ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Sequential Monte Carlo: A Unified Review

Adrian G. Wills[1] and Thomas B. Schön[2]

[1]School of Engineering, University of Newcastle, Callaghan, New South Wales, Australia; email: adrian.wills@newcastle.edu.au

[2]Department of Information Technology, Uppsala University, Uppsala, Sweden

Annu. Rev. Control Robot. Auton. Syst. 2023. 6:159–82

First published as a Review in Advance on January 9, 2023

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-042920-015119

Keywords

sequential Monte Carlo, particle filter, nonlinear state-space model, state estimation, system identification

Abstract

Sequential Monte Carlo methods—also known as particle filters—offer approximate solutions to filtering problems for nonlinear state-space systems. These filtering problems are notoriously difficult to solve in general due to a lack of closed-form expressions and challenging expectation integrals. The essential idea behind particle filters is to employ Monte Carlo integration techniques in order to ameliorate both of these challenges. This article presents an intuitive introduction to the main particle filter ideas and then unifies three commonly employed particle filtering algorithms. This unified approach relies on a nonstandard presentation of the particle filter, which has the advantage of highlighting precisely where the differences between these algorithms stem from. Some relevant extensions and successful application domains of the particle filter are also presented.

1. INTRODUCTION

Decision-making in the presence of uncertainty is a fundamental aspect of our modern world. For example, consider an autonomous car that is faced with an obstacle in its path. Assuming that it is important to avoid colliding with obstacles, a decision is required to determine the best course of action for the steering, braking, and acceleration. Importantly, these decisions must be made without complete knowledge of the environment, the response of the vehicle, or even the location and orientation of the car. Another example is the many decisions surrounding the best way to manage a disease epidemic. These decisions are inevitably made without perfect knowledge of their impact.

In any case, it is wise to consider any prior knowledge and available evidence when making decisions. For example, using prior knowledge of vehicle dynamics and measurements from sensors (such as GPS, lidar, radar, inertial measurement units, and cameras) has proven to be hugely successful in autonomous vehicle applications (1). Similarly, dynamic models of disease spread and infection rate measurements have been successfully combined to help predict outbreaks and therefore allow decision makers to take evasive action (2).

In many important cases, new evidence is obtained sequentially. Therefore, it is also prudent to repeat the decision-making process as the sequence evolves; otherwise, actions may lead to devastating consequences. For example, forward-looking vehicle lidar measurements may indicate the presence of an obstacle that was previously (in time) occluded from view. Ignoring this new information may have catastrophic effects.

Combining prior knowledge with evidence that is revealed sequentially is the topic of this article. In a general sense, these types of problems can be considered within a so-called filtering framework. It is important to recognize that there are many different theoretical frameworks for considering this filtering problem. This article is concerned with a probabilistic approach, where tools from probability theory and statistics are used to describe levels of belief or uncertainty.

We will consider a general filtering problem that is expressed in so-called probabilistic state-space form (3). In principle, the solution to this filtering problem is already known and relies on the sequential use of Bayes' rule, among other probabilistic identities. There are two fundamental problems with this general solution: (*a*) It is impossible to obtain closed-form solutions in general, and (*b*) the solution requires evaluation of potentially large-dimensional integrals.

Due to the importance of filtering and the difficulty associated with solving the general problem, enormous research attention has been directed toward various approximations. In many cases, these approximations require implicitly or explicitly modifying the problem from its original form in order to employ a particular approximation method (this essentially simplifies the integration problem). Two well-known examples include the extended Kalman filter (4) and unscented Kalman filter (5) approaches. These approaches are very popular and attractive due to their performance and low computational complexity. A potential drawback of these methods is that the approximation accuracy cannot, in general, be arbitrarily improved.

This article reviews a popular alternative approximation where the problem remains unaltered from its stated form and the approximation of the required integrals can be made arbitrarily accurate. The key mechanism relies on Monte Carlo integration techniques, which are applicable to a broad set of problems. Since the filtering problem requires a sequential update based on newly available evidence, we employ a specialized form of Monte Carlo integration called sequential Monte Carlo (SMC) methods. The SMC methods are also often called particle filters.

These ideas have their origins in works by Gordon et al. (6), Kitagawa (7), and Stewart & McCarty (8), and there have been many relevant reviews (9–11) and important monographs (12–16). Importantly, the efficacy of SMC methods has been demonstrated in many and disparate situations, from autonomous vehicles (1) to disease modeling (17) to machine learning (18) to

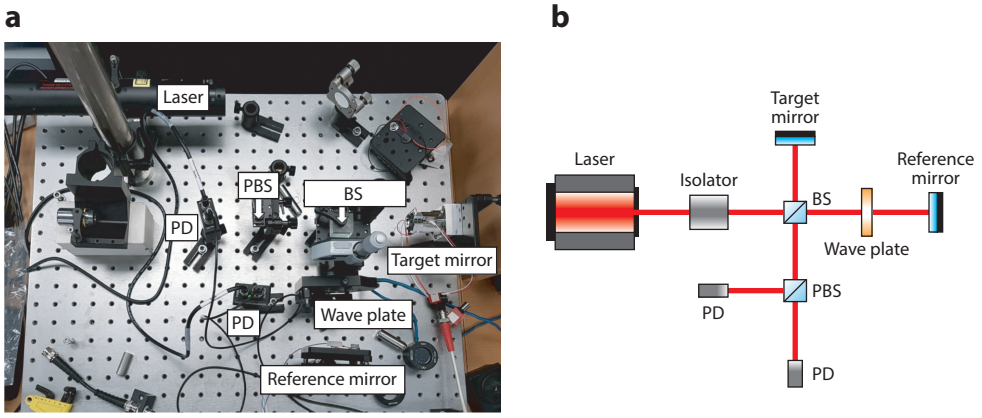

Figure 1

(*a*) A homodyne Michelson interferometer setup. (*b*) A schematic of the interferometer shown in panel *a*. Abbreviations: BS, beam splitter; PBS, polarizing beam splitter; PD, photodetector. Figure adapted from Reference 66 with permission from Petter Ersbo.

searching for the MH370 aircraft (19). This article is therefore not intended to argue the case for SMC methods. Rather, we are interested in achieving two primary goals: First, we intend to introduce the key ideas in an intuitive manner using a pedagogical example of estimating the position and velocity of a target mirror using the interferometer apparatus shown in **Figure 1*a***, and second, we abstract from the particulars of this example and present the details of three popular particle filters in a unified manner.

In particular, we use a nonstandard presentation of the particle filter in order to unify three commonly used algorithms. This perspective highlights the different ways of employing Monte Carlo integration and the impact of these choices within the filter.

The article is organized as follows. Section 2 presents a motivating example and the essential ideas underlying the particle filter. Section 3 details the key technical ideas and a unifying framework to consider several popular particle filtering variants. Section 4 shows how the particle filters play an important role in other related fields and how they can be generalized to a broad class of problems. Section 5 provides some concluding remarks. It is assumed that the reader has a basic working knowledge of probability theory and statistics, but for those who do not, potentially helpful resources include books by Gut (20) and Gelman et al. (21).

2. A PEDAGOGICAL EXAMPLE

This section introduces the key principles and assumptions underlying SMC methods. To make these ideas concrete, we introduce the essential components by way of a pedagogical example. It is important to note that this example is intended for illustrative purposes and is not, in any way, intended to represent state-of-the-art estimation in this case.

More specifically, we consider a problem of estimating the position and velocity of a moving mirror using an interferometer apparatus (shown in **Figure 1*a***). Interferometry dates back more than 100 years and was vital in accurately measuring flatness, an essential ingredient in the development of modern science and engineering (22). Perhaps the most famous interferometers today are the two large instruments that form the Laser Interferometer Gravitational-Wave Observatory (LIGO). This apparatus was used in the 2015 verification of gravitational waves (23), for which Rainer Weiss, Barry Barish, and Kip Thorne were awarded the 2017 Nobel Prize in Physics.

The particular setup in **Figure 1*a*** is a homodyne Michelson interferometer that is commonly used for very-high-precision displacement measurements (see, e.g., 24). These displacement measurements have found application in many areas, including vibration measurement, gas flow analysis, high-precision gyroscopes, and high-precision position control, to name just a few (25).

The basic principle of interferometry is to use the wave properties of light to measure distance. Splitting a coherent light source into two separate paths and then recombining them will generate an interference pattern. Differences in the path length for the split light will produce changes in the interference pattern, which is the observed quantity. That is, assuming a light wavelength of λ, path-length differences that are integer multiples of λ result in constructive interference, which results in high levels of measured light intensity. Fractional path-length differences result in lower levels of measured light intensity, and fully destructive interference results in the lowest levels of measured light intensity.

To make these ideas more concrete, the apparatus shown in **Figure 1*a*** can be represented by the schematic in **Figure 1*b***. The laser, with wavelength λ, provides a coherent light source that is split in two using a beam splitter. The split light is then reflected by two separate mirrors, one that is fixed (the top mirror, called the reference mirror) and one that is allowed to move (the rightmost mirror, called the target mirror). The moving mirror then provides a time-dependent path-length difference. Between the beam splitter and the target mirror is a wave plate, which adds a $\lambda/8$ phase shift between the horizontal and vertical polarization directions of the light. This wave plate adds the same phase shift on the returning light, for a total $\lambda/4$ phase shift between polarizations. The polarizing beam splitter then splits the incident light depending on the polarization direction, and the newly split light intensity is detected by the two photodetectors.

If we label the two measured intensities from the photodetectors as $\mathbf{y}_1$ and $\mathbf{y}_2$, then we can model the measured signals as

$$y_1 = \alpha_1 + \beta_1 \cos(\kappa d) + v_1, \qquad 1.$$

$$y_2 = \alpha_2 + \beta_2 \sin(\kappa d) + v_2, \qquad 2.$$

where d is the position of the target mirror and $\kappa \triangleq 2\pi/\lambda$. The sine and cosine terms stem from the wave properties of light and the phase shift introduced by the wave plate. The parameters $\alpha_{1,2}$ and $\beta_{1,2}$ account for the offsets and gain terms of the measured intensities, respectively. The noise terms $v_{1,2}$ account for uncertainty in the measured intensity. **Figure 2*a*** shows a segment of the measured light intensities for the simulated target mirror position shown in **Figure 3*b***, and **Figure 2*b*** shows a scatter plot of the measurements for the full data sequence. (**Figure 3*a*** is further discussed below, and **Figure 3*b–d*** is further explained at the end of this section.)

