monte Carlo nature, variance is an important criterion, e.g. (co)variance of estimate and variance of importance weights, both of which are calculated based on Monte Carlo averaging results (say 100 $\sim$ 1000 independent runs). This requirement is deemed necessary when comparing different particle filters' performance, otherwise it is unfair to say one is better than the others or the opposite. Other evaluation issues include sampling and resampling efficiency, trade-off between performance and computational complexity, parallel architecture, ease of implementation, etc.

The implementation issue of particle filters also deserves special attention, though it is not formally discussed before in the literature. As discussed earlier, for certain particle filter, e.g. SIS filter, does allow the parallel implementation since the simulated particles are independent, but the resampling step usually makes the parallelization unfriendly because it requests all of the information of importance weights. Nevertheless, we do can consider parallel implementation in another perspective. Let's consider a parallel particle filter structure (see Fig. 16) that comprises of a bunch of (say m) particle filters, each particle filter is run independently with different initial conditions (e.g., different seeds for the same random generator, different dynamic noises), different simulated samples for the same proposal distribution, different proposal distributions, or different resampling schemes. The estimated result is based on the average of the estimates from m particle filters, namely

$$\hat{\mathbf{x}}_n = \sum_{k=1}^{m} c_k \hat{\mathbf{x}}_{n(k)}$$

where $\sum_{k=1}^{m} c_k = 1$, c_k can be a same constant $1/m$ or be different, which allows on-line estimation (for instance, c_k can be associated to the filtered error of the k-th particle filter). The complexity is proportional to the number of particle filters, but different particle filters can be implemented in different processors or computers. The structure of parallel particle filters is somewhat similar to the interacting multiple models (to be discussed in Section VII).

Finally, we would like to point out couple research resources about Kalman filter, particle filters, and Monte Carlo methods available in the Internet, an increasingly growing database and resource open for researchers. We deem it very important for multidisciplinary research intersection, quick access of research results, open discussion, as well as result comparison and justification.

- Kalman filters and particle filters: We particularly refer the reader to a Kalman/particle filter Matlab[87] toolbox "ReBEL" (Recursive Bayesian Estimation Library), developed by Rudolph van der Merwe, which is available on line for academic purpose http://varsha.ece.ogi.edu/rebel/index.html. The toolbox cover many state-of-the-art Kalman/particle filtering methods, including joint/dual estimation, UKF, UPF and their extensions. Demos and data sets are also available.
- Monte Carlo methods: A website dedicated to the sequential Monte Carlo approaches (including softwares), maintained by Nando de Freitas, is available on line http://www.cs.ubc.ca/~nando/smc/index.html. A shareware package called BUGS (Bayesian inference Using Gibbs Sampling) is available on line http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml. A website dedicated to MCMC methods is available on line http://www.statslab.cam.ac.uk/~mcmc.

VII. Other Forms of Bayesian Filtering and Inference

A. Conjugate Analysis Approach

One of important Bayesian filtering techniques is the conjugate method, which admits the nonlinear filtering/inference in a close finite-dimensional form. In particular, when prior information about the model is limited, the prior distribution is often chosen from a parametric family $\mathcal{P}$. The families $\mathcal{P}$ that are closely under sampling (that is for every prior $p \in \mathcal{P}$, the posterior distribution also belongs to $\mathcal{P}$) are of particular interest. These families are called *conjugate* families and the associated priors are called conjugate priors, which can only belong to the exponential family according to the Pitman-Koopman Lemma. The main motivation for using conjugate priors is their analytical tractability and ease of interpretation.

In [469], Vidoni introduced a finite-dimensional nonlinear and non-Gaussian filtering method for exponential family of state space models. Specifically, he defined a *conjugate latent process*, in which the likelihood belongs to an exponentially family, initial state density is conjugate to the likelihood, and the transition prior also remains conjugate in the prediction step. The update and inference in each step follows a Bayes rule. Examples of exponential families include Gaussian, Gamma, Poisson, binomial, inverse Gaussian, Laplace, etc.

B. Differential Geometrical Approach

Statistical inference has an intrinsic link with differential geometry [9], [10]. A family of probability distributions corresponds to a geometric structure as a certain manifold with a Riemannian metric. By transforming the statistical models to the geometric manifold, information geometry

[87] Matlab © is the trade mark of MathWorks, Inc.

provides many new insights to Bayesian filtering and inference.

In a series of papers [276]-[281], Kulhavý explored the idea of recursive Bayesian parameter estimation using differential geometry method. The basic idea is to approximate the true point by orthogonal projection onto a tangent surface. He suggested to use an invariant metric called conditional inaccuracy as error criterion, and formulated the inverse problem to an approximation problem; the true density is assumed to come from a parameterized known family, and the filtered density is approximated by the empirical density given the observations. This methodology was also further extended to state estimation problem [279], [225]. In particular, Iltis [225] used the disjoint basis function (similar to the Haar basis) to represent the posterior density, the filtering density is an affine transformation of the state vector; and the filtering problem is reduced to fit the model density in each step to the true posterior. Instead of using L_2 norm, the KL divergence (cross-entropy) criterion is used to measure the approximation accuracy with the reduced statistics.[88] The algorithm works under several assumptions [225]: (i) the transition density is approximated by a piecewise constant function; (ii) the arithmetic mean of posterior is close to the geometric mean; and (iii) the bias in the affine approximation is constant.

Brigo [55]-[57], and Brigo *et al.* [53], [54] also applied the differential geometry approach to the finite-dimensional filtering. By using the notion of projection filter [202], they projected the infinite-dimensional Kushner-Stratonovich equation onto a tangent space of a finite-dimensional manifold of square root of probability density (from exponential family) according to the Fisher information metric, where the optimal filter is further sought in the tangent space. More details can be found in the thesis of Brigo [55].

C. Interacting Multiple Models

One of important Bayesian filtering methods in literature is the multiple models, e.g., generalized pseudo-Bayesian (GPB) [1], interacting multiple models (IMM) [27], which are widely used in the data association and target tracking [501], [28]. The intuition of using multiple models is to tackle the multiple hypotheses problem. For instance, in target tracking, the dynamic system can switch under different modes (so-called switching dynamics). A single linear/nonlinear filter thus is not sufficient to characterize the underlying dynamics, once the filter loses the target, the risk might be unaffordable. In order to tackle this situation, multiple filters are run in parallel to track the target, each one responsible to match a different target motion. The final estimate is calculated based on the weighted results from the multiple filters, with the weighting probability determined by the posterior probability of each hypothesis. Usually it is assumed the target switch from one mode to another with a known transition probability (via prior knowledge or estimatation from data), all of decisions are *soft* and fit a perfect niche for Bayesian filtering.

In the conventional IMM, the assumption is limited by the linearity and Gaussianity which allows to use Kalman filter or EKF for each potential hypothesis. However, this is not realistic in the real world. For the nonlinear non-Gaussian multiple-model problem, the estimate from EKF's are not accurate. Naturally, particle filtering can be used straightforward in IMM for target tracking [326]. Applications of particle filters in multiple models were also found in computer vision and visual tracking [43], [356].

D. Bayesian Kernel Approaches

Recently, kernel methods have attracted much attention in machine learning [405]. We will briefly discuss some popular Bayesian kernel methods, the reader is strongly referred to [405] for more details. The discussions here are applicable to parameter as well as state estimation.

From Bayesian point of view, instead of defining a prior on the parameter space, kernel methods directly define a prior on the functional space, choosing a kernel K is equivalent to assuming a Gaussian prior on the functional, with a normalized covariance kernel being K. On the other hand, instead of working on raw data space, kernel learning works in the high-dimensional feature space by a "kernel trick".

- Gaussian Process, as a well-studied stochastic process, is one of the popular kernel machines for regression [489]. The covariance of the random variables $\{f(\mathbf{x}_1), \cdots, f(\mathbf{x}_\ell)\}$ are defined by a symmetric positive definite kernel $K \approx \mathrm{Cov}\{f(\mathbf{x}_1), \cdots, f(\mathbf{x}_\ell)\}$ with $K_{ij} = \mathrm{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)], (i, j = 1, \cdots, \ell)$. An on-line algorithm for Gaussian processes for sequential regression has been developed [508], [109].
- Laplacian Process, which uses the Laplacian prior as regularization functional, admits a sparse approximation for regression. The kernel is a Laplacian radial basis function.
- Relevance vector machine (RVM) [454], is a kind of kernel method to obtain sparse solutions while maintaining the Bayesian interpretability. The basic idea is the use the hyperparameters to determine the priors on the individual expansion coefficients. RVM also allows on-line estimation.

E. Dynamic Bayesian Networks

In the Bayesian perspective, many dynamic state-space models can be formalized into the so-called belief networks or dynamic Bayesian networks (DBN) (e.g., [183], [184]), which covers the following HMM and switching state-space model as special cases.[89] Bayesian statistics has provided a principled approach for probabilistic inference, with incorporation of prior, causal, or domain knowledge. Recently, particle filtering has been applied in DBN [262], [263], [145], [344], a detailed treatment was also given in [162].

[88] Opposed to the sufficient statistics for original posterior estimation problem, reduced statistics is used for seeking an equivalent class of posterior, thereby making the inference more flexible.

[89] A Matlab toolbox of DBN is available on line http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html.

HMM Filters. Hidden Markov models (HMM), or HMM filters [380], [379], [90] can be viewed as a finite discrete-valued state space model.[91] Given continuous-valued observations $\mathbf{y}_{0:n}$, the HMM filters are anticipated to estimate the discrete state $\boldsymbol{z}_n$ ($\boldsymbol{z} \in \mathbb{N}^{N_z} = \{1, 2, \cdots, N_z\}$) given the model parameters (transition probability matrix $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})$, emission probability matrix $p(\mathbf{y}_n|\boldsymbol{z}_n)$, and initial state distribution $p(\boldsymbol{z}_0)$).[92] In contrast to the Kalman filtering, there are two popular algorithms used to train HMM filters[93]

- Viterbi algorithm [470], [170]: It is used to calculate the MAP estimate of the path through the *trellis*, that is, the sequence of discrete states that maximize the probability of the state sequence given the observations.
- Baum-Welch algorithm [379], [381]: It is used to to calculate the probability of each discrete state at each epoch given the entire data sequence.

Recently, many algorithms have been developed for non-stationary HMM in Monte Carlo framework [**?**], [390], [136]. Specific particle filtering algorithms were also developed for HMM [142], [162].

Switching State-Space Models. Switching state-space model share the same form as the general state-space model (1a)(1b) but with a jump Markov dynamics (either in state model or measurement model), which can be linear/nonlinear and Gaussian/non-Gaussian. It might also have mixed states consisting of both continuous and discrete components. Many exact or approximate inference methods were proposed:

- Exact inference: e.g. switching Kalman filter and switching AR model [343] via EM algorithm.
- Monte Carlo simulation: e.g., random sampling approach [6], state estimation of jump Markov linear systems (JMLS) using [146], [147], multi-class mixed-state dynamics [43], [356] via EM combined with particle filtering.
- Variational approximation [236], [241], [237] and mean-field approximation [241], [401]: variational Kalman filter [30], variational switching state space models [213], variational DBN [183], [184], variational Bayesian inference [22], variational Rao-Blackwellized particle filter [23], variational MCMC [121].

With no doubt, there is still much research space for further exploration along these lines.

VIII. Selected Applications

Bayesian filtering and Bayesian inference have found numerous applications in different areas. Due to space constraint, here we can only shortly describe several representative and well-studied problems in Bayesian learning community. However, the idea rooted in these applications can be extended to many scientific and engineering problems.

A. Target Tracking

Target tracking is one of the most important applications of sequential state estimation, which naturally admits Kalman filters and particle filters as the main tools. Many papers have been published with particle filtering applications in this field [193], [192], [24], [35], [48]. Bearings-only tracking and multiple-target tracking [313], [216], [217], [302], [362] are both well addressed. Some performance bounds for multiple-target tracking were also given [218]. In addition, particle filters were extensively used for visual-based human motion tracking or audio-based speaker localization/tracking. In [88], we give some quantitative comparisons of different particle filters on several tracking problems.

B. Computer Vision and Robotics

The pioneering work applying particle filtering in computer vision is due to Isard and Blake [229], [230], [228], where they called CONDENSATION for their algorithm. Since then, many papers have been published along this line [231], [232], [313], [44], [43], [131], [457], [458], [94]. The motion and sensor models correspond to the state and measurement equations, respectively.

Another important application area of particle filter in artificial intelligence is robot navigation and localization [447], [448], [171], [332], [288], which refers to the ability of a robot to predict and maintain its position and orientation within its environment.

C. Digital Communications

Particle filter and Monte Carlo methods have also found numerous applications in digital communications, including blind deconvolution [303], [83], demodulation [378], channel equalization [97], estimation and coding [84], [507], and wireless channel tracking [215], [88]. Some reviews of Monte Carlo methods in wireless communication are also found in [415] and [477], [85].

- In [98], a fixed-lag particle smoothing algorithm was used for blind deconvolution and equalization.
- In [476], the delayed-pilot sampling (which uses future observations for generating samples) was used in MKF for detection and decoding in fading channels.
- In [499], particle filter was used as blind receiver for orthogonal frequency-division multiplexing (OFDM) system in frequency-selective fading channels.
- The time-varying AR(p) process was used for Rayleigh fast-fading wireless channel tracking, where particle filtering was applied for improving symbol detector [269]. In [93], APF was used for semi-blind MIMO channel tracking.
- Jump Markov linear systems (JMLR) [94] has many

[90] Kalman filter is also a HMM filter, except that the state space is continuous-valued.

[91] An excellent review paper on hidden Markov processes was given in [160].

[92] Note that particle filter is more computationally efficient than the HMM. Suppose we discretize the continuous state-space for formulate the HMM filter with N_z discrete states, the complexity of HMM filter is $\mathcal{O}(N_z^2)$, as opposed to $\mathcal{O}(N_z)$ for particle filter.

[93] Some on-line algorithms were also developed for HMM [26], [429].

[94] Jump Markov system is referred to the system whose parameters

implications in communications, where particle filters can be applied [147].

D. Speech Enhancement and Speech Recognition

The speech signal is well known for its non-Gaussianity and non-stationarity, by accounting for the existence of non-Gaussian noise in real life, particle filter seems a perfect candidate tool for speech/audio enhancement and noise cancellation. Lately, many research results have been reported within this framework [467], [466], [169], [500]. It was also proposed for solving the audio source separation or (restricted and simplified version of) cocktail party problem [4].

It would be remiss of us to overlook the important application of HMM filters in automatic speech recognition (ASR). Within the Bayesian framework, HMM filters have been extensively used in speech recognition (see e.g. [380], [379], [381], [219], [220]) and speech enhancement [159], in which the latent states are discrete and finite, which correspond to the letters in the alphabet.

E. Machine Learning

The Kalman filtering methodology has been extensively used in neural networks training (see [206] and the references therein), especially in the area of real-time signal processing and control. On the other hand, in recent decade, Bayesian inference methods have been widely applied to machine learning, probabilistic inference, and neural networks. Many papers can be found in the literature [58], [317], [120], [323], including a number of Ph.D. theses [316], [346], [118], [333]. Applying Monte Carlo methods especially sequential Monte Carlo techniques also attracted many researchers' attention [120], [145], [262], [263]. In particular in [120], a novel hybrid SIR (HySIR) algorithm was developed for training neural networks, which used a EKF update to move the particles towards the gradient descent direction and consequently speech up the convergence. To generalize the generic state-space model, a more powerful learning framework will be the dynamic Bayesian networks that admit more complex probabilistic graphical models and include Fig. 2 as a special case. Another interesting branch is the Bayesian kernel machines that are rooted in the kernel method [405], which can tackle the high-dimensional data and don't suffer the curse of dimensionality. How to explore the (sequential) Monte Carlo methods to this area is still an open topic.

F. Others

It is impossible to include all of applications of Bayesian filtering and sequential Monte Carlo estimation, the literature of them is growing exponentially nowadays. We only list some of them available within our reach:

- fault diagnosis [119], [338]
- tempo tracking [76], speaker tracking [464], direction of arrival (DOA) tracking [290]
- spectral estimation [148]
- positioning and navigation [35], [196]
- time series analysis [484], financial analysis [310]
- economics and econometrics [436], [437], [443]
- biology sequence alignment [306]
- beamforming [478]
- source separation [23]
- automatic control [200], [5], [6]

G. An Illustrative Example: Robot-Arm Problem

At the end of this section, we present a simple example to illustrate the practical use of the particle filter discussed thus far. Consider the kinematics of a two-link robot arm, as shown in Fig. 17(a). For given the values of pair angles (α_1, α_2), the end effector position of the robot arm is described by the Cartesian coordinates as follows:

$$y_1 = r_1 \cos(\alpha_1) - r_2 \cos(\alpha_1 + \alpha_2), \tag{158a}$$

$$y_2 = r_1 \sin(\alpha_1) - r_2 \sin(\alpha_1 + \alpha_2), \tag{158b}$$

where $r_1 = 0.8$, $r_2 = 0.2$ are the lengths of the two links of the robot arm; $\alpha_1 \in [0.3, 1.2]$ and $\alpha_2 \in [\pi/2, 3\pi/2]$ are the joint angles restricted in specific region. The solid and dashed lines in Fig. 17(a) show the "elbow up" and "elbow down" situation, respectively. Finding the mapping from (α_1, α_2) to (y_1, y_2) is called as *forward kinematics*, whereas the *inverse kinematics* is referred to the mapping from (y_1, y_2) to (α_1, α_2). The inverse kinematics is not a one-to-one mapping, namely the solution is not unique (e.g. the "elbow up" and "elbow down" in Fig. 17(a) both give the same position). Now we want to formulate the problem as a state space model and solve the inverse kinematics problem. Let α_1 and α_2 are augmented into a state vector, denoted as $\mathbf{x} \equiv [\alpha_1, \alpha_2]^T$, the measurement vector is given by $\mathbf{y} = [y_1, y_2]^T$. Equations (158a) and (158b) are rewritten in the following form of state space model

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{d}_n, \\ \mathbf{y}_n &= \begin{bmatrix} \cos(\alpha_{1,n}) & -\cos(\alpha_{1,n} + \alpha_{2,n}) \\ \sin(\alpha_{1,n}) & -\sin(\alpha_{1,n} + \alpha_{2,n}) \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \mathbf{v}_n. \end{aligned}$$

The state equation is essentially a random-walk with assumed white Gaussian noise $\mathbf{d} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}\{0.008^2, 0.08^2\})$, the measurement equation is nonlinear with measurement noise $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, 0.005 \times \mathbf{I})$. As observed in Fig. 17(b), the state trajectories of α_1 and α_2 are independent, thus $p(\alpha_1, \alpha_2|\mathbf{y}) = p(\alpha_1|\mathbf{y})p(\alpha_2|\mathbf{y})$. α_1 is a a slowly increasing process with periodic random walk, α_2 is a periodic fast linearly-increasing/decreasing process. The SIR filter are used in our experiment.[95] Considering the fast monotonically increasing behavior of α_2, random walk model is not efficient. To be more accurate, we can roughly model the states as a time-varying first or second-order (or higher-order if necessary) AR process with unknown parameter $\mathbf{A}_n$, namely $\boldsymbol{\alpha}_{n+1} = \mathbf{A}_n \boldsymbol{\alpha}_n + \mathbf{d}_n$. The uncertainty of

evolve with time according to a finite-state Markov chain. It is also called switching Markov dynamics or switching state space model.