Assuming that the intensities can be measured at regular time intervals spaced Δ seconds apart, we use the notation $y_1(k)$ and $y_2(k)$ to indicate the measured intensities at time instant $k\Delta$, where k is an integer and is called the discrete-time index. We can further collect these measurement at index k in the so-called output variable $\mathbf{y}_k$, defined as

$$\mathbf{y}_k \triangleq \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \alpha_1 + \beta_1 \cos(\kappa d) \\ \alpha_2 + \beta_2 \sin(\kappa d) \end{bmatrix} + \mathbf{e}_k, \qquad \mathbf{e}_k \triangleq \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix}. \qquad 3.$$

Given these intensity measurements, the aim is to estimate both the position d and the associated velocity $\dot{d}$ at each discrete-time index k. We can assume for simplicity that the accelerations causing the mirror to move are unknown, and we can therefore model the discrete-time evolution of

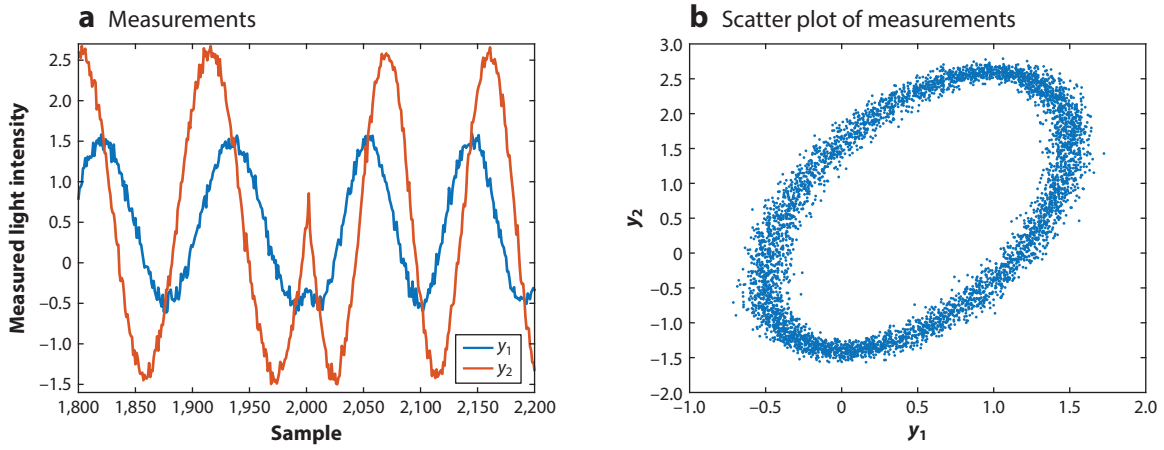

Figure 2

(*a*) Segment of measured light intensities. (*b*) Scatter plot of measured light intensities.

position and velocity via a simple kinematic, stochastic state-space model,

$$\underbrace{\begin{bmatrix} d(k+1) \\ \dot{d}(k+1) \end{bmatrix}}_{\triangleq \mathbf{x}_{k+1}} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} d(k) \\ \dot{d}(k) \end{bmatrix}}_{\triangleq \mathbf{x}_k} + \boldsymbol{\eta}_k, \tag{4.}$$

where the implicit definition of $\mathbf{x}_k$ is the so-called model state vector, and $\boldsymbol{\eta}_k$ is a random variable used to model uncertainty in the state evolution model.

Restating the aim, we wish to estimate the state $\mathbf{x}_k$ (the position and velocity of the target mirror) given all the measurements $\mathbf{y}_{1:k} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$. There are many ways to attack this, but here we follow a probabilistic approach. The main idea is to model our belief of the state using a probability density function. It is important to clarify that the actual position and velocity are not uncertain; rather, it is our knowledge that is uncertain. In particular, we are interested in providing a distribution of the state conditioned on all of the available data—that is, we seek to find

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}). \tag{5.}$$

It is perhaps not immediately clear how we can arrive at this distribution, which we now turn our attention to. Suppose that our belief of the state at time $k = 1$, prior to obtaining any measurements, can be described via the distribution

$$\mathbf{x}_1 \sim p(\mathbf{x}_1). \tag{6.}$$

For example, suppose that we believe that $p(\mathbf{x}_1)$ can be adequately modeled via a multivariate normal distribution according to

$$p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_1, \mathbf{P}_1) = \det(2\pi\mathbf{P}_1)^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \mathbf{P}_1^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1)}, \tag{7.}$$

where $\boldsymbol{\mu}_1$ is the mean and $\mathbf{P}_1$ is the covariance. The actual values employed for each depends on prior knowledge. **Figure 3*a*** shows samples (also referred to as particles) from $p(\mathbf{x}_1)$.

Then we are at time $k = 1$ provided with light intensity measurements $\mathbf{y}_1$, and we would like to update our belief of the state—that is, we want $p(\mathbf{x}_1 \mid \mathbf{y}_1)$. To this end, we can employ Bayes' rule to obtain

$$p(\mathbf{x}_1 \mid \mathbf{y}_1) = \frac{p(\mathbf{y}_1 \mid \mathbf{x}_1)\, p(\mathbf{x}_1)}{p(\mathbf{y}_1)}. \tag{8.}$$

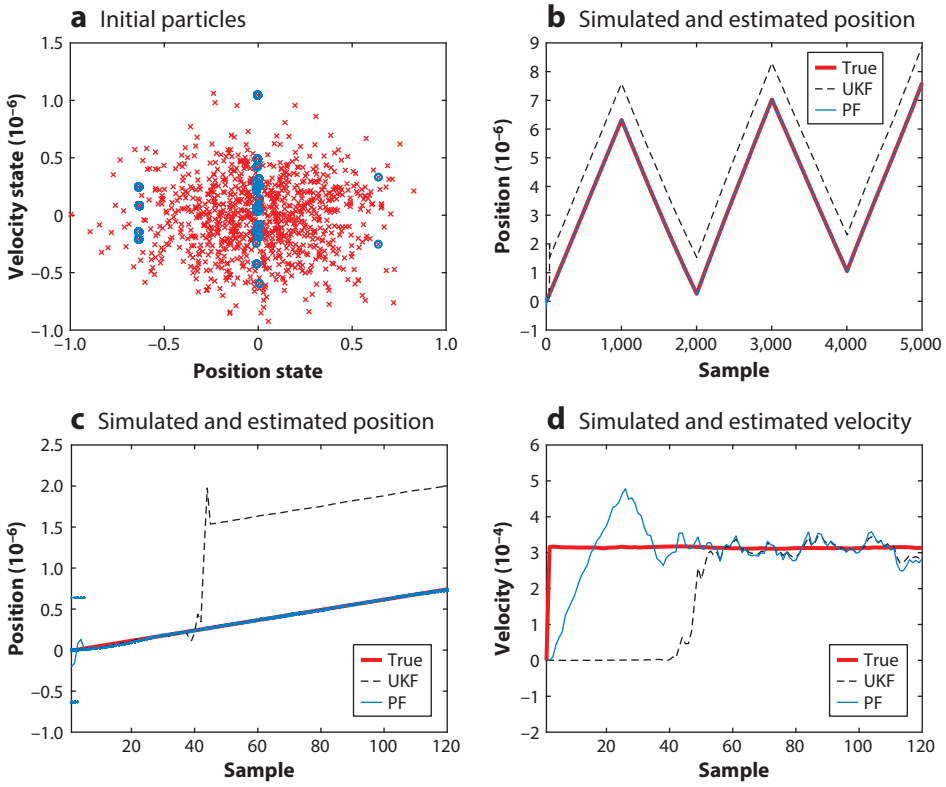

Figure 3

(*a*) Initial samples (particles) $\mathbf{x}_1^i$, shown as red x's, drawn from a multivariate normal with mean $\boldsymbol{\mu}_1 = 0$ and covariance $\mathbf{P}_1 = \frac{1}{4}\lambda^2\mathbf{I}_{2\times 2}$. Resampled particles based on $\mathbf{y}_1$ are shown as blue circles, where three vertical bands are roughly λ distance apart. (*b*) Simulated (true) target mirror position (*solid red line*), the mean position estimate from an unscented Kalman filter (UKF) (*dashed black line*), and the mean position estimate from the particle filter (PF) (*solid blue line*). The transient is shown in panel *c*. (*c*) Simulated (true) target mirror position (*solid red line*), UKF mean estimate (*dashed black line*), and PF mean estimate (particles shown as *blue dots*), showing more detail at the start of the simulation. (*d*) Simulated (true) target mirror velocity (*solid red line*), UKF mean estimate (*dashed black line*), and PF mean estimate (particle mean shown as a *solid blue line*), again showing more detail at the start of the simulation.

For this to be useful, we need to consider each of the distributions on the right-hand side. We start by noting that the denominator term $p(\mathbf{y}_1)$ is a normalizing constant that ensures a proper density function, and it may be safely ignored for now. Concerning $p(\mathbf{x}_1)$, notice that this is the prior distribution available in Equation 7. We now concentrate on $p(\mathbf{y}_1 \mid \mathbf{x}_1)$, which describes our belief of the measurements $\mathbf{y}_1$ given the state. This is a fundamentally important object, known as the measurement likelihood model, relating the state to the measurements.

We already have such a relationship described in Equation 3, but it is not quite in the required probabilistic form $p(\mathbf{y}_1 \mid \mathbf{x}_1)$. To remedy this, we will further assume a distribution for the measurement noise term e_k, which is independent over k and where for each k,

$$\mathbf{e}_k \sim \mathcal{N}(\mathbf{e}_k; 0, \mathbf{R}). \qquad 9.$$

With this in place, we can then say that the distribution of $\mathbf{y}_1$ given $\mathbf{x}_1$ is multivariate normal via

$$p(\mathbf{y}_1 \mid \mathbf{x}_1) = \mathcal{N}(\mathbf{y}_k; \mathbf{g}(\mathbf{x}_k), \mathbf{R}) = \det(2\pi R)^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{y}_1 - \mathbf{g}(\mathbf{x}_1))^T \mathbf{R}^{-1}(\mathbf{y}_1 - \mathbf{g}(\mathbf{x}_1))}, \qquad 10.$$

where $\mathbf{g}(\mathbf{x}_k)$ is defined as (refer to Equation 3)

$$\mathbf{g}(\mathbf{x}_k) = \begin{bmatrix} \alpha_1 + \beta_1 \cos(\kappa \mathbf{x}_k(1)) \\ \alpha_2 + \beta_2 \sin(\kappa \mathbf{x}_k(1)) \end{bmatrix}, \quad 11.$$

where $\mathbf{x}_k(1)$ is used to indicate the position state at discrete-time index k. Put another way, we believe that given the state $\mathbf{x}_1$, the measurements $\mathbf{y}_1$ are a realization from $p(\mathbf{y}_1 \mid \mathbf{x}_1)$, and that if the target mirror was fixed, then repeated measurements would just be different realizations from $p(\mathbf{y}_1 \mid \mathbf{x}_1)$.

In principle, with the measurement likelihood model in Equation 10 and prior in Equation 7 in place, we can describe the posterior $p(\mathbf{x}_1 \mid \mathbf{y}_1)$ via Equation 8. At the same time, it is generally not possible to express this posterior in closed form.[1] A major benefit of SMC methods is that they approximate the posterior using a finite number of terms. The essential idea is to represent the filtered density $p(\mathbf{x}_1 \mid \mathbf{y}_1)$ as an empirical distribution represented using a finite weighted sum of point-mass distributions (Dirac delta functions),

$$p(\mathbf{x}_1 \mid \mathbf{y}_1) \approx \sum_{i=1}^{M} w_1^i \delta(\mathbf{x}_1 - \mathbf{x}_1^i), \quad 12.$$

where the locations of the point masses are determined by the so-called particles $\mathbf{x}_1^i$. The associated weight w_1^i represents the relative importance of the ith particle.

To be more specific, consider each of the $M = 1{,}000$ samples for the initial state $\mathbf{x}_1 \sim p(\mathbf{x}_1)$ from **Figure 3*a***. Intuitively, we can determine the importance of each particle $\mathbf{x}_1^i$ by pretending that the target mirror had this position and velocity and then generate a virtual measurement using this state via Equation 11. This can then be compared with the actual measurement, where strong agreement would result in high importance, and strong disagreement would result in low importance. This comparison is neatly captured by our likelihood model (Equation 10), which will have larger values when $\mathbf{y}_1 - \mathbf{g}(\mathbf{x}_1^i)$ is small and small values when $\mathbf{y}_1 - \mathbf{g}(\mathbf{x}_1^i)$ is large. Therefore, intuitively speaking, we could weight each of the particles by the following so-called importance weight:

$$w_1^i \triangleq p(\mathbf{y}_1 \mid \mathbf{x}_1^i). \quad 13.$$

For the current example, **Figure 3*a*** also shows that 99.99% of the importance comes from just 30 particles. This indicates that most of the prior states are extremely unlikely, and we have effectively reduced our options to just 30 possibilities. Also note that the likely particles occur in bands, spaced roughly λ distance in the position state, which stems from the cyclic nature of the measurements.

Suppose, for the moment, that at time $k = 2$ we are presented with new light intensity measurements $\mathbf{y}_2$ and wish to provide a distribution of the state $\mathbf{x}_2$ based on all the measurements so far (i.e., $\mathbf{y}_1, \mathbf{y}_2$). That is, we seek

$$p(\mathbf{x}_2 \mid \mathbf{y}_{1:2}) = \frac{p(\mathbf{y}_2 \mid \mathbf{x}_2, \mathbf{y}_1)\, p(\mathbf{x}_2 \mid \mathbf{y}_1)}{p(\mathbf{y}_2 \mid \mathbf{y}_1)}, \quad 14.$$

where the right-hand side again stems from application of Bayes' rule. These distributions require some discussion. The denominator term is again a normalizing constant and can be safely ignored. The term $p(\mathbf{y}_2 \mid \mathbf{x}_2, \mathbf{y}_1)$ is similar to the measurement likelihood previously defined but includes

[1] That is, in general, it is not possible to express the posterior using a finite number of elementary mathematical operations and constants (6, 8).

conditioning on $\mathbf{y}_1$. Intuitively, it may be argued that if the state $\mathbf{x}_2$ was given, then the current intensity measurements $\mathbf{y}_2$ should not depend on previous measurements. This is often called a conditional independence assumption and results from the Markov nature of state-space systems (see, e.g., 3), where we assume that $\mathbf{y}_k$ is conditionally independent of the past when given the state $\mathbf{x}_k$—that is, we will henceforth assume that

$$p(\mathbf{y}_k \mid \mathbf{x}_k, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k \mid \mathbf{x}_k). \tag{15.}$$

Considering the remaining term $p(\mathbf{x}_2 \mid \mathbf{y}_1)$ from Equation 14, we have at our disposal the posterior $p(\mathbf{x}_1 \mid \mathbf{y}_1)$, which from the law of total probability affords

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) = \int p(\mathbf{x}_2, \mathbf{x}_1 \mid \mathbf{y}_1)\, \mathrm{d}\mathbf{x}_1 \tag{16.}$$

$$= \int p(\mathbf{x}_2 \mid \mathbf{x}_1, \mathbf{y}_1)\, p(\mathbf{x}_1 \mid \mathbf{y}_1)\, \mathrm{d}\mathbf{x}_1, \tag{17.}$$

where the second equality stems from an application of conditional probability. In principle, $p(\mathbf{x}_1 \mid \mathbf{y}_1)$ is already known via Equation 8. The remaining term, $p(\mathbf{x}_2 \mid \mathbf{x}_1, \mathbf{y}_1)$, is a distribution of the state at $k = 2$ given the previous state $\mathbf{x}_1$ and past measurements; this is another fundamentally important distribution, known as the transition distribution. A commonly employed modeling assumption is that $\mathbf{x}_k$ is independent of past data given $\mathbf{x}_{k-1}$, and we say that $\mathbf{x}_k$ satisfies the Markov property (see, e.g., 3)—that is, we assume

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}). \tag{18.}$$

Therefore, the distribution $p(\mathbf{x}_2 \mid \mathbf{y}_1)$ (known as the prediction distribution since it involves predicting the state based on previous data) can be expressed as

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) = \int p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{x}_1 \mid \mathbf{y}_1)\, \mathrm{d}\mathbf{x}_1. \tag{19.}$$

For the current example, the transition distribution $p(\mathbf{x}_2 \mid \mathbf{x}_1)$ is a model of the dynamic behavior of the target mirror over the time interval between points $k = 1$ and $k = 2$. We already have such a model in Equation 4, but again it is not in the required form $p(\mathbf{x}_2 \mid \mathbf{x}_1)$. In a similar manner to before, we remedy this by imposing an assumption on the distribution for η_k (the state noise); specifically, we assume that

$$\boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{\eta}_k; 0, \mathbf{Q}). \tag{20.}$$

With this in place, the distribution for $\mathbf{x}_{k+1}$ given $\mathbf{x}_k$ is given by

$$p(\mathbf{x}_{k+1} \mid \mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{f}(\mathbf{x}_k), \mathbf{Q}), \tag{21.}$$

where $\mathbf{f}(\mathbf{x}_k)$ is defined as (refer to Equation 4)

$$\mathbf{f}(\mathbf{x}_k) \triangleq \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} \mathbf{x}_k. \tag{22.}$$

In principle, we have all the terms required to provide the prediction distribution from Equation 19. At the same time, we face the same challenge as before, in that we cannot express this solution in closed form in general. We also face another challenge in that the integral is intractable for even modest dimensions of $\mathbf{x}_k$ (for a detailed discussion of these ideas, see 26, 27). A

major benefit of the SMC approach is that it overcomes both of these problems. In essence, we can simply substitute the approximation from Equation 12 into Equation 19 to reveal

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \int p(\mathbf{x}_2 \mid \mathbf{x}_1) \sum_{i=1}^{M} w_1^i \delta(\mathbf{x}_1 - \mathbf{x}_1^i)\, d\mathbf{x}_1$$

$$= \sum_{i=1}^{M} w_1^i p(\mathbf{x}_2 \mid \mathbf{x}_1^i). \qquad 23.$$

Therefore, the SMC approach results in an approximation of the prediction density that involves a weighted combination of the transition model (i.e., our dynamics model for the target mirror).

Faced with a new measurement $\mathbf{y}_2$, we can in principle repeat the above process to arrive at an approximation $p(\mathbf{x}_3 \mid \mathbf{y}_{1:2})$. More specifically, we note that

$$p(\mathbf{x}_3 \mid \mathbf{y}_{1:2}) = \int p(\mathbf{x}_3 \mid \mathbf{x}_2)\, p(\mathbf{x}_2 \mid \mathbf{y}_{1:2})\, d\mathbf{x}_2 \qquad 24.$$

and that the filter distribution $p(\mathbf{x}_2 \mid \mathbf{y}_{1:2})$ is given by Equation 14. Following a similar line of argument to the above discussion, we can see that the essential idea is to once again approximate this filter distribution as

$$p(\mathbf{x}_2 \mid \mathbf{y}_{1:2}) \approx \sum_{i=1}^{M} w_2^i \delta(\mathbf{x}_2 - \mathbf{x}_2^i), \qquad 25.$$

where the particles $\mathbf{x}_2^i$ are this time samples from the prior $\mathbf{x}_2^i \sim p(\mathbf{x}_2 \mid \mathbf{y}_1)$ and the weights w_2^i are defined as the likelihood $p(\mathbf{y}_2 \mid \mathbf{x}_2^i)$; this has the same interpretation as previously, in that particles that are highly likely will have higher weight values and vice versa. Substitution of Equation 25 into Equation 24 reveals that

$$p(\mathbf{x}_3 \mid \mathbf{y}_{1:2}) \approx \int p(\mathbf{x}_3 \mid \mathbf{x}_2) \sum_{i=1}^{M} w_2^i \delta(\mathbf{x}_2 - \mathbf{x}_2^i)\, d\mathbf{x}_2$$

$$= \sum_{i=1}^{M} w_2^i p(\mathbf{x}_3 \mid \mathbf{x}_2^i). \qquad 26.$$

These steps can be repeated for each new sample time k as new measurements $\mathbf{y}_k$ become available. It is important to remember that the above discussion is using finite sample approximations. Under some mild conditions (see Section 3.6), the resulting approximation is guaranteed to converge to the true underlying distribution as $M \to \infty$.

As a preview of the utility of this approach, using $M = 1{,}000$ particles and a variant of Algorithm 1 [using the sequential importance resampling (SIR) choice], given in Section 3.4, results in the position estimates shown in **Figure 3*b***. For the purposes of simple comparison, an unscented Kalman filter estimate is also provided. **Figure 3*c*,*d*** provides more detail around the start of the simulation, where the particle locations demonstrate the multimodal nature of the state distribution; three regions of support are provided by the particles, each separated by a single wavelength (refer to **Figure 3*a***). Note that the unscented Kalman filter mean jumps several wavelengths around the 40th sample, which is perfectly well supported by the measurements.