[95] The Matlab code for generating robot-arm problem data and a SIR filter demo are available on line http://soma.crl.mcmaster.ca/~zhechen/demo_robot.m.

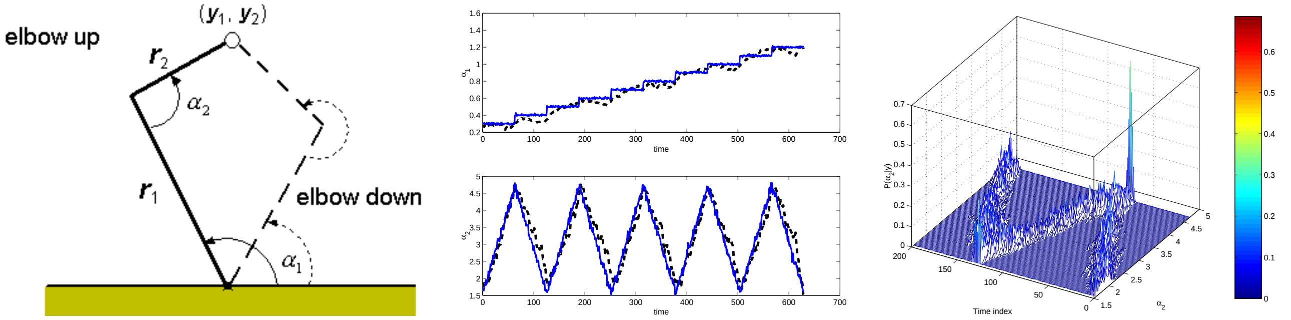

Fig. 17. Schematic illustration of a two-link robot arm in two dimensions. (a) **Left:** for given joint angles (α_1, α_2), the position of the end effector (circle symbol), described by the Cartesian coordinates (y_1, y_2), is uniquely determined. (b) **Middle:** the state trajectories (solid) of (α_1, α_2) in experiment. The dotted lines are the estimates given by SIR filter (multinomial resampling) using a random-walk model with $N_p = 200$. (c) **Right:** the pdf evolution of α_2 in the first 200 steps.

$\mathbf{A}_n = [a_{1,n}, b_{1,n}, a_{2,n}, b_{2,n}]^T$ is augmented into the state for joint estimation (to be discussed in next section). In this context, the new augmented state equation becomes

$$\mathbf{x}^a_{n+1} = \mathbf{F}_{n+1,n}\mathbf{x}^a_n + \mathbf{d}_n$$

where

$$\mathbf{x}^a_{n+1} = [\alpha_{1,n+1}, \alpha_{1,n}, \alpha_{2,n+1}, \alpha_{2,n}, a_{1,n+1}, b_{1,n+1}, a_{2,n+1}, b_{2,n+1}]^T,$$

and

$$\mathbf{F}_{n+1,n} = \begin{bmatrix} a_{1,n} & b_{1,n} & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{2,n} & b_{2,n} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Since $\mathbf{A}_n$ doesn't enter the likelihood, by conditioning on $\boldsymbol{\alpha}$, $\mathbf{A}$ is a linear Gaussian model, therefore it can be estimated separately by other methods, such as gradient descent, recursive least-squares (RLS), or Rao-Blackwellization.[96] Namely, the joint estimation problem is changed to a dual estimation problem (see next section). It can be also solved with the EM algorithm, in which E-step uses Bayesian filtering/smoothing for state estimation, and M-step estimates the AR parameters via ML principle. The marginalization approach allows particle filter to work in a lower-dimensional space, thereby reducing the variance and increasing the robustness. Hence, the Kalman filter update is embedded in every iteration for every particle. The detailed derivation and comparative experimental results will be given elsewhere.

IX. Discussion and Critique

A. Parameter Estimation

The parameter estimation problem arises from the fact that we want to construct a parametric or nonparametric model to fit the observed data, and the Bayesian procedure is used for model selection (not discussed here), hyperparameter selection (specifying priors or regularization coefficient, not discussed here), and probabilistic inference (of the unknown parameters). Parameter estimation has been extensively used in off-line Bayesian estimation [272], Bayesian learning (e.g. for neural networks) [58], [316], [346], [118], or Bayesian identification [366], [367], [280]. It is also related to Bayesian modeling and time series analysis [480], [483], [484], [372], [373].

Parameter estimation can be also treated in an on-line estimation context. Formulated in a state space model, the transition density of the parameters is a random-walk (or random-field) model, the likelihood is often described by a parametric model (e.g. a neural network). It is also possible to use the gradient information to change the random-walk behavior to accelerate the convergence in a dynamic environment, as illustrated in [**?**]. Recently, many authors have applied particle filters or sequential Monte Carlo methods for parameter estimation or static model [310], [13], [95]. In many cases, particle filters are also combined with other inference techniques such as data augmentation [13], EM [43], or gradient-based methods. However, there are two intrinsic open problems arising from parameter estimation using particle filtering technique. (i) The pseudo state is neither "causal" nor "ergodic", the convergence property is lost; (ii) The state space can be very large (order of hundreds), where the curse of dimensionality problem might be very severe. These two problems can somehow be solved with MCMC techniques, some papers are devoted to this direction [13], [16].

B. Joint Estimation and Dual Estimation

If one encounters some parameter uncertainty in state estimation, the problem of state estimation and parameter (either fixed parameter or time-varying parameter) estimation simultaneously arises. Generally, there is no unique optimal solution for this problem. Hence we are turn into finding a suboptimal solution. One way is to treat the unknown parameters $\boldsymbol{\theta}$ as part of the states, by this trick one can use conventional filtering technique to infer the parameter and state simultaneously. This is usually called

[96] This arises from the fact that $p(\mathbf{A}_n|\boldsymbol{\alpha}_{0:n}, \mathbf{y}_{0:n})$ is Gaussian distributed which can be estimated a Kalman filter, and $p(\mathbf{A}_n, \boldsymbol{\alpha}_n|\mathbf{y}_{0:n})$ can be obtained from $p(\boldsymbol{\alpha}_{0:n}|\mathbf{y}_{0:n})$.

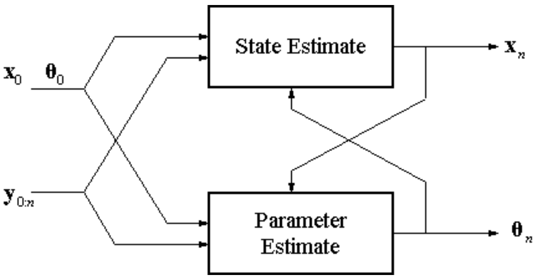

Fig. 18. A suboptimal solution of dual estimation problem.

joint estimation [473]. The problem of joint estimation is to find out the joint probability distribution (density) of the unknown parameters and states, $p(\mathbf{x}_n, \boldsymbol{\theta}|\mathbf{y}_{0:n})$, which usually has no analytic form. Another problem of joint estimation using particle filtering is that, when the parameter is part of the state, the augmented state space model is not ergodic, and the uniform convergence result doesn't hold any longer [102]. An alternative solution is dual estimation, which uses an iterative procedure to estimate the state and parameters alternatingly. Dual estimation was first suggested in [12], and was lately studied in detail in [473], [352], with some new development. The idea of dual estimation is illustrated in Fig. 18, where a suboptimal sequential estimation solution is sought. Dual estimation can be understood as a generalized EM algorithm: E-step uses Kalman or particle filter for state estimation; whereas M-step performs model parameter estimation. The iterative optimization process guarantees the algorithm to converge to the suboptimal solution.

C. Prior

In the Bayesian estimation (filtering or inference) context, choosing an appropriate prior (quantitatively and qualitatively) is a central issue.[97] In the case where no preferred prior is available, it is common to choose a noninformative prior. It was called because the prior can be merely determined from the data distribution which is the only available information. The purpose of noninformative priors is to attain an "objective" inference within the Bayesian framework.[98]

Laplace was among the first who used noninformative methods ([388], chap. 3). In 1961, Jeffrey first proposed a kind of noninformative prior based on Fisher information, which is the so-called Jeffrey's prior [388], [38]

$$\pi(\boldsymbol{\theta}) \propto |\mathbf{H}(\boldsymbol{\theta})|^{1/2}, \tag{159}$$

where

$$|\mathbf{H}(\boldsymbol{\theta})|_{ij} = -\int p(\mathbf{x}|\boldsymbol{\theta}) \frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} \tag{160}$$

is a Fisher information matrix. The logarithmic divergence locally behaves like the square of a distance, determined by a Riemannian metric with a natural length element $|\mathbf{H}(\boldsymbol{\theta})|^{1/2}$, the natural length elements of Riemannian metric are *invariant* to reparameterization. The Jeffrey's prior has a nice geometrical interpretation: the natural volume elements generate "uniform" measures on the manifolds, in the sense that equal mass is assigned to regions of equal volume, which makes Lebesque measure intuitively appealing. Another approach to construct a noninformative prior is the so-called "reference priors" [38], [389], which maximize asymptotically the expected KL divergence.

In order to use conjugate approach in Bayesian filtering or inference, conjugate priors are often chosen [388], [38], which can be of a single or a mixture form. the mixture conjugate priors allows us to have much freedom in modeling the prior distribution. Within the conjugate approach-based filtering, the inference can be tackled analytically. Dirichlet prior is an important conjugate prior in the exponential family and widely used in Bayesian inference. In addition, priors can be designed in the robust priors framework [226], e.g. the ϵ-contaminated robust priors.

D. Localization Methods

The intuition of localization idea is that, realizing the fact that it is infeasible to store the whole state trajectories or data due to limited storage resource in practice, instead of ambitiously finding an optimal estimate in a global sense, we are turn to find a locally optimal estimate by taking account of most important observations or simulated data. Mathematically, we attempt to find a locally unbiased but with minimum variance estimator. This idea is not new and has been widely used in machine learning [50], [463], control [337], signal processing (e.g. forgetting factor), and statistics (e.g. kernel smoothing). Localization can be either time localization or space localization. By time localization, it is meant that in the time scale, a local model is sought to characterize the most recent observation data, or the data are introduced with an exponential discounting/forgetting factor. By space localization, it is referred to in any time instant, the sparse data are locally represented, or the data are smoothed in a predefined neighborhood around the current observation, among the whole data space.

The localization idea has been used for Monte Carlo sampling [304], [3]. In the context of filtering, the forgetting factor has been introduced for particle filter [137]. Bearing in mind that we encounter the risk that the particle filters might accumulate the estimate inaccuracy along the time, it is advisable to take the localization approach w.r.t. the trajectory. Namely, in order to estimate $\hat{\mathbf{x}}_n$ at time n, we only use partial observations, i.e. the posterior reduces to $p(\mathbf{x}_n|\mathbf{y}_{n-\tau:n})$ $(1 \leq \tau \leq n)$ instead of $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. Kernel-based smoothing is one of the popular localization methods, and it is straightforward to apply it to particle filters. The candidate kernel can be Gaussian or Epanechnikov. In addition to the disadvantage of introducing bias (see Section VI-G), another shortcoming of kernel smoothing is the curse of dimensionality, and it cannot be updated sequentially.

[97] When a flat prior is chosen, the Bayesian result reduces to the frequentist approach.

[98] Maximum-likelihood based methods essentially ignore the priors, or regard the priors as uniform.

E. Dimensionality Reduction and Projection

Many state space models usually satisfy $N_{\mathbf{y}} \leq N_{\mathbf{x}}$. When $N_{\mathbf{y}} > N_{\mathbf{x}}$ (e.g., the observation is an image), some dimensionality reduction or feature extraction techniques are necessary. In this case, the observation data are usually sparely distributed, we can thus project the original high-dimensional data to a low-dimensional subspace. Such techniques include principal component analysis (PCA), SVD, factor analysis, nearest-neighborhood model. For example, in visual tracking, people attempted to perform the sampling in a subspace, namely to find a 2D image space for the 3D object motion. Likewise in robot localization, the sensor information is usually high-dimensional with an unknown measurement model, in on-line processing the sensor information arrives much faster than the update of the filter, not to mention the audio-visual data association problem. In order to handle such situation, dimensionality reduction becomes a must-be,[99] either for a fixed measurement model or a nonparametric model [471].

Projection idea is to project the object (data, distribution, or function) to a subspace which is "well-posed", this geometrical insight has been widely used in filtering, learning, and inference. The idea of projection can be also considered for the proposal distribution. The basic intuition is to assume that the the current posterior $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ is close to the previous posterior $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$, the only update arises from the new observation $\mathbf{y}_n$. In order to draw samples from proposal $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})$, we project the previous posterior to the subspace (called proposal space) by marginalization (see Fig. 19). In the subspace we draw the samples $\{\mathbf{x}_n^{(i)}\}$ and use Bayes rule to update the posterior. Usually the update will deviate again from the subspace (but not too far away), hence it is hoped that in the next step we can project it back to the proposal space. The reason behind it is that the subspace is usually simpler than the true posterior space and it is also easy to sample. To do this, we can use data augmentation technique discussed earlier in Section VI-H. Suppose at time step n we have the approximate posterior $\hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$, given new observation $\mathbf{y}_n$, we use the marginalization approach to alternatingly generate the augmented $\mathbf{z}^{(i)}$ (they are thus called the "imputations" of the observations). First we assume

$$\begin{aligned} q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n}) &= q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}, \mathbf{y}_n) \\ &\approx \hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, \mathbf{y}_n). \end{aligned}$$

By viewing the new observation as an augmented data $\mathbf{z}$, we can draw the samples from the proposal through the marginalized density

$$q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n}) \approx \int \hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, \mathbf{z}) p(\mathbf{z}|\mathbf{y}_{0:n-1}) d\mathbf{z},$$

$$p(\mathbf{z}|\mathbf{y}_{0:n-1}) = \int p(\mathbf{z}|\mathbf{x}_{n-1}, \mathbf{y}_{0:n-1}) \hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}.$$

Since $\mathbf{z}$ is supposed to be independent of the previous observations, hence $p(\mathbf{z}|\mathbf{y}_{0:n-1})$ reduces to $p(\mathbf{z})$ and we further have

$$\begin{aligned} q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n}) &\approx \int \hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, \mathbf{z}) p(\mathbf{z}) d\mathbf{z}, \\ p(\mathbf{z}) &= \int p(\mathbf{z}|\mathbf{x}_{n-1}) \hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1} \\ &= \frac{1}{N_p} \sum_{i=1}^{N_p} p(\mathbf{z}|\mathbf{x}_{n-1}^{(i)}), \end{aligned}$$

where $\hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ is the previous posterior estimate represented by a discrete set $\{\mathbf{x}_{n-1}^{(i)}\}_{i=1}^{N_p}$. Let $\mathbf{z}^{(0)} = \mathbf{y}_n$, we can use the similar sampling procedure discussed in Section VI-H.2. The details of the methodology will be presented elsewhere [**?**]. Our idea of projection filtering[100] is similar but not identical to the one in [51], in which they used marginalization idea for the belief update in the DBN, but their method involved neither data augmentation nor Bayesian sampling.

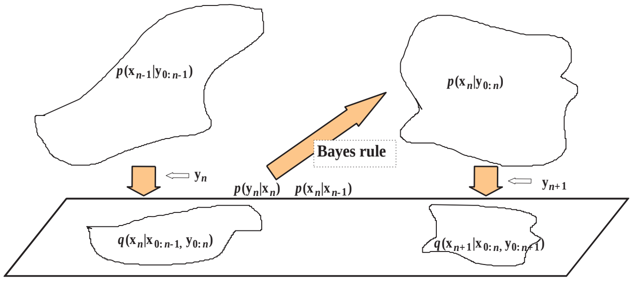
$p(\mathrm{x}_{n-1}|\mathrm{y}_{0:n-1})$
$p(\mathrm{x}_n|\mathrm{y}_{0:n})$
Bayes rule
y_n
$p(\mathrm{y}_n|\mathrm{x}_n)$
$p(\mathrm{x}_n|\mathrm{x}_{n-1})$
y_{n+1}
$q(\mathrm{x}_n|\mathrm{x}_{0:n-1}, \mathrm{y}_{0:n})$
$q(\mathrm{x}_{n+1}|\mathrm{x}_{0:n}, \mathrm{y}_{0:n+1})$

Fig. 19. A geometrical illustration of projection/marginalization of Bayesian filtering.

F. Unanswered Questions

Having discussed many features of particle filters, at this position, a question naturally occurring to us is:

Does particle filtering have free lunch?

In particular, we feel that the following issues have not been satisfactorily addressed in the literature.

First, how to choose effective particles still lacks rigorous theoretical justification. How many independent samples (or antithetic variables) are needed in the sequential Monte Carlo methods? Is it possible to get some upper and lower bounds of necessity of number of particles (see an attempted effort in [171]), though they are usually quite loose and are problem-dependent? Of course, we can blindly increase the number of particles to improve the approximation accuracy, however, it will also inevitably increase the variance (due to the bias-variance dilemma, we cannot make bias and variance simultaneously small according to the *Uncertainty Principle*), not to mention the increasing computational effort and sampling inefficiency (No free lunch!). Albeit many techniques were used to improve the degenerate problem, it seems to the authors that none of them are totally satisfactory. On the other hand, how to

[99] Another novel method called real-time particle filter [288] has been lately proposed to address the same problem in a different way.

[100] Note that the term "projection filter" has been abused in the literature with different meanings.

seek an adaptive procedure of choosing/adding *informative* particles (or "support particles"), still remains an open problem.[101] This issue becomes crucial when we encounter the *scaling* problem: the algorithm remains computationally feasible when dimensionality of $N_{\mathbf{x}}$ is order of hundreds or thousands. In addition, the number of sufficient particles depends largely on the chosen proposal distribution, with a good choice, the error might vanish at a linear rate of the increasing N_p; with a bad choice, the error might increase exponentially with increasing $N_{\mathbf{x}}$ no matter how large N_p is.

Second, the cumulative error due to the inaccuracy of the simulated samples at each iteration may grow exponentially. For SIR or SIS filters, bias and variance will both increases along the time; for rejection particle filter, the variance also increases given a moderate number of particles. In addition, as recalled in the discussion of convergence behavior, the uniform convergence cannot be assured unless N_p increases over the time or the particle filter has the capability to forget the error exponentially. A good example is given in [361]:

> Suppose the transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ is uniform and independent of $\mathbf{x}_{n-1}$, the likelihood is binary with $p(\mathbf{y}_n = 1|\mathbf{x}_n)$ if $\mathbf{x}_n < 0.2$ and $p(\mathbf{y}_n = 0|\mathbf{x}_n)$ otherwise. If the true states happen to stay in $[0, 0.2)$ so that $\mathbf{y}_n = 1$ for all n. However, the probability of having no particles (which are binomially distributed) within $[0, 0.2)$ in any one of n time steps is $1 - (1 - 0.8^{N_p})^n$, which converges to 1 exponentially with increasing n; in other words, the particle filter almost loses the true trajectory completely.