3. SEQUENTIAL MONTE CARLO: A UNIFIED APPROACH

The above discussion of the position estimation problem highlighted a number of key elements when considering a probabilistic estimation approach. The first key element was that the quantities

of interest at time k were collected into a vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$, which is called the state. The second was the availability of a belief about the state $\mathbf{x}_1$ prior to collecting any measurements (but possibly conditioned on any other relevant knowledge of the problem). The third was the availability of a measurement likelihood model that relates the state $\mathbf{x}_k$ to the measurements $\mathbf{y}_k$. And finally, the fourth was the availability of a state transition model that predicts the distribution of the state one step into the future $\mathbf{x}_{k+1}$ based on knowledge of the current state $\mathbf{x}_k$.

In this section, we abstract from the particulars of this pedagogical example and return to it when pertinent to do so. Our focus in this section is to consider the same type of problem from the example—namely, that at time k we seek the distribution of the state based on all the available data so far. This is known as the Bayesian filtering problem for state-space models. Using slightly more formal notation than in the example, the state at time index k is treated as a random variable and denoted as $\mathbf{X}_k$. We assume that the state evolves over time according to the following conditional distribution (akin to the dynamic model from the example):

$$(\mathbf{X}_{k+1} \mid \mathbf{X}_k = \mathbf{x}_k) \sim p(\mathbf{x}_{k+1} \mid \mathbf{x}_k). \tag{27.}$$

We note that this state transition model does not explicitly rely on inputs or other possibly important knowledge. This is purely for ease of exposition, and the above model is allowed to depend on inputs and other knowledge where appropriate. We further assume that the measured outputs are a realization of the random variable $\mathbf{Y}_k$ and that the state is related to $\mathbf{Y}_k$ via the following conditional distribution (the measurement likelihood model):

$$(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}_k) \sim p(\mathbf{y}_k \mid \mathbf{x}_k). \tag{28.}$$

The so-called Bayes filter aims to provide the state distribution at time k conditioned on a collection of output measurements $\mathbf{y}_{1:k} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$. That is, the filter distribution is

$$(\mathbf{X}_k \mid \mathbf{Y}_{1:k} = \mathbf{y}_{1:k}) \sim p(\mathbf{x}_k \mid \mathbf{y}_{1:k}). \tag{29.}$$

This filtered distribution can be expressed using the measurement likelihood $p(\mathbf{y}_k \mid \mathbf{x}_k)$ and a prior on $\mathbf{X}_k$ given the previous data $\mathbf{y}_{1:k-1}$ via[2]

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})}, \tag{30.}$$

where the distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ is known as the prediction distribution and describes the distribution of the state given all measurements except $\mathbf{y}_k$—that is,

$$(\mathbf{X}_k \mid \mathbf{Y}_{1:k-1} = \mathbf{y}_{1:k-1}) \sim p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}). \tag{31.}$$

This prediction distribution can also be linked to the previous filtered distribution and the state transition model $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ via[3]

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_{k-1}. \tag{32.}$$

The filtering and prediction stages can be combined to provide a recursion from one prediction distribution to the next via

$$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k}) = \int \frac{p(\mathbf{y}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})}\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_k. \tag{33.}$$

Equation 33 serves as the central object throughout the remainder of this article.

[2]This relies on the Markov property of state-space systems and Bayes' theorem.
[3]This relies on the law of total probability and the Markov property.

Without loss of generality, we can assume that the initial state $\mathbf{X}_1$ is distributed according to a finite mixture distribution:

$$\mathbf{X}_1 \sim p(\mathbf{x}_1) \triangleq \sum_{i=1}^{M} w_0^i \, p(\mathbf{x}_1 \mid \mathbf{x}_0^i), \qquad w_0^i \geq 0, \qquad \sum_{i=1}^{M} w_0^i = 1. \tag{34.}$$

The distributions $p(\mathbf{x}_1 \mid \mathbf{x}_0^i)$ are allowed to depend on regular variables $\mathbf{x}_0^i$, in case this is useful. Furthermore, the distributions $p(\mathbf{x}_1 \mid \mathbf{x}_0^i)$ should not be confused with the state transition model (although this is technically allowed and possibly useful in some circumstances). As a concrete example, the prior from the interferometry example can be recovered with $M = 1$, $w_0^1 = 1$, and $p(\mathbf{x}_1 \mid \mathbf{x}_0^i) \triangleq p(\mathbf{x}_1)$. We note that this level of generality may not be warranted, but it does allow for a parsimonious presentation of the various SMC methods without any special consideration regarding the initial distribution.

Armed with the initial state distribution and presented with a single observation $\mathbf{y}_1$, the application of the Bayes filter equations results in a prediction distribution $p(\mathbf{x}_2 \mid \mathbf{y}_1)$ via[4]

$$\begin{aligned} p(\mathbf{x}_2 \mid \mathbf{y}_1) &= \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} p(\mathbf{x}_1)\, \mathrm{d}\mathbf{x}_1 \\ &\approx \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} \sum_{i=1}^{M} w_0^i \, p(\mathbf{x}_1 \mid \mathbf{x}_0^i)\, \mathrm{d}\mathbf{x}_1 \\ &= \sum_{i=1}^{M} \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i \, p(\mathbf{x}_1 \mid \mathbf{x}_0^i)\, \mathrm{d}\mathbf{x}_1. \end{aligned} \tag{35.}$$

As mentioned previously, it is not generally possible to obtain closed-form solutions to the above sum integral, except in some well-known and important cases (see, e.g., 13, 28). Therefore, we seek instead to arrive at asymptotic solutions that exhibit convergence to the desired solution, with a guaranteed convergence rate.

Toward this goal, the primary mechanism employed in SMC methods relies on the law of large numbers (LLN), often called Monte Carlo integration, whereby the sample mean converges to the expected value:

$$\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \mathbf{f}(\mathbf{z}^i) \to \int \mathbf{f}(\mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z}. \tag{36.}$$

The above assumes that $\mathbf{z}^i \sim p(\mathbf{z})$ are independent and identically distributed (i.i.d.) for $i = 1, \ldots$ and that $\mathbf{f}(\cdot)$ is a measurable function. In general, it may be difficult to sample directly from $p(\mathbf{z})$ but possible to evaluate it pointwise. If we suppose that it is instead straightforward to sample from another distribution $q(\mathbf{z})$, it follows under mild conditions that

$$\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \frac{\mathbf{f}(\mathbf{z}^i) p(\mathbf{z}^i)}{q(\mathbf{z}^i)} \to \int \frac{\mathbf{f}(\mathbf{z}) p(\mathbf{z})}{q(\mathbf{z})} q(\mathbf{z}) \mathrm{d}\mathbf{z} = \int \mathbf{f}(\mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z}, \tag{37.}$$

where $\mathbf{z}^i \sim q(\mathbf{z})$ are i.i.d. Furthermore, and importantly, the LLN can be used to approximate expectations for joint discrete–continuous random variables. Indeed,

$$\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \frac{\mathbf{f}(j^i, \mathbf{z}^i) p(j^i, \mathbf{z}^i)}{q(j^i, \mathbf{z}^i)} \to \sum_j \int \frac{\mathbf{f}(j, \mathbf{z}) p(j, \mathbf{z})}{q(j, \mathbf{z})} q(j, \mathbf{z}) \mathrm{d}\mathbf{z} = \sum_j \int \mathbf{f}(j, \mathbf{z}) p(j, \mathbf{z}) \mathrm{d}\mathbf{z}, \tag{38.}$$

where $(j^i, \mathbf{z}^i)$ are i.i.d. with the joint discrete–continuous distribution $q(j, \mathbf{z})$.

[4]Here we have implicitly applied the monotone convergence theorem in order to exchange the order of integration and summation for the last equality.

The various incarnations of particle filters can all be explained by observing that each employs the LLN to approximate various integrals and summations within the Bayes filter equations. As a summary of what follows, all SMC methods considered in this article result in an approximation of the prediction distribution in the form of a mixture,

$$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k}) \approx \sum_{i=1}^{M} w_k^i p(\mathbf{x}_{k+1} \mid \mathbf{x}_k^i). \tag{39.}$$

Toward this result, in the subsequent sections we introduce the key ideas behind three popular particle filters and then discuss a unified algorithm that encapsulates all three. We remark that this presentation is somewhat nonstandard in that our focus is directed toward the prediction distribution rather than the filter $p(x_k \mid y_{1:k})$, which is more commonly discussed. Due to its importance, we address the filter distribution following the discussion on various particle filter methods.

3.1. The Sequential Importance Sampling Particle Filter

The sequential importance sampling (SIS) particle filter has a long history in the field of Bayesian filtering (29). In the framework of the current article, the essential idea behind the SIS approach is to approximate the integral in

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) = \sum_{i=1}^{M} \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i\, p(\mathbf{x}_1 \mid \mathbf{x}_0^i) \mathrm{d}\mathbf{x}_1 \tag{40.}$$

for each i via Monte Carlo integration, but, importantly, using only a single sample.

We have a number of options at this point. We could draw a sample from $p(\mathbf{x}_1 \mid \mathbf{x}_0^i)$ (which is the most common choice in the literature),[5] or, as mentioned in the previous section (see Equations 37 and 38), it may be beneficial to draw a sample from a more general proposal distribution,

$$q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1). \tag{41.}$$

It is important to reflect on this proposal for a moment. Notice that the proposal distribution has the flexibility to be conditioned on the most recent observation $\mathbf{y}_1$; this flexibility can help in proposing particles in more likely locations since we are allowed to use the most recent data. We can therefore express the required sum integral as

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) = \sum_{i=1}^{M} \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i \frac{p(\mathbf{x}_1 \mid \mathbf{x}_0^i)}{q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)} q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1) \mathrm{d}\mathbf{x}_1. \tag{42.}$$

The SIS particle filter approximates the above integral for each i using a single sample (also called a particle) from a proposal distribution $\mathbf{x}_1^i \sim q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)$ in order to approximate

$$\int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i\, p(\mathbf{x}_1 \mid \mathbf{x}_0^i) \mathrm{d}\mathbf{x}_1 \approx \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1^i)\, p(\mathbf{y}_1 \mid \mathbf{x}_1^i)}{p(\mathbf{y}_1)} \frac{p(x_1^i \mid \mathbf{x}_0^i)}{q(\mathbf{x}_1^i \mid \mathbf{x}_0^i, \mathbf{y}_1)} w_0^i. \tag{43.}$$

This leads to the following approximation for the prediction distribution:

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \sum_{i=1}^{M} p(\mathbf{x}_2 \mid \mathbf{x}_1^i) \underbrace{\frac{p(\mathbf{y}_1 \mid \mathbf{x}_1^i)}{p(\mathbf{y}_1)} \frac{p(x_1^i \mid \mathbf{x}_0^i)}{q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)} w_0^i}_{\triangleq \tilde{w}_1^i}. \tag{44.}$$

[5] Reflecting on the interferometry problem from Section 2, we can see that this amounts to using the multivariate normal distribution from Equation 21.

We note that the above-defined weights[6] $\tilde{w}_1^i$ are not guaranteed to sum to one, which unfortunately implies that the above approximation is not a proper distribution. A simple way to correct this is to replace $\tilde{w}_1^i$ with a new normalized weight, defined as

$$w_1^i \triangleq \frac{\tilde{w}_1^i}{\sum_{j=1}^M \tilde{w}_1^j}, \quad \Longrightarrow \quad \sum_{i=1}^M w_1^i = 1. \tag{45.}$$

Here we notice that the action of normalizing removes the dependence on $p(\mathbf{y}_1)$ since it cancels in the fraction; we can therefore safely ignore this term. Combining Equation 44 and the normalized weights in Equation 45 allows the prediction distribution $p(\mathbf{x}_2 \mid \mathbf{y}_1)$ to be approximated by the following mixture distribution:

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \sum_{i=1}^M w_1^i \, p(\mathbf{x}_2 \mid \mathbf{x}_1^i). \tag{46.}$$

It is important to highlight that the mixture relies only on weights w_1^i and particles $\mathbf{x}_1^i$. Interestingly, our approximation of the prediction distribution is also a mixture distribution, which was the assumed form for the prior $p(\mathbf{x}_1) = \sum_{i=1}^M w_0^i \, p(\mathbf{x}_1 \mid \mathbf{x}_0^i)$. Therefore, at least in principle, we could replace the unknown prediction distribution with this approximation and repeat the above process. To see this, note that the next prediction density $p(\mathbf{x}_3 \mid \mathbf{y}_{1:2})$ is given by

$$p(\mathbf{x}_3 \mid \mathbf{y}_{1:2}) = \int \frac{p(\mathbf{x}_3 \mid \mathbf{x}_2) p(\mathbf{y}_2 \mid \mathbf{x}_2)}{p(\mathbf{y}_2 \mid \mathbf{y}_1)} p(\mathbf{x}_2 \mid \mathbf{y}_1) \, \mathrm{d}\mathbf{x}_2 \tag{47.}$$

$$\approx \sum_{i=1}^M \int \frac{p(\mathbf{x}_3 \mid \mathbf{x}_2) p(\mathbf{y}_2 \mid \mathbf{x}_2)}{p(\mathbf{y}_2 \mid \mathbf{y}_1)} w_1^i p(\mathbf{x}_2 \mid \mathbf{x}_1^i) \, \mathrm{d}\mathbf{x}_2, \tag{48.}$$

where the final approximation comes from substituting Equation 46 into Equation 47. This is the same form as Equation 40, and we can therefore repeat the same reasoning to arrive at the mixture

$$p(\mathbf{x}_3 \mid \mathbf{y}_{1:2}) \approx \sum_{i=1}^M w_2^i \, p(\mathbf{x}_3 \mid \mathbf{x}_2^i). \tag{49.}$$

The sequential nature of using Monte Carlo integration in this way reveals the source of the term sequential Monte Carlo. The SIS particle filter is summarized in Algorithm 1 (see Section 3.4) with the SIS choice.

3.2. The Sequential Importance Resampling Particle Filter

A potential issue with the SIS approach is that employing only one sample for each i in the LLN approximation generally leads to high variance in the estimate (i.e., a poor approximation). In practice, the lack of particle diversity caused by using a single sample often means that the current measurement is not well supported, and most (if not all) particles will be very unlikely. Since it is typical that the variance is reduced as the number of samples increases, we could just use more samples for each i so that

$$\int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1) \, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i \, p(\mathbf{x}_1 \mid \mathbf{x}_0^i) \, \mathrm{d}\mathbf{x}_1 \approx \frac{1}{N} \sum_{j=1}^N \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1^j) \, p(\mathbf{y}_1 \mid \mathbf{x}_1^j)}{p(\mathbf{y}_1)} \frac{p(\mathbf{x}_1^j \mid \mathbf{x}_0^i)}{q(\mathbf{x}_1^j \mid \mathbf{x}_0^i, \mathbf{y}_1)} w_0^i, \tag{50.}$$

[6] We recommend computing these weights in log form, since this helps alleviate floating-point underflow.

where $\mathbf{x}_1^j \sim q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)$. Following similar reasoning as before, we arrive at the following approximation of the prediction distribution with NM components:

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \sum_{j=1}^{N} \sum_{i=1}^{M} w_1^{i,j}\, p(\mathbf{x}_2 \mid \mathbf{x}_1^j). \tag{51.}$$

While this is perfectly fine, repeating this approach will ultimately produce an exponential growth in the number of mixture components. To combat this, we could take a slightly different approach and use the LLN to approximate both the integral and the sum:

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) = \sum_{i=1}^{M} \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i\, p(\mathbf{x}_1 \mid \mathbf{x}_0^i)\, \mathrm{d}\mathbf{x}_1. \tag{52.}$$

Introducing the distribution $q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)$ provides

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) = \sum_{i=1}^{M} \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} \frac{p(\mathbf{x}_1 \mid \mathbf{x}_0^i)}{q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)} w_0^i\, q(\mathbf{x}_1 \mid \mathbf{x}_0^i, \mathbf{y}_1)\, \mathrm{d}\mathbf{x}_1. \tag{53.}$$

In contrast to the SIS case, where we used a single sample for each i, here we choose to sample both the index i and $\mathbf{x}_1$ jointly; this added flexibility allows the possibility of ignoring certain indices and concentrating on others of higher utility. It will be important later that we have knowledge of which index i was sampled. Therefore, we denote the jth sample of i as the integer a_1^j to indicate that this was the jth sample at $k = 1$; this is called an ancestor index. We sample $(a_1^j, \mathbf{x}_1^j)$ jointly from

$$(a_1^j, \mathbf{x}_1^j) \;\sim\; q(a_1, \mathbf{x}_1), \tag{54.}$$

where we choose the joint distribution to be defined as

$$q(a_1, \mathbf{x}_1) \triangleq \underbrace{q(\mathbf{x}_1 \mid \mathbf{x}_0^{a_1}, \mathbf{y}_1)}_{q(\mathbf{x}_1 \mid a_1)} \underbrace{w_0^{a_1}}_{q(a_1)}. \tag{55.}$$

Importantly, we can straightforwardly sample from the joint by first sampling a_1^j from the categorical distribution[7]

$$a_1^j \;\sim\; q(a_1) \triangleq \mathrm{Cat}(\{w_0^i\}_{i=1}^M). \tag{56.}$$

Choosing the index a_1^j based on the previous weights $\{w_0^i\}_{i=1}^M$ is known as resampling. In contrast to the SIS case, where all indices were explicitly considered, the effect of resampling is to choose indices at random with a probability proportional to the weights w_0^i. One interpretation of this effect is that resampling concentrates attention to mixture components with higher weights. This is not always well justified since the utility of a mixture component relates to both the weight w_0^i and the distribution $p(\mathbf{x}_1 \mid \mathbf{x}_0^i)$.

Nevertheless, having sampled a_1^j, we then sample $\mathbf{x}_1$ conditioned on a_1^j via

$$\mathbf{x}_1^j \sim q\left(\mathbf{x}_1 \mid \mathbf{x}_0^{a_1^j}, \mathbf{y}_1\right). \tag{57.}$$

[7]The categorical distribution $\mathrm{Cat}(\{w^i\}_{i=1}^M)$ is parameterized by the nonnegative numbers $w^i \geq 0$, with $\sum_{i=1}^M w^i = 1$, and the probability mass $\mathbb{P}(j = i) = w^i$. There are many ways to sample from this distribution, each with slightly different properties (30).

Using this approach, we arrive at the following approximation by using N samples of $(a_1^j, \mathbf{x}_1^j) \sim q(\mathbf{x}_1 \mid \mathbf{x}_0^{a_1}, \mathbf{y}_1)\, w_0^{a_1}$:

$$\sum_{i=1}^{M} \int \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{y}_1 \mid \mathbf{x}_1)}{p(\mathbf{y}_1)} w_0^i\, p(\mathbf{x}_1 \mid \mathbf{x}_0^i) \mathrm{d}\mathbf{x}_1 \approx \frac{1}{N} \sum_{j=1}^{N} \frac{p(\mathbf{x}_2 \mid \mathbf{x}_1^j)\, p(\mathbf{y}_1 \mid \mathbf{x}_1^j)}{p(\mathbf{y}_1)} \frac{p(\mathbf{x}_1^j \mid \mathbf{x}_0^{a_1^j})}{q(\mathbf{x}_1^j \mid \mathbf{x}_0^{a_1^j}, \mathbf{y}_1)}. \quad 58.$$

It is common, but not essential, that $N = M$, which leads to the following approximation for the prediction distribution:

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{x}_2 \mid \mathbf{x}_1^i) \frac{p(\mathbf{y}_1 \mid \mathbf{x}_1^i)}{p(\mathbf{y}_1)} \frac{p(x_1^i \mid \mathbf{x}_0^{a_1^i})}{q(\mathbf{x}_1^i \mid \mathbf{x}_0^{a_1^i}, \mathbf{y}_1)}. \quad 59.$$

Similar to the SIS case, the expression on the right-hand side is not guaranteed to be a distribution for $\mathbf{x}_2$ since it may not have unit area. We can apply the same normalization strategy to arrive at the normalized weights via

$$w_1^i \triangleq \frac{\tilde{w}_1^i}{\sum_{j=1}^{M} \tilde{w}_1^j}, \qquad \tilde{w}_1^i \triangleq p(\mathbf{y}_1 \mid \mathbf{x}_1^i) \frac{p(x_1^i \mid \mathbf{x}_0^{a_1^i})}{q(\mathbf{x}_1^i \mid \mathbf{x}_0^{a_1^i}, \mathbf{y}_1)}. \quad 60.$$

Once again, we have that the prediction density is given by

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \sum_{i=1}^{M} w_1^i\, p(\mathbf{x}_2 \mid \mathbf{x}_1^i). \quad 61.$$

As with the SIS case, we can therefore repeat the above steps to arrive at the SIR particle filter provided in Algorithm 1 with the SIR choice.