Although this is an extreme example which might never happen in the real life, it does convince us that the inaccuracy will bring a "catastrophic" effect as time evolves such that the filter either diverges or deviates far away from the true states. In this sense, *"Bayesian statistics without tears"* will be probably rephrased as *"particle filtering with tears"*. Although the above example is a special toy problem, it does make us realize the importance of the robustness issue posed earlier. On the other hand, it is noted that convergence behavior is a transient phenomenon, nothing is said about the error accumulation in a long run. Does error approach a steady state? How to characterize the steady-state behavior of particle filter? To our best knowledge, theoretical results are still missing.

Third, Bayesian principle is not the only induction principle for statistical inference. There might also exist other principles, e.g. minimax (worst case analysis), SRM (structural risk minimization), MDL (minimum description length), or Occam's razor. Is Bayesian solution always optimal in any sense? The answer is no. The Bayesian method makes sense only when the *quantitative* prior is correct [463]. In other words, in the situation lack of a priori knowledge, Bayesian solution is possibly misleading. In fact, the conflict between SRM and Bayesianism has been noticed in the machine learning literature (e.g. [463]). In the context of Bayesian filtering, the quantitative prior will be the chosen proposal distribution, initial state density $p(\mathbf{x}_0)$ and noise statistics. Unfortunately, none of them of is assured in practice. To our best knowledge, this question has not been addressed appropriately in the literature. Nevertheless, it is suspected that we might benefit from the rigorous theoretical results established in the dependency estimation and statistical/computational learning literature [463], many notions such as metric entropy, VC dimension, information complexity, are potentially useful for establishing strong mathematical results for Monte Carlo filtering. For example, since the integrand is known, how do we incorporate the prior knowledge into Monte Carlo sampling?[102] Is it possible to introduce structural hypothesis class for proposal distribution? Is it possible to establish a upper bound or lower bound for particular Monte Carlo integration (i.e. a problem-dependent bound that is possibly much tighter than the generic Cramér-Rao bound)?

Particle filters certainly enjoy some free lunches in certain special circumstances, e.g. partially observable Gaussian model, decoupled weakly Gaussian model. However, answering the all of concerns of a general problem, unfortunately, have no free lunch. It was felt that the current status of particle filter research is very similar to the situation encountered in the early 1990s of neural networks and machine learning. Such examples include the bootstrap technique, asymptotic convergence result, bias-variance dilemma, curse of dimensionality, and NFL theorem. In no doubt, there are still a lot of space left for theoretical work on particle filters. As firstly addressed in the theoretic exposition [128], the theories of interacting particle systems [300], large deviation theory [59], [126], Feller semigroups, limit theorem, etc. are the heart of Monte Carlo or particle filtering theory. But they are certainly not the whole story.

One of theoretical issue, for example, is about the abuse the information in Monte Carlo simulation, since it is usually hard to verify quantitatively the information we use and ignore. Recently, Kong *et al.* [267] have partially approached this question, in which they formulated the problem of Monte Carlo integration as a statistical model with simulation draws as data, and they further proposed a semi-parametric model with the *baseline measure* as a parameter, which makes explicit what information is ignored and what information is retained in the Monte Carlo methods; the parameter space can be estimated by the ML approach.

It is also noteworthy to keep in mind that the classic Monte Carlo methods belong to the frequentist procedure, a question naturally arising is: *Can one seek a Bayesian version of Monte Carlo method?* [318]. Lately, this question has been partially tackled by Rasmussen and Ghahramani [382], in which they proposed a Bayesian Monte Carlo (BMC) method to incorporate prior knowledge (e.g.

[101] This issue was partially addressed in the paper [88].

[102] As matter of fact, as we discussed earlier in importance sampling, the proposal distribution can be chosen in a smart way to even lower down the true variance.

smoothness) of the integrand to the Monte Carlo integration: Given a large number of samples, the integrand $\{f(\mathbf{x}^{(i)})\}_{i=1}^{N_p}$ is assumed to be a Gaussian process (i.e. the prior is defined in the functional space instead of data space) [489], their empirical experimental results showed that the BMC is much superior to the regular Monte Carlo methods. It would be beneficial to introduce this technique to the on-line filtering context. Besides, in real-life applications, the noise statistics of dynamical systems are unknown, which are also needed to be estimated within Bayesian framework via introducing hyperparameters; thus the hierarchical Bayesian inference are necessary. To summarize, there can be several levels of Bayesian analysis for different objects: data space, parameter/hyperparameter space, and functional space.

Currently, we are investigating the average/worst case of Monte Carlo filtering/inference. The objective is to attempt to find the upper/lower bounds using variational methods [241], [237], [236]. The potential applications combining deterministic variational Bayesian approximation and stochastic Monte Carlo approximation are very promising, which are also under investigation.

X. Summary and Concluding Remarks

In this paper, we have attempted to present a tutorial exposition of Bayesian filtering, which covers such topics as stochastic filtering theory, Bayesian estimation, and Monte Carlo methods. Within the sequential state estimation framework, Kalman filter reduces to be a special case of Bayesian filtering in the LQG scenario; particle filter, rooted deeply in Bayesian statistics and Monte Carlo technique, comes up as a powerful solution candidate for tackling the real-life problems in the physical world where the nonlinearity and non-Gaussianity abound.

It is our purpose to provide the reader a complete picture of particle filters originated from stochastic filtering theory. Besides Monte Carlo filtering, other Bayesian filtering or Bayesian inference procedures are also addressed. It is obvious that the theory of Bayesian filtering presented here has a lot of potentials in variety of scientific and engineering areas, thus suitable for a wide circle of readers. Certain applications in artificial intelligence, signal processing, communications, statistics, and machine learning, have been already mentioned in Section VIII. In addition to the sequential Monte Carlo nature of estimation, another attractive property of particle filter is that it allows flexibility design and parallel implementation. On the other hand, it should be cautioned that particle filters are not the panacea, designing special particle filter in practice is problem dependent and requires a good understanding of the problem at hand. We should also be borne in mind that this area is far from mature and has left a lot of space for theoretical work.

In summary, most of research issues of particle filters focused on (and will still concentrate on) the following:

- Choices of proposal distribution;
- Choices of resampling scheme and schedule;
- Efficient use of simulated samples and monitoring the sample efficiency;
- Exploration of smoothing, regularization, data augmentation, Rao-Blackwellization, and MCMC variations.
- Exploration of of different (or new) Monte Carlo integration rules for efficient sampling.

Another promising future direction seems to be combining particle filtering with other inference methods to produce a fruitful outcome. The geometrical and conjugate approaches provide many insights for application of Rao-Blackwellization and data augmentation.

In no doubt, modern Monte Carlo methods have opened the door to more realistic and complex probabilistic models. For many complex stochastic processes or dynamics where the posterior distributions are intractable, various approximate inference methods other than Monte Carlo approximation come in (e.g., mean-field approximation, variational approximation), or they can be combined to use together (e.g. [121]). Alternatively, one can also simplify the complex stochastic processes by the ways of decomposition, factorization, and modulation for the sake of inference tractability. For the higher-order Markov dynamics, mixture or hierarchical structure seems necessary and efficient approximation inference are deemed necessary. To conclude, *from the algorithm to practice, it is a rocky road*, but there is no reason to disbelieve that we can pave the way forward.

Appendix A: A Proof

Assuming that $\mathbf{x}^{(i)}$ $(i = 1, \cdots, N_p)$ are N_p i.i.d. samples, $\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}]$ and $\hat{\boldsymbol{\mu}} = \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}^{(i)}$ are the expected mean and sample mean, respectively. The covariance of sample estimate $\hat{\boldsymbol{\mu}}$ is calculated as

$$\begin{aligned}
\mathrm{Cov}[\hat{\boldsymbol{\mu}}] &= \mathbb{E}[(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T] \\
&= \mathbb{E}[\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\mathbb{T}}] - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathbb{T}} \\
&= \mathbb{E}\Big[(\frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}^{(i)})(\frac{1}{N_p}\sum_{j=1}^{N_p}\mathbf{x}^{(j)})^T\Big] - \boldsymbol{\mu}\boldsymbol{\mu}^T \\
&= \frac{1}{N_p^2}\sum_{i=1}^{N_p}\sum_{j=1}^{N_p}\mathbb{E}[\mathbf{x}\mathbf{x}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T \\
&= \frac{N_p\mathbb{E}[\mathbf{x}\mathbf{x}^T] + (N_p^2 - N_p)\boldsymbol{\mu}\boldsymbol{\mu}^T}{N_p^2} - \boldsymbol{\mu}\boldsymbol{\mu}^T \\
&= \frac{\mathbb{E}[\mathbf{x}\mathbf{x}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T}{N_p} = \frac{1}{N_p}\mathrm{Cov}[\mathbf{x}]
\end{aligned}$$

where $\mathrm{Cov}[\mathbf{x}]$ is the covariance of random vector $\mathbf{x}$, the fourth step in above equation uses the *independence assumption* of $\mathbf{x}$

$$\mathbb{E}\Big[(\mathbf{x}^{(i)})(\mathbf{x}^{(j)})^T\Big] = \begin{cases} \mathbb{E}[\mathbf{x}\mathbf{x}^T] & i = j \\ \mathbb{E}[\mathbf{x}^{(i)}]\mathbb{E}[\mathbf{x}^{(j)}]^T = \boldsymbol{\mu}\boldsymbol{\mu}^T & i \neq j \end{cases}$$

Appendix B: Convergence of Random Variables

Definition 8: Almost Convergence (or Convergence with Probability 1): A sequence of $\{X_n\}$ is said to converge to a random variable X with probability 1 if for any $\zeta > 0$, $\epsilon > 0$

$$\Pr\{\omega : |X_n(\omega) - X(\omega)| < \epsilon\} > 1 - \zeta$$

is satisfied for all $n > N$ where N may depend on both ζ and ϵ. Or equivalently,

$$\Pr\{\omega : | \lim_{n\to\infty} X_n(\omega) = X(\omega)\} = 1.$$

Definition 9: Mean Square Convergence: A sequence of $\{X_n\}$ of random variables is said to converge to a random variable X in the mean-square sense if

$$\mathbb{E}[(X_n(\omega) - X(\omega))^2] \to 0 \quad (n \to \infty)$$

or $\lim_{n\to\infty} \mathbb{E}[(X_n(\omega) - X(\omega))^2] = 0$.

Definition 10: Convergence in Probability: A sequence of $\{X_n\}$ of random variables converges in probability to the random variable X if for every $\epsilon > 0$

$$\lim_{n\to\infty} \Pr\{|X_n(\omega) - X(\omega)| \geq \epsilon\} = 0.$$

Definition 11: Convergence in Distribution: A sequence of $\{X_n\}$ of random variables is said to converge to a random variable X in distribution if the distribution functions $F_n(x)$ of X_n converge to the distribution function $F(x)$ of X at all points of continuity of F, namely,

$$\lim_{n\to\infty} F_n(x) = F(x)$$

for all x at which $F(x)$ is continuous.

Appendix C: Random Number Generator

In what follows, we briefly discuss some popular random number generators. Strictly speaking, we can only construct the pseudo-random or quasi-random number generators, which are deterministic in nature but the samples they generated exhibit the same or similar statistical properties as the true random samples. For standard distributions such as uniform, Gaussian, exponential, some exact random sampling algorithms exist. Other standard distributions are generally obtained by passing an inverse of the cumulative distribution function (cdf) with a pseudo-random sequence, the resulting distributions are mostly approximate rather than exact.

Theorem 6: [168] Let $\{F(z), a \leq z \leq b\}$ denote a distribution function with an inverse distribution function as

$$F^{-1}(z) = \inf\{z \in [a,b] : F(z) \geq u,\ 0 \leq u \leq 1\}.$$

Let u denote a random variable from $\mathcal{U}(0,1)$, then $z = F^{-1}(u)$ has the distribution function $F(z)$.

Reader is referred to [168], [389], [386], [132] for more information. For simulation purpose, the Matlab user can find many random number generators for various distributions in the Statistics Toolbox (MathWorks Inc.).

Uniform distribution

The uniform random variable is the basis on which the other random number generators (other than uniform distribution) are constructed. There are many uniform random number generators available [386]. The following routine is a one based on the congruencial method

- Start with an arbitrary seed x_0;
- $x_n = (69069 x_{n-1} + 1) \bmod 2^{32}$,
- $u_n = 2^{-32} x_n$.

where the sequence u_n can be regarded as the i.i.d. uniform random variables drawn from $\mathcal{U}(0,1)$. Some uniform distribution random number generator functions in Matlab are `rand`, `unifrnd`, and `unidrnd`.

Normal (Gaussian) distribution

Suppose u_1 and u_2 are two random variables uniformly distributed in $\mathcal{U}(0,1)$, by taking

$$\begin{aligned} x_1 &= \mu + \sigma\sqrt{-2\log(u_1)}\cos(2\pi u_2), \\ x_2 &= \mu + \sigma\sqrt{-2\log(u_1)}\sin(2\pi u_2), \end{aligned}$$

then x_1 and x_2 can be regarded as two independent draws from $\mathcal{N}(\mu, \sigma^2)$; this algorithm is exact [389].

It can be also generated by the transformation method by calculating the cdf

$$\begin{aligned} F(x) &= \int_0^x \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(\xi-\mu)^2}{\sigma^2}) d\xi \\ &= \frac{1}{2}\Big[1 + \mathrm{erf}\Big(\frac{x-\mu}{\sqrt{2\sigma^2}}\Big)\Big], \end{aligned}$$

then the random number can be generated by the inverse function

$$x = F^{-1}(u) = \mu + \sqrt{2\sigma^2}\,\mathrm{erf}^{-1}(2u - 1).$$

Some normal distribution random number generator functions in Matlab include `mvnrnd` or `normrnd` or `randn` (for $\mathcal{N}(\mathbf{0}, \mathbf{I})$).

Exponential and Logistic distribution

Let u be one random variable uniformly distributed in $\mathcal{U}(0,1)$, by taking $x = -\log(u)/\lambda$, then x can be regarded as a draw from exponential distribution $\texttt{Exponential}(\lambda)$; by calculating $x = \log\frac{u}{1-u}$, then x can be regarded as a draw from logistic distribution $\texttt{Logistic}(0,1)$ [389]. An exponential distribution random number generator function in Matlab is `exprnd`.

Cauchy distribution

To generate the Cauchy distribution, we can use the transformation method. The pdf of zero-mean Cauchy distribution is given by

$$p(x) = \frac{\sigma}{\pi}\frac{1}{\sigma^2 + x^2}$$

where σ^2 is the variance. The cdf of Cauchy distribution is

$$F(x) = \int_{-\infty}^x \frac{\sigma}{\pi}\frac{1}{\sigma^2 + \xi^2} d\xi = \frac{1}{\pi}\arctan(\frac{x}{\sigma}) + \frac{1}{2}.$$

The transformation is then given by the inverse transform $x = F^{-1}(u)$:

$$F^{-1}(u) = \sigma\tan(\pi(u - \frac{1}{2})) = -\sigma\cot(\pi u).$$

Hence given some uniform random numbers $u \in \mathcal{U}(0,1)$, we can use above relationship to produce the Cauchy random numbers by $x = -\sigma \cot(\pi u)$. The acceptance-rejection sampling approach to generate Cauchy distribution proceeds as follows [168]:

- repeat
- generate u_1 and u_2 from $\mathcal{U}(-1/2, 1/2)$
- until $u_1^2 + u_2 \leq 1/4$
- return u_1/u_2.

Laplace distribution

Laplace distribution is also called double exponential distribution. It is the distribution of differences between two independent variates with identical exponential distributions. The pdf of Laplace distribution is given by

$$p(x) = \frac{1}{2\sigma} \exp\Big(-\frac{|x|}{\sigma}\Big)$$

where σ is a positive constant. The distribution function of Laplace distribution is

$$F(x) = \begin{cases} \frac{1}{2}\exp(\frac{x}{\sigma}) & x < 0 \\ 1 - \frac{1}{2}\exp(\frac{-x}{\sigma}) & x \geq 0 \end{cases},$$

and the inverse transform $x = F^{-1}(u)$ is given by

$$F^{-1}(u) = \begin{cases} \sigma \ln(2u) & 0 < u < 1/2 \\ -\sigma \ln(2 - 2u) & 1/2 \leq u < 1 \end{cases}.$$

Given some uniform random numbers $u \in \mathcal{U}(0,1)$, we can use above relationship to produce the Laplace distributed random variables $x = F^{-1}(u)$.

Appendix D: Control Variate and Antithetic Variate

Control variate and antithetic variate are two useful variance-reduction techniques by exploring the knowledge of integrand. To illustrate the idea, only one-dimensional variable is considered here.

Suppose we want to estimate an integral of interest

$$\theta = \int \phi(x)p(x)dx \equiv \int f(x)dx.$$

To achieve this, we use another known statistics

$$\mu = \int \phi(x)q(x)dx \equiv \int h(x)dx$$

to further construct an equivalent integral

$$\theta = \mu + \int (f(x) - h(x))dx,$$

where μ is a known constant, $h(x)$ is called as a "control variate", which is usually chosen to be close to $f(x)$.

In order to reduce the variance (i.e. the right-hand side is no more than the left-hand side), we need to show $\mathrm{Var}[f(x)] \geq \mathrm{Var}[f(x) - h(x)]$, which is equivalent to $\mathrm{Var}[h(x)] < 2\mathrm{Cov}[f(x), h(x)]$, where

$$\mathrm{Cov}[f(x), h(x)] = \int (f(x) - \theta)(h(x) - \mu)dx.$$

Suppose $\hat{\theta}$ is an unbiased Monte Carlo estimate obtained from exact draws, namely $\mathbb{E}[\hat{\theta}] = \theta$. We can find another unbiased estimator $\hat{\mu}$ ($\mathbb{E}[\hat{\mu}] = \mu$), as a control variate, to construct a new estimator

$$\theta' = \hat{\theta} + \mu - \hat{\mu}.$$

It is obvious that θ' is also an unbiased estimate of θ. The variance of this new estimator is given by

$$\begin{aligned} \mathrm{Var}[\theta'] &= \mathrm{Var}[\hat{\theta} - \hat{\mu}] \\ &= \mathrm{Var}[\hat{\theta}] + \mathrm{Var}[\hat{\mu}] - 2\mathrm{Cov}[\hat{\theta}, \hat{\mu}], \end{aligned}$$

hence $\mathrm{Var}[\theta'] < \mathrm{Var}[\hat{\theta}]$ if $\mathrm{Var}[\hat{\mu}] < 2\mathrm{Cov}[\hat{\theta}, \hat{\mu}]$. In some sense, controlled variate can be understood as a kind of variational method.

Antithetic variate is a variance-reduction method exploiting the negative correlation. Suppose $\hat{\theta}$ and θ' are two unbiased estimates of θ, we construct another unbiased estimate as

$$\hat{\mu} = \frac{\hat{\theta} + \theta'}{2},$$

whose variance is given by

$$\mathrm{Var}[\hat{\mu}] = \frac{1}{4}\mathrm{Var}[\hat{\theta}] + \frac{1}{4}\mathrm{Var}[\theta'] + \frac{1}{2}\mathrm{Cov}[\hat{\theta}, \theta'].$$

Suppose $\hat{\theta}$ and θ' are two Monte Carlo estimates obtained from exact draws, if θ' is chosen s.t. $\mathrm{Cov}[\hat{\theta}, \theta'] < 0$ (i.e. the Monte Carlo samples are negatively correlated instead of independent; a.k.a. correlated sampling), variance reduction is achieved.

For example, if the integrand is a symmetric function w.r.t. $\frac{a+b}{2}$ over the region $[a, b]$, we can write $f(x) = \frac{f(x)+f(a+b-x)}{2}$ (when $-a = b$, it reduces to an even function). Thus we can introduce negative correlation since generally $\mathrm{Cov}[f(x), f(a + b - x)] < 0$; if $a = 0, b = 1$ and $f(x) \sim \mathcal{U}(0, 1)$, then $\mathrm{Cov}[f(x), f(1 - x)] = -1$.

More generally, if $f(\cdot)$ is a *monotonically increasing/decreasing* function, then $f(x)$ and $f(1 - x)$ are negatively correlated. Hence in order to reduce the variance, one may construct a Monte Carlo estimate

$$\frac{1}{2N_p}\sum_{i=1}^{N_p}(f(x^{(i)}) + f(1 - x^{(i)})),$$

instead of using the naive estimates $\frac{1}{N_p}\sum_{i=1}^{N_p} f(x^{(i)}$ or $\frac{1}{N_p}\sum_{i=1}^{N_p} f(1 - x^{(i)}$.

Example 2: To give a more specific example, consider drawing the samples from a zero mean Cauchy distribution discussed in Appendix C. Given uniform random variables $u \sim \mathcal{U}(0, 1)$, we can produce the Cauchy random numbers by $x_1 = -\sigma \cot(\pi u)$. On the other hand, noting that $1 - u$ are also uniformly distributed that is negatively correlated with u. Utilizing this symmetry property, we can generate another set of Cauchy random numbers $x_2 = -\sigma \cot(\pi(1 - u)) = \sigma \tan(\pi u)$. Obviously, x_1 and x_2 are slightly negatively correlated, their covariance is also usually negative. By drawing $N_p/2$ samples of x_1 and $N_p/2$ samples of x_2, we obtain some negatively correlated samples from Cauchy distribution. Alternatively, by constructing N_p samples $x = (x_1 + x_2)/2$, we have $\text{Var}[x] < \max\{\text{Var}[x_1], \text{Var}[x_2]\}$, and $\text{Var}[x]$ is expected to reduce, compared to the two independent runs for x_1 and x_2. The sample estimate of x is unbiased, i.e., $\mathbb{E}[x] = \mathbb{E}[x_1] = \mathbb{E}[x_2]$. Also note that when x_1 and x_2 are negatively correlated, $f(x_1)$ and $f(x_2)$ are usually negatively correlated when $f(\cdot)$ is a monotonic function.

This approach can be utilized in any transformation-based random number generation technique (Appendix C) whenever applicable (i.e., using uniform random variable, and F^{-1} being a monotonic function). Such examples include exponential distribution, logistic distribution, and Laplace distribution.

Appendix E: Unscented Transformation Based on SVD

There are many types of matrix factorization techniques [42], e.g. Cholesky factorization, U-D factorization, LDU^T factorization.[103] Hence we can use different factorization methods to implement the unscented transformation (UT). The basic idea here is to use singular value decomposition (SVD) instead of Cholesky factorization in the UT. In Table X, the state estimation procedure is given, the extension to parameter estimation is straightforward and is omitted here. As to the notations, $\mathbf{P}$ denotes the state-error correlation matrix, $\mathbf{K}$ denotes the Kalman gain, ρ is a scaling parameter (a good choice is $1 \leq \rho \leq \sqrt{2}$) for controlling the extent of covariance,[104] κ is a small tuning parameter. The computational complexity of SVD-KF is the same order of $\mathcal{O}(N_{\mathbf{x}}^3)$ as UKF.

TABLE X

The SVD-based Derivative-free Kalman Filtering for State Estimation.

Initialization

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0], \quad \hat{\mathbf{P}}_0 = \mathbb{E}[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T].$$

Compute the SVD and eigen-point covariance matrix

$$\begin{aligned}
\mathbf{P}_n &= \mathbf{U}_n \mathbf{S}_n \mathbf{V}_n^T \\
\boldsymbol{\mathcal{X}}_{0,n-1} &= \hat{\mathbf{x}}_{n-1} \\
\boldsymbol{\mathcal{X}}_{i,n-1} &= \hat{\mathbf{x}}_{n-1} + \rho \mathbf{U}_{i,n} \sqrt{s_{i,n}}, \quad i = 1, \cdots, N_{\mathbf{x}} \\
\boldsymbol{\mathcal{X}}_{i,n-1} &= \hat{\mathbf{x}}_{n-1} - \rho \mathbf{U}_{i,n} \sqrt{s_{i,n}}, \quad i = N_x + 1, \cdots, 2N_{\mathbf{x}}
\end{aligned}$$

Time updates

$$\begin{aligned}
\boldsymbol{\mathcal{X}}_{i,n|n-1} &= \mathbf{f}(\boldsymbol{\mathcal{X}}_{i,n-1}, \mathbf{u}_n), \quad i = 0, 1, \cdots, 2N_{\mathbf{x}} \\
\hat{\mathbf{x}}_{n|n-1} &= \boldsymbol{\mathcal{X}}_{0,n|n-1} + \sum_{i=1}^{2N_{\mathbf{x}}} \mathcal{W}_i^{(m)} (\boldsymbol{\mathcal{X}}_{i,n|n-1} - \boldsymbol{\mathcal{X}}_{0,n|n-1}) \\
\mathbf{P}_{n|n-1} &= \sum_{i=0}^{2N_{\mathbf{x}}} \mathcal{W}_i^{(c)} (\boldsymbol{\mathcal{X}}_{i,n|n-1} - \hat{\mathbf{x}}_{n|n-1})(\boldsymbol{\mathcal{X}}_{i,n|n-1} - \hat{\mathbf{x}}_{n|n-1})^T + \Sigma_{\mathbf{d}} \\
\mathcal{Y}_{i,n|n-1} &= \mathbf{g}(\boldsymbol{\mathcal{X}}_{i,n|n-1}, \mathbf{u}_n), \quad i = 0, 1, \cdots, 2N_{\mathbf{x}} \\
\hat{\mathbf{y}}_{n|n-1} &= \mathcal{Y}_{0,n|n-1} + \sum_{i=1}^{2N_{\mathbf{x}}} \mathcal{W}_i^{(m)} (\mathcal{Y}_{i,n|n-1} - \mathcal{Y}_{0,n|n-1})
\end{aligned}$$

Measurement updates

$$\begin{aligned}
\mathbf{P}_{\hat{\mathbf{y}}_n \hat{\mathbf{y}}_n} &= \sum_{i=0}^{2N_{\mathbf{x}}} \mathcal{W}_i^{(c)} (\mathcal{Y}_{i,n|n-1} - \hat{\mathbf{y}}_{n|n-1})(\mathcal{Y}_{i,n|n-1} - \hat{\mathbf{y}}_{n|n-1})^T + \Sigma_{\mathbf{v}} \\
\mathbf{P}_{\hat{\mathbf{x}}_n \hat{\mathbf{y}}_n} &= \sum_{i=0}^{2N_{\mathbf{x}}} \mathcal{W}_i^{(c)} (\boldsymbol{\mathcal{X}}_{i,n|n-1} - \hat{\mathbf{x}}_{n|n-1})(\mathcal{Y}_{i,n|n-1} - \hat{\mathbf{y}}_{n|n-1})^T \\
\mathbf{K}_n &= \mathbf{P}_{\hat{\mathbf{x}}_n \hat{\mathbf{y}}_n} \mathbf{P}_{\hat{\mathbf{y}}_n \hat{\mathbf{y}}_n}^{-1} \\
\hat{\mathbf{x}}_n &= \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n (\mathbf{y}_n - \hat{\mathbf{y}}_{n|n-1}) \\
\mathbf{P}_n &= \mathbf{P}_{n|n-1} - \mathbf{K}_n \mathbf{P}_{\hat{\mathbf{y}}_n \hat{\mathbf{y}}_n} \mathbf{K}_n^T
\end{aligned}$$

$$\text{Weights}: \quad \mathcal{W}_i^{(m)} = \frac{1}{2N_{\mathbf{x}}}, \quad \mathcal{W}_0^{(c)} = \frac{\kappa}{N_{\mathbf{x}} + \kappa}, \quad \mathcal{W}_i^{(c)} = \frac{1}{2N_{\mathbf{x}} + 2\kappa}$$

Appendix F: No Free Lunch Theorem

The no-free lunch (NFL) [105] theorems basically claim that no learning algorithms can be *universally* good; in other words, an algorithm that performs exceptionally well in certain situations will perform comparably poorly in other situations. For example, NFL for optimization [493], for cross-validation, for noise prediction, for early stopping, for bootstrapping, to name a few (see also [87] for some discussions on NFL in the context of regularization theory). The implication of NFL theorem is that, given two random based algorithms `Algorithm A` and `Algorithm B`, suppose `Algorithm A` is superior to `Algorithm B` averaged on some set of target S, then `Algorithm B` must be superior to `Algorithm A` if averaging over all target not in S. Such examples also include sampling theory and Bayesian analysis [491].

For the particle filters (which certainly belong to random based algorithm class), the importance of prior knowledge is very crucial. Wolpert [491], [492] has given a detailed mathematical treatment of the issues of existence and lack of prior knowledge in machine learning framework. But the discussions can be certainly borrowed to stochastic filtering context. In Monte Carlo filtering methods, the most valuable and important prior knowledge is the proposal distribution. No matter what kind of particle filter is used, an appropriately chosen proposal is directly related to the final performance. The choice of proposal is further related to the functions $\mathbf{f}$ and $\mathbf{g}$, the likelihood model or measurement noise density. Another crucial prior knowledge is the noise statistics, especially the dynamical noise. If the $\Sigma_{\mathbf{d}}$ is small, the weight degeneracy problem is severe, which requires us to either add "jitter" or choose regularization/smoothing technique. Also, the prior knowledge of

[103] The factorization is not unique but the factorization techniques are related, they can be used to develope various forms of square-root Kalman filters [42], [247].

[104] In one-dimensional Gaussian distribution, variance σ^2 accounts for 95% covering region of data ($2\sigma^2$ for 98%, $3\sigma^2$ for 99%).

[105] The term was first used by David Haussler.

the model structure is helpful for using data augmentation and Rao-Blackwellization techniques.

Appendix G: Notations

Symbol	**Description**
$\mathbb{N}$	integer number set
$\mathbb{R}(\mathbb{R}^+)$	(positive) real-valued number set
$\mathbf{u}$	input vector as driving force
$\mathbf{x}$	continuous-valued state vector
$\boldsymbol{z}$	discrete-valued state vector
$\mathbf{y}$	measurement vector
$\mathbf{z}$	augmented (latent) variable vector
$\mathbf{e}$	state-error error (innovations)
$\boldsymbol{\omega}$	Wiener process
$\mathbf{d}$	dynamical noise vector
$\mathbf{v}$	measurement noise vector
$\Sigma_{\mathbf{d}}, \Sigma_{\mathbf{v}}$	covariance matrices of noises
$\mathbf{P}$	correlation matrix of state-error
$\mathbf{I}$	identity matrix
$\mathbf{J}$	Fisher information matrix
$\mathbf{K}$	Kalman gain
$\mathbf{f}(\cdot)$	nonlinear state function
$\mathbf{g}(\cdot)$	nonlinear measurement function
$\mathbf{F}$	state transition matrix
$\mathbf{G}$	measurement matrix
$\mathbf{H}$	Hessian matrix
$l(\mathbf{x})$	logarithm of optimal proposal distribution
$\boldsymbol{\mu}$	true mean $\mathbb{E}[\mathbf{x}]$
$\hat{\boldsymbol{\mu}}$	sample mean from exact sampling
Σ	true covariance
$\hat{\Sigma}$	sample covariance
$\hat{f}_{N_p}$	Monte Carlo estimate from exact sampling
$\hat{f}$	Monte Carlo estimate from importance sampling
$\mathbf{x}^{(i)}$	the i-th simulated sample (particle)
$\tilde{\mathbf{x}}_n$	prediction error $\mathbf{x}_n - \hat{\mathbf{x}}_n(\mathcal{Y}_n)$
$\emptyset$	empty set
S	set
f, g, ϕ	generic nonlinear functions
F	distribution function
$\mathrm{sgn}(\cdot)$	signum function
$\mathrm{erf}(\cdot)$	error function
$\lfloor \cdot \rfloor$	floor function
$\delta(\cdot)$	Dirac delta function
$\mathbb{I}(\cdot)$	indicator function
$K(\cdot, \cdot)$	kernel function
$\alpha(\cdot, \cdot)$	probability of move
$\mathrm{Pr}(\cdot)$	probability
$\mathcal{P}$	parametric probability function family
P, Q	probability distribution
p	probability density (mass) function
q	proposal distribution, importance density
π	(unnormalized) density/distribution
$\mathcal{E}$	energy
$\mathcal{K}$	kinetic energy
$N_{\mathbf{x}}$	the dimension of state
$N_{\mathbf{y}}$	the dimension of measurement
N_p	the number of particles
N_z	the number of discrete states
N_{eff}, N'_{eff}	the number of effective particles
N_T	the threshold of effective particles
N_{KL}	$\mathrm{KL}(q\|p)$ estimate from important weights
m	the number of mixtures
c	mixture coefficient
C	constant
W	importance weight
$\tilde{W}$	normalized importance weight
ξ	auxiliary variable
t	continuous-time index
n	discrete-time index
τ	time delay (continuous or discrete)
X, Y, Z	sample space
$\mathcal{X}_n$	equivalent to $\mathbf{x}_{0:n} \equiv \{\mathbf{x}_0, \cdots, \mathbf{x}_n\}$
$\mathcal{Y}_n$	equivalent to $\mathbf{y}_{0:n} \equiv \{\mathbf{y}_0, \cdots, \mathbf{y}_n\}$
$\mathscr{X}$	sigma points of $\mathbf{x}$ in unscented transformation
$\mathscr{Y}$	sigma points of $\mathbf{y}$ in unscented transformation
$\mathscr{W}$	sigma weights in unscented transformation
$\mathbb{E}[\cdot]$	mathematical expectation
$\mathrm{Var}[\cdot], \mathrm{Cov}[\cdot]$	variance, covariance
$\mathtt{tr}(\cdot)$	trace of matrix
$\mathtt{diag}$	diagonal matrix
$\mathbf{A}^T$	transpose of vector or matrix $\mathbf{A}$
$\lvert \cdot \rvert$	determinant of matrix
$\\| \cdot \\|$	norm operator
$\\| \cdot \\|_{\mathbf{A}}$	weighted norm operator
$\mathcal{E}$	loss function
$\Psi(\cdot)$	sufficient statistics
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Normal distribution with mean $\boldsymbol{\mu}$ and covariance Σ
$\mathcal{U}(0, 1)$	uniform distribution in the region $(0, 1)$
$(\Omega, \mathcal{F}, P)$	probability space
$\mathcal{O}(\cdot)$	order of
$\sim$	sampled from
$\boldsymbol{A}$	operator
$\tilde{\boldsymbol{A}}$	adjoint operator
$\boldsymbol{L}$	differential operator
$\boldsymbol{T}$	integral operator
a.k.a.	also known as
a.s.	almost sure
e.g.	exempli gratia
i.e.	id est
i.i.d.	identically and independently distributed
s.t.	such that
w.r.t.	with respect to

Acknowledgement

This paper would not be possible without the contributions of numerous researchers in this ever growing field. The author would like to thank Drs. Simon Haykin and Thia Kirubarajan (McMaster) for reading the manuscript and providing many feedbacks. We are also grateful to Dr. Fred Daum (Raytheon) for sharing his unpublished papers with us as well as many helpful comments, and Dr. Yuguo Chen (Duke) for providing his Stanford Ph.D. thesis in the early stage for better understanding sequential Monte Carlo methods. We also thank Dr. David J.C. MacKay (Cambridge) for allowing us to reproduce a figure in his paper. Finally, special thanks are also due to Prof. Rudolph E. Kalman for his seminal contribution that directly motivated the writing of this paper.

References

[1] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Automat. Contr.*, vol. 15, pp. 10–17, 1970.

[2] S. L. Adler, "Over-relaxation method for the Monte-Carlo evaluation of the partition function for multiquuadratic actions," *Phys. Rev. D*, vol. 23, no. 12, pp. 2901–2904, 1981.

[3] M. Aerts, G. Claeskens, N. Hens, and G. Molenberghs, "Local multiple imputation," *Biometrika*, vol. 89, no. 2, pp. 375–388.

[4] A. Ahmed, "Signal separation," Ph.D. thesis, Univ. Cambridge, 2000. Available on line http://www-sigproc.eng.cam.ac.uk/publications/theses.html

[5] H. Akashi and H. Kumamoto, "Construction of discrete-time nonlinear filter by Monte Carlo methods with variance-reducing techniques," *Systems and Control*, vol. 19, pp. 211–221, 1975 (in Japanese).

[6] ———, "Random sampling approach to state estimation in switching environments," *Automatica*, vol. 13, pp. 429–434, 1977.

[7] D. F. Allinger and S. K. Mitter, "New results in innovations problem for nonlinear filtering," *Stochastics*, vol. 4, pp. 339–348, 1981.

[8] D. L. Alspach, and H. W. Sorenson, "Nonlinear Bayesian estimation using gaussian sum approximation," *IEEE Trans. Automat. Contr.*, vol. 20, pp. 439–447, 1972.

[9] S. Amari, *Differential Geometrical Methods in Statistics*, Lecture Notes in Statistics, Berlin: Springer, 1985.

[10] S. Amari and H. Nagaoka, *The Methods of Information Geometry*, New York: AMS and Oxford Univ. Press, 2000.

[11] B. D. O. Anderson and J. B. Moore, "The Kalman-Bucy filter as a true time-varying Wiener filter," *IEEE Trans. Syst., Man, Cybern.*, vol. 1, pp. 119–128, 1971.