3.3. The Auxiliary Particle Filter

In the SIR case, we chose the joint distribution according to Equation 55, but this is not essential. We can, in fact, use any reasonable joint distribution $q(a_1, \mathbf{x}_1)$. This flexibility was first employed within the auxiliary particle filter (APF) approach (31). In particular,

$$q(a_1, \mathbf{x}_1) \triangleq \underbrace{q(\mathbf{x}_1 \mid \mathbf{x}_0^{a_1}, \mathbf{y}_1)}_{q(\mathbf{x}_1 \mid a_1)} \underbrace{v_0^{a_1}}_{q(a_1)}, \quad 62.$$

where the main difference compared with Equation 55 is that we allow more flexibility in choosing the probability masses v_0^i for the indices a_1. Following similar reasoning to the SIS and SIR cases, we arrive at the following mixture approximation of the prediction distribution:

$$p(\mathbf{x}_2 \mid \mathbf{y}_1) \approx \sum_{i=1}^{M} w_1^i\, p(\mathbf{x}_2 \mid \mathbf{x}_1^i), \quad 63.$$

where the normalized weights for the APF case are

$$w_1^i \triangleq \frac{\tilde{w}_1^i}{\sum_{j=1}^{M} \tilde{w}_1^j}, \qquad \tilde{w}_1^i \triangleq p(\mathbf{y}_1 \mid \mathbf{x}_1^i) \frac{w_0^{a_1^i}\, p(x_1^i \mid \mathbf{x}_0^{a_1^i})}{v_0^{a_1^i}\, q(\mathbf{x}_1^i \mid \mathbf{x}_0^{a_1^i}, \mathbf{y}_1)}. \quad 64.$$

A commonly used choice for the proposal is to set

$$v_0^i = w_0^i\, p(y_1 \mid x_0^i), \quad 65.$$

$$q(x_1 \mid x_0^i, y_1) = p(x_1 \mid x_0^i, y_1). \quad 66.$$

The argument for this choice stems from the fact that[8]

$$p(y_1 \mid x_1)p(x_1 \mid x_0) = p(y_1, x_1 \mid x_0) = p(y_1 \mid x_0)p(x_1 \mid x_0, y_1). \tag{67.}$$

Therefore,

$$\tilde{w}_1^i = \frac{p(\mathbf{y}_1 \mid \mathbf{x}_1^i)}{p(\mathbf{y}_1)} \frac{w_0^{a_1^i}\, p(x_1^i \mid \mathbf{x}_0^{a_1^i})}{v_0^{a_1^i}\, q(\mathbf{x}_1^i \mid \mathbf{x}_0^{a_1^i}, \mathbf{y}_1)} = \frac{p(\mathbf{y}_1 \mid \mathbf{x}_1^i)}{p(\mathbf{y}_1)} \frac{w_0^{a_1^i}\, p(x_1^i \mid \mathbf{x}_0^{a_1^i})}{w_0^{a_1^i} p(y_1 \mid x_0^{a_1^i})\, p(x_1^i \mid x_0^{a_1^i}, y_1)} = \frac{1}{p(\mathbf{y}_1)}, \tag{68.}$$

which implies that all particles are equally important, and we therefore minimize wasted effort. This approach is known as the fully adapted APF. In general, it is not possible to compute v_0 and draw from $q(\cdot)$ defined in this way, in which case several authors have proposed replacing $p(y_1 \mid x_0)$ and $p(x_1 \mid x_0, y_1)$ with approximations (31). This leads to the so-called partially adapted APF. At any rate, we arrive back at another mixture distribution for the prediction density, and the above can be repeated sequentially to deliver the APF summarized in Algorithm 1 with the APF choice.

3.4. A Unified View

The above discussion shows that three important variants of the particle filter—SIS, SIR, and APF—all employ Monte Carlo integration to approximate various expectations. These algorithms may be differentiated on the basis of which part of the expectation they approximate and on the basis of the proposal they employ. These similarities and differences can be summarized in Algorithm 1, which highlights that the practical difference in implementation is very subtle. It essentially reduces to how the so-called ancestor index a_k^i is chosen. In summary, (*a*) the SIS filter selects all ancestor indices, even if they are extremely unlikely; (*b*) the SIR filter chooses ancestor indices randomly according to the weights w_{k-1}^i; (*c*) the APF chooses ancestor indices randomly according to a more flexible set of weights v_{k-1}^i, and this added flexibility can be exploited to minimize wasted effort; and (*d*) all filters allow for the use of a general proposal $q(\mathbf{x}_k \mid \mathbf{x}_k^i, \mathbf{y}_k)$.

Algorithm 1 (unified particle filter: generate particles and weights $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^M, \forall k$).
Require: $M > 0$ and the particle filter variant to use (SIS, SIR, or APF).
for $k = 1, \ldots, N$ **do**
  for $i = 1, \ldots, M$ **do**
    if SIS **then**
      Set $a_k^i = i$ and $v_{k-1}^{a_k^i} = 1$.
    else if SIR **then**
      Draw a sample a_k^i according to $a_k^i \sim \text{Cat}(\{w_{k-1}^j\}_{j=1}^M)$ and set $v_{k-1}^{a_k^i} = w_{k-1}^{a_k^i}$.
    else if APF **then**
      Draw a sample a_k^i according to $a_k^i \sim \text{Cat}(\{v_{k-1}^j\}_{j=1}^M)$.
    end if
    Draw a sample $\mathbf{x}_k^i$ according to $\mathbf{x}_k^i \sim q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{a_k^i}, \mathbf{y}_k)$.
    Compute weights $\tilde{w}_k^i$ according to $\tilde{w}_k^i \triangleq p(\mathbf{y}_k \mid \mathbf{x}_k^i) \frac{w_{k-1}^{a_k^i} p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^{a_k^i})}{v_{k-1}^{a_k^i} q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^{a_k^i}, \mathbf{y}_k)}$.
  end for
  Compute normalized weights $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^M \tilde{w}_k^j}$ for $i = 1, \ldots, M$.
end for

[8] Using conditional probability and the Markov property.

3.5. The Bootstrap Particle Filter

We would be remiss not to mention that choosing the proposal $q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^{a_k^i}, \mathbf{y}_k)$ as

$$q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^{a_k^i}, \mathbf{y}_k) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i) \tag{69.}$$

leads to the celebrated bootstrap particle filter, where some terms cancel to reveal

$$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k}) \approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{x}_{k+1} \mid \mathbf{x}_k^i) \underbrace{\frac{p(\mathbf{y}_k \mid \mathbf{x}_k^i)}{\sum_{j=1}^{M} p(\mathbf{y}_k \mid \mathbf{x}_k^j)}}_{\triangleq w_k^i}. \tag{70.}$$

3.6. Sequential Monte Carlo Convergence

The convergence of SMC methods has received significant attention (for a full theoretical treatment, see, e.g., 16). In essence, provided that the particle filter can correct errors in the initial state particles (a type of forgetting behavior), it can be shown that the particle filter converges. To be a little more specific, we define two expectations, I_k and $\hat{I}_k^M$, via

$$I_k \triangleq \int \psi(\mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{x}_k, \qquad \hat{I}_k^M \triangleq \int \psi(\mathbf{x}_k)\, \hat{p}^M(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{x}_k, \tag{71.}$$

where $\hat{p}^M(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ denotes the M-particle approximation of the prediction distribution at time k, and ψ satisfies $|\psi(\mathbf{x}_k)| \leq 1$ for all $\mathbf{x}_k$. Then it can be shown that (16)

$$\sup_{k \geq 0} \left| \mathrm{E}\left[\hat{I}_k^M\right] - I_k \right| \leq \frac{a}{M}, \qquad \sup_{k \geq 0} \mathrm{E}\left[\left(\hat{I}_k^M - I_k\right)^2\right] \leq \frac{b}{M}, \tag{72.}$$

where a and b are constants that do not depend on the number of observations N; this is the reason why particle methods can be used in online applications for state estimation.

Unfortunately, while this is promising in terms of the data length N, the same does not necessarily hold for the state dimension n_x. When naively implemented, particle filters are known to be impractical for state dimensions greater than $n_x \approx 10$ (see, e.g., 26, 27). In short, the main reason for this is that the terms a and b grow at an exponential rate in the state dimension; this is certainly the case for a standard implementation of the bootstrap particle filter (27).

This phenomenon is well known to the research community and to practitioners of SMC methods and has recently received theoretical underpinnings (26, 27). In particular, Rebeschini & van Handel (27) outlined a theoretical argument for constructing algorithms that do not suffer from the same exponential growth in state dimension, which fits within the unified presentation presented in this section. This was the motivation behind the work of Andersson et al. (32), who employed local SMC methods to adapt the proposal distribution within the broader SMC framework for large-dimension spatiotemporal systems.

4. PROBLEMS SUCCESSFULLY SOLVED BY SEQUENTIAL MONTE CARLO METHODS

The aim of this section is to provide a rough overview of problems where SMC has been—and will most likely continue to be—useful. The problem areas of system identification (Section 4.1) and state estimation (Section 4.2) are at the heart of both control and robotics, and we will see in Section 4.3 that SMC is in fact much more generally applicable than most of us thought at first.

4.1. System Identification

In system identification, we build mathematical models of dynamical systems from measured data (33). SMC is a useful component in solving the problem when the dynamics are nonlinear (for overviews providing ample entry points into this development, see, e.g., 34, 35). The commonly used maximum likelihood formulation,

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{y}_{1:N}), \tag{73.}$$

requires the likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}_{1:N})$—and often also its derivatives—to be available for evaluation. By writing the likelihood in the form

$$p_{\boldsymbol{\theta}}(\mathbf{y}_{1:N}) = \int p_{\boldsymbol{\theta}}(\mathbf{y}_{1:N}, \mathbf{x}_{1:N}) \mathrm{d}\mathbf{x}_{1:N}, \tag{74.}$$

we better see its relationship to the latent states. The mathematical motivation for Equation 74 is marginalization, with the interpretation that we average $p_{\boldsymbol{\theta}}(\mathbf{y}_{1:N}, \mathbf{x}_{1:N})$ over all possible state sequences $\mathbf{x}_{1:N}$. Equivalently, we can write

$$p_{\boldsymbol{\theta}}(\mathbf{y}_{1:N}) = \prod_{k=1}^{N} p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) = \prod_{k=1}^{N} \int p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{x}_k) p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{x}_k, \tag{75.}$$

where $\mathbf{y}_{1:0} = \emptyset$. This intractable integral explains why SMC is so natural for maximum likelihood estimation of the parameters in nonlinear state-space models. When SMC is used to approximate the likelihood in the integral in Equation 75, it is quite remarkable that the resulting likelihood estimate is in fact unbiased (16, 36). However, it is still a stochastic quantity, implying that the resulting optimization problem in Equation 73 is stochastic.