[12] ———, *Optimal Filtering*, Prentice-Hall, 1979.

[13] C. Andrieu and A. Doucet, "Recursive Monte Carlo algorithms for parameter estimation in general state space models" in *Proc. IEEE Signal Processing Workshop on Statistical Signal Processing*, pp. 14–17, 2001.

[14] ———, "Particle filtering for partially observed Gaussian state space models," *J. Roy. Statist. Soc., Ser. B*, vol. 64, pp. 4, pp. 827–836, 2002.

[15] C. Andrieu, N. de Freitas, and A. Doucet, "Rao-Blackwellised particle filtering via data augmentation," in *Adv. Neural Inform. Process. Syst. 14*, Cambridge, MA: MIT Press, 2002.

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

[17] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems", in *Proc. IEEE ICASSP2002*, vol. 2, pp. 1625–1628.

[18] ———, "Improved auxiliary particle filtering: Application to time-varying spectral analysis", in *Proc. IEEE Signal Processing Workshop on Statistical Signal Processing*, 2001, pp. 14–17.

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

[20] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous time nonlinear systems from discrete noisy measurements," *IEEE Trans. Automat. Contr.*, vol. 13, pp. 504–514, 1968.

[21] H. Attias, "Inferring parameters and structure of latent variable models by variational Bayes," in *Proc. 15th Conf. UAI, UAI'99*, 1999.

[22] ———, "A variational Bayesian framework for graphical models," in *Adv. Neural Inform. Process. Syst., 12*, Cambridge, MA: MIT Press, 2000.

[23] ———, "Source separation with a microphone array using graphical models and subband filtering," in *Adv. Neural Inform. Process. Syst., 15*, Cambridge, MA: MIT Press, 2003.

[24] D. Avitzour, "A stochastic simulation Bayesian approach to multitarget tracking," *IEE Proc.-F*, vol. 142, pp. 41–44, 1995.

[25] B. Azimi-Sadjadi, "Approximate nonlinear filtering with applications to navigation," Dept. Elect. Comput. Engr., Univ. Maryland, College Park, 2001.

[26] P. Baldi and Y. Chauvin, "Smooth on-line learning algorithms for hidden Markov models," *Neural Comput.*, vol. 6, pp. 307–318, 1994.

[27] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, New York: Academic Press, 1988.

[28] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorihtms, and Software*, New York: Wiley, 2001.

[29] T. R. Bayes, "Essay towards solving a problem in the doctrine of chances," *Phil. Trans. Roy. Soc. Lond.*, vol. 53, pp. 370–418, 1763. Reprinted in *Biometrika*, vol. 45, 1958.

[30] M. Beal and Z. Ghahramani, "The variational Kalman smoother," Tech. Rep., GCNU TR2001-003, Gatsby Computational Neuroscience Unit, Univ. College London, 2001.

[31] E. R. Beadle and P. M. Djurič, "Fast weighted boodstrap filter for non-linear state estimation," *IEEE Trans. Aerosp. Elect. Syst.*, vol. 33, pp. 338–343, 1997.

[32] Ya. I. Belopolskaya and Y. L. Dalecky, *Stochastic Equations and Differential Geometry*, Kluwer Academic Publishers, 1990.

[33] V. E. Beneš, "Exact finite-dimensional for certain diffusions with nonlinear drift," *Stochastics*, vol. 5, no. 1/2, pp. 65–92, 1981.

[34] ———, "New exact nonlinear filters with large Lie algebras," *Syst. Contr. Lett.*, vol. 5, pp. 217-221, 1985.

[35] N. Bergman, "Recursive Bayesian estimation: Navigation and tracking applications," Ph.D. thesis, Linköping Univ., Sweden, 1999.

[36] ———, "Posterior Cramér-Rao bounds for sequential estimation," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[37] N. Bergman, A. Doucet, and N. Gordon, "Optimal estimation and Cramer-Rao bounds for partial non-Gaussian state space models," *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 97–112, 2001.

[38] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*, 2nd ed., New York: Wiley, 1998.

[39] D. P. Bertsekas and I. B. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Trans. Automat. Contr.*, vol. 16, pp. 117–128, 1971.

[40] C. Berzuini, N. G. Best, W. Gilks, and C. Larizza, "Dynamic conditional independent models and Markov chain Monte Carlo methods," *J. Amer. Statist. Assoc.*, vol. 92, pp. 1403–1412, 1997.

[41] C. Berzuini and W. Gilks, "RESAMPLE-MOVE filtering with cross-model jumps," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[42] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, New York: Academic Press, 1977.

[43] A. Blake, B. North, and M. Isard, "Learning multi-class dynamics," in *Adv. Neural Inform. Process. Syst. 11*, pp. 389–395, Cambridge, MA: MIT Press, 1999.

[44] A. Blake, B. Bascle, M. Isard, and J. MacCormick, "Statistical models of visual shape and motion," *Proc. Roy. Soc. Lond. Ser. A*, vol. 356, pp. 1283–1302, 1998.

[45] B. Z. Robrovsky, E. Mayer-Wolf, and M. Zakai, "Some classes of global Cramér-Rao bounds," *Ann. Statist.*, vol. 15, pp. 1421–1438, 1987.

[46] H. W. Bode and C. E. Shannon, "A simplified derivation of linear least square smoothing and prediction theory," *Proc. IRE*, vol. 38, pp. 417–425, 1950.

[47] Y. Boers, "On the number of samples to be drawn in particle filtering," *Proc. IEE Colloquium on Target Tracking: Algorithms and Applications*, Ref. No. 1999/090, 1999/215, pp. 5/1–5/6, 1999.

[48] Y. Boers and J. N. Driessen, "Particle filter based detection for tracking," in *Proc. Amer. Contr. Conf.*, vol. 6, pp. 4393–4397, 2001.

[49] E. Bølviken, P. J. Acklam, N. Christophersen, J-M. Størdal, "Monte Carlo filters for nonlinear state estimation," *Automatica*, vol. 37, pp. 177–183, 2001.

[50] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Comput.*, vol. 4, pp. 888–900, 1992.

[51] X. Boyen, and D. Koller, "Tractable inference for complex stochastic process," in *Proc. 14th Conf. Uncertainty in AI, UAI'98*, pp. 33–42, 1998.

[52] P. Boyle, M. Broadie, and P. Glasserman, "Monte Carlo methods for security pricing," *J. Economic Dynamics and Control*, vol. 3, pp. 1267–1321, 1998.

[53] D. Brigo, B. Hanzon, and F. LeGland, "A differential geometric approach to nonlinear filtering: the prjection filter," *IEEE Trans. Automat. Contr.*, vol. 43, no. 2, pp. 247–252, 1998.

[54] ———, "Approximate nonlinear filtering by projection on ex-

ponential manifolds of densities," *Bernoulli*, vol. 5, no. 3, pp. 495–534, 1999.

[55] D. Brigo, "Filtering by projection on the manifold of exponential densities," Ph.D. thesis, Dept. Economics and Econmetrics, Free University of Amsterdam, the Netherlands, 1996. Available on line http://www.damianobrigo.it/.

[56] ———, "Diffusion processes, manifolds of exponential densities, and nonlinear filtering," in *Geometry in Present Day Science*, O. E. Barndorff-Nielsen and E. B. V. Jensen, Eds., World Scientific, 1999.

[57] ———, "On SDE with marginal laws evolving in finite-dimensional exponential families," *Statist. Prob. Lett.*, vol. 49, pp. 127–134, 2000.

[58] W. L. Bruntine and A. S. Weigend, "Bayesian back-propagation," *Complex Syst.*, vol. 5, pp. 603–643, 1991.

[59] J. A. Bucklew, *Large Deviation Techniques in Decision, Simulations, and Estimation*, Wiley, 1990.

[60] R. S. Bucy and P. D. Joseph, *Filtering for Stochastic Processes with Applications to Guidance*, New York: Wiley, 1968.

[61] R. S. Bucy, "Linear and nonlinear filtering," *Proc. IEEE*, vol. 58, no. 6, pp. 854–864, 1970.

[62] ———, "Bayes theorem and digital realization for nonlinear filters," *J. Astronaut. Sci.*, vol. 17, pp. 80–94, 1969.

[63] R. S. Bucy and K. D. Senne, "Digital synthesis of non-linear filters," *Automatica*, vol. 7, pp. 287–298, 1971.

[64] R. S. Bucy and H. Youssef, "Nonlinear filter representation via spline functions," in *Proc. 5th Symp. Nonlinear Estimation*, 51–60, 1974.

[65] Z. Cai, F. LeGland, and H. Zhang, "An adaptive local grid refinement method for nonlinear filtering," Tech. Rep., INRIA, 1995.

[66] F. Campillo, F. Ceŕou, and F. LeGland, "Particle and cell approximation for nonlinear filtering," Tech. Rep. 2567, INIRA, 1995.

[67] B. P. Carlin, N. G. Polson, and D. S. Stoffer, "A Monte Carlo approach to non-normal and non-linear state-space modelling," *J. Amer. Statist. Assoc.*, vol. 87, pp. 493–500, 1992.

[68] C. Cargnoni, P. Müller, and M. West, "Bayesian forecasting of multinomial time series through conditionally gaussian dynamic models," *J. Amer. Statist. Assoc.*, vol. 92, pp. 587–606, 1997.

[69] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proc. -F Radar, Sonar Navig.*, vol. 146, no. 1, pp. 2–7, 1999.

[70] ———, "Building robust simulation-based filters for evolving data sets," Tech. Rep., Statist. Dept., Oxford Univ., 1998. Available on line http://www.stats.ox.ac.uk/~clifford/particles/.

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

[72] ———, "Markov chain Monte Carlo in conditionally Gaussian state-space models," *Biometrika*, vol. 83, no. 3, pp. 589–601, 1996.

[73] G. Casella and E. George, "Explaining the Gibbs sampler," *Am. Statist.*, vol. 46, pp. 167–174, 1992.

[74] G. Casella and C. P. Robert, "Rao-Blackwellization of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996.

[75] G. Casella, "Statistical inference and Monte Carlo algorithms," *Test*, vol. 5, pp. 249–344, 1997.

[76] A. T. Cemgil and B. Kappen, "Rhythm quantization and tempo tracking by sequential Monte Carlo," in *Adv. Neural Inform. Process. Syst. 14*, Cambridge, MA: MIT Press, 2002.

[77] F. Cérou and F. LeGland, "Efficient particle methods for residual generation in partially observed SDE's," in *Proc. 39th Conf. Decision and Control*, pp. 1200–1205, 2000.

[78] S. Challa and Y. Bar-Shalom, "Nonlinear filter design using Fokker-Planck-Kolmogorov probability density evolutions," *IEEE Trans. Aero. Elect. Syst.*, vol. 36, no. 1, pp. 309–315, 2000.

[79] C. D. Charalambous and S. M. Diouadi, "Stochastic nonlinear minimax filtering in continous-time," in *Proc. 40th IEEE Conf. Decision and Control*, vol. 3, pp. 2520–2525, 2001.

[80] G. Chen, Ed. *Approximate Kalman Filtering*, Singapore: World Scientific, 1993.

[81] M.-H. Chen and B. W. Schmeiser, "Performances of the Gibbs, hit-and-run, and Metropolis samplers," *J. Comput. Graph. Stat.*, vol. 2, pp. 251–272, 1993.

[82] M.-H. Chen, Q.-M. Shao, and J. G. Ibrahim, *Monte Carlo Methods in Bayesian Computation*, Springer, 2000.

[83] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. Roy. Statist. Soc., Ser. B*, vol. 62, pp. 493–508, 2000.

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

[85] R. Chen, J. S. Liu, and X. Wang, "Convergence analyses and comparison of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 255–269, Feb. 2002.

[86] Y. Chen, "Sequential importance sampling with resampling: Theory and applications," Ph.D. thesis, Stanford Univ., 2001.

[87] Z. Chen and S. Haykin, "On different facets of regularization theory," *Neural Comput.*, vol. 14, no. 12, pp. 2791–2846, 2002.

[88] Z. Chen and K. Huber,, "Robust particle filters with applications in tracking and communications", Tech. Rep., Adaptive Systms Lab, McMaster University, 2003.

[89] J. Cheng and M. J. Druzdzel, "AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks," *J. Artif. Intell. Res.*, vol. 13, pp. 155–188, 2000.

[90] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *Am. Stat.*, vol. 49, pp. 327–335, 1995.

[91] Y. T. Chien and K. S. Fu, "On Bayesian learning and stochastic approximation," *IEEE Trans. Syst. Sci. Cybern.*, vol. 3, no. 1, pp. 28-38.

[92] W. H. Chin, D. B. Ward, and A. G. Constantinides, "Semi-blind MIMO channel tracking using auxiliary particle filtering," in *Proc. GLOBECOM*, 2002.

[93] K. Choo and D. J. Fleet, "People tracking with hybrid Monte Carlo filtering," in *Proc. IEEE Int. Conf. Comp. Vis.*, vol. II, pp. 321–328, 2001.

[94] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, Aug. 2002.

[95] C. K. Chui and G. Chen, *Kalman Filtering: With Real-Time Applications*, 2nd ed., Berlin: Springer-Verlag, 1991.

[96] T. Clapp, "Statistical methods in the processing of communications data," Ph.D. thesis, Dept. Eng., Univ. Cambridge, U.K., 2000. Available on line http://www-sigproc.eng.cam.ac.uk/publications/theses.html

[97] T. Clapp and S. J. Godsill, "Fixed-lag smoothing using sequential importance sampling," in *Bayesian Statistics 6*, J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith Eds. pp. 743–752, Oxford: Oxford Univ. Press, 1999.

[98] M. K. Cowles and B. P. Carlin, "Markov chain Monte Carlo convergence diagnostics — A comparative review," *J. Amer. Statist. Assoc.*, vol. 91, pp. 883–904, 1996.

[99] F. G. Cozman, "An informal introduction to quasi-Bayesian theory," Tech. Rep., CMU-RI-TR 97-24, Robotics Institute, Carnegie Mellon Univ., 1997.

[100] ———, "Calculation of posterior bounds given convex sets of prior probability measures and likelihood functions," *J. Comput. Graph. Statist.*, vol. 8, no. 4, pp. 824–838, 1999.

[101] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practioners," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.

[102] D. Crisan, J. Gaines, T. Lyons, "Convergence of a branching particle method to the solution of the Zakai equation," *SIAM J. Appl. Math.*, vol. 58, no. 5, pp. 1568–1598, 1998.

[103] D. Crisan, P. Del Moral, T. Lyons, "Interacting particle systems approximations of the Kushner Stratonovitch equation," *Adv. Appl. Prob.*, vol. 31, no. 3, pp. 819–838, 1999.

[104] ———, "Non-linear filtering using branching and interacting particle systems," *Markov Processes Related Fields*, vol. 5, no. 3, pp. 293–319, 1999.

[105] D. Crisan, "Particle filters - A theoretical perspective," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[106] ———, "Exact rates of convergence for a branching particle approximation to the solution of the Zakai equation," *Ann. Prob.*, vol. 32, April 2003.

[107] ———, "A direct computation of the Benes filter conditional density," *Stochastics and Stochastic Reports*, vol. 55, pp. 47–54, 1995.

[108] L. Csató and M. Opper, "Sparse on-line Gaussian process," *Neural Comput.*, vol. 14, pp. 641–668, 2002.

[109] A. I. Dale, *A History of Inverse Probability: From Thomas Bayes to Karl Pearson*, New York: Springer-Verlag, 1991.

[110] F. E. Daum, "Exact finite dimensional nonlinear filters," *IEEE Trans. Automat. Contr.*, vol. 31, no. 7, pp. 616–622, 1986.

[111] ———, "New exact nonlinear filters," in *Bayesian Analysis of Time Series and Dynamic Models*, J. C. Spall, Ed. New York: Marcel Dekker, 1988, pp. 199–226.

[112] ———, "Industrial strength nonlinear filters," in *Proc. Estimation, Tracking, and Fusion Workshop: A Tribute to Prof. Yaakov Bar-Shalom*, 2001.

[113] ———, "Solution of the Zakai equation by separation of variables," *IEEE Trans. Automat. Contr.*, vol. 32, no. 10, pp. 941–943, 1987.

[114] ———, "Dynamic quasi-Monte Carlo for nonlinear filters," in *Proc. SPIE*, 2003.

[115] F. E. Daum and J. Huang, "Curse of dimensionality for particle filters," submitted paper preprint.

[116] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed. New York: Academic Press, 1984.

[117] J. F. G. de Freitas, "Bayesian methods for neural networks," Ph.D. thesis, Dept. Eng., Univ. Cambridge, 1998. Available on line http://www.cs.ubc.ca/~nando/publications.html.

[118] ———, "Rao-Blackwellised particle filtering for fault diagnosis," in *Proc. IEEE Aerospace Conf.*, vol. 4, pp. 1767–1772, 2002.

[119] J. F. G. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet, "Sequential Monte Carlo methods to train neural network models," *Neural Comput.*, vol. 12, no. 4, pp. 955–993, 2000.

[120] J. F. G. de Freitas, P. Højen-Sørensen, M. Jordan, and S. Russell, "Variational MCMC," Tech. Rep., UC Berkeley, 2001.

[121] P. Del Moral, "Non-linear filtering using random particles," *Theo. Prob. Appl.*, vol. 40, no. 4, pp. 690–701, 1996.

[122] ———, "Non-linear filtering: Interacting particle solution," *Markov Processes Related Fields*, vol. 2, no. 4, pp. 555–580, 1996.

[123] P. Del Moral and G. Salut, "Particle interpretation of non-linear filtering and optimization," *Russian J. Mathematical Physics*, vol. 5 , no. 3, pp. 355–372, 1997.

[124] P. Del Moral and A. Guionnet, "Central limit theorem for nonlinear filtering and interacting particle systems," *Ann. Appl. Prob.*, vol. 9, pp. 275–297, 1999.

[125] ———, "Large deviations for interacting particle systems: Applications to nonlinear filtering problems" *Stochast. Process. Applicat.*, vol. 78, pp. 69–95, 1998.

[126] P. Del Moral and M. Ledoux, "On the convergence and the applications of empirical processes for interacting particle systems and nonlinear filtering," *J. Theoret. Prob.*, vol. 13, no. 1, pp. 225–257, 2000.

[127] P. Del Moral and L. Miclo, "Branching and interacting particle systems approximations of Feynamkac formulae with applications to nonlinear filtering," in *Seminaire de Probabilities XXXIV*, Lecture Notes in Mathematics, no. 1729, pp. 1–145, Berlin: Springer-Verlag, 2000.

[128] P. Del Moral, J. Jacod, and Ph. Protter, "The Monte-Carlo method for filtering with discrete-time observations," *Probability Theory and Related Fields*, vol. 120, pp. 346–368, 2001.

[129] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorihtm," *J. Roy. Statist. Soc., Ser. B*, vol. 39, pp. 1–38, 1977.

[130] J. Deutscher, A. Blake and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2000, vol. 2, pp. 126–133.

[131] L. Devroye, *Non-uniform Random Variate Generation*, Berlin: Springer, 1986.

[132] X. Dimakos, "A guide to exact simulation," *Int. Statist. Rev.*, vol. 69, 27–48, 2001.

[133] P. M. Djurić, Y. Huang, and T. Ghirmai, "Perferct sampling: A review and applications to signal processing," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 345–356, 2002.

[134] P. M. Djurić, J. H. Kotecha, J.-Y. Tourneret, and S. Lesage, "Adaptive signal processing by particle filters and discounting of old measurements," in *Proc. ICASSP'01*, vol. 6, pp. 3733–3736, 2001.

[135] P. M. Djurić and J-H. Chun, "An MCMC sampling approach to estimation of nonstationary hidden Markov models," *IEEE Trans. Signal Processing*, vol. 50, no. 5, pp. 1113–1122, 2002.

[136] P. M. Djurić and J. H. Kotecha, "Estimation of non-Gaussian autoregressive processes by particle filter with forgetting factors," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 2001.

[137] P. C. Doerschuk, "Cramér-Rao bounds for discrete-time nonlinear filtering problems," *IEEE Trans. Automat. Contr.*, vol. 40, no. 8, pp. 1465–1469, 1995.

[138] J. L. Doob, *Stochastic Processes.* New York: Wiley, 1953.

[139] H. Doss, J. Sethuraman, and K. B. Athreya, "On the convergence of the Markov chain simulation," *Ann. Statist.*, vol. 24, pp. 69–100, 1996.

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

[141] A. Doucet, "Monte Carlo methods for Bayesian estimation of hidden Markov models: Application to radiation signals," Ph.D. thesis, Univ. Paris-Sud Orsay, 1997.

[142] ———, "On sequential simulation-based methods for Bayesian filtering," Tech. Rep., Dept. Engineering, CUED-F-TR310, Cambridge Univ., 1998.

[143] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, pp. 197–208, 2000.

[144] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proc. UAI2000*, pp. 176–183, 2000.

[145] A. Doucet, N. Gordon, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 188– , Jan. 2000.

[146] ———, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, pp. 613–624, Mar. 2001.

[147] A. Doucet, S. J. Godsill, and M. West, "Monte Carlo filtering and smothing with application to time-varying spectral estimation," in *Proc. ICASSP2000*, vol. 2, pp. 701–704, 2000.

[148] ———, "Maximum a posteriori sequence estimation using Monte Carlo particle filters," *Ann. Inst. Stat. Math.*, vol. 52, no. 1, pp. 82–96, 2001.

[149] A. Doucet and V. B. Tadic, "Parameter estimation in general state-space models using particle methods," *Ann. Inst. Stat. Math.*, 2003.

[150] A. Doucet, C. Andrieu, and M. Davy, "Efficient particle filtering for jump Markov systems - Applications to time-varying autoregressions," *IEEE Trans. Signal Processing*, 2003.

[151] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Phys. Lett. B*, vol. 195, pp. 216–222, 1987.

[152] J. Durbin and S. J. Koopman, "Monte Carlo maximum likelihood estimation for non-Gaussian state space models," *Biometrika*, vol. 84, pp. 1403–1412, 1997.

[153] ———,, "Time series analysis of non-gaussian observations based on state space models from both classical and Bayesian perspectives," *J. Roy. Statist. Soc., Ser. B*, vol. 62, pp. 3–56, 2000.

[154] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, pp. 1–26, 1979.

[155] ———, *The Bookstrap, Jackknife and Other Resampling Plans*, Philadephia: SIAM, 1982.

[156] B. Efron and R. J. Tibshirani, *An Introduction to the Bookstrap*, London: Chapman & Hall, 1994.

[157] G. A. Einicke and L. B. White, "Robust extended Kalman filter," *IEEE Trans. Signal Processing*, vol. 47, no. 9, pp. 2596–2599, Sept. 1999.

[158] Y. Ephraim, "Bayesian estimation approach for speech enhancement using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 725–735, April 1992.

[159] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Informat. Theory*, vol. 48, no. 6, pp. 1518–1569, June 2002.

[160] R. Everson and S. Roberts, "Particle filters for non-stationary ICA," in *Advances in Independent Component Analysis*, pp. 23-41, Springer, 2000.

[161] P. Fearnhead, "Sequential Monte Carlo methods in filter theory," Ph.D. thesis, Univ. Oxford, 1998. Available on line http://www.stats.ox.ac.uk/~fhead/thesis.ps.gz.

[162] ———,, "Particle filters for mixture models with unknown number of components," paper preprint, 2001. Available on line http://www.maths.lancs.ac.uk/~fearnhea/.

[163] ———, "MCMC, sufficient statistics, particle filters," *J. Comput. Graph. Statist.*, vol. 11, pp. 848–862, 2002.

[164] P. Fearnhead and P. Clifford, "Online inference for well-log data," *J. Roy. Statist. Soc. Ser. B.*, paper preprint, 2002.

[165] L. A. Feldkamp, T. M. Feldkamp, and D. V. Prokhorov, "Neural network training with the nprKF," in *Proc. IJCNN01*, pp. 109–114.

[166] M. Ferrante and W. J. Runggaldier, "On necessary conditions for existence of finite-dimensional filters in discrete time," *Syst. Contr. Lett.*, vol. 14, pp. 63–69, 1990.

[167] G. S. Fishman, *Monte Carlo - Concepts, Algorithms and Applications*, New York, Springer, 1996.

[168] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Caro smoothing with application to audio signal processing," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 438–448, Feb. 2002.

[169] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[170] D. Fox, "KLD-sampling: Adaptive particle filters," in *Adv. Neural Inform. Process. Syst. 14*, Cambridge, MA: MIT Press, 2002.

[171] S. Frühwirth-Schnatter, "Applied state space modelling of non-Gaussian time series using integration-based Kalman filtering," *Statist. Comput.*, vol. 4, pp. 259–269, 1994.

[172] D. Gamerman, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, London: Chapman & Hall, 1997.

[173] A. Gelb, Ed. *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.

[174] A. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating mariginal densities," *J. Amer. Statist. Assoc.*, vol. 85, pp. 398–409, 1990.

[175] A. Gelman and D. B. Rubin, "Inference from iterative algorithms (with discussions)," *Statist. Sci.*, vol. 7, pp. 457–511, 1992.

[176] A. Gelman and X.-L. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Statist. Sci.*, vol. 13, pp. 163–185, 1998.

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

[178] J. E. Gentle, *Random Number Generation and Monte Carlo*, 2nd ed., Berlin: Springer-Verlag, 2002.

[179] J. Geweke, "Bayesian inference in Econometrics models using Monte Carlo integration," *Econometrica*, vol. 57, pp. 1317–1339, 1989.

[180] J. Geweke and H. Tanizaki, "On Markov chain Monte Carlo methods for nonlinear and non-gaussian state-space models," *Commun. Stat. Simul. C*, vol. 28, pp. 867–894, 1999.

[181] C. Geyer, "Practical Markov chain Monte Carlo (with discussions)," *Statist. Sci.*, vol. 7, pp. 473–511, 1992.

[182] Z. Ghahramani, "Learning dynamic Bayesian networks," in *Adaptive Processing of Sequence and Data Structure*, C. L. Giles and M. Gori, Eds. Lecture Notes in Artificial Intelligence, Springer-Verlag, 1998, pp. 168–197.

[183] ———, "An introduction to hidden Markov models and Bayesian networks," *Int. J. Pattern Recognition and Artificial Intelligience*, vol. 15, no. 1, pp. 9–42, 2001.

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

[185] W. R. Gilks and C. Berzuini, "Following a moving target – Monte Carlo inference for dynamic Bayesian models," *J. Roy. Statist. Soc., Ser. B*, vol. 63, pp. 127–1546, 2001.

[186] W. R. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *J. Roy. Statist. Soc. Ser. C*, vol. 41, pp. 337–348, 1992.

[187] R. D. Gill and B. Y. Levit, "Application of the van Trees inequality: A Bayesian Cramér-Rao bound," *Bernoulli*, vol. 1, no. 1/2, pp. 59–79, 1995.

[188] S. Godsill and T. Clapp, "Improved strategies for Monte Carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[189] S. Godsill, A. Doucet, and M. West, "Maximum a posteriori sequence estimation using Monte Carlo particle filters," in *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 82–96, 2001.

[190] N. Gordon, "Bayesian methods for tracking," Ph.D. thesis, Univ. London, 1993.

[191] ———, "A hybrid bootstrap filter for target tracking in clutter," *IEEE Trans. Aerosp. Elect. Syst.*, vol. 33, pp. 353–358, 1997.

[192] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proc. -F Radar, Sonar Navig.*, vol. 140, pp. 107–113, 1993.

[193] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination" *Biometrika*, vol. 82, pp. 711–732, 1995.

[194] M. S. Grewal, *Kalman Filtering: Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[195] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssel, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 425–436, 2002.

[196] J. H. Halton, "A retrospective and prospective survey of the Monte Carlo method," *SIAM Rev.*, vol. 12, pp. 1–63, 1970.

[197] J. M. Hammersley and K. W. Morton, "Poor man's Monte Carlo," *J. Roy. Statist. Soc. Ser. B*, vol. 16, pp. 23–38, 1954.

[198] J. M. Hammersley and D. C. Hanscomb, *Monte Carlo Methods*, London: Chapman & Hall, 1964.

[199] J. E. Handschin and D. Q. Mayne, "Monte Carlo techniques to estimate conditional expectation in multi-state non-linear filtering," *Int. J. Contr.*, vol. 9, no. 5, pp. 547–559, 1969.

[200] J. E. Handschin, "Monte Carlo techniques for prediction and filtering of non-linear stochastic processes," *Automatica*, vol. 6, pp. 555–563, 1970.

[201] B. Hanzon, "A differential-geometric approach to approximate nonlinear filtering," in *Geometrization of Statistical Theory*, C. T. J. Dodson, Ed., Univ. Lancaster: ULMD Pub., pp. 219–223, 1987.

[202] P. J. Harrisons and C. F. Stevens, "Bayesian forecasting (with discussion)," *J. Roy. Statist. Soc. Ser. B*, vol. 38, pp. 205–247, 1976.

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

[204] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[205] ———, Ed., *Kalman Filtering and Neural Networks*, New York: Wiley, 2001.

[206] S. Haykin and B. Widrow, Eds., *Least-Mean-Squares Filters*, New York: Wiley, 2003.

[207] S. Haykin and N. de Freitas, Eds, *Sequential State Estimation*, forthcoming special issue *Proc. IEEE*, 2003.

[208] S. Haykin, P. Yee, and E. Derbez, "Optimum nonlinear filter," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2774-2786, 1997.

[209] S. Haykin, K. Huber, and Z. Chen, "Bayesian sequential state estimation for MIMO wireless communication," submitted to *Proc. IEEE.*

[210] D. M. Higdon, "Auxiliary variable methods for Markov chain Monte Carlo with applications," *J. Amer. Statist. Assoc.*, vol. 93, pp. 585–595, 1998.

[211] T. Higuchi, "Monte Carlo filter using the genetic algorithm operators," *J. Statist. Comput. Simul.*, vol. 59, no. 1, pp. 1–23, 1997.

[212] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Automat. Contr.*, vol. 9, pp. 333–339, Oct. 1964.

[213] A. Honkela, "Nonlinear switching state-space models," Master Thesis, Helsinki Univ. Technology, 2001.

[214] P. Huber, *Robust Statistics*, New York: Wiley, 1981.

[215] K. Huber and S. Haykin, "Application of particle filters to MIMO wireless communications," in *Proc. IEEE Int. Conf. Commu., ICC2003*, pp. 2311–2315.

[216] C. Hue, J. Le Cadre, and P. Pérez, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 309–325, 2002.

[217] ———, "Tracking multiple objects with particle filtering," *IEEE Trans. Aero. Electr. Syst.*, vol. 38, no. 3, pp. 791-812, 2002.

[218] ———, "Performance analysis of two sequential Monte Carlo methods and posterior Cramér-Rao bounds for multi-target tracking," Tech. Rep., no. 4450, INRIA, 2002.

[219] Q. Huo, and C.-H. Lee, "On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 161–172, 1997.

[220] ———, "A Bayesian predictive approach to robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 200–204, 2000.

[221] M. Hürzeler, "Statistical methods for general state-space models," Ph.D. thesis, Dept. Math., ETH Zürich, Zürich, 1998.

[222] M. Hürzeler and H. R. Künsch, "Monte Carlo approximations for general state-space models ," *J. Comput. Graphical Statist.*, vol. 7, no. 2, pp. 175–193, 1998.

[223] ———, "Approximating and Maximising the likelihood for a general state-space model," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[224] Y. Iba, "Population Monte Carlo algorihtms," *Trans. Japanese Soc. Artificial Intell.*, vol. 16, no. 2, pp. 279–286, 2001.

[225] R. A. Iltis, "State estimation using an approximate reduced statistics algorithm," *IEEE Trans. Aero. Elect. Syst.*, vol. 35, no. 4, pp. 1161–1172, Oct. 1999.

[226] D. R. Insua and F. Ruggeri, Eds. *Robust Bayesian Analysis*, Lecture Note in Statistics 152, Berlin: Springer, 2000.

[227] M. Irwin, N. Cox, and A. Kong, "Sequential imputation for multilocus linkage analysis," *Proc. Natl. Acad. Sci.*, vol. 91, pp. 11684–11688, 1994.

[228] M. Isard, "Visual motion analysis by probabilistic propagation of conditional density," D.Phil. Thesis, Oxford Univ., 1998. Avaiable on line http://research.microsoft.com/users/misard/

[229] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. 4th European Conf. Computer Vision*, vol. 1, pp. 343–356, 1996.

[230] ———, "CONDENSATION: conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[231] ———, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," in *Proc. 5th European Conf. Computer Vision*, vol. 1, pp. 893–908, 1998.

[232] ———, "A smoothing filter for Condensation," in *Proc. 5th European Conf. Computer Vision*, vol. 1, pp. 767–781, 1998.

[233] Kiyosi Itô, "On a formula concerning stochastic differentials," *Nagoya Math. J.*, vol. 3, pp. 55–65, 1951.

[234] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Automat. Contr.*, vol. 45, no. 5, pp. 910-927, 2000.

[235] K. Ito, "Approximation of the Zakai equation for nonlinear filtering," *SIAM J. Contr. Optim.*, vol. 34, pp. 620–634, 1996.

[236] T. Jaakkola, "Tutorial on variational approximation methods," in *Advanced Mean Field Methods: Theory and Practice*, D. Saad and M. Opper, Eds. Cambridge, MA: MIT Press, 2001.

[237] T. Jaakkola and M. Jordan, "Bayesian parameter estimation via variational methods," *Statist. Comput.*, vol. 10, pp. 25–37, 2000.

[238] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.

[239] F. V. Jensen, *An Introduction to Bayesian Networks*, New York: Springer-Verlag, 1996.

[240] ———, *Bayesian Networks and Decision Graphs*, Berlin: Springer, 2001.

[241] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[242] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense*, 1997.

[243] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Automat. Contr.*, vol. 45, no. 3, pp. 477–482, 2000.

[244] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 146–181, 1974.

[245] ———, "The innovations approach to detection and estimation theory," *Proc. IEEE*, vol. 58, pp. 680–695, 1970.

[246] ———, *Lecture on Wiener and Kalman Filtering*, New York: Springer-Verlag, 1981.

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

[248] G. Kallianpur, *Stochastic Filtering Theory*, New York: Springer-Verlag, 1980.

[249] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Trans. ASME, Ser. D, J. Basic Eng.*, vol. 83, pp. 95–107, 1961.

[250] R. E. Kalman, "A new approach to linear filtering and prediction problem" *Trans. ASME, Ser. D, J. Basic Eng.*, vol. 82, pp. 34–45, 1960.

[251] ———, "When is a linear control system optimal?" *Trans. ASME, Ser. D, J. Basic Eng.*, vol. 86, pp. 51–60, 1964.

[252] ———, "Mathematical description of linear dynamical systems" *SIAM J. Contr.*, vol. 1, pp. 152–192, 1963.

[253] ———, "New methods in Wiener filtering theory," in *Proc. 1st Symp. Engineering Applications of Random Function Theory and Probability* J. Bogdanoff and F. Kozin, Eds., pp. 270–388, New York: Wiley, 1963.

[254] K. Kanazawa, D. Koller, and S. Russel, "Stochastic simulation algorithms for dynamic probabilistic networks," in *Proc. 11th Conf. UAI*, pp. 346–351, 1995.

[255] S. A. Kassam and H. V. Poor, "Robust statistics for signal processing," *Proc. IEEE*, vol. 73, no. 3, pp. 433–481, 1985.

[256] J. K. Kim, "A note on approximate Bayesian bootstrap imputation," *Biometrika*, vol. 89, no. 2, pp. 470–477, 2002.

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

[258] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series," *J. Amer. Statist. Assoc.*, vol. 82, pp. 503–514, 1987.

[259] ———, "Monte Carlo filter and smoother for non-gaussian nonlinear state space models," *J. Comput. Graph. Statist.*, vol. 5, no. 1, pp. 1–25, 1996.

[260] ———, "Self-organising state space model," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1203–1215, 1998.

[261] G. Kitagawa and W. Gersch, *Smoothness Priors Analysis of Time Series*, Lecture Notes in Statistics, 116, New York: Springer-Verlag, 1996.

[262] D. Koller and R. Fratkina, "Using learning for approximation in stochastic processes," in *Proc. 15th Int. Conf. Machine Learning*, 1998, pp. 287–295.

[263] D. Koller and U. Lerner, "Sampling in Factored Dynamic Systems," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J.F.G. de Freitas, and N. Gordon, Eds., Springer-Verlag, 2001.

[264] A. N. Kolmogorov, "Stationary sequences in Hilbert spaces," *Bull. Math. Univ. Moscow* (in Russian), vol. 2, no. 6, p. 40, 1941.

[265] ———, "Interpolation and extrapolation of stationary random sequences," *Izv. Akad. Nauk USSR, Ser. Math.*, vol. 5, no. 5, pp. 3–14, 1941.

[266] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *J. Amer. Statist. Assoc.*, vol. 89, pp. 278–288, 1994.

[267] A. Kong, P. McCullagh, D. Nicolae, Z. Tan and X.-L. Meng, "A theory of statistical models for Monte Carlo integration," *J. Roy. Statist. Soc. Ser. B*, vol. 65, 2003.

[268] J. H. Kotecha and P. M. Djurić, "Gaussian sum particle filtering for dynamic state space models," in *Proc. ICASSP2001*, pp. 3465–3468, 2001.

[269] ———, "Sequential Monte Carlo sampling detector for Rayleigh fast-fading channels," in *Proc. ICASSP2000*, vol. 1, pp. 61–64, 2000.

[270] S. C. Kramer, "The Bayesian approach to recursive state estimation: Implementation and application," Ph.D. thesis, UC San Diego, 1985.