One way around this stochastic optimization problem is to make use of the expectation maximization algorithm (37), which amounts to iteratively solving

$$\boldsymbol{\theta}_{i+1} = \arg\max_{\boldsymbol{\theta}} \int \ln p_{\boldsymbol{\theta}}(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}) p_{\boldsymbol{\theta}_i}(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N}) \mathrm{d}\mathbf{x}_{1:N} \tag{76.}$$

for $i = 1, 2, \ldots$, initialized as $\boldsymbol{\theta}_0$. The sequence $\{\boldsymbol{\theta}_i\}_{i \geq 0}$ computed in this way is guaranteed to not decrease the log-likelihood (37) [i.e., $\ln p_{\boldsymbol{\theta}_{i+1}}(\mathbf{y}_{1:N}) \geq \ln p_{\boldsymbol{\theta}_i}(\mathbf{y}_{1:N})$], which explains why expectation maximization can be used in solving maximum likelihood problems. The additional challenge with the formulation in Equation 76 is that the smoothing distribution $p_{\boldsymbol{\theta}_i}(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$ is not available in closed form, but we can employ SMC-based methods to approximate this quantity, effectively replacing the integral with the following tractable sum over all particles:

$$\boldsymbol{\theta}_{i+1} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{M} w^i \ln p_{\boldsymbol{\theta}}(\mathbf{x}^i_{1:N}, \mathbf{y}_{1:N}). \tag{77.}$$

The particles $\{\mathbf{x}^i_{1:N}\}_{i=1}^M$ and their weights $\{w^i_{1:N}\}_{i=1}^M$ are computed for a model parameterized by the current iteration $\boldsymbol{\theta}_i$. Details on this solution are available in papers by Olsson et al. (38) and Schön et al. (39) and were later further improved by Lindholm & Lindsten (40).

A more direct solution to the maximum likelihood problem is to acknowledge its stochastic nature in the first place and make use of stochastic optimization algorithms. These algorithms have—due to their importance in solving deep learning problems—experienced a good development over the past decade (see, e.g., 41). Stochastic optimization algorithms rely on a Markov chain of the form

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha_i \mathbf{d}_i, \tag{78.}$$

providing an iterative updating mechanism for the parameters. Here, $\mathbf{d}_i$ is the search direction, and $\alpha_i > 0$ denotes the so-called step length, also referred to as the learning rate. Wills & Schön (42) developed a second-order stochastic optimization algorithm specifically tailored for nonlinear system identification using SMC.

The joint distribution of all random variables used in the model—sometimes referred to as the full probabilistic model—within the maximum likelihood formulation is given by

$$p_\theta(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}) = \prod_{k=1}^{N} \underbrace{p_\theta(\mathbf{y}_k \mid \mathbf{x}_k)}_{\text{observation}} \prod_{k=1}^{N} \underbrace{p_\theta(\mathbf{x}_k \mid \mathbf{x}_{k-1})}_{\text{dynamics}} p_\theta(\mathbf{x}_1), \tag{79.}$$

where the latent states follow a prior distribution according to model specification. The maximum likelihood formulation implies that we assume the unknown parameters $\boldsymbol{\theta}$ to be modeled as deterministic variables. Depending on the problem setting, a Bayesian formulation (21) might be more useful. Here, the unknown parameters are instead modeled as random variables, implying that we need to complement the model with an assumption of this prior $p(\boldsymbol{\theta})$ as well. The full probabilistic model is now instead given by $p(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}, \boldsymbol{\theta}) = p(\mathbf{x}_{1:N}, \mathbf{y}_{1:N} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$, where the first term is—with a slight change of notation—given in Equation 79. The goal is then to compute the posterior distribution

$$p(\boldsymbol{\theta} \mid \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{y}_{1:k})}. \tag{80.}$$

The first thing to note is that the likelihood $p(\mathbf{y}_{1:k} \mid \boldsymbol{\theta})$ takes center stage in the Bayesian formulation as well. Over the past decade, we have seen a most interesting and useful development when it comes to Bayesian solutions based on SMC. To a large extent, this all started with the particle Markov chain Monte Carlo (MCMC) construction (43). Andrieu et al. (43) introduced both Metropolis–Hastings (44, 45) and Gibbs (46) constructions. The resulting MCMC algorithms are exact in the sense that the target distribution of interest—typically $p(\boldsymbol{\theta} \mid \mathbf{y}_{1:N})$ or $p_\theta(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$—is the stationary distribution of the Markov chain, even though it makes use of an SMC-based approximation of the likelihood in evaluating the acceptance probability. This has resulted in the somewhat peculiar but descriptive term for this class of algorithms, namely exact approximations.

The particle Metropolis–Hastings sampler makes use of SMC to guide a Metropolis–Hastings method through the parameter space (for a tutorial introduction, see 47). Slightly more specifically, it makes use of a nonnegative and unbiased likelihood estimate provided by SMC, which is a version of pseudomarginal Metropolis–Hastings (48) applied to this particular setting. A possibly underappreciated fact is that the particle Metropolis–Hastings algorithm provides a solution to the smoothing problem as well.

When it comes to the particle Gibbs construction, SMC is used as a high-dimensional proposal mechanism on the space of state trajectories $\mathbf{x}_{1:N}$. The original particle Gibbs construction has been improved via the addition of a so-called ancestor sampling step (49).

4.2. State Estimation

We have already discussed aspects of the state estimation problem to a great extent in the earlier sections of this review, via the practical example in Section 2, after which we used the nonlinear filtering problem to derive the SMC method in Section 3. The information about the state is represented using probability density functions of the form $p(\mathbf{x}_{r:k} \mid \mathbf{y}_{1:s})$. Depending on the relationships between the time indices r, k, and s, this problem falls into one of three main categories: filtering, prediction, or smoothing. **Table 1** provides a more detailed enumeration of the most commonly encountered state inference problems.

Table 1 Commonly used filtering and smoothing densities[a]

Name	Notation
Marginal filtering	$p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$
Joint filtering	$p(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k}),\ k = 0, 1, \ldots, N-1$
Prediction	$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})$
l-step prediction	$p(\mathbf{x}_{k+l} \mid \mathbf{y}_{1:k}),\ l \geq 1$
Joint smoothing	$p(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$
Marginal smoothing ($k \leq N$)	$p(\mathbf{x}_k \mid \mathbf{y}_{1:N})$
Fixed-interval smoothing ($s < k \leq N$)	$p(\mathbf{x}_{s:k} \mid \mathbf{y}_{1:N})$
Fixed-lag smoothing (l fixed)	$p(\mathbf{x}_{k-l+1:k} \mid \mathbf{y}_{1:k})$

[a]These densities may appear either on their own or as components in a larger algorithm.

When it comes to smoothing, the joint smoothing density $p(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$ is the main object we are after. The other smoothing solutions are marginal densities with respect to this density. One of the most commonly used strategies for computing smoothing solutions is to first run a (forward) filter and then perform a backwards pass to carry the information from the future backward in time. An early and well-used example of this strategy is provided by the Rauch–Tung–Striebel smoother (50) for linear Gaussian state-space models. The equivalent idea for particle filters was introduced in 2000 (51), but it was not practical due to its high computational cost. However, since then we have seen very useful developments—in particular, when it comes to backward simulation, relying on a backward pass, where the states are simulated backward in time, resulting in (uncorrelated) samples from $p(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$. Lindsten & Schön (52) provided a tutorial introduction to the backward simulation idea.

Since the introduction of the particle MCMC construction (43) roughly a decade ago, we have seen the emergence of a completely different class of very capable smoothing algorithms—namely, those based on carefully engineered Markov kernels. The idea is to run SMC methods within an outer MCMC construction, implying an iterative algorithm. Svensson et al. (53) provided a concrete example of such a construction.

4.3. Sequential Monte Carlo Is Generally Applicable

SMC is useful not only when it comes to nonlinear state-space models, but also for a much broader class of models. It can be used whenever the model contains a sequential structure, be it natural or artificial. Concrete examples include the class of probabilistic graphical models (54) and the even more general programmatic model (55) offered by probabilistic programming languages (56). The combined use of variational inference (57) and SMC has also seen useful developments recently (see, e.g., 58–60). Naesseth et al. (10) provided a tutorial introduction to SMC in this more general setting.

5. CONCLUSIONS AND FUTURE WORK

While the use of SMC to solve nonlinear filtering and system identification problems is starting to mature, its use in more general settings is only just starting to emerge. In this article, we have focused on providing a nonstandard and hopefully intuitive presentation of the SMC method when it is used to solve the nonlinear filtering problem.

A clear trend is that SMC methods are increasingly being used as components in various larger (often iterative) algorithms. Examples include Bayesian learning, where SMC is used within an MCMC algorithm (43); iterated filtering (61); maximum likelihood using stochastic

optimization (42); maximum likelihood via expectation maximization (39); and more elaborate blends of variational inference and SMC (58–60), to name a few.

A rewarding way forward is likely offered by the two main design choices of SMC—the intermediate target distributions and the proposal distribution—which pave the way for new algorithm development. By multiplying the intermediate target in each step by a so-called twisting potential, one can utilize this design freedom to obtain a better approximation (62). Here, there are interesting possibilities in the use of deterministic algorithms to approximate the twisting potential. While algorithms based on variational inference are fast, the resulting estimates suffer from biases that are hard to quantify. The Monte Carlo methods are the other way around, in that they enjoy asymptotic consistency and are well supported by theory but can instead suffer from a high computational cost. Hence, it is natural to follow the path started by Maddison et al. (58), Naesseth et al. (59), and Le et al. (60) and develop solutions that blend variational inference and SMC to achieve fast algorithms with theoretical guarantees.

Various deep architectures have recently also proved highly useful when it comes to nonlinear dynamics [see, e.g., the ordinary differential equation variational autoencoder (ODE^{2}VAE) (63) and Kalman variational autoencoder (64) constructions]. Including SMC in this setting is, naturally, an interesting next step. The main roadblock is arguably the fact that the essential resampling step is not differentiable with respect to the model and SMC parameters. However, recent developments (65) have introduced a first differentiable particle filter, effectively opening the way for end-to-end training.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

We would like to acknowledge the very kind feedback and support from Dr. Alejandro Donaire, Dr. Christopher Renton, Dr. Jack Umenberger, and Dr. Johannes Hendriks. This research was financially supported by the projects NewLEADS – New Directions in Learning Dynamical Systems (contract 621-2016-06079) and Deep Probabilistic Regression – New Models and Learning Algorithms (contract 2021-04301), both funded by the Swedish Research Council and the Kjell and Märta Beijers Foundation.