[271] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica*, vol. 24, pp. 789–901, 1988.

[272] ———, "Bayesian parameter estimation," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 217–222, 1988.

[273] A. J. Krener, "Kalman-Bucy and minimax filtering," *IEEE Trans. Automat. Contr.*, vol. 25, pp. 291–292, 1980.

[274] R. Kress, *Linear Integral Equations* (2nd ed.), Berlin: Springer-Verlag, 1999.

[275] V. Krishnan, *Nonlinear Filtering and Smoothing: An Introduction to Martingales, Stochastic Integrals and Estimation*, New York: Wiley, 1984.

[276] R. Kulhavý, "Recursive nonlinear estimation: A geometric approach," *Automatica*, vol. 26, no. 3, pp. 545–555, 1990.

[277] ———, "Recursive nonlinear estimation: Geometry of a space of posterior densities," *Automatica*, vol. 28, no. 2, pp. 313–323, 1992.

[278] ———, *Recursive Nonlinear Estimation: A Geometric Approach.* Lecture Notes in Control and Information Sciences, 216, London: Springer-Verlag, 1996.

[279] ———, "On extension of information geometry of parameter estimation to state estimation," in *Mathematical Theory of Networks and Systems*, A. Beghi, L. Finesso and G. Picci (Eds), pp. 827–830, 1998.

[280] ———, "Quo vadis, Bayesian identification?" *Int. J. Adaptive Control and Signal Processing*, vol. 13, pp. 469–485, 1999.

[281] ———, "Bayesian smoothing and information geometry," in *Learning Theory and Practice*, J. Suykens Ed, IOS Press, 2003.

[282] H. J. Kushner, "On the differential equations satisfied by conditional probability densities of Markov processes with applications," *SIAM J. Contr.*, vol. 2, pp. 106–119, 1965.

[283] ———, "Approximations to optimal nonlinear filters," *IEEE Trans. Automat. Contr.*, vol. 12, pp. 546–556, Oct. 1967.

[284] ———, "Dynamical equations for optimal nonlinear filtering," *J. Differential Equations*, vol. 3, pp. 179–190, 1967.

[285] ———, *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations*, New York: Academic Press, 1977.

[286] H. J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, New York: Springer-Verlag, 1992.

[287] H. Kushner and A. S. Budhiraja, "A nonlinear filtering algorithm based on an approximation of the conditional distribution," *IEEE Trans. Automat. Contr.*, vol. 45, no. 3, pp. 580–585, March 2000.

[288] C. Kwok, D. Fox, and M. Meila, "Real-time particle filter," in *Adv. Neural Inform. Process. Syst. 15*, Cambridge, MA: MIT Press, 2003.

[289] D. G. Lainiotis, "Optimal nonlinear estimation," *Int. J. Contr.*, vol. 14, no. 6, pp. 1137–1148, 1971.

[290] J-R. Larocque, J. P. Reilly, and W. Ng, "Particle filters for tracking an unknown number of sources," *IEEE Trans. Signal Processing*, vol. 50, no. 12, pp. 2926–2937, 2002.

[291] D. S. Lee and N. K. Chia, "A particle algorithm for sequential Bayesian parameter estimation and model selection," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 326–336, Feb. 2002.

[292] F. LeGland, "Monte-Carlo methods in nonlinear filtering," in *Proc. IEEE Conf. Decision and Control*, pp. 31–32, 1984.

[293] ———, "Stability and approximation of nonlinear filters: An information theoretic approach," in *Proc. 38th Conf. Decision and Control*, pp. 1889–1894, 1999.

[294] F. LeGland, and N. Oudjane, "Stability and uniform approximation of nonlinear filters using the Hilbert metric, and application to particle filters," in *Proc. 39th Conf. Decision and Control*, pp. 1585-1590, 2000.

[295] P. L'Ecuyer and C. Lemieux, "Variance reduction via lattice rules," *Management Sci.*, vol. 46, pp. 1214–1235, 2000.

[296] C. Lemieux and P. L'Ecuyer, "Using lattice rules for variance reduction in simulation," in *Proc. 2000 Winter Simulation Conf.*, 509–516, 2000.

[297] N. Levinson, "The Wiener rms (root-mean-square) error criterion in filter design and prediction," *J. Math. Phys.*, vol. 25, pp. 261–278, Jan. 1947.

[298] F. Liang, "Dynamically weighted importance sampling in Monte Carlo computation" *J. Amer. Statist. Assoc.*, vol. 97, 2002.

[299] J. G. Liao, "Variance reduction in Gibbs sampler using quasi random numbers," *J. Comput. Graph. Statist.*, vol. 7, no. 3, pp. 253–266, 1998.

[300] T. M. Liggett, *Interacting Particle Systems*, Springer-Verlag, 1985.

[301] T-T. Lin and S. S. Yau, "Bayesian approach to the optimization of adaptive systems," *IEEE Trans. Syst. Sci. Cybern.*, vol. 3, no. 2, pp. 77–85.

[302] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, "Comparison of EKF, pseudomeasurement and particle filters for a bearings-only target tracking problem," in *Proc. SPIE on Signal and Data Processing of Small Targets*, vol. 4728, 2002.

[303] J. S. Liu and R. Chen, "Blind deconvolution via sequential imputation," *J. Amer. Statist. Assoc.*, vol. 90, pp. 567–576, 1995.

[304] ———, "Sequential Monte Carlo methods for dynamical systems," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1032–1044, 1998.

[305] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statist. Comput.*, vol. 6, pp. 113–119, 1996.

[306] ———, *Monte Carlo Strategies in Scientific Computing*, Berlin: Springer, 2001.

[307] J. S. Liu, R. Chen, and W. H. Wong, "Rejection control and sequential importance sampling," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1022–1031, 1998.

[308] J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling with resampling," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[309] J. S. Liu, F. Liang, and W. H. Wong, "A theory for dynamic weighting in Monte Carlo computation," *J. Amer. Statist. Assoc.*, vol. 96, pp 561–573, 2001.

[310] 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. J. Gordon, Eds. New York: Springer, 2001.

[311] S. V. Lototsky, and B. L. Rozovskii, "Recursive nonlinear filter for a continuous-discrete time model: Separation of parameters and observations," *IEEE Trans. Automat. Contr.*, vol. 43, no. 8, pp. 1154–1158, 1996.

[312] S. V. Lototsky, R. Mikulevicius, and B. L. Rozovskii, "Nonlinear filtering revisited: A spectral approach," *SIAM J. Contr. Optim.*, vol. 35, pp. 435–461, 1997.

[313] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," in *Proc. Int. Conf. Comput. Vision*, 1999, pp. 572–578.

[314] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," Tech. Rep., Dept. Eng. Sci., Univ. Oxford, 2000.

[315] S. N. MacEachern, M. Clyde, and J. S. Liu, "Sequential importance sampling for nonparametric Bayes models: The next generation," *Canadian J. Statist.*, vol. 27, pp. 251–267, 1999.

[316] D. J. C. MacKay, "Bayesian methods for adaptive models," Ph.D. thesis, Dept. Computation and Neural Systems, Caltech, 1992. Available on line http://wol.ra.phy.cam.ac.uk/mackay/.

[317] ———, "Probable networks and plausible predictions - A review of practical Bayesian methods for supervised neural networks," *Network*, vol. 6, pp. 469–505, 1995.

[318] ———, "Introduction to Monte Carlo methods," in *Learning in Graphical Models*, M. Jordan Ed., Kluwer Academic Publishers, 1998.

[319] ———, "Choice of basis for Laplace approximation," *Machine Learning*, vol. 33, no. 1, pp. 77–86, 1998.

[320] D. M. Malakoff, "Bayes offers 'new' way to make sense of numbers," *Science*, vol. 286, pp. 1460–1464, 1999.

[321] B. Manly, *Randomization, Bootstrap and Monte Carlo Methods in Biology*, 2nd ed., CRC Press, 1997.

[322] Z. Mark and Y. Baram, "The bias-variance dilemma of the Monte Carlo method," in *Artificial Neural Networks (ICANN2001)*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Berlin: Springer-Verlag, 2001.

[323] ———, "Manifold stochastic dynamics for Bayesian learning," *Neural Comput.*, vol. 13, pp. 2549–2572, 2001.

[324] A. Marshall, "The use of multi-stage sampling schemes in Monte Carlo computations," in *Symposium on Monte Carlo Methods*, M. Meyer Ed. New York: Wiley, pp. 123–140, 1956.

[325] S. Maskell, Orton, and N. Gordon, "Efficient inference for conditionally Gaussian Markov random fields", Tech. Rep. CUED/F-INFENG/TR439, Cambridge Univ., August 2002.

[326] S. McGinnity and G. W. Irwin, "Manoeuvring target tracking using a multiple-model bootstrap filter" in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. J. Gordon, Eds. New York: Springer, 2001.

[327] I. W. McKeague and W. Wefelmeyer, "Markov Chain Monte Carlo and Rao-Blackwellization," *Statistical Planning and Inference*, vol. 85, pp. 171–182, 2000.

[328] X.-L. Meng and D. A. van Dyk, "Seeking efficient data augmentation schemes via conditional and marginal augmentation," *Biometrika*, vol. 86, pp. 301–320, 1999.

[329] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1091, 1953.

[330] N. Metropolis and S. Ulam, "The Monte Carlo method," *J. Amer. Statist. Assoc.*, vol. 44, pp. 335–341, 1949.

[331] J. Miguez and P. M. Djuric, "Blind equalization by sequential importance sampling," in *Proc. IEEE Symp. Circuit Syst., ISCAS'02*, vol. 1, pp. 845–848, 2002.

[332] A. Milstein, J. Sánchez, and E. T. Williamson, "Robust global localization using clustered particle filtering," in *Proc. 8th AAAI*, 2002.

[333] T. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. thesis, Department of Computer Science and Electrical Engineering, MIT, 2001. Available on line http://www.stat.cmu.edu/~minka/.

[334] ———, "Expectation propagation for approximate Bayesian inference," in *Proc. UAI'2001*, 2001.

[335] ———, "Using lower bounds to approximate integrals," Tech. Rep., Dept. Statist., CMU, 2001.

[336] A. Mira, J. Møller, and G. Roberts, "Perfect slice samplers," *J. Roy. Statist. Soc., Ser. B*, vol. 63, pp. 593–606, 2001.

[337] A. W. Moore, C. G. Atkeson, and S. A. Schaal, “Locally weighted learning for control,” *Artificial Intell. Rev.*, vol. 11, pp. 75–113, 1997.

[338] R. Morales-Menendez, N. de Freitas, and D. Poole, “Real-time monitoring of complex industrial processes with particle filters,” in *Adv. Neural Info. Process. Syst. 15*, Cambridge, MA: MIT Press, 2003.

[339] D. R. Morrell and W. C. Stirling, “Set-valued filtering and smoothing,” *IEEE Trans. Syst. Man Cybern.*, vol. 21, pp. 184–193, 1991.

[340] K. Mosegaard and M. Sambridge, “Monte Carlo analysis of inverse problems,” *Inverse Problems*, vol. 18, pp. 29–54, 2002.

[341] P. Müller, “Monte Carlo integration in general dynamic models,” *Contemporary Mathematics*, vol. 115, pp. 145–163, 1991.

[342] ———, “Posterior integration in dynamic models,” *Comput. Sci. Statist.*, vol. 24, pp. 318–324, 1992.

[343] K. Murphy, “Switching Kalman filter,” Tech. Rep., Dept. Comput. Sci., UC Berkeley, 1998.

[344] ———, “Dynamic Bayesian networks: Representation, inference and learning,” Ph.D. thesis, Dept. Comput. Sci., UC Berkeley, 2002. Available on line http://www.ai.mit.edu/~murphyk/papers.html.

[345] C. Musso, N. Oudjane, and F. LeGland, “Improving regularised particle filters,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. J. Gordon, Eds. New York: Springer, 2001.

[346] R. Neal, *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics, 118, Berlin: Springer, 1996.

[347] ———, “An improved acceptance procedure for the hybrid Monte Carlo,” *J. Comput. Phys.*, vol. 111, pp. 194–203, 1994.

[348] ———, “Sampling from multimodal distributions using tempered transitions,” *Statist. Comput.*, vol. 6, pp. 353–366, 1996.

[349] ———, “Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation,” in *Learning in Graphical Models*, M. I. Jordan, Ed, pp. 205–228, Kluwer Academic Publishers, 1998.

[350] ———, “Annealed importance sampling,” *Statist. Comput.*, vol. 11, pp. 125–139, 2001.

[351] ———, “Slice sampling (with discussions),” *Ann. Statist.*, vol. 31, no. 3, June 2003.

[352] A. T. Nelson, “Nonlinear estimation and modeling of noisy time series by dual Kalman filtering methods,” Ph.D. thesis, Dept. Elect. Comput. Engin., Oregon Graduate Institute, 2000.

[353] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia, PA: SIAM, 1992.

[354] H. Niederreiter and J. Spanier, Eds. *Monte Carlo and Quasi-Monte Carlo Methods*, Berlin: Springer-Verlag, 2000.

[355] M. Norgaard, N. Poulsen, and O. Ravn, “Adavances in derivative-free state estimation for nonlinear systems,” Tech. Rep., Technical Univ. Denmark, 2000. Available on-line http://www.imm.dtu.dk/nkp/.

[356] B. North, A. Blake, M. Isard, and J. Rittscher, “Learning and classification of complex dynamics,” *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 22, no. 9, pp. 1016–1034, Sept. 2000.

[357] J.P. Norton and G. V. Veres, “Improvement of the particle filter by better choice of the predicted sample set,” in *Proc. 15th IFAC*, pp. 904–909, 2002.

[358] G. W. Oehlert, “Faster adaptive importance sampling in low dimensions,” *J. Comput. Graph. Statist.*, vol. 7, pp. 158–174, 1998.

[359] M.-S. Oh, “Monte Carlo integration via importance sampling: Dimensionality effect and an adaptive algorithm,” *Contemporary Mathematics*, vol. 115, pp. 165–187, 1991.

[360] B. Oksendal, *Stochastic Differential Equations* (5th ed.), Berlin: Springer, 1998.

[361] D. Ormoneit, C. Lemieux and D. Fleet, “Lattice particle filters,” in *Proc. UAI2001*, 2001, pp. 395–402.

[362] M. Orton and W. Fitzgerald, “A Bayesian approach to tracking multiple targets using sensor arrays and particle filters,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 216–223, Feb. 2002.

[363] M. Ostland and B. Yu, “Exploring quasi Monte Carlo for marginal density approximation,” *Statist. Comput.*, vol. 7, pp. 217–228, 1997.

[364] N. Oudjane and C. Musso, “Progressive correction for reguarlized particle filters,” in *Proc. 3rd Int. Conf. Inform. Fusion*, 2000, Paris, ThB2-2.

[365] N. Oudjane, “Stabilité et approximations particulaires en filtrage non-linéaire. Application au pistage,” Ph.D. thesis (in French), Université de Rennes, 2000.

[366] V. Peterka, “Bayesian approach to system identification,” in *Trends and Progress in System Identification*, pp. 239–304, Pergamon Press, 1981.

[367] ———, “Bayesian system identification,” *Automatica*, vol. 17, pp. 41–53, 1981.

[368] V. Philomin, R. Duraiswami, and L. Davis, “Quasi-random sampling for condesation,” in *Proc. Euro. Conf. Comp. Vis.*, vol. II, pp. 134–149, 2000.

[369] M. Pitt and N. Shephard, “A fixed lag auxillary particle filter with deterministic sampling rules,” unpublished paper, 1998.

[370] ———, “Filtering via simulation: Auxillary particle filter,” *J. Amer. Statist. Assoc.*, vol. 94, pp. 590–599, 1999.

[371] ———, “Auxiliary variable based particle filters,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[372] A. Pole, M. West, and P. J. Harrison, *Applied Bayesian Forecasting and Time Series Analysis.* New York: Chapman-Hall, 1994.

[373] ———, “Non-normal and non-linear dynamic Bayesian modelling,” in *Bayesian Analysis of Time Series and Dynamic Models*, J. C. Spall Ed., pp. 167–198, New York: Marcel Dekker, 1988.

[374] N. G. Polson, B. P. Carlin, and D. S. Stoffer, “A Monte-Carlo approach to non-normal and nonlinear state-space modelling,” *J. Amer. Statist. Assoc.*, vol. 87, pp. 493–500, 1992.

[375] S. J. Press, *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications* (2nd ed.), New York: Wiley, 2003.

[376] W. H. Press and G. R. Farrar, “Recursive stratified sampling for multidimensional Monte Carlo integration,” *Computers in Physics*, vol. 4, pp. 190–195, 1990.

[377] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed., Cambridge Univ. Press, 1997.

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

[379] L. R. Rabiner, “A tutorial on hidden Markov models and selected applictions in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[380] L. R. Rabiner and B.-H. Juang, “An introduction to hidden Markov models,” *IEEE Acoust., Speech, Signal Processing Mag.*, pp. 4–16, Jan. 1986.

[381] ———, *Fundamentals of Speech Recognition*, Englewood Cliffs, NJ: Prentice Hall, 1993.

[382] C. E. Rasmussen and Z. Ghahramani, “Bayesian Monte Carlo,” in *Adv. Neural Inform. Process. Syst. 15*, Cambridge, MA: MIT Press, 2003.

[383] H. E. Rauch, “Solutions to linear smoothing problem,” *IEEE Trans. Automat. Contr.*, vol. 8, pp. 371–372, 1963.

[384] H. E. Rauch, T. Tung, and T. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA J.*, vol. 3, pp. 1445–1450, 1965.

[385] I. B. Rhodes, “A tutorial introduction to estimation and filtering,” *IEEE Trans. Automat. Contr.*, vol. 16, pp. 688–707, 1971.

[386] B. Ripley, *Stochastic Simulation*, New York: Wiley, 1987.

[387] H. Risken, *The Fokker-Planck Equation* (2nd ed.), Berlin: Springer-Verlag, 1989.

[388] C. P. Robert, *The Bayesian Choice: A Decision-Theoretic Motivation* (2nd ed.), New York: Springer, 2001.

[389] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Berlin: Springer, 1999.

[390] C. P. Robert, T. Rydén, and D. M. Titterington, “Bayesian inference in hidden Markov models through the reverse jump Markov chain Monte Carlo method,” *J. Roy. Statist. Soc., Ser. B*, vol. 62, pp. 57–75, 2000.

[391] C. P. Robert, C. Celeux, and J. Diebolt, “Bayesian estimation of hidden Markov chains: A stochastic implementation,” *Statist. Probab. Lett.*, vol. 16, pp. 77–83, 1993.

[392] G. O. Roberts and J. S. Rosenthal, “Markov chain Monte Carlo: Some practical implications of theoretical results,” *Can. J. Stat.*, vol. 25, pp. 5–31, 1998.