LITERATURE CITED

1. Thrun S, Burgard W, Fox D. 2005. *Probabilistic Robotics*. Cambridge, MA: MIT Press
2. Duncan S, Gyöngy M. 2006. Using the EM algorithm to estimate the disease parameters for smallpox in 17th century London. In *2006 IEEE International Conference on Control Applications*, pp. 3312–17. Piscataway, NJ: IEEE
3. Jazwinski AH. 1970. *Stochastic Processes and Filtering Theory*. New York: Academic
4. Smith GL, Schmidt SF, McGee LA. 1962. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. Tech. Rep. TR R-135, Natl. Aeronaut. Space Adm., Washington, DC
5. Julier SJ, Uhlmann JK. 2004. Unscented filtering and nonlinear estimation. *Proc. IEEE* 92:401–22
6. Gordon NJ, Salmond DJ, Smith AFM. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F* 140:107–13
7. Kitagawa G. 1993. A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models. In *Proceedings of the 2nd US-Japan Joint Seminar on Statistical Time Series Analysis*, pp. 110–31. N.p.

8. Stewart L, McCarty P. 1992. The use of Bayesian belief networks to fuse continuous and discrete information for target recognition and discrete information for target recognition, tracking, and situation assessment. In *Signal Processing, Sensor Fusion and Target Recognition*, ed. V Libby, I Kadar pp. 177–85. Proc. SPIE 1699. Bellingham, WA: SPIE
9. Doucet A, Johansen AM. 2011. A tutorial on particle filtering and smoothing: fifteen years later. In *Nonlinear Filtering Handbook*, ed. D Crisan, B Rozovsky, pp. 656–704. Oxford, UK: Oxford Univ. Press
10. Naesseth AC, Lindsten F, Schön TB. 2019. Elements of sequential Monte Carlo. *Found. Trends Mach. Learn.* 12:307–92
11. Cappé O, Godsill S, Moulines E. 2007. An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. IEEE* 95:899–924
12. Chopin N, Papaspiliopoulos O. 2020. *An Introduction to Sequential Monte Carlo*. Cham, Switz.: Springer
13. Särkkä S. 2013. *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge Univ. Press
14. Douc R, Moulines E, Stoffer D. 2014. *Nonlinear Time Series: Theory, Methods, and Applications with R Examples*. Boca Raton, FL: CRC
15. Cappé O, Moulines E, Rydén T. 2005. *Inference in Hidden Markov Models*. Berlin: Springer
16. Del Moral P. 2004. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer
17. Endo A, van Leeuwen E, Baguelin M. 2019. Introduction to particle Markov chain Monte Carlo for disease dynamics modellers. *Epidemics* 29:100363
18. Naesseth AC, Lindsten F, Schön TB. 2019. High-dimensional filtering using nested sequential Monte Carlo. *IEEE Trans. Signal Process.* 67:4177–88
19. Davey S, Gordon N, Holland I, Rutten M, Williams J. 2016. *Bayesian Methods in the Search for MH370*. Singapore: Springer
20. Gut A. 1995. *An Intermediate Course in Probability*. New York: Springer
21. Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. 2013. *Bayesian Data Analysis*. Boca Raton, FL: CRC. 3rd ed.
22. Perot A, Fabry C. 1899. On the application of interference phenomena to the solution of various problems of spectroscopy and metrology. *Astrophys. J.* 9:87
23. Abbott BP, Abbott R, Abbott TD, Abernathy MR, Acernese F, et al. 2016. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.* 116:061102
24. Fricke TT. 2011. *Homodyne detection for laser-interferometric gravitational wave detectors*. PhD Thesis, La. State Univ., Baton Rouge
25. Hariharan P. 2010. *Basics of Interferometry*. Amsterdam: Elsevier
26. Snyder C, Bengtsson T, Bickel P, Anderson J. 2008. Obstacles to high-dimensional particle filtering. *Mon. Weather Rev.* 136:4629–40
27. Rebeschini P, van Handel R. 2015. Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab.* 25:2809–66
28. Kailath T, Sayed AH, Hassibi B. 2000. *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall
29. Doucet A, de Freitas N, Gordon N, ed. 2001. *Sequential Monte Carlo Methods in Practice*. New York: Springer
30. Hol J, Schön TB, Gustafsson F. 2006. On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pp. 79–82. Piscataway, NJ: IEEE
31. Pitt MK, Shephard N. 1999. Filtering via simulation: auxiliary particle filters. *J. Am. Stat. Assoc.* 94:590–99
32. Andersson C, Ribeiro AH, Tiels K, Wahlström N, Schön TB. 2019. Deep convolutional networks in system identification. In *2019 IEEE 58th Conference on Decision and Control*, pp. 3670–76. Piscataway, NJ: IEEE
33. Ljung L. 1999. *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice Hall. 2nd ed.
34. Schön TB, Lindsten F, Dahlin J, Wågberg J, Naesseth AC, et al. 2015. Sequential Monte Carlo methods for system identification. *IFAC-PapersOnLine* 48(28):775–86
35. Kantas N, Doucet A, Singh SS, Maciejowski JM, Chopin N. 2015. On particle methods for parameter estimation in state-space models. *Stat. Sci.* 30:328–51
36. Pitt MK, dos Santos Silva R, Giordani R, Kohn R. 2012. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *J. Econom.* 171:134–51

37. Dempster A, Laird N, Rubin D. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *J. R. Stat. Soc. B* 39:1–38
38. Olsson J, Douc R, Cappé O, Moulines E. 2008. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models. *Bernoulli* 14:155–79
39. Schön TB, Wills A, Ninness B. 2011. System identification of nonlinear state-space models. *Automatica* 47:39–49
40. Lindholm A, Lindsten F. 2019. Learning dynamical systems with particle stochastic approximation EM. arXiv:1806.09548 [stat.CO]
41. Bottou L, Curtis FE, Nocedal J. 2018. Optimization methods for large-scale machine learning. *SIAM Rev*. 60:223–311
42. Wills AG, Schön TB. 2021. Stochastic quasi-Newton with line-search regularisation. *Automatica* 127:109503
43. Andrieu C, Doucet A, Holenstein R. 2010. Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. B* 72:269–342
44. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equations of state calculations by fast computing machine. *J. Chem. Phys.* 21:1087–92
45. Hastings WK. 1970. Monte Carlo simulation methods using Markov chains and their applications. *Biometrica* 57:97–109
46. Geman S, Geman D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-6:721–41
47. Schön TB, Svensson A, Murray LM, Lindsten F. 2018. Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo. *Mech. Syst. Signal Process.* 104:866–83
48. Andrieu C, Roberts GO. 2009. The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Stat.* 37:697–725
49. Lindsten F, Jordan MI, Schön TB. 2014. Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.* 15:2145–84
50. Rauch HE, Tung F, Striebel CT. 1965. Maximum likelihood estimates of linear dynamic systems. *AIAA J.* 3:1445–50
51. Doucet A, Godsill SJ, Andrieu C. 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* 10:197–208
52. Lindsten F, Schön TB. 2013. Backward simulation methods for Monte Carlo statistical inference. *Found. Trends Mach. Learn.* 6:1–143
53. Svensson A, Schön TB, Kok M. 2015. Nonlinear state space smoothing using the conditional particle filter. *IFAC-PapersOnLine* 48(28):975–80
54. Naesseth AC, Lindsten F, Schön TB. 2014. Sequential Monte Carlo for graphical models. In *Advances in Neural Information Processing Systems 27*, ed. Z Ghahramani, M Welling, C Cortes, N Lawrence, KQ Weinberger, pp. 1862–70. Red Hook, NY: Curran
55. Murray LM, Schön TB. 2018. Automated learning with a probabilistic programming language: Birch. *Annu. Rev. Control* 46:29–43
56. Wood F, Meent JW, Mansinghka V. 2014. A new approach to probabilistic programming inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, ed. S Kaski, J Corander, pp. 1024–32. Proc. Mach. Learn. Res. 33. N.p.: PMLR
57. Blei D, Kucukelbir A, McAuliffe J. 2017. Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* 112:859–77
58. Maddison CJ, Lawson J, Tucker G, Heess N, Norouzi M, et al. 2017. Filtering variational objectives. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, U Von Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 6574–84. Red Hook, NY: Curran
59. Naesseth AC, Linderman S, Ranganath R, Blei D. 2018. Variational sequential Monte Carlo. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ed. A Storkey, F Perez-Cruz, pp. 968–77. Proc. Mach. Learn. Res. 84. N.p.: PMLR
60. Le TA, Igl M, Rainforth T, Jin T, Wood F. 2018. Auto-encoding sequential Monte Carlo. In *Proceedings of the 2018 International Conference on Learning Representations*. La Jolla, CA: Int. Conf. Learn. Represent. **https://openreview.net/pdf?id=BJ8c3f-0b**

61. Ionides EL, Bhadra A, Atchadé Y, King AA. 2011. Iterated filtering. *Ann. Stat.* 39:1776–802
62. Guarniero P, Johansen AM, Lee A. 2016. The iterated auxiliary particle filter. *J. Am. Stat. Assoc.* 112:1636–47
63. Yildiz C, Heinonen M, Lahdesmaki H. 2019. ODE^{2}VAE: deep generative second order ODEs with Bayesian neural networks. In *Advances in Neural Information Processing Systems 32*, ed. H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, R Garnett, pp. 13366–75. Red Hook, NY: Curran
64. Fraccaro M, Kamronn S, Paquet U, Winther O. 2017. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, U Von Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 3602–11. Red Hook, NY: Curran
65. Corenflos A, Thornton J, Deligiannidis G, Doucet A. 2021. Differentiable particle filtering via entropy-regularized optimal transport. In *Proceedings of the 38th International Conference on Machine Learning*, ed. M Meila, T Zhang, pp. 2100–11. Proc. Mach. Learn. Res. 139. N.p.: PMLR
66. Ersbo P. 2018. *Displacement estimation for homodyne Michelson interferometers based on particle filtering*. MS Thesis, Uppsala Univ., Uppsala, Swed.

Annual Review of Control, Robotics, and Autonomous Systems

Volume 6, 2023

Contents

Errata

An online log of corrections to *Annual Review of Control, Robotics, and Autonomous Systems* articles may be found at http://www.annualreviews.org/errata/control