[393] M. N. Rosenbluth and A. W. Rosenbluth, “Monte Carlo calculation of the average extension of molecular chains,” *J. Chem. Phys.*, vol. 23, pp. 356–359, 1955.

[394] D. B. Rubin, “Multiple imputations in sample survey: A phenomeonological Bayesian approach to nonresponse,” in *Proc. Sur-*

vey Res. Meth. Sect. Am. Statist. Assoc., Washington DC: American Statistical Association, pp. 20–28, 1978.

[395] ———, *Multiple Imputation for Nonresponse in Surveys*, New York: Wiley, 1987.

[396] ———, "Comment on 'The calculation of posterior distributions by data augmentation' by M. A. Tanner and W. H. Wong," *J. Amer. Statist. Assoc.*, vol. 82, pp. 543–546, 1987.

[397] ———, "Using the SIR algorithm to simulate posterior distributions," in *Bayesian Statistics 3*, J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, Eds. pp. 395–402, Oxford Univ. Press, 1988.

[398] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Proc. CVPR 2001*, vol. II, pp. 786–793, 2001.

[399] W. J. Runggaldier and F. Spizzichino, "Finite dimensionality in discrete time nonlinear filtering from a Bayesian statistics viewpoint," in *Stochastic Modeling and Filtering*, A. German Ed., Lecture Notes in Control and Information Science, 91, pp. 161–184, Berlin: Springer, 1987.

[400] J. S. Rustagi, *Variational Methods in Statistics*, New York: Academic Press, 1976.

[401] D. Saad and M. Opper, Eds. *Advanced Mean Field Method — Theory and Practice*, Cambridge, MA: MIT Press, 2001.

[402] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, McGraw-Hill, 1973.

[403] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Mag.*, vol. 11, pp. 18–60, 1994.

[404] M. Schetzen, "Nonlinear system modeling based on the Wiener theory," *Proc. IEEE*, vol. 69, pp. 1557–1572, 1981.

[405] B. Schölkopf, and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge, MA: MIT Press, 2002.

[406] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proc. 2001 IEEE Int. Conf. Robotics & Automation*, pp. 1665-1670, 2001.

[407] L. Shan and P. C. Doerschuk, "Performance bounds for nonlinear filters," *IEEE Trans. Aerosp. Elect. Syst.*, vol. 33, no. 1, pp. 316–318, 1997.

[408] J. Shao and D. Tu. *The Jackknife and the Bootstrap.* Springer, 1996.

[409] N. Shephard and M. K. Smith, "Likelihood analysis of non-Gaussian measurement time series," *Biometrika*, vol. 84, pp. 653–667, 1997.

[410] M. Šimandl, J. Královec, and P. Tichavský, "Filtering, predictive, and smoothing Cramér-Rao bounds for discrete-time nonlinear dynamic systems," *Automatica*, vol. 37, pp. 1703–1716, 2001.

[411] M. Šimandl and O. Straka, "Nonlinear estimation by particle filters and Cramér-Rao bound," in *Proc. 15th IFAC'2002*, 2002.

[412] I. N. Sinitsyn, "Ill-posed problems of on-line conditionally optimal filtering," in *Ill-Posed Problems in Natural Sciences*, A. Tikhonov, Ed., VSP/TVP, The Netherlands, 1992.

[413] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration*, Oxford: Clarendon Press, 1994.

[414] A. F. M. Smith and A. E. Gelfand, "Bayesian statistics without tears: A sampling-resampling perspective," *Am. Stat.*, vol. 46, no. 4, pp. 84–88, 1992.

[415] P. J. Smith, M. Shafi, and H. Gao, "Quick simulation: A review of importance sampling techniques in communications systems," *IEEE J. Selected Areas Commu.*, vol. 15, no. 4, pp. 597–613, 1997.

[416] A. F. M. Smith and G. Roberts, "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods," *J. Roy. Statist. Soc., Ser. B*, vol. 55, no. 1, pp. 3–23, 1993.

[417] R. L. Smith, "The hit-and-run sampler: A globally reaching Markov chain sampler for generating arbitrary multivariate distributions," in *Proc. 28th Conf. Winter Simulation*, pp. 260–264, New York: ACM Press, 1996.

[418] K. Sobczyk, *Stochastic Differential Equations: With Applications to Physics and Engineering*, Kluwer Academic Publishers, 1991.

[419] Solomon, H. "Buffon needle problem, extensions, and estimation of π,' in *Geometric Probability*, chap. 1, pp. 1–24, Philadelphia, PA: SIAM, 1978.

[420] H. W. Sorenson and A. R. Stubberud, "Nonlinear filtering by approximation of the a posteriori density," *Int. J. Contr.*, vol. 8, pp. 33–51, 1968.

[421] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[422] H. W. Sorenson, "On the development of practical nonlinear filters," *Inform. Sci.*, vol. 7, pp. 253–270, 1974.

[423] ———, Ed. *Kalman Filtering: Theory and Application*, IEEE Press, 1985.

[424] ———, "Recursive estimation for nonlinear dynamic systems," in *Bayesian Analysis of Time Series and Dynamic Models*, J. C. Spall, Ed., pp. 127–165, New York: Marcel Dekker, 1988.

[425] J. Spanier and E. H. Maize, "Quasi-random methods for estimating integrals using relatively small samples," *SIAM Rev.*, vol. 33, no. 1, pp. 18–44, 1994.

[426] J. Spragins, "A note on the iterative application of Bayes' rule. *IEEE Trans. Informa. Theory*, vol. 11, no. 4, pp. 544–549, 1965.

[427] K. Sprinivasan, "State estimation by orthogonal expansion of probability distributions," *IEEE Trans. Automat. Contr.*, vol. 15, no. 1, pp. 3–10, 1970.

[428] P. Stavropoulos and D. M. Titterington, "Improved particle filters and smoothing," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, N. J. Gordon, Eds. Berlin: Springer Verlag, 2001.

[429] J. C. Stiller and G. Radons, "Online estimation of hidden Markov models," *IEEE Signal Process. Lett.*, vol. 6, no. 8, pp. 213–215, 1999.

[430] R. L. Stratonovich, "Conditional Markov processes," *Theor. Prob. Appl.* (USSR), vol. 5, pp. 156–178, 1960.

[431] ———, *Conditional Markov Processes and Their Application to the Theory of Optimal Control*, New York: Elsevier, 1968.

[432] G. Storivik, "Particle filters for state-space models with the presence of unknown statistic parameters," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 281–289, Feb. 2002.

[433] V. B. Svetnik, "Applying the Monte Carlo method for optimum estimation in systems with random disturbances," *Automation and Remote Control*, vol. 47, pp. 818–825, 1986.

[434] P. Swerling, "A proposed stagewise differential correction procedure for satellite tracking and prediction," Tech. Rep. P-1292, Rand Corporation, 1958.

[435] ———, "Modern state estimation methods from the viewpoint of the method of least squares," *IEEE Trans. Automat. Contr.*, vol. 16, pp. 707–720, 1971.

[436] H. Tanizaki and R. S. Mariano, "Prediction, filtering and smoothing in non-linear and non-normal cases using Monte Carlo integration," *J. Appl. Econometrics*, vol. 9, no. 2, pp. 163–179, 1994.

[437] ———, "Nonlinear filters based on Taylor series expansion," *Commu. Statist. Theory and Methods*, vol. 25, no. 6, pp. 1261–1282, 1996.

[438] ———, "Nonlinear and non-Gaussian state-space modeling with Monte Carlo integration," *J. Econometrics*, vol. 83, no. 1/2, pp. 263–290, 1998.

[439] H. Tanizaki, *Nonlinear Filters: Estimation and Applications*, 2nd ed., New York: Springer-Verlag, 1996.

[440] ———, "Nonlinear and non-Gaussian state estimation: A quasi-optimal estimator," *Commu. Statist. Theory and Methods*, vol. 29, no. 12, 1998.

[441] ———, "On the nonlinear and non-normal filter using rejection sampling," *IEEE Trans. Automat. Contr.*, vol. 44, no. 2, pp. 314–319, 1999.

[442] ———, "nonlinear and non-normal filter using importance sampling: Antithetic Monte-Carlo integration," *Commu. Statist. Simu. and Comput.*, vol. 28, no. 2, pp. 463–486, 1999.

[443] ———, "Nonlinear and non-Gaussian state-space modeling with Monte Carlo techniques: A survey and comparative study," in *Handbook of Statistics*, C. R. Rao and D. N. Shanbhag, Eds., North-Holland, 2000.

[444] ———, "Nonlinear and non-Gaussian state space modeling using sampling techniques," *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 63–81, 2001.

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

[446] M. A. Tanner, *Tools for Statistical Inference: Methods for Exploration of Posterior Distributions and Likelihood Functions*, 3rd ed., Berlin: Springer Verlag, 1996.

[447] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, May 2001.

[448] S. Thrun, J. Langford, and V. Verma, "Risk sensitive particle filters," in *Adv. Neural Inform. Process. Syst. 14*, Cambridge, MA: MIT Press, 2002.

[449] S. Thrun, J. Langford, and D. Fox, "Monte Carlo hidden Markov models: Learning non-parameteric models of partially observable stochastic processes," in *Proc. Int. Conf. Machine Learning*, 1999.

[450] S. Thrun, "Particle filters in robotics," in *Proc. UAI02*, 2002.

[451] P. Tichavský, C. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Processing*, vol. 46, no. 5, pp. 1386–1396, 1998.

[452] L. Tierney, "Markov chains for exploring posterior distributions (with discussion)," *Ann. Statist.*, vol. 22, pp. 1701–1762, 1994.

[453] L. Tierney, R. E. Kass, and J. B. Kadane, "Approximate marginal densities of nonlinear functions," *Biometrika*, vol. 76, pp. 425–433, 1989.

[454] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[455] E. Tito, M. Vellasco, and M. Pacheco, "Genetic particle filter: An evolutionary perspective of SMC methods," Paper preprint, available on line http://www.ica.ele.puc-rio.br/cursos/download/TAIC-GPFilter.pdf

[456] P. Torma and C. Szepesvári, "LS-N-IPS: An improvement of particle filters by means of local search," in *Proc. Nonlinear Control Systems*, 2001.

[457] ———, "Combining local search, neural networks and particle filters to achieve fast and realiable contour tracking," Paper preprint, 2002. Available on line http://www.mindmaker.hu/~szepes/research/onlinepubs.htm

[458] ———, "Sequential importance sampling for visual tracking reconsidered," in *Proc. 9th Workshop AI and Statistics*, 2003.

[459] R. van der Merwe, J. F. G. de Freitas, A. Doucet, and E. Wan, "The unscented particle filter," Tech. Rep. CUED/F-INFENG/TR 380, Cambridge Univ. Engineering Dept., 2000. Also in *Adv. Neural Inform. Process. Syst. 13*, Cambridge, MA: MIT Press, 2001.

[460] R. van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter estimation," in *Proc. ICASSP'01*, vol. 6, pp. 3461–3464.

[461] D. A. van Dyk and X.-L. Meng, "The art of data augmentation (with discussion)," *J. Comput. Graph. Statist.*, vol. 10, pp. 1–111, 2001.

[462] H. L. Van Trees, *Detection, Estimation and Modulation Theory*, New York: Wiley, 1968.

[463] V. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.

[464] J. Vermaak, M. Gangnet, A. Blake, and P. Pérez, "Sequential Monte Carlo fusion of sound and vision for speaker tracking," in *Proc. 8th IEEE Int. Conf. Comput. Vision, ICCV'01*, 2001, vol. 1, pp. 741–746.

[465] J. Vermaak and A. Blake, "Nonlinear filtering for speaker tracking in noisy and reverberant environment," in *Proc. ICASSP01*, 2001, vol. 5, pp. 3021–3024.

[466] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, "Particle methods for Bayesian modelling and enhancement of speech signals," *IEEE Trans. Audio Speech Processing*, vol. 10, no. 3, pp. 173–185, March 2002.

[467] J. Vermaak, "Bayesian modelling and enhancement of speech signals," Ph.D. thesis, Cambridge Univ., 2000. Available on line at http://svr-www.eng.cam.ac.uk/~jv211/publications.html.

[468] J. Vermaak, N. D. Lawrence, and P. Pérez, "Variational inference for visual tracking," paper preprint, 2002.

[469] P. Vidoni, "Exponential family state space models based on a conjugate latent process," *J. Roy. Statist. Soc., Ser. B*, vol. 61, pp. 213–221, 1999.

[470] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Informa. Theory*, vol. 13, pp. 260–269, 1967.

[471] N. Vlassis, B. Terwijn, and B. Kröse, "Auxiliary particle filter robot localization from high-dimensional sensor observations," in *Proc. 2002 IEEE Int. Conf. Robot. Automat.*, pp. 7–12, 2002.

[472] J. von Neurmann, "Various techniques used in connection with random digits," *National Bureau of Standards Applied Mathematics*, vol. 12, pp. 36–38, 1959.

[473] E. Wan and A. Nelson, "Dual extended Kalman filter methods," in *Kalman Filtering and Neural Networks* (chap. 5), S. Haykin Ed. New York: Wiley, 2001.

[474] E. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks* (chap. 7), S. Haykin Ed. New York: Wiley, 2001.

[475] A. H. Wang and R. L. Klein, "Optimal quadrature formula nonlinear estimators," *Inform. Sci.*, vol. 16, pp. 169–184, 1978.

[476] X. Wang, R. Chen, and D. Guo, "Delayed-pilot sampling for mixture Kalman filter with application in fading channels," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 241–253, Feb. 2002.

[477] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo signal processing for wireless communications," *J. VLSI Signal Processing*, vol. 30, no. 1–3, pp. 89-105, 2002.

[478] D. B. Ward and R. C. Williamson, "Particle filter beamforming for acoustic source localization in a reverberant environment," in *Proc. ICASSP'2002*, vol. II, pp. 1777–1780, 2002.

[479] D. B. Ward, E. A. Lehmann, and R. C. Williamson,"Particle filtering algorithms for acoustic source localization," *IEEE Trans. Speech Audio Processing*, 2003 (to appear).

[480] M. West, "Robust sequential approximate Bayesian estimation," *J. Roy. Statist. Soc., Ser. B*, vol. 43, pp. 157–166, 1981.

[481] ———, "Mixture models, Monte Carlo, Bayesian updating and dynamic models," *Comput. Sci. Statist.*, vol. 24, pp. 325–333, 1992.

[482] ———, "Modelling with mixtures," in *Bayesian Statistics 4*, London: Clarendon Press, 1992.

[483] M. West, P. J. Harrison, and H. S. Migon, "Dynamic generalised linear models and Bayesian forecasting (with discussion)," *J. Amer. Statist. Assoc.*, vol. 80, pp. 73–97, 1985.

[484] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, 2nd ed., New York: Springer, 1997.

[485] B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," in *IRE Wescon Conv. Record*, Pt. 4, pp. 96–104, 1960.

[486] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prenitice-Hall, 1985.

[487] N. Wiener and E. Hopf, "On a class of singular integral equations," in *Proc. Prussian Acad. Math. – Phys. Ser.*, p. 696, 1931.

[488] N. Wiener, *Extrapolation, Interpolation and Smoothing of Time Series, with Engineering Applications*, New York: Wiley, 1949. Originally appears in 1942 as a classified National Defense Research Council Report. Also published under the title *Time Series Analysis* by MIT Press.

[489] C. K. I. Williams, "Prediction with Gaussian processes: From linear regression to linear prediction and beyond," in *Learning in Graphical Models*, M. Jordan Ed., Kluwer Academic Publishers, 1998.

[490] D. B. Wilson, "Annotated bibliography of perfectly random sampling with Markov chain," in *Microsurveys in Discrete Probability*, D. Aldous and J. Propp Eds., pp. 209–220, Providence: American Math. Society, 1998.

[491] D. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Comput.*, vol. 8, pp. 1341–1390, 1996.

[492] ———, "The existence of a priori distinctions between learning algorithms," *Neural Comput.*, vol. 8, pp. 1391–1420, 1996.

[493] ———, "No free lunch theorems for optimization," *IEEE Trans. Evolu. Comput.*, vol. 1, pp. 77–82, 1997.

[494] W. H. Wong and F. Liang, "Dynamic importance weighting in Monte Carlo and optimization," *Proc. Natl. Acad. Sci.*, vol. 94, pp. 14220–14224, 1997.

[495] W. S. Wong, "New classes of finite-dimensional nonlinear filters," *Syst. Contr. Lett.*, vol. 3, pp. 155–164, 1983.

[496] W. M. Wonham, "Some applications of stochastic differential equations to optimal nonlinear filtering," *SIAM J. Contr.*, vol. 2, pp. 347–369, 1965.

[497] ———, "Random differential equations in control theory," in *Probabilistic Methods in Applied Mathematics*, A. T. Bharucha-Reid Ed., vol. 2, pp. 131–212, New York: Academic Press, 1970.

[498] H. Wozniakowski, "Average case complexity of mulitvariate integration," *Bull. Amer. Math. Soc.*, vol. 24, pp. 185–194, 1991.

[499] Z. Yang and X. Wang, "A sequential Monte Caro blind receiver for OFDM systems in frequency-selective fading channels," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 271–280, Feb. 2002.

[500] K. Yao and S. Nakamura, "Sequential noise compensation by sequential Monte Carlo method," in *Adv. Neural Inform. Process. Syst. 14*, Cambridge, MA: MIT Press, 2002.

[501] M. Yeddanapudi, Y. Bar-Shalom, and K. R. Pattipati, "IMM estimation for multitarget-multisensor air traffic surveillance," *Proc. IEEE*, vol. 85, no. 1, 80–94, 1997.

[502] L. A. Zadeh and J. R. Ragazzini, "An extension of Wiener's theory of prediction," *J. Appl. Phys.*, vol. 21, pp. 644–655, 1950.

[503] L. A. Zadeh, “Optimum nonlinear filters,” *J. Appl. Phys.*, vol. 24, pp. 396–404, 1953.

[504] M. Zakai and J. Ziv, “Lower and upper bounds on the optimal filtering error of certain diffusion processes,” *IEEE Trans. Inform. Theory*, vol. 18, no. 3, pp. 325–331, 1972.

[505] M. Zakai, “On the optimal filtering of diffusion processes,” *Zeitschrift für Wahrscheinlichkeitstheorie und verwande Gebiete*, vol. 11, no. 3, pp. 230–243, 1969.

[506] V. S. Zaritskii, V. B. Svetnik, and L. I. Shimelevich, “Monte Carlo technique in problems of optimal data processing,” *Autom. Remote Control*, vol. 12, pp. 95-103, 1975.

[507] J. Zhang and P. M. Djuric, “Joint estimation and decoding of space-time Trellis codes,” *EURASIP J. Appl. Signal Processing*, no. 3, pp. 305–315, March 2002.

[508] H. Zhu and R. Rohwer, “Bayesian regression filters and the issue of priors,” *Neural Comput. Appl.*, vol. 4, pp. 130–142, 1996